CREXDATA

Critical Action Planning over Extreme-Scale Data

Deliverable D3.2 Final Report on System Architecture, Integration, and Released Software Stacks

Version 1.3

Documentation Information

Contract Number	**101092749**
Project Website	https://crexdata.eu/
Contractual Deadline	M36 / 31.12.2025
Dissemination Level	Public
Nature	Report
Author	Ralf Klinkenberg (Altair RapidMiner (RM))
Contributors	Giannis Spiliopoulos (UAEGEAN), Dimitris Zissis (UAEGEAN), Elias Xidias (UAEGEAN), Christina Tsikou (UAEGEAN), Jenny Papaziou (UAEGEAN), Nikos Sapountzis (UAEGEAN), Georgia Spyridaki (UAEGEAN), Arnau Montagud (BSC), Thalia Diniaco (BSC), Jonathan Orama (BSC), Miguel Ponce de León (BSC), Beatriz Eguzkitza (BSC), Jordi Roca (HYDS), Georgios Grigoropoulos (Kpler), Konstantina Bereta (Kpler), Marios Vodas (Kpler), Ilias Chamatidis (Kpler) Ourania Ntouni (TUC), Dimitrios Banelas (TUC), Nikos Giatrakos (TUC), Antonios Deligiannakis (TUC), Jens Pottebaum (UPB), Marcel Ebel (UPB), Deniz Özcan (UPB), Niklas Döhner (UPB), Edin Klapic (RM), Enid Vyshka (RM)
Reviewer	Nikos Giatrakos (TUC)
Keywords	System Architecture, System Integration, Use Cases, Data Integration, Data Processing Workflows, Visual Workflow Design, Distributed Data Stream Processing, Cloud-to-Edge Processing, Federated Data Processing

CREXDATA has received funding from the European Union's Horizon Europe programme under grant agreement number 101092749.

Change Log

Version	Author	Date	Description of Change
V0.1	Ralf Klinkenberg (RM)	20/11/2025	Draft document incl. table of content (ToC), introductory chapters about system architecture and system integration (and uses cases), including Chapter 4 on the CREXDATA system components and their integration
V0.2	Giannis Spiliopoulos (UAEGEAN), Dimitris Zissis (UAEGEAN)	09/12/2025	Section 4.2 on Multi-Modal Data Fusion added
V0.3	Ourania Ntouni (TUC), Dimitrios Banelas (TUC), Nikos Giatrakos (TUC), Antonios Deligiannakis (TUC)	09/12/2025	Section 4.10 on Edge-to-Cloud Processing added
V0.4	Ralf Klinkenberg (RM), Edin Klapic (RM), Enid Vyshka (RM), Jonathan (BSC)	09/12/2025	Sections 4.1, 4.5, 4.8, 4.9, and 4.11 on the system components and their integratrion added, Chapter 5 updated; Sections 4.3, 4.4, 4.6, 4.7, 4.9, 4.12, 4.13, 4.14, and 4.15 started.
V0.5	Ralf Klinkenberg (RM),Giannis Spiliopoulos (UAEGEAN)	16/12/2025	Updates in Chapter 4
V0.6	Nikos Giatrakos (TUC)	21/12/2025	Internal Review completed
V0.9	Edin Klapic (RM), Ralf Klinkenberg (RM)	23/12/2025	Finalization of Chapters 1, 2, 3, 4, 5. Updates based on the internal review
V1.0	Antonios Deligiannakis (TUC)	29/12/2025	Final pass and approval by the Coordinator.

(Table continued on the next page)

Version	Author	Date	Description of Change
V1.1	Spiliopoulos Giannis (UAEGEAN)	10/03/2026	Post-review - added section 4.2.2 with state-of-the-art and contributions for MMDF
V1.2	Ralf Klinkenberg (RM), Marcel Eibel (UPB), Nikolaos Giatrakos (TUC), Vasilis Samoladas (TUC), Salvatore Rinzivillo (CNR)	20/03/2026	Section 7 on related work and state-of-the-art and contributions of the CREXDATA platform added
V1.3	Antonios Deligiannakis (TUC)	13/04/2026	Finalized all sections

Contents

List of Figures

List of Tables

Executive Summary

The vision of CREXDATA was to develop a generic platform for real-time critical situation management including flexible action planning and agile decision making over data of extreme scale and complexity. CREXDATA developed the algorithmic apparatus, software architecture, and tools for federated predictive analytics and forecasting under uncertainty. The framework boosts proactive decision making providing highly accurate and transparent short- and long-term forecasts to end-users, explainable via advanced visual analytics and accurate, real-time, off and on-site augmented reality facilities.

This document describes the CREXDATA system architecture, the system components, the system integration, and the released software stacks of the CREXDATA project. The system architecture of the prototype conceptualized and developed in the CREXDATA project and utilized by the CREXDATA pilots and simulators includes a graphical tool for designing processing workflows. This graphical design tool interacts with and influences the system architecture. The system architecture can handle multi-modal data fusion from dispersed sources and of different types (extreme scale data streams, time series data, text data, image data, video data, weather data, etc.) and includes the overall system integration.

This document also describes the use cases, their scenarios, and requirements in relation to the system components that are involved to realize those scenarios as data processing workflows. Hence this documents also describes the use cases and their scenarios, the system components involved, and workflow level aspects. The CREXDATA system is adopted per use case in terms of demonstrator systems. This includes communication interfaces to use case domain-specific systems, implementation of custom data fusion operators, as well as configuration of CREXDATA components and visualization schemes in case of specific User Interfaces (UIs). The CREXDATA use cases include:

- weather-induced emergencies, with pilots in Dortmund, Austria and Finland, as well as an initial application scenario on pluvial flooding,
- health, with two application scenarios on epidemiology and multiscale lung infection,
- maritime, with two application scenarios on Collision Forecasting and Hazardous Weather Rerouting.

In these use case scenarios, data streams are produced and collected over inherently distributed, networked architectures composed of several sites, including edge devices operating in the field, relay nodes and one or more clouds or computer clusters. Continuously accumulating all the extreme scale streaming data at a single site/cloud for executing analytics is severely suboptimal resource-wise, and it compromises the real-time nature of the target applications. Therefore, novel algorithms and frameworks for federated learning and forecasting solutions optimize the use of the underlying hardware resources while delivering analytics as a service. CREXDATA exploits in-situ processing capabilities of on-site and remote devices and distributively allocates the computational load in-network exploiting the capacity of each available resource.

The deliverable supports alignment of interfaces between the integrated systems and use case-specific demonstrators, mapping of terminologies across use cases, and preparation of cross-impact evaluation. As simulators are use case specific, these are covered as an extension of CREXDATA system interfaces.

1 Introduction

The project CREXDATA addressed planning and decision making based on data of extreme scale and complexity. Three use cases of real-time critical situation management were carried out involving numerous datasets of very different nature. Those datasets, that are either input or output of algorithms, software tools, and hardware devices, are integrated in a Prediction-as-a-Service (PaaS) system as outcome of the project. The CREXDATA use cases are:

- The Weather Emergencies Use Case is devoted to improving situational awareness in weather emergency situations, so that informed decisions are taken by civil protection avoiding disaster impacts. Two levels of data sources are considered: (i) local environment data including stationary sensors' systems, as well as mobile robotic sensor platforms deployed in case of an incident and (ii) global environment data referring to current and forecasted weather data issued by Copernicus, covering specific fields such as forest fires, river banks, etc.
- The Health Crisis Use Case considers two different scales: (i) epidemiological models are used to build digital twins of the movement and infection of populations and (ii) mechanistic multiscale models simulate different treatments of patients as drug administration or the use of mechanical ventilators in severe patients with collapsed lungs among other interventions.
- The Maritime Use Case provides vessel routing and route forecasting solutions in (weather or other) emergency situations that perform for all vessels of a fleet simultaneously (instead of on-demand request per vessel).

This document describes the CREXDATA system architecture, the system components, the system integration, and the released software stacks of the CREXDATA project. This document also serves to capture all aspects of the use cases and their scenarios in relation to the system components that are involved to realize those scenarios as data processing workflows. Hence, this document also describes the use cases and their scenarios, the system components involved, and workflow level aspects.

Within the CREXDATA project, WP3 defined and implemented the system architecture prototype that materializes the application scenarios on which the CREXDATA prototype is based. The objectives were:

- Define an architecture for extreme-scale analytics that enables the seamless interaction and interconnection between its data processing components.
- Enable the graphical design of data processing workflows, in order to facilitate the data analyst in specifying and correctly configuring the processing that the data analyst wishes to perform on the data.
- Enable different visual analytic tools and interfaces to be easily plugged in.
- Materialize the architecture into a working prototype.

For each use case, WP2 developed or enhanced the simulators that were integrated into demonstrators and used in the pilots of CREXDATA. The system provided through WP3 (covering elements of WP4-5) was adopted per use case in terms of demonstrators. This included communication interfaces to use case specific systems (like ARGOS in T2.1), implementation of custom data fusion operators (WP3), as well as configuration of CREXDATA components and visualization schemes in case of specific user interfaces (UIs) (especially for Augmented Reality (AR), in WP5).

The system architecture of the prototype conceptualized and developed in the CREXDATA project and utilized by the CREXDATA pilots and simulators includes a graphical tool for designing processing workflows. This graphical design tool interacts with and influences the system architecture. The system architecture can handle multi-modal data fusion from dispersed sources of different types (extreme scale data streams, time series data, text data, image data, video data, weather data, etc.), perform distributed data processing, machine learning and federated learning in cloud / fog / edge / mixed compute environments, e.g. for complex event forecasting, and includes the overall system integration. In the use case scenarios of this project, data streams were produced and collected over inherently distributed, networked architectures composed of several sites, including edge devices operating in the field, relay nodes and one or more clouds or computer clusters. Continuously accumulating all the extreme scale streaming data at a single site/cloud for executing analytics is severely suboptimal resource-wise, and it compromises the real-time nature of the target applications. Therefore, novel algorithms and frameworks for workflow optimization, federated learning and forecasting solutions optimize the use of the underlying hardware resources while delivering analytics as a service. CREXDATA exploits in-situ processing capabilities of on-site and remote devices and distributively allocates the computational load in-network exploiting the capacity of each available resource.

The CREXDATA system is adopted per use case in terms of demonstrator systems. This includes communication interfaces to use case domain-specific systems, implementation of custom data fusion operators to be mapped to generic ones, as well as configuration of CREXDATA components and visualization schemes in case of specific User Interfaces (UIs).

The CREXDATA use cases include:

- weather-induced emergencies, with pilots in Dortmund, Austria and Finland, as well as an initial application scenario on pluvial flooding,
- health, with two application scenarios on epidemiology and multiscale lung infection,
- maritime, with two application scenarios on Collision Forecasting and Hazardous Weather Rerouting.

The deliverable supports alignment of interfaces between the integrated systems and use-case-specific demonstrators, mapping of terminologies across use cases, and cross-impact evaluation, especially in terms of consolidation of application program interfaces (APIs). As simulators are use case specific, these are integrated via extensions of the CREXDATA system interfaces.

1.1 Relation to Other Project Documents

[GA] Grant Agreement (no. 101092749) with its

[DoA] Description of Actions (DoA, part of the [GA])

[CA] Consortium Agreement

[D1.1] Deliverable D1.1 Quality Assurance Plan

[D1.2] Deliverable D1.2 Data Management Plan

[D1.3] Deliverable D1.3 Ethics Manual

[D2.1] Deliverable D2.1 Data Handling and Use Case Scenario Specifications

[D2.2] Deliverable D2.2 Initial Use Case Evaluation, Pilots, Demonstrators and Simulation Models and Tools

[D2.3] Deliverable D2.3 Final Use Case Evaluation, Pilots, Demonstrators and Simulation Models and Tools

[D3.1] Deliverable D3.1 Initial Report on System Architecture, Integration Approach, and Released Software Stacks

[D4.1] Deliverable D4.1 Initial Report on Complex Event Forecasting, Learning, and Analytics

[D4.2] Deliverable D4.2 Final Report on Complex Event Forecasting, Learning, and Analytics

[D5.2] Deliverable D5.2 Final Report on Explainable AI, Visual Analytics, and Augmented Reality

[D6.3] Deliverable D6.3 Final Report on the Dissemination, Exploitation, and Business Plan

1.2 Contribution and Structure of this Document

This document describes the system architecture, the system components, the system integration, and the released software stacks of the CREXDATA project. The document first describes the CREXDATA system architecture, system components, system integration approach and application programming interfaces (APIs) and then the three CREXDATA use cases, their requirements for the system architecture (e.g. with regards to various data types, data formats, and data volumes to integrated by the system), the use-case-specific system components for the use cases (e.g. various simulators), and the data processing flow and integration aspects per use case. At the end, the document provides a summarizing overview of the initial software stacks of the CREXDATA system and its components.

Chapters 1 to 3 are updated versions of the corresponding chapters in Deliverable D3.1 (Initial Report on System Architecture, Integration Approach, and Released Software Stacks). The main updates relate to updates to the system architecture. Chapter 4 is new and describes the system components and their integration. Chapters 5 and 6 have been significantly extended, especially related to system components and software stacks released by the CREXDATA project.

1.3 Target Audience

- CREXDATA technology partners (“developers”)
- CREXDATA use case partners
    - (end users)
    - technology partners providing demonstrator components
- CREXDATA project reviewers
- Third party researchers in system- and application-oriented research domains, especially
    - software architecture and system design
    - system design for extreme-scale data stream processing and predictive analytics
    - software architecture and system design for distributed federated data stream processing and federated machine learning
    - handling of weather emergencies
    - handling of health crisis
    - maritime event detection and handling
    - security
- Third party software architects and software developers of software systems for the above domains

1.4 Glossary

Abbreviation	Expression	Explanation
UC	*Use Case*	Applications of CREXDATA technology resp. the CREXDATA system in real-world scenarios. Within the project three use cases are defined: weather induced emergencies, health, and maritime.
	Pilot (site)	Conceptual term to describe a set of stakeholders within their context like spatial environment, equipment, data sources etc. For each Use Case, several Pilot (sites) can be specified (for instance, Dortmund and Austria in the emergency use case).
	Application Scenario	Procedural and structural description of potential uptake of CREXDATA technologies in Use Cases (for instance, flooding and forest fires in the emergency case).
	CREXDATA System	Output of WP3, integrating technologies created in WP4 and WP5 without use case-specific customizations. It includes customization and configuration functionality, esp. through graphical workflow management.
	Demonstrator (System)	Technical system based on the CREXDATA system, which is customized and configured for specific Use Cases, Pilots and/or Application Scenarios. The Demonstrator might include additional components both as data sources and sinks (for instance, legacy systems of end users or the ARGOS system in the emergency use case).
	(Data Processing) Workflow	A Data Processing Workflow (or short Workflow) describes the steps to be performed from data ingestion from the data source systems (e.g. sensors, IoT and edge devices), data processing, data analysis, model generation with machine learning (if applicable), predictions and forecasting, to model deployment and visualization. Workflows can be designed and edited visually in the Workflow designer/editor.
	(Data Processing) Operator	A step in a Data Processing Workflow is called an Operator. Operators are functional modules or building blocks.

2 CREXDATA System Architecture

This chapter describes the conceptual system architecture, its layers and components (Section 2.1). The more technology-focused Section 2.2 describes system integration and chosen platforms on which the CREXDATA system was implemented. Section 2.3 elaborates on the graphical workflow flow designer. The APIs (Application Programming Interfaces) for the seamless interconnection of all components and their integration is described in Section 2.4. Domain specific simulators of the project's simulation HyperSuite are included in Section 2.5.

2.1 CREXDATA Concept and Reference System Architecture for Use Case Demonstrators

An initial system architecture was provided in the Description of Action (DoA) of the CREXDATA project [DoA, part B, p.16] and in CREXDATA Deliverable D3.1. The core CREXDATA system was developed and configured through an integrated development environment (IDE). It comprises application logics of all technologies that were subject of WP3 to WP5. Figure 1 provides an overview of the major elements of the CREXDATA system and a general view on interfaces (input and output).

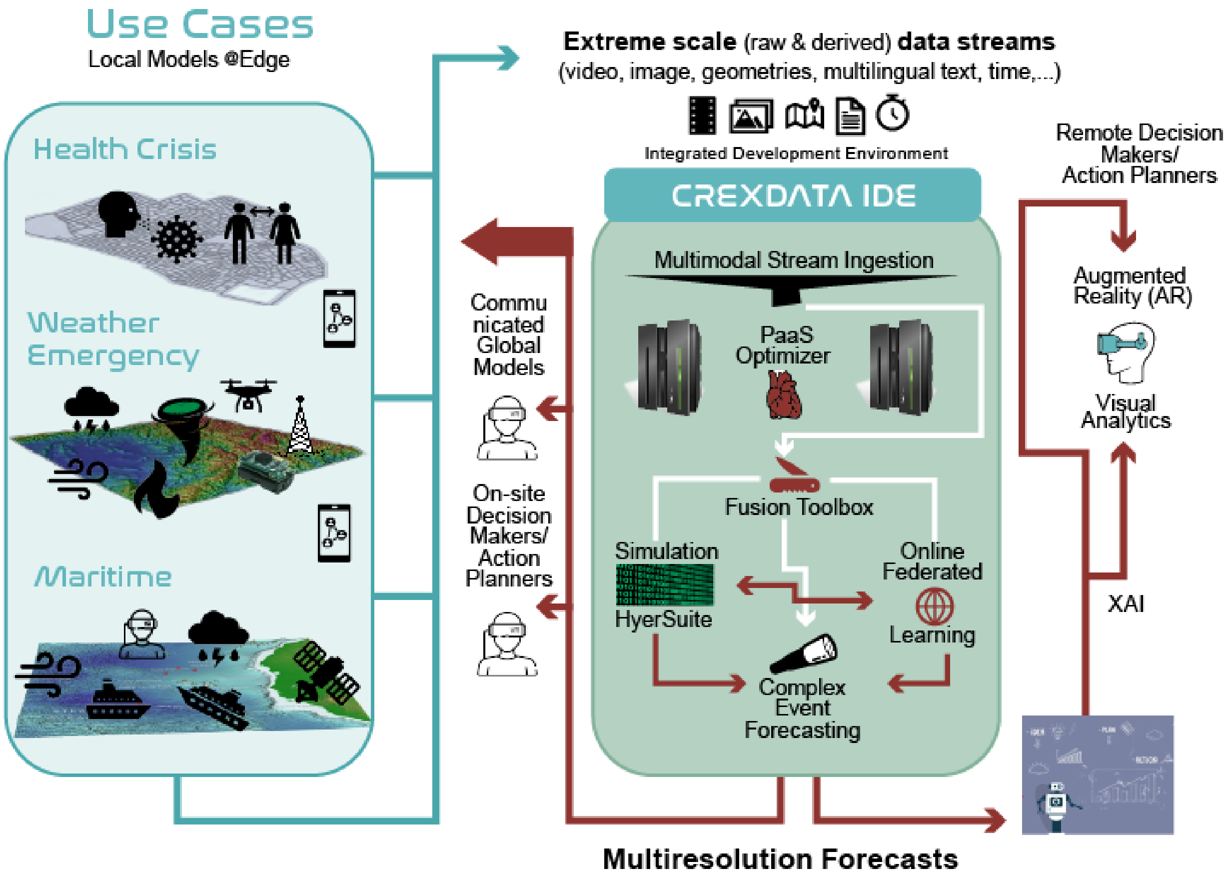


Figure 1: CREXDATA Concept Overview (www.crexdata.eu, based on [DoA, part B, p.16]).

The core CREXDATA system was integrated in work package WP3. The resulting system itself was made available for integration in demonstrators per use case. Thus, it can be understood as a sub-system of these demonstrator systems. Demonstrator system architectures are described per use case in Sections 3.1.2 (weather emergency case), 3.2.3 (health crisis) and 3.3.4 (maritime use case).

The following layers can be identified in the CREXDATA core system architecture:

Multi-Modal Data Stream Processing

The CREXDATA system consumes data from a variety of data sources. For real-time data processing, streaming data from various, potentially distributed data sources needs to be handled and fused. Therefore, technical solutions like Apache Kafka or RabbitMQ, general data formats like JSON and XML as well as domain specific data schemas need to be considered. Use case-specific data sources are identified in Section 3 and detailed in Deliverable D1.2. In CREXDATA, these requirements were implemented in a Multi-Modal Data Fusion Toolbox and by the Altair RapidMiner Data Science and AI Platform via data access modules (RapidMiner operators, data connectors, and project-specific extensions) in combination with use-case-specific systems (e.g. ARGOS in the CREXDATA weather emergency use case), leveraging Kafka for the interaction between system components and for data streaming between system components. Altair RapidMiner AI Studio (desktop) with CREXDATA-specific extensions were developed for visually designing the data processing workflows, which are then executed on distributed servers and compute clusters (leveraging e.g. Altair RapidMiner AI Hub (server or cloud instances)) as well as distributed and deployed on IoT and edge devices (using Altair RapidMiner Real-Time Scoring Agents (RTSA) at the edge, on IoT devices). Key components of the multi-modal data stream processing system architecture:

- Multi-modal Stream Ingestion Modeler (offline): Graphical editor for visual data processing workflow design and modelling, operators from data stream ingestion through data analysis to visualization and forecasting are needed. Therefore, extensions were implemented for data sources (per use case), for demonstrator system elements (like algorithms implemented in ARGOS in the CREXDATA emergency use case), and especially for CREXDATA algorithms developed in WP4/WP5.
- Prediction-as-a-Service (PaaS) Optimizer: Operators representing Prediction-as-a-Service (PaaS) functionality can be included in logical workflows that specify the application logic, but are deprived of physical execution details. The Optimizer receives such logical plans and outputs physical plans, assigning workflow operators and respective computations across available resources (e.g., from drones to base stations to cloud services in the emergency use case).
- Fusion Toolbox: Toolbox capable of executing entire data processing workflows and models specified by the Multi-Modal Stream Ingestion Modeler in real-time based on streaming data.

The definition of operators (functional modules and data connectors) was both driven by use cases (data sources, demonstrator elements) and technology developments (WP4-WP5). For modularity and loose-coupling purposes, there is no algorithmic interdependency of PaaS optimizations with other CREXDATA technologies, but workflows are seamlessly optimized and distributed across different computational devices/services to enable distributed data processing (e.g. for federated machine learning) and distributed use-case scenarios.

Simulation HyperSuite

The Simulation HyperSuite was created with the intention of a domain-specific integration of relevant simulators. Thus, it can be seen as a frame for use case-specific simulator sub-systems extensible to other domains. The Simulator HyperSuite was developed in WP2, providing data to algorithms developed in WP4-WP5. Details are provided in Section 2.7.

Machine Learning

Detailed use cases of Machine Learning (ML) typically require streams for training, besides inference, purposes. Regarding stakeholder requirements elicitation, a guiding question is “Are there situations in which you think that data is available, but so far there is no way to make sense out of it?”. For training models, there needs to be an option to create or acquire labelled data streams (ground truth).

- Online Federated Machine Learning: “online” and “federated” are two attributes that focus the solution space with regards to ML in CREXDATA. Sequential steps can help requirements elicitation, from general relevancy of ML to online ML, and then even federated ML.
- Interactive Learning for simulation exploration: Simulators are part of demonstrators in all three use cases. In contrast to ML models which need to be built via online, federated training, simulation models are prebuilt on transparently known mathematical principles and methodologies, but need to be guided towards an optimum to identify best fitting parameter sets. In CREXDATA, ML-based algorithms are integrated to guide and reduce the efforts for such an exploration of the parameter space. Thus, relevant simulation types were identified, and the principles of parameter exploration were determined.

These ML algorithms are not dependent on other algorithms in WP4. Interactive learning regarding simulations needs to be aligned with the Simulation HyperSuite developments to cover relevant simulations.

Complex Event Processing

Complex Event Processing (CEP) includes both, Complex Event Recognition (CER) and Complex Event Forecasting (CEF). Based on an input data stream, events are recognized and forecasted based on, for instance, logical expressions. An event might mean a single point in time (e. g., threshold at river gauge reached) or it might have a duration (e. g., person endangered by rising river gauge). Event patterns are either learned from data (cf. ML) or defined based on semantics of a domain of interest (i.e., the three use cases).

- CEF: Event patterns are modelled by logical formulae and transition systems. Simple events represent events close to data sources, while complex events hold as soon as relevant simple events occur. Inputs to event forecasting need to be provided as multi-variable time-series data, for instance, through a single Kafka topic.
- Text Mining: In the context of CREXDATA, text mining is understood as a synonym for Natural Language Processing (NLP). For instance, RapidMiner includes a Text Processing operator. The intention is to learn from social media streams or to detect events in social media streams. As a rough differentiation, both identification of highly relevant postings and analysis across a large number of postings are relevant. While in the first case, input for CEF might be generated, in the latter one this might be coupled with simulation exploration (verifying simulated futures).

CEF can be based on data streams created by other CREXDATA components (e.g., online federated learning and especially text mining) but does not need such interdependent setups.

Explainability and Visualization

Explainable Artificial Intelligence (Explainable AI (XAI)) and Visual Analytics require models and data as input to extend them with explainability and visualization layers. Thus, these algorithms are strongly dependent on ML and Evolutionary Programming (EP) algorithms.

- Explainable AI: A focus was set to ML models as input, independent of specific use cases (online, federated, interactive). The intention was not to implement own ML models, but to extend existing ones.
- Visual Analytics: Prerequisites are similar to XAI. For Visual Analytics, event streams (recognized and/or forecasted) are of similar interest like ML models. There is a direct interconnection with XAI models.
- Uncertainty Visualization: Situational awareness in decision-making situations includes an understanding of uncertainty in available data resp. information. For human actors, visualization of uncertainty semantics is required. Such semantics can be based, for instance, on established concepts of data/information quality dimensions, criteria and indicators (Data Quality/Information Quality (DQ/IQ)).

Algorithms adding explainability and visual features require ML and/or CEP models as a basis. Therefore, the system architecture requirements target the cascade from data sources through ML/CEP to XAI/Visual Analytics.

Augmented Reality

Augmented Reality (AR) in CREXDATA was tested by Head-Mounted Displays (HMDs) like HoloLens 2. AR provides very special means to visualize data and information. It is based on the principle that reality is enriched by superimposed information (i.e., not only virtual objects like in Virtual Reality/VR). Therefore, specific use cases need to involve stakeholders close to an environment or to objects of interest. Information is visualized with reference to such real-world perceptions. AR is implemented in terms of AR applications (e.g., using the development environment Unity3D). Interfaces to other services (i.e., algorithms) need to be incorporated into such standalone applications.

AR is relevant in very specific situations where real objects are of interest. In CREXDATA, AR is meant to be one possible User Interface (UI) to access results from data processing pipelines. Thus, AR might be added as a sink in graphical workflows, visualizing results from ML, CEP, XAI and Visual Analytics.

Figure 2 provides a detailed outline of the above discussion and specific component interconnection in a flowchart-like representation.

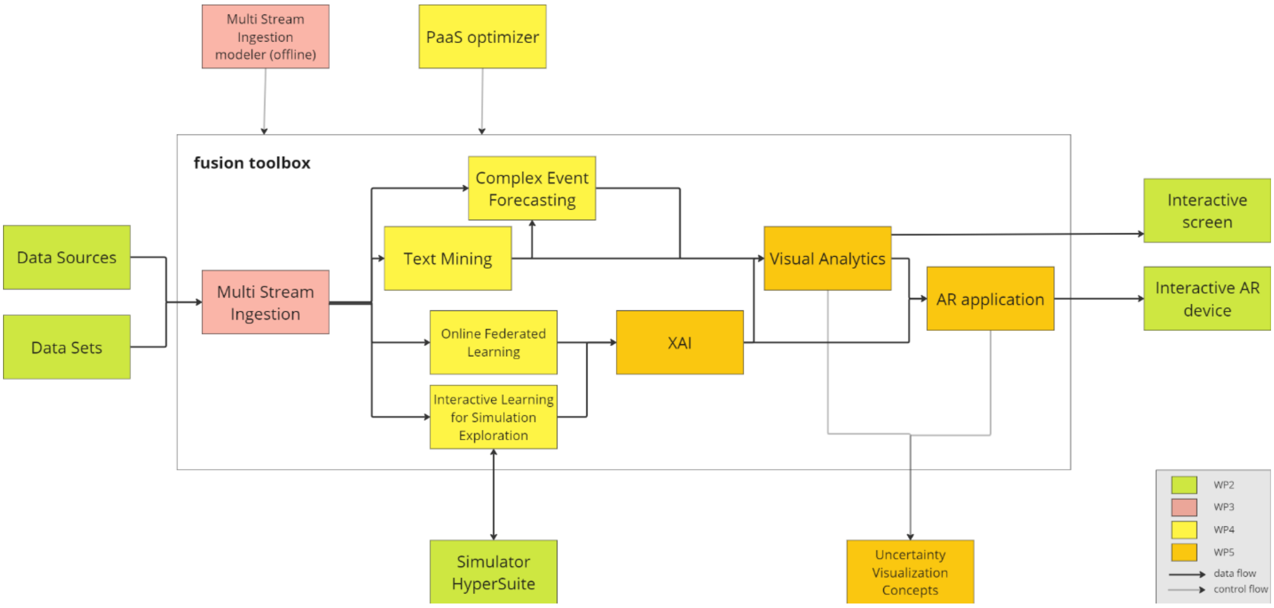


Figure 2: Elements of the CREXDATA system (integrated in demonstrator systems).

The following diagram visualizes the work package (WP) level interaction in CREXDATA:

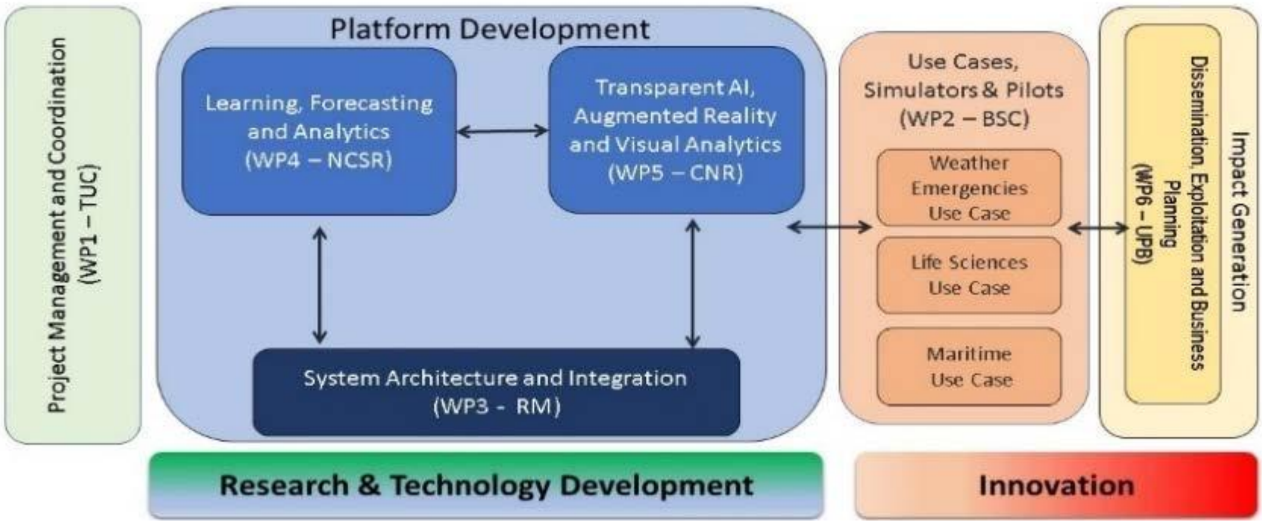


Figure 3: CREXDATA system architecture and integration with respect to work packages (WP) and WP level interaction (from [DoA] CREXDATA Project Proposal, part B, page 30, figure 3).

2.2 CREXDATA System Integration Architecture

This section describes the CREXDATA system integration architecture and the technologies and platforms chosen for the implementation of the CREXDATA prototype system.

The main objectives for the CREXDATA architecture were

- enable graphical data processing workflow design and automated deployment
  - low / no code workflow development through visual workflow design (ease of use by visual data processing workflow design without the need to code)
  - platform-agnostic workflows for supported streaming platforms (ease of use by abstraction)
  - automated deployment of the visually designed workflows on the execution platforms, including servers, clusters, cloud, edge and IoT devices (ease of use via automated distribution and execution)
- extensible design to integrate project-, use-case-specific components and tools
  - pluggable connections to data sources and sinks
  - ingest data from components into workflows
  - export data out of workflows (from intermediary or final step)
- enable extreme scale data analytics
  - supported backends (for stream execution or batch execution) are scalable and support distribution.

The CREXDATA system is able to integrate data in heterogeneous formats from distributed data sources, both static data as well as continuous data streams, process the data either on the edge/IoT devices or on a central server, compute clusters, or cloud or distributed and in parallel across different system components (e.g. for distributed data processing and distributed machine learning). Hence, we chose underlying platforms that enable the CREXDATA system to fulfil all these requirements.

- For the visual design of data processing workflows, we chose and extended the visual data processing workflow designer Altair RapidMiner AI Studio.
- For central data processing, we chose and extended the server version of the Altair RapidMiner data science platform, i.e. Altair RapidMiner AI Hub.
- For distributed and parallel data processing, we leverage a special version of Altair RapidMiner AI Hub optimized for low system resource requirements, low latency, and high throughput, namely Altair RapidMiner Real-Time Scoring Agents (RTSA), which can be deployed on distributed IoT devices, edge devices, remote servers, and distributed compute clusters.
- For the system integration, the system components use
  - web services for sending requests and responses (control flow) and
  - data streams and event streams (data flow).
- For the web services, Application Programming Interfaces (APIs) specify the exact format of the supported web services requests and responses in JSON format (see Section 2.6 for more details).
- For the event streams' and data streams' ingestion, we leverage the de-facto standard for stream ingestion which is Apache Kafka.

- For the simulations in the various use case scenarios, domain- and use-case-specific simulators were developed or extended and used (see Sections 2.7 and 3 for more details).

For the integration and interaction of the system components, we use web services as Application Programming Interface (API). The web service requests and responses and their JSON formats are described in more detail in Section 2.6.

For exchanging data between the system components, we use Kafka data/event streams. Data sources (data producers) write their data (e.g. sensor data from IoT devices or the other edge devices) into Kafka streams, from which other system components can read these data streams (see Figure 4 and Figure 5 below) and write their responses to other Kafka streams (e.g. events detected or predicted by automated complex event detection and prediction (CEP), weather or flood forecasts, maritime vessel collision predictions, etc.).

The following diagrams (Figure 4 and Figure 5) show the overall technical CREXDATA system architecture and system integration approach leveraging JSON-based Web Service APIs and Kafka data/event streams. They also show how the CREXDATA system components interact (via web service requests and responses) and exchange data (by writing data to and reading data from Kafka streams) and thereby how users can define data processing workflows using the CREXDATA system.

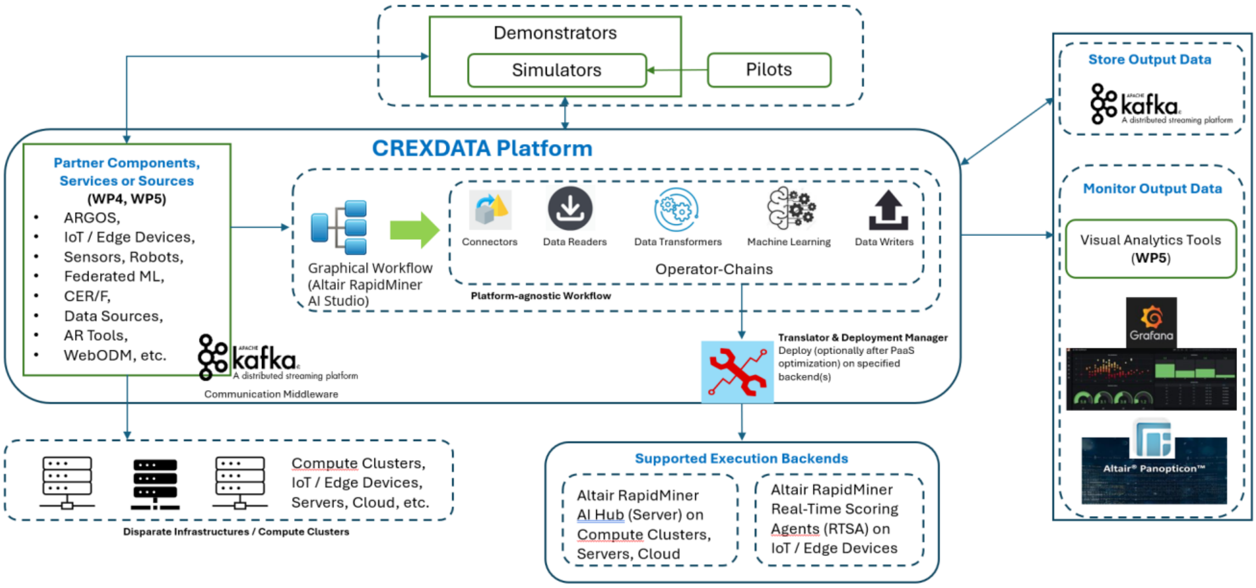


Figure 4: Technical CREXDATA system architecture and system integration approach via web services (control flow) and Kafka clusters (data flow, e.g. for sensor data streams from IoT devices and edge devices, as well as for stream of analysis results and predictions).

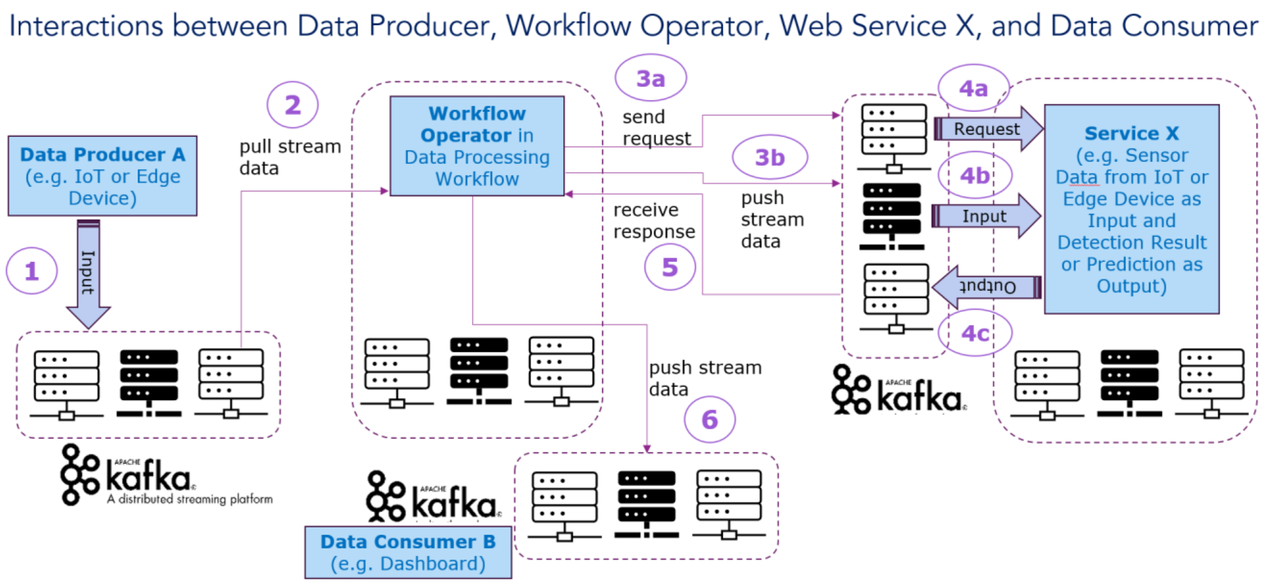


Figure 5: Interactions between data producers, workflow operator, web service, and data consumer: process control flow and data flows via web services and Kafka data/event streams.

The following sections provide more details about the APIs (Section 2.6), Kafka as communication middleware (Section 2.4), how the users can define data processing workflows using the visual workflow editor of the CREXDATA system (Section 2.3), and how the workflows can be applied on streaming data across different compute resources from edge to cloud (Section 2.5 and in more detail in Section 4.10).

2.3 Graphical Workflow Designer and Data Processing Platform

This section describes the graphical workflow flow designer of the CREXDATA platform, a graphical tool that enables non-expert programmers and workflow designers to visually specify complex data processing workflows. The graphical workflow designer is implemented as an extension of the graphical RapidMiner workflow designer (Altair RapidMiner AI Studio). The workflow designer supports the integration and fusion of heterogeneous data from dispersed data sources. In CREXDATA, we

- specified and implemented new and customized data processing and data fusion operators, where operators are functional modules or process steps represented as graphical elements (boxes) within the graphically represented data processing workflow, and
- specified processing tasks as data processing workflows (using operators), and
- these designed workflows are automatically translated to code that runs in the underlying CREXDATA architecture.

Altair RapidMiner is a unified data analytics and artificial intelligence (AI) software platform for all stages of data analytics – from data ingestion from heterogeneous data sources, data integration and fusion, data (pre-)processing, machine learning and modelling, model validation to model operationalization (deployment), and orchestration of data processing workflows, which can handle static data as well as continuous data streams and which can be executed in distributed processing nodes on servers, clusters, clouds, desktops, IoT and edge devices (see Figure 6 to Figure 11).

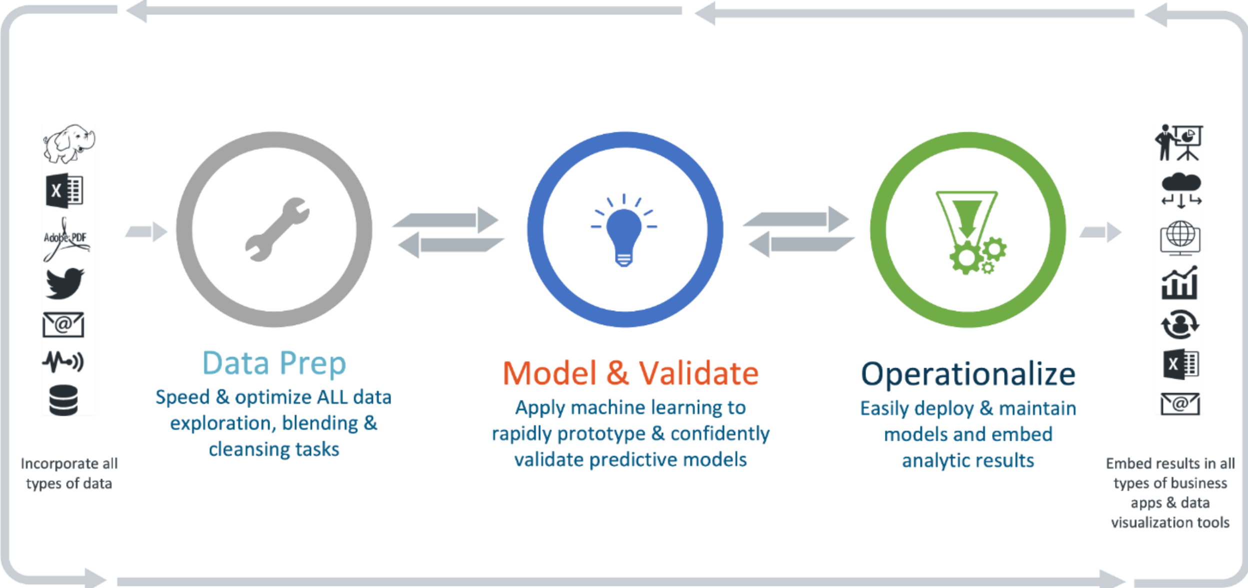


Figure 6: Altair RapidMiner platform as unified data platform from data integration from heterogeneous data sources, data (pre-)processing, modelling with machine learning, and model validation to model deployment (operationalization).

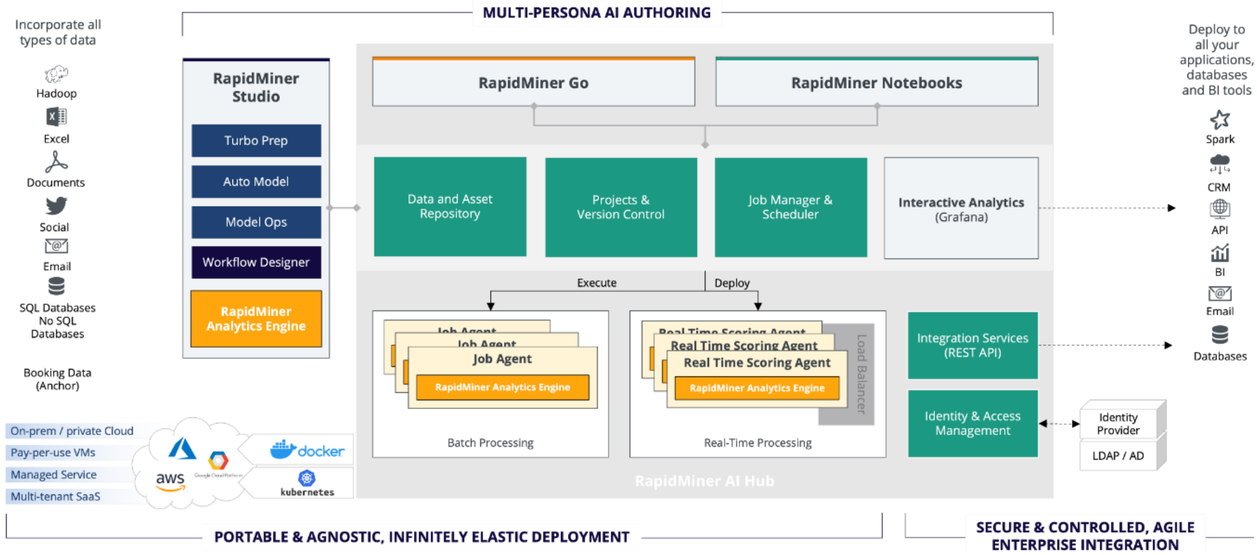


Figure 7: The RapidMiner platform overview including the visual workflow designer Altair RapidMiner AI Studio, the server Altair RapidMiner AI Hub, and the Altair RapidMiner Real-Time Scoring Agents (RTSA) for the process execution on the edge or on IoT devices.

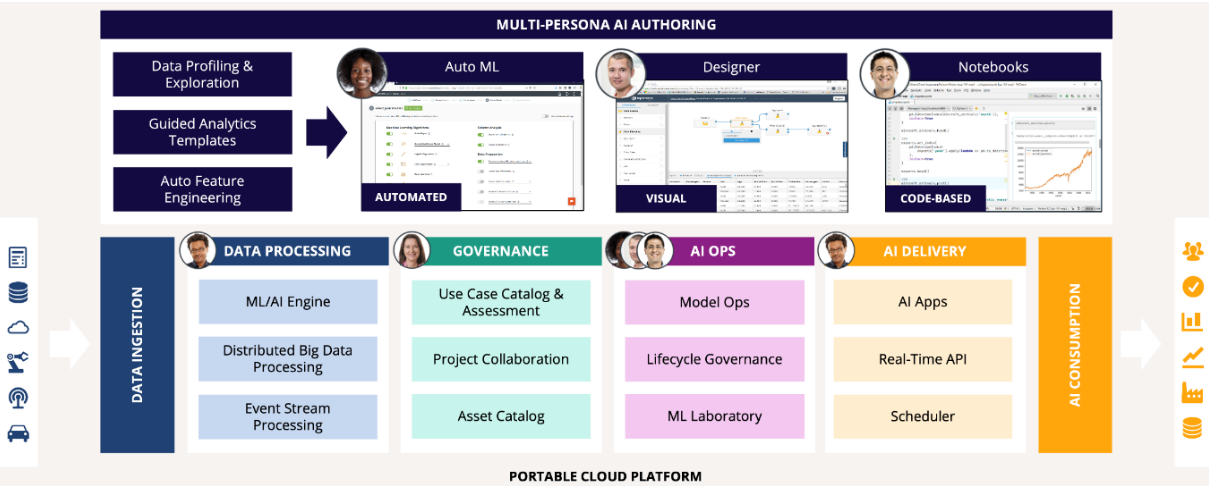


Figure 8: RapidMiner cloud platform architecture and server architecture.

Altair RapidMiner AI Studio's visual editor allows for designing complete data analytics workflows (see Figure 9). AI Studio supports a rich collection of over 100 machine learning (ML) algorithms and statistical functions out-of-the-box and is extensible using a plugin-based framework allowing to flexibly extend the platform and to integrate other tools and ML libraries. Process workflows are visually designed in AI Studio and can then be executed anywhere, locally on the desktop (in AI Studio), remotely on servers (on AI Hub) or in the cloud (on AI Hub) or on compute clusters or distributed on IoT or edge devices (on an RTSA). However, prior to CREXDATA this execution involved manual/human expert decisions and was restricted at the level of one or few operators.

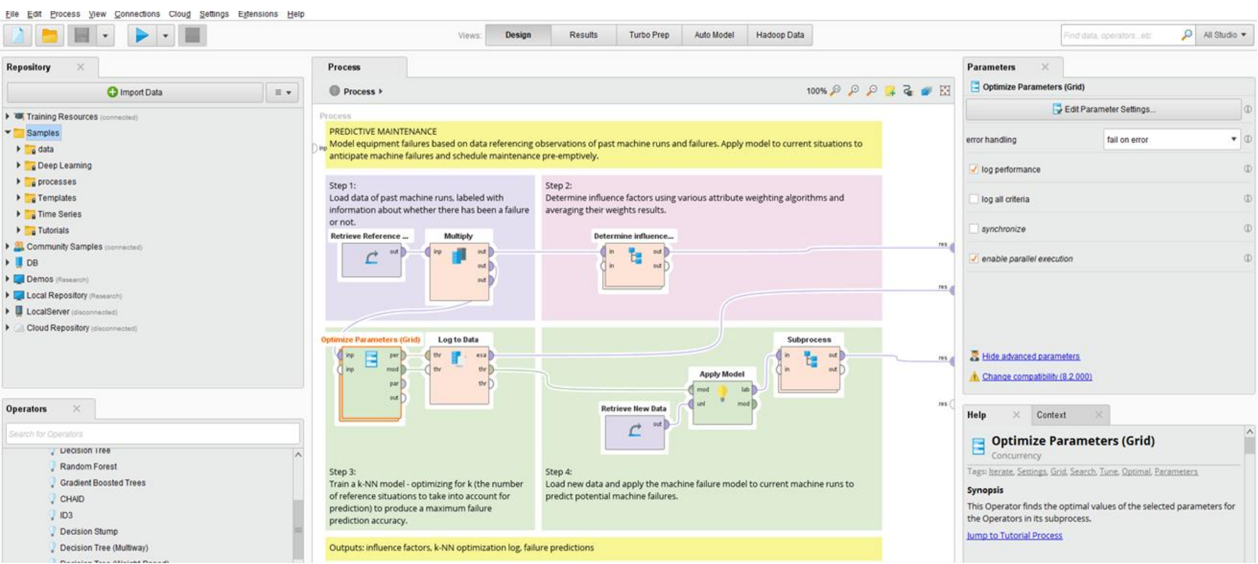


Figure 9: Graphical User Interface (GUI) of RapidMiner AI Studio for graphically designing data processing workflows with an exemplary visually defined data processing workflow (here for predictive maintenance, i.e. for predicting and avoiding machine failures before they occur).

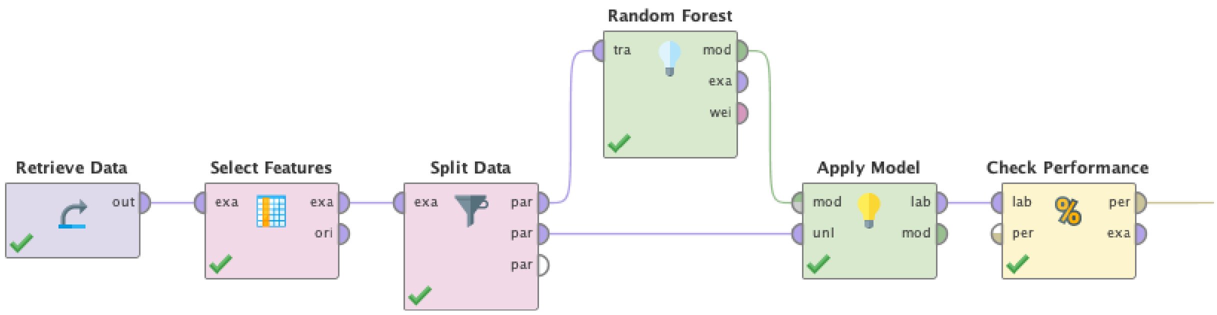


Figure 10: Exemplary RapidMiner data processing workflow for training, application, and testing (validation) of a machine learned model (ML model).

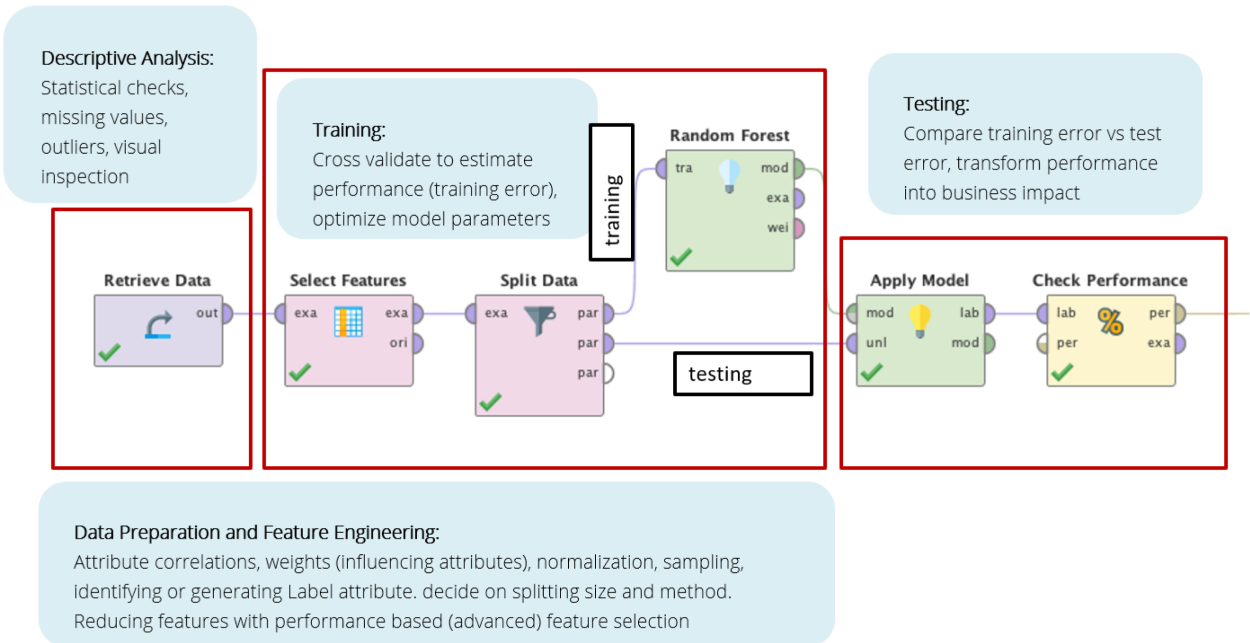


Figure 11: Exemplary RapidMiner data processing workflow for training, application, and testing (validation) of a machine learned model (ML model) with explanations for the process steps.

2.4 Kafka as Communication Middleware

Kafka is leveraged as a communication middleware to integrate external components like IoT and edge devices, sensors, event emitting servers, etc. Kafka data streams (topics) are used to pass data between the systems and their components (i.e. to read and write streaming data). The following figure (Figure 12) shows an example streaming analysis workflow to integrate external components via Kafka topics, i.e. for a fictitious 'critical action planning' scenario: plan a rescue action by interpreting current weather in context of historical risks:

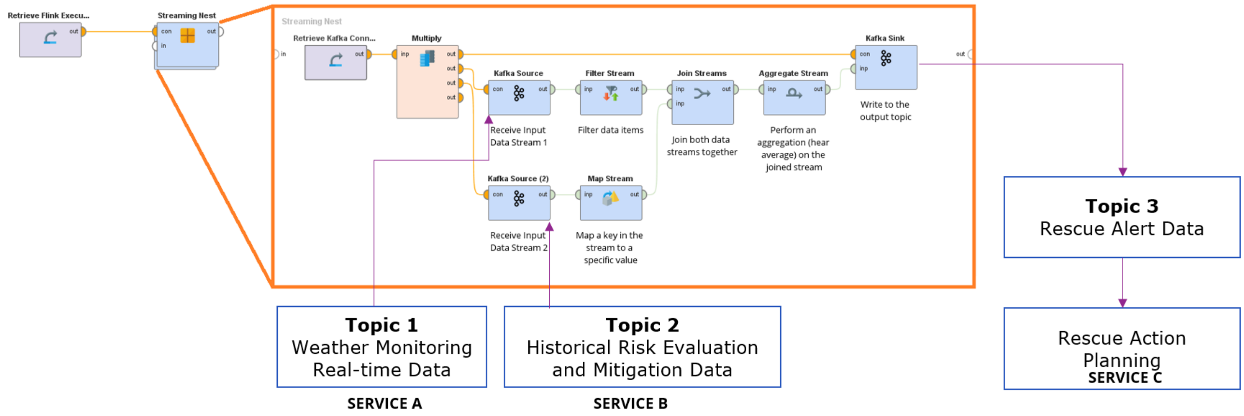


Figure 12: Example streaming data analysis workflow to integrate external components via Kafka topics, i.e. for a fictitious 'critical action planning' scenario: plan a rescue action by interpreting current weather in context of historical risks.

2.5 Process Execution from Edge to Cloud

The RapidMiner-based CREXDATA system architecture offers different approaches for executing data stream processing workflows on the CREXDATA platform, covering diverse compute resources from the cloud, compute clusters, compute servers to edge devices.

A first option, depicted in Figure 12, is via the RapidMiner operator "*Streaming Nest*", which translates visual data processing workflows into executable code and runs them on the big data stream processing platforms like Flink on compute clusters and/or distributed servers. The "*Streaming Nest*" operator, which was developed in the EU-funded project INFORE, automatically translates its inner (sub)process (workflow) within the *Streaming Nest* to Flink code and distributes this code to and executes it in parallel on big data clusters running Flink. This requires that every machine or device has Flink installed, which may be too resource-intensive for IoT and edge devices. It also only works for RapidMiner operators, for which automated translations to Flink exist.

The second option, described in more detail in Section 4.10, is via the newly developed RapidMiner operator "*Edge Processing Nest*" which allows for running big data stream processing workflows on remote RapidMiner Real-Time Scoring Agents (RTSA), e.g. on edge nodes or IoT devices, e.g. for distributed data processing or federated machine learning, without the necessity to install Flink on edge devices and without the need to translate visual workflows to Flink code and thereby allowing to execute any RapidMiner operator on the RTSA(s). More precisely, the "*Edge Processing Nest*" operator was developed in the CREXDATA project as part of the Edge-To-Cloud Extension (see Section 4.10). It significantly adds the option of distributing operators to Real-Time Scoring Agents, which require only limited resources on the target machines and devices, and which can execute every RapidMiner operator and process at fog or edge devices.

For CREXDATA, the "*Edge Processing Nest*" is a key enabler for cloud-to-edge deployments in order to exploit the processing power of the entire computing continuum. "*Edge Processing Nest*", combined with the workflow optimization facilities developed in the project, can push computation at the edge, alleviating network latencies and communication costs, and still keep heavy load computation at the cloud. The "*Edge to Cloud Optimizer*" determines the distribution of the (abstract) visual data processing workflow to the distributed (physical) compute resources (from cloud machines and compute clusters to IoT and edge devices) for parallel execution on these resources (see Section 4.10 for details) exploiting "*Edge Processing Nest*" to break up workflows to sub-parts destined to be deployed on various sites.

The following figure (Figure 13) shows the general interaction of the CREXDATA platform components from visual workflow design in RapidMiner AI Studio (desktop) via RapidMiner Server to the workload distribution on various RapidMiner Real-Time Scoring Agents (RTSA) at different edge sites:

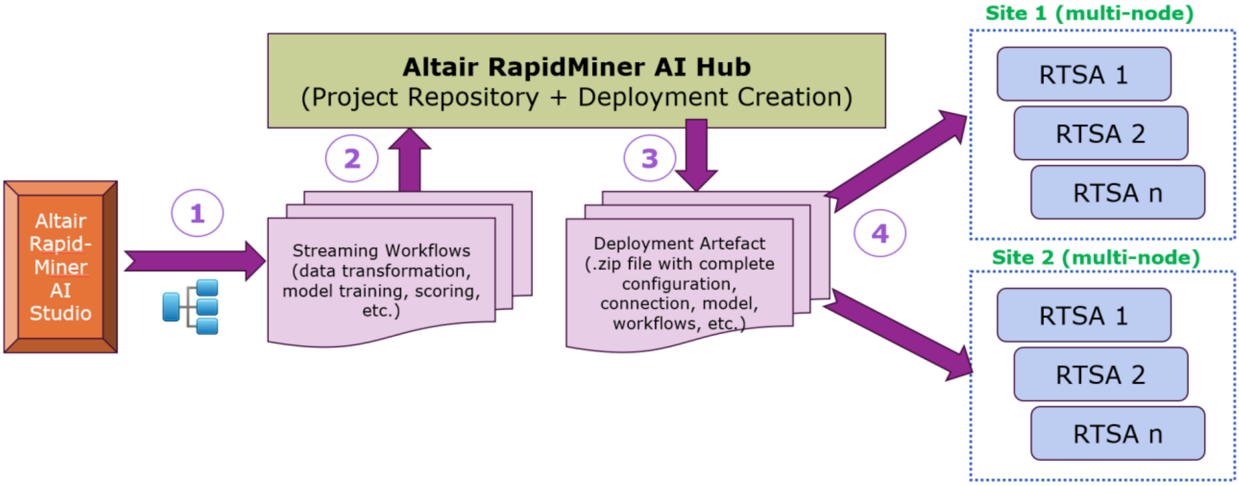


Figure 13: General interaction between the visual workflow designer (RapidMiner AI Studio), the data processing server (RapidMiner AI Hub), and numerous execution nodes (RTSA 1..n) at different edge sites, supporting executing processes at the edge, on IoT devices.

This interaction flow is for example used for the federated data processing use case with distributed IoT and edge devices. Altair RapidMiner AI Hub automates the deployment creation and dispatches the deployment artefacts to the different sites, where these artefacts are executed on Altair RapidMiner RTSAs. The RTSAs are preconfigured environments to support workflow execution, making connections between Kafka data sources and Kafka data sinks, and optionally can also run Python code (scripts) if needed.

The following figure (Figure 14) provides an example of a use-case-oriented data processing workflow, which was visually designed in RapidMiner AI Studio and could then be distributed to RapidMiner Real-Time Scoring Agents (RTSA).

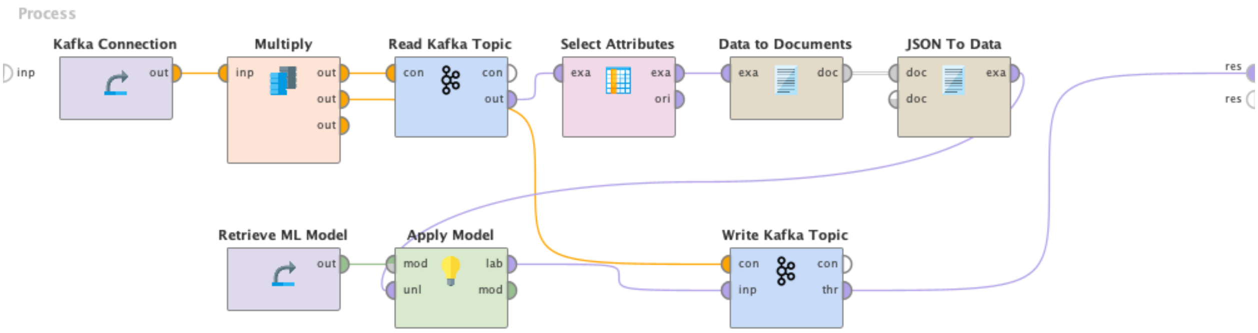


Figure 14: Use case workflow example for executing processes on RTSA node(s). This example workflow reads input data from a Kafka data stream (Kafka topic) using a pre-defined "Kafka Connection" and the "Read Kafka" operator, converts the data from JSON text format to a data format, retrieves a machine learned model (ML Model) and applies this model to the input data read from the Kafka data stream. The prediction result of this ML model is then written to the Kafka output data stream (Kafka topic) using the "Write Kafka" operator. This workflow is an example of a workflow that could be distributed to and executed on different machines and devices (e.g. IoT and edge devices or compute clusters or servers or cloud computers).

RapidMiner workflows can also be used for the infrastructure management processes and services, e.g. for defining and exposing web service endpoints. Infrastructure management workflows are exposed as RapidMiner AI Hub endpoints like shown in the screenshot of the AI Hub web service endpoint management user interface in Figure 15:

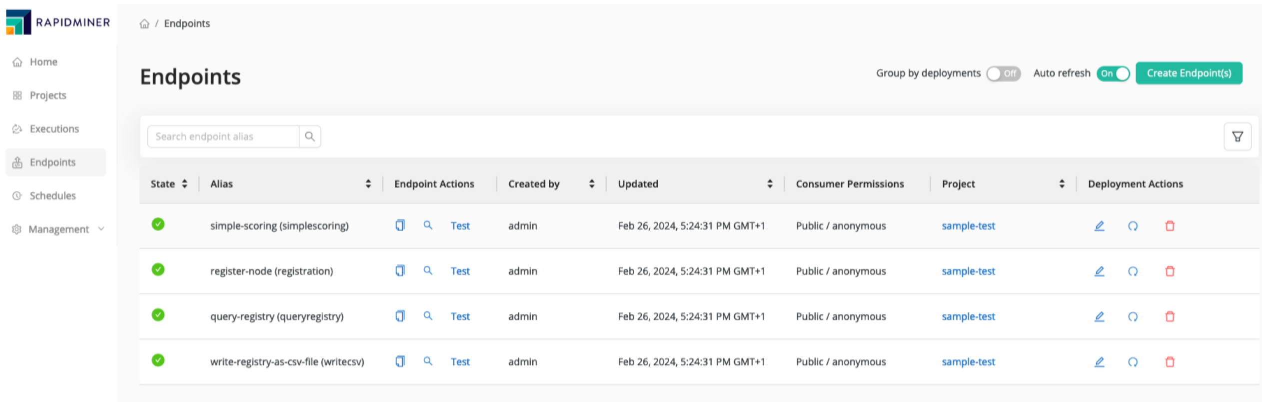


Figure 15: Infrastructure management workflows are exposed as RapidMiner AI Hub web service endpoints (screenshot of the AI Hub web service endpoint management user interface).

Each AI Hub web service is provided by an underlying RapidMiner process (workflow). Accordingly, each CREXDATA infrastructure management web service implemented as an AI Hub web service endpoint has an underlying infrastructure management workflow, which can also be visually designed like any other data processing workflow using AI Studio.

2.6 CREXDATA APIs for System Component Integration

The previous section described the technologies and platforms chosen on which the CREXDATA system is implemented. This section describes the APIs (Application Programming Interfaces) for the seamless interconnection of all components and their integration in more detail. For the integration, the CREXDATA system uses web services (in JSON format) and Kafka data streams (Kafka topics).

Since the CREXDATA use case scenarios involve continuous data streams, the CREXDATA platform needs to be able to not only handle static data or batch data, but also continuous data streams. Hence, the use-case-specific system components to be integrated into the CREXDATA platform also need to be stream-enabled. A blueprint for the API of each component is defined leveraging web services for the control flow and Kafka for the data (stream) exchange and for the control flow, using Kafka data streams (Kafka topics) for input data, output data, and supported requests for each component (in JSON format).

For the CREXDATA project, we followed the following approach to map a use case scenario to a data processing workflow:

- Use case scenario description as a sequence of steps involving system components:
    - Scenario tells a story as a sequence of steps.
    - Scenario involves 1 or more system components.
- Component to scenario mapping:
    - Component developers assist in deciding which scenario suits them.
    - Use case leaders help in this mapping exercise.
- Implement scenario as visual RapidMiner workflow:
    - Scenario defines interaction sequence.
    - Component API is stream-enabled (i.e. provides a standard mechanism to interface via web services and Kafka), so that it can be integrated into the CREXDATA platform (via web services and Kafka).
    - Use-case-specific operators are implemented in RapidMiner extensions.

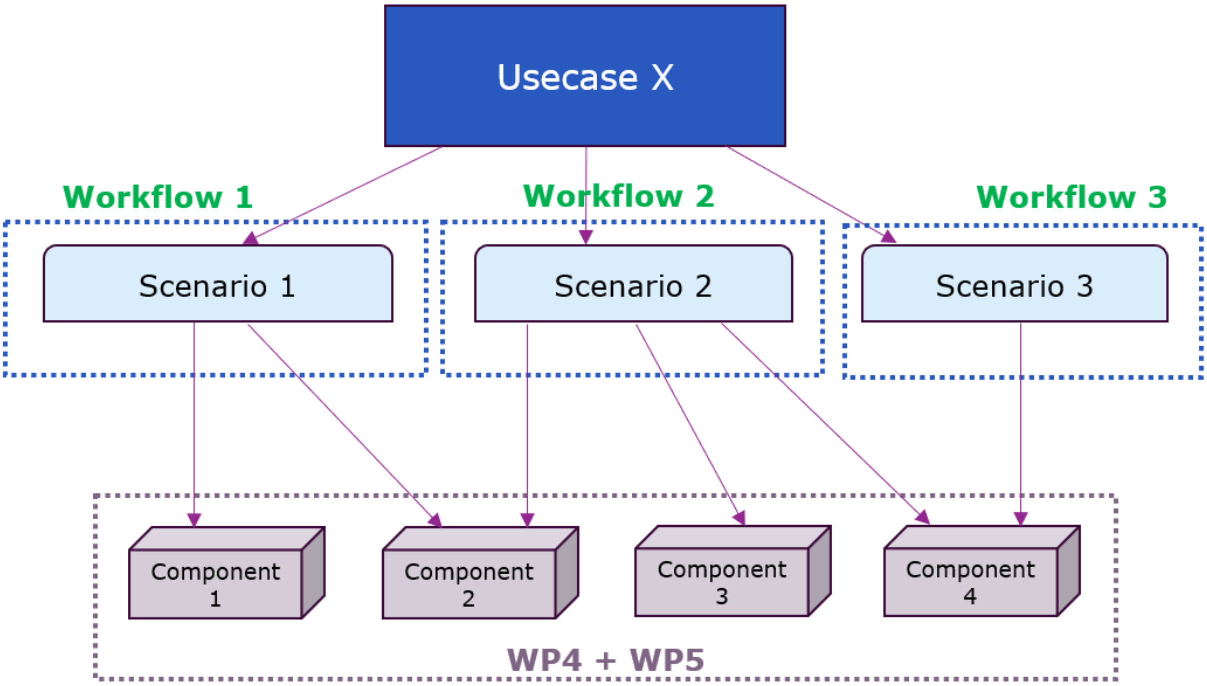


Figure 16: Approach to map use case scenarios to process steps and system components.

A CREXDATA reference API was used to stream-enable the use-case-specific system components and to define the web service interfaces (APIs). The reference API leverages web services, which process requests in JSON format, and Kafka data streams (Kafka topics). The following figures and paragraphs explain the structure of the API and its key parameters using a simple calculator as an example for illustration.

Following this CREXDATA system component reference API structure, APIs are defined for all use cases and for all integrated components.

The basic structure of the CREXDATA system component reference API describes the possible web service request(s), i.e. the method names (Request Topic) and their signatures, i.e. parameters, inputs (Input Data Topic), and outputs (Output Response Topic). Figure 17 shows this basic structure for a simple calculator web service example for summing up input data values.

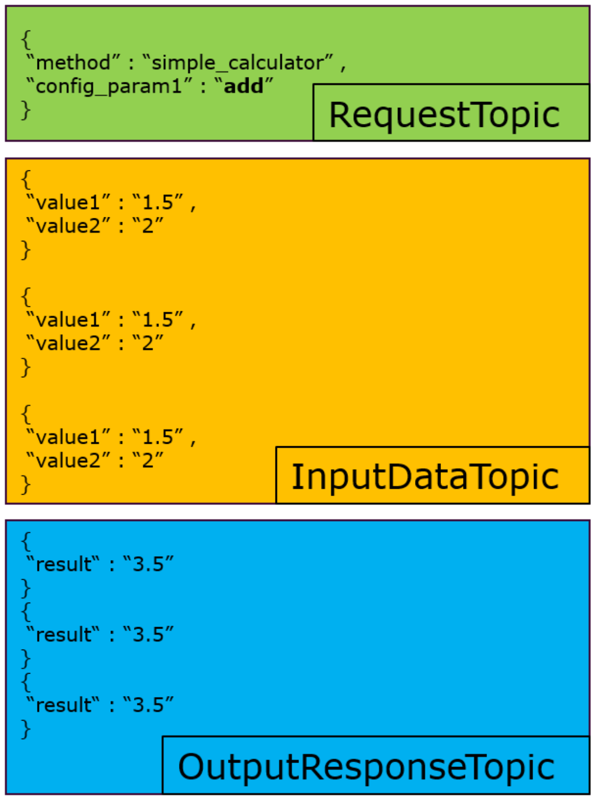


Figure 17: Basic structure of the CREXDATA system component API using web services (with data in JSON format) and Kafka data streams (Kafka Topics), here illustrated with a simple calculator example.

The CREXDATA system component API is implemented via web services (in JSON format) using Kafka data streams (Kafka topics) for web service requests, input data, and output data streams:

- Implementation of a basic reference service (job), here with
  - 2 Input Kafka Topics: RequestTopic, InputDataTopic
  - 1 Output Kafka Topic: OutputResponseTopic
- The service implements an API and exposes its signature:
  - Example Signature:
    - string simple calculator(string methodNameConfig)
      - methodNameConfig : **add, subtract, multiply, divide**
  - Behaviour:
    - request : read request in JSON format from RequestTopic
    - input : read JSON data to be used for this request from InputDataTopic
    - output : write result in JSON format to OutputResponseTopic
- Implementation of a client of this API to test the API method defined above:
  - Write JSON to Kafka topics using a Kafka console script or using the "Write Kafka Topic" operator in the visual process design GUI (RapidMiner AI Studio)
  - Read the method output using a Kafka console script or the "Read Kafka Topic" operator in the visual process design GUI (RapidMiner AI Studio)

In order to allow web services and input data topics with multiple input data streams (e.g. multiple sensors providing one sensor measurement data stream each), the above API is refined with a unique identifier for the data streams ("datasetKey"), so that each input data stream can be identified uniquely.

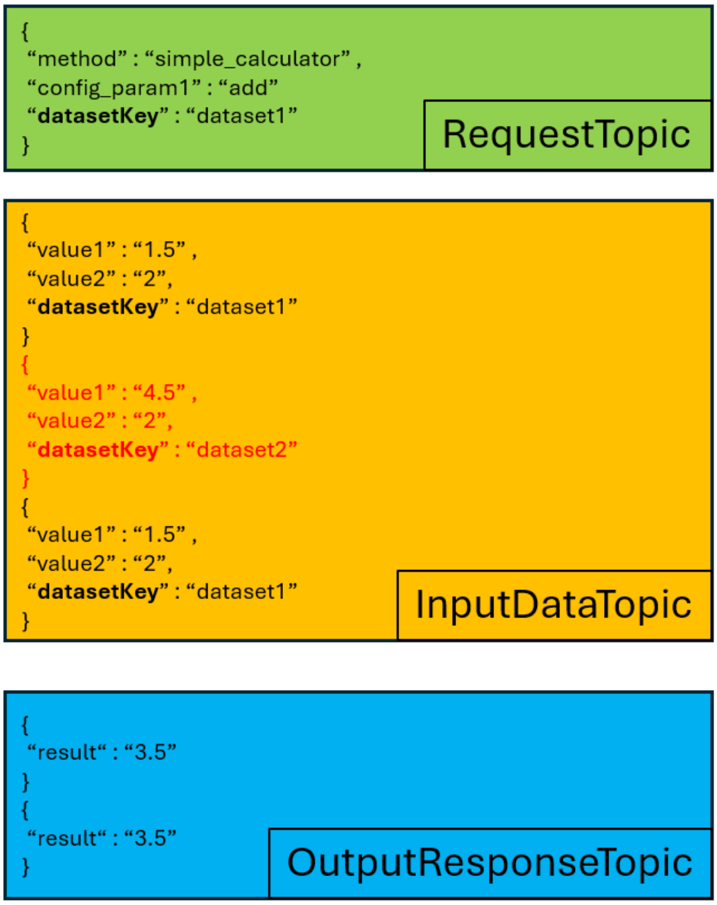


Figure 18: Refined CREXDATA system component API allowing multiple input data streams using data stream identifiers (i.e. a key called "datasetKey"). In this example only input data values from the data stream with the "datasetKey" value "dataset1" are considered for the summation of the simple calculator web service.

The refined CREXDATA system component API uses data stream identifiers (i.e. a key called "datasetKey") to support multiple data streams from different workflows:

- Different workflows write data on the same input data topic.
- To differentiate input data (events) coming in from different workflows:
  - **datasetKey**: add a key called "**datasetKey**" in both, request and input data (JSON payload).
- Behaviour:
  - A request is only processed for data that have matching "datasetKey".
  - This allows for multiple workflows to use the same backend service, where each workflow submits an add request for a different dataset, i.e. using different dataset keys.

In order to enable data processing workflows not only on single values from data streams, but also on time windows from data streams, the CREXDATA system component API was further refined to support time windows on the input data, either requiring a minimum number of events ("minEvents") (data points) or specified waiting time ("waitingTime") to complete a time window and to start processing the web service request:

- Some web services may need to process requests for a batch of data (time window).
- Hence, if less or more data comes in via the input data stream, the input data can be collected in batches (time windows) for a specified time or requiring a specified minimum number of events (data points).
- Collecting input data events by windowing by either requiring a specified minimum number of events per time window ("minEvents") or covering a "waitingTime" span:
    - minEvents: add a key called "minEvents" to the request to require a specified minimum number of events (data points) per time window (batch of data).
    - waitingTime: add a key called "waitingTime" to the request to collect all events (data points) within the specified waiting time into one time window (batch of data).
- Behaviour:
    - Collect events based on number of events or time passed parameters before computing the web service request (here the request "add").
    - Since the windowing may block the response from the web service (due to the waiting time until the specified time has passed or until the specified minimum number of events (data points) could be collected from the input data stream), appropriate values should be used to avoid undesired web service blockages.
    - In case of concurrent requests from other workflows, these other requests should be delegated to other task managers.

Figure 19 shows an example of a summation process for a time window of (at least) three events ("minEvents" : "3"), i.e. waiting until three input data values have been received from the input data stream (InputDataTopic), before computing the sum of these three input data values and writing the sum to the output data stream (OutputResponseTopic):

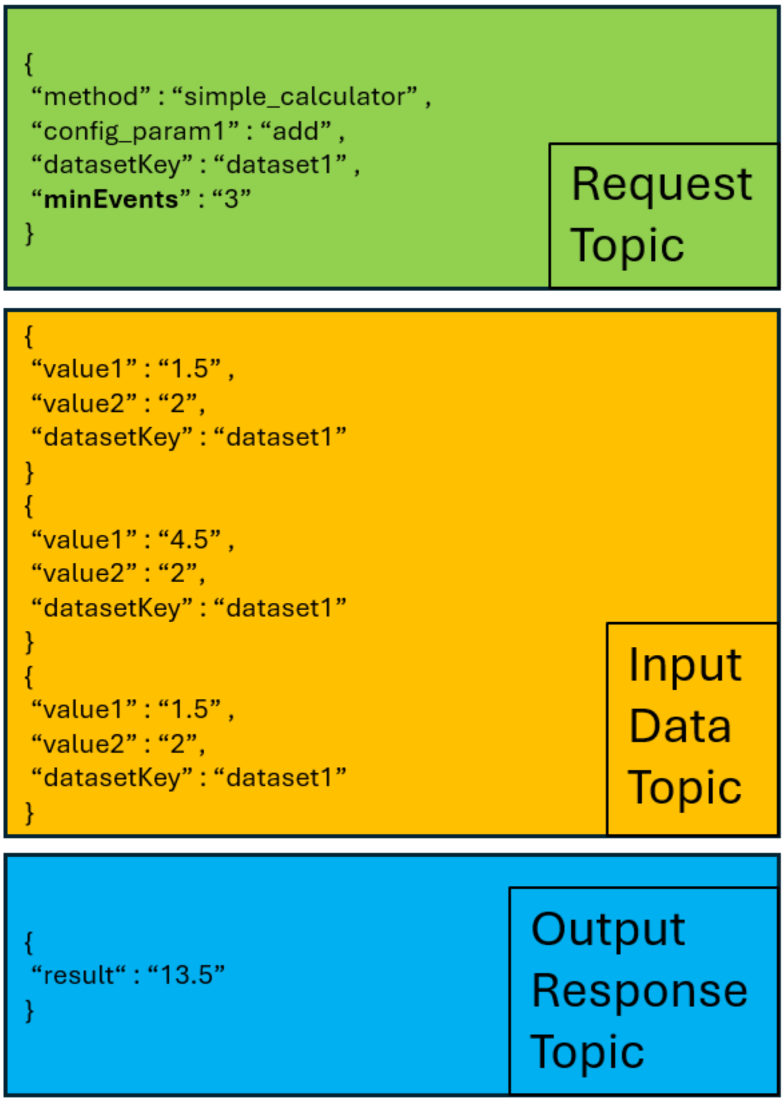


Figure 19: Refined CREXDATA system component API supporting time windows on the input data streams (InputDataTopic), either requiring a minimum number of events ("minEvents") (data points) or specified waiting time ("waitingTime") to complete a time window, to start processing the web service request, and to writing the result to the output data stream (OutputResponseTopic).

In order to avoid that web services produce output data *too frequently*, e.g. too much output data to handle for other components, the output frequency of a web service can be limited, i.e. specifying the frequency of the output generation in milliseconds between two consecutive outputs using the request parameter "outputFrequencyMiliSec":

- A web service may produce output data *too frequently*. In this case:
- Control the frequency of generating output data:
  - **outputFrequency**: add a key called "**outputFrequency**" to limit the frequency with which output (response) events are generated.
- Behaviour:
  - A service may expose method configuration on the frequency of its output.

- This could be suitable for services which would otherwise generate output data too frequently, especially if the output data is too large.

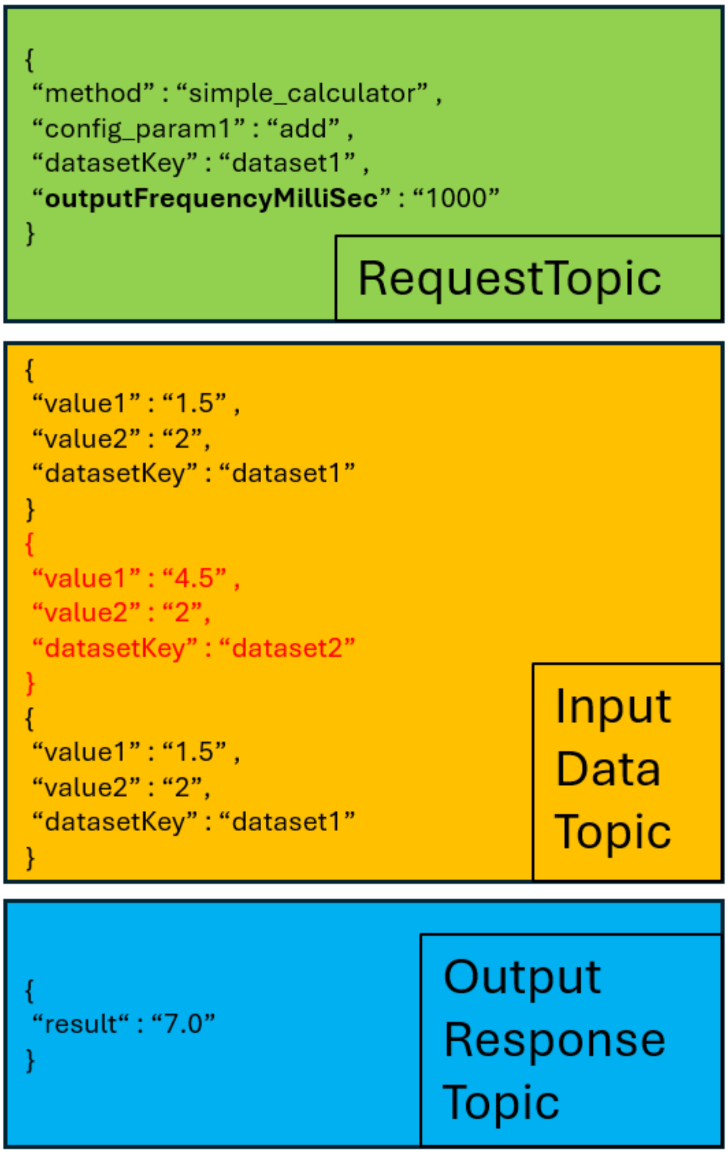


Figure 20: Refined CREXDATA system component API allowing to limit the output frequency of a web service via the parameter “outputFrequency” specified in milliseconds.

The CREXDATA system component API was further refined to allow several clients (CREXDATA system nodes or web services) to use the same datasets (e.g. input data streams for their process inputs and output data streams for their process outputs (Kafka response topic)), request identifiers (“requestID”) can be used to identify which web service output value in the output data stream corresponds to which web service request:

- In cases where different clients (CREXDATA system nodes or web services) use the same input and output dataset(s) and where most configuration parameters for the web service requests are also the same, it is necessary to be able to differentiate which web service outputs belong to which of these different web service requests.

- Adding a **requestID**: Instead of adding all web service request parameters to the output data stream, just a key called "**requestID**" is added in the request and in the output event(s).
- This requires each client to do book-keeping of its requestID(s) in a separate file.

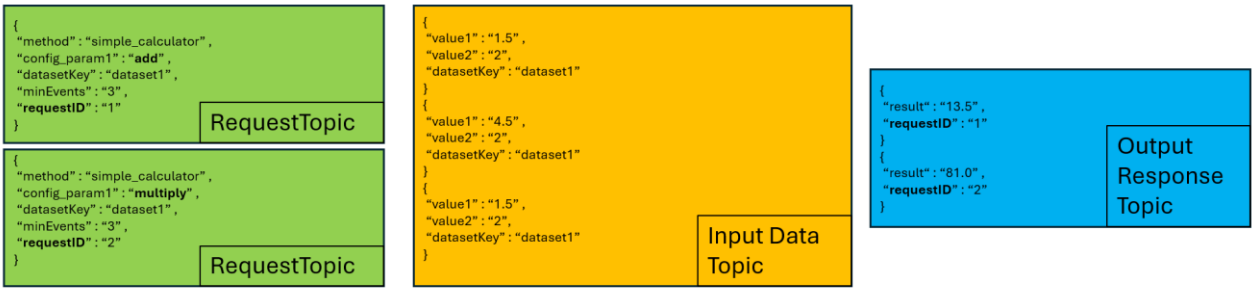


Figure 21: Refined CREXDATA system component API allowing multiple clients to use the same input and output data streams, leveraging request identifiers ("requestID") to specify which web service output value corresponds to which web service request (client).

The data in the web service input data streams and output data streams must be formatted in the following JSON format to the processable by the Altair RapidMiner Data Streaming extension:

Data type and format:

JSON formats are supported where the format of each datapoint is non-hierarchical. JSON objects should look like the following:

```
{key1: value1, key2: value2, key3: value3}
```

Concrete example:

```
"{
        "time": "02/19/2019 06:07:14",
        "StockID": "ForexALLNoExpiry",
        "price": "110.11",
        "streamID": "7040"
}"
```

Importing data from the system components / Altair RapidMiner AI Studio into Kafka:

Altair RapidMiner has strong data integration capabilities (Extract Transform Load (ETL). Altair RapidMiner can read in data from a variety of file formats (like CSV, XML, JSON, Excel, text files, PDF documents, etc.), reshape them (header adjustment, etc.) and transform them into the RapidMiner format. Datasets in RapidMiner are called ExampleSet (e.g. training examples for a machine learning algorithm). RapidMiner can then insert an ExampleSet directly into a

Kafka topic (data stream) in the supported JSON format (shown above). For each CREXDATA system component to be integrated into the CREXDATA platform (and hence with Altair RapidMiner), a sample of their input data is used to implement an ETL pipeline (automatable data processing workflow) to retrieve data (which can also read files from folders, etc.) and to populate data in a Kafka topic (e.g. output response data stream).

For the CREXDATA project and its use cases, additional data connectors are implemented as RapidMiner extensions to support the required additional data formats (e.g. for images and videos as well as for the use-case-specific simulators and system components).

Note:

- This standardization of the JSON data formats for the integration of the components is necessary for the data streaming extension operators to do their job in a common way.
- Once a component/simulator/etc. receives output from a RapidMiner workflow, it can internally of course arrange the flat JSON into a complex hierarchical format as required internally.

2.7 Domain-Specific Simulators

Simulation models and tools were developed in Task T2.4 specifically per Use Case, as they usually represent domain specific phenomena. From a technical perspective, simulators are components which are part of the Demonstrator systems developed per Use Case. They are integrated with the CREXDATA system based on the overall system architecture of WP3, incorporating the research outcomes of T2.1-T2.3.

2.7.1 Simulator for the Weather Emergency Use Case

For the weather emergency case, no integrated solution for the different types of “simulation” was available. By nature, very different influence factors need to be considered in an emergency. In general, a separation is made between own and foreign situation. Resources are deployed (own situation) to cope with natural and man-made events evolving into an emergency to be mitigated (foreign situation). Different approaches of simulation can be recognised, but also very different understandings of the term “simulation”. In the CREXDATA Simulation HyperSuite for emergency management, the following elements were or can be integrated:

- Weather related simulation: Typically, in meteorology there is no simulation as such, but nowcasting and forecasting. As a very specific phenomenon, fires depend on a trigger event that starts the fire (see, e. g., the Propagator forest fire simulation model [10]). Similarly, flood simulators (e.g. DeepWaive by FloodWaive) model the actual flow of water in urban surroundings or even the diffusion of water into buildings.
- Weather data archives: Additionally, there are large archives accessible through data services to load satellite images and forecasts from a certain point in time and location in the past (for instance, flooding events in Germany in 2021). So, instead of simulating a certain weather condition, it might be relevant to just select a weather situation in the past. This complies with typical behaviours of experience-based decision-making in the domain of emergency management.
- Robotic simulation models: Physical robots can be extended by digital twins in a virtual environment. By simulating environmental effects in such environments like Gazebo,

"what-if" scenarios can be tested in such a safe space. By incorporating parameters representing the actual environment such as wind speed relevant for UAVs measuring flood levels, action planning regarding deployment and routing of drones can be supported.

- Domain specific simulation models: In requirements elicitation sessions, mainly two types of simulation models were indicated: a) people movements and b) spread of hazardous goods. People movement is relevant both for understanding a situation which is not controlled by Public Protection and Disaster Relief (PPDR) organisations, as well as situations in which evacuation is conducted under control of authorities. Spread of hazardous goods refers to information that is required to plan actions especially regarding spatial parameters. For instance, a contamination might be transferred through the air or, in case of a flooding, through water.
- Event injection: There are several approaches in the field of emergency simulation used to train emergency management staff where scenario editors are used to "inject" incident data into simulated or real-time data streams. In case data from the field is missing, it could be injected through such kind of a tool. If this is done, corresponding uncertainty needs to be considered.

For such an integration, a core component is required that provides functionality for the orchestration and configuration of multi-domain services like those mentioned above.

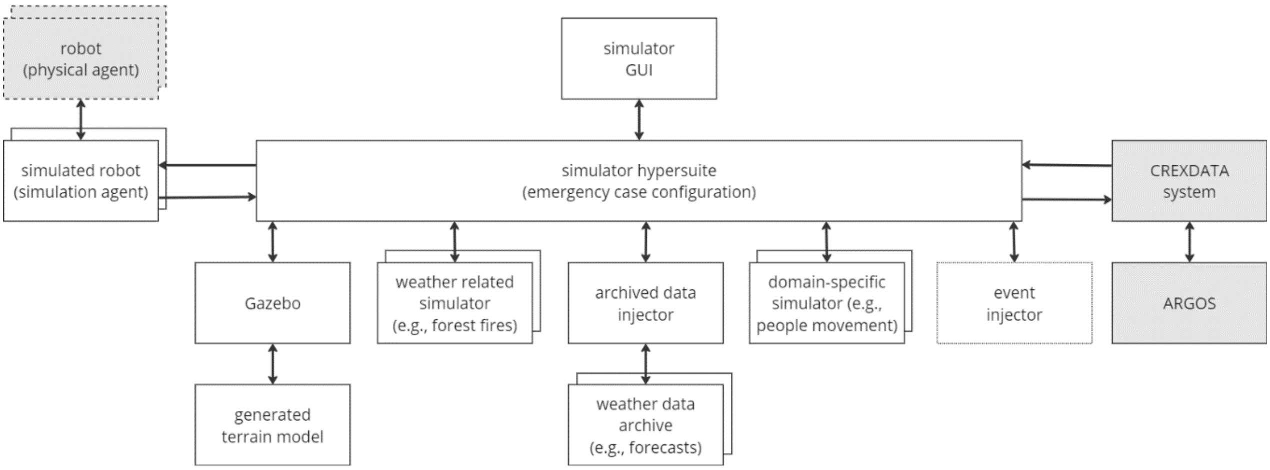


Figure 22: Architecture of the simulator system for the weather emergency case (Source: CREXDATA Deliverable D2.1).

2.7.2 Simulators for the Health Emergency Use Case

The health case is separated into two scenarios, each requiring specific types of simulation models. For the epidemiological scenario, the MMCACovid19 simulator was adopted [11]. For the multiscale lung infection scenario, the Alya [12] and PhysiBoSS simulators were coupled [13, 14].

2.7.2.1 Simulator for the Epidemiology Scenario

The *MMCACovid19* simulator is a flexible package written in Julia language that allows simulating the spread of a disease in a metapopulation with multiple agent types. The package implements a general Susceptible-Exposed-Infected-Recovered (SEIR) model and simulates an epidemic process using the Micro-Markov-Chain-Approach. Moreover, the simulator incorporates a wide range of data, including population demographics, epidemiological parameters, population mobility, healthcare system capacity, and intervention strategies, including confinements, mobility reduction and vaccination. Importantly, MMCACovid19 can be extended beyond COVID-19 to encompass general epidemiology by performing a calibration of the epidemiological parameters. Overall, the MMCACovid19 simulator serves as a powerful tool for decision-makers, public health officials, and researchers, enabling them to simulate, analyse, and optimize strategies in response to health crises like the COVID-19 pandemic. By providing valuable insights, it supports evidence-based decision-making and aids in mitigating the impact of the crisis on the population and healthcare systems.

emews-mmca-covid19 is HPC-based high-throughput model exploration workflow designed for running iterative calibration of epidemiological models and design of optimal interventions based on EMEWS, which provides a sophisticated platform for large-scale model exploration and optimization in HPC infrastructures. By combining it with the MMCACovid19 simulator, users can leverage the capabilities of both tools to enhance the understanding of the spreading patterns of an epidemiological outbreak and inform decision-making processes.

2.7.2.2 Simulator for the Multiscale Lung Infection Scenario

The Lung infection scenario is based on the coupling of two simulators, an organ-level simulator Alya and the cell-level simulator PhysiBoSS. This coupling allows for a mechanistic multiscale model that encompasses the pulmonary alveoli sacks' structure and cells, the vascular and lymphatic system around these, the airflow that comes in these sacks and the virus infection and cells' interactions.

Technically, PhysiBoSS simulates cell-agents with mechanistic models as well as their interactions and Alya simulates the air pressure arriving to the lungs and the perfusion of the vascular and lymphatic vessels.

2.7.3 Simulator for the Maritime Use Case

The simulator for the Maritime Use Case was implemented by Kpler. This simulator presents and simulates a high-level overview of the Maritime Use Case test cases for the end user operators. The simulated events are presented to the end users via a graphical user interface (GUI) for evaluation and testing. In this use case, the simulator is not open source.

- MarineTraffic (Kpler) creates streams of simulated vessel positions, events, and weather conditions in order to simulate and optimise the navigation of a vessel under certain conditions (e.g., traffic, weather conditions).
- MarineTraffic (Kpler) developed a simulator system for the needs of the Maritime Use Case. The system uses synthetic simulated data and simulates specific scenarios of emergency events.

More details are described in CREXDATA Deliverable D2.3.

3 Use Case Specifications, Data Sources, Use-Case-Specific Requirements for the CREXDATA System Architecture, and Use Case Demonstrator Architectures

This section provides an overview of the real-word CREXDATA application use cases, i.e. their specifications and resulting requirements for the CREXDATA system architecture and its capability to handle large amounts of multi-modal data of various types from a variety of data sources including extremely large data streams and its capability for federated real-time data processing of extremely large data streams.

This section summarizes the parts of CREXDATA Deliverable D2.1 relevant to the understanding of the use cases, their scenarios, and their requirements with regards to the data sources (variety, volumes, frequencies, etc.), data processing capabilities (multi-modal data integration, real-time data stream processing, federated data processing, federated machine learning, etc.), the overall CREXDATA system architecture, the use case demonstrator architectures, as well as use-case-specific simulators, systems, and system components. For more details on these topics see CREXDATA Deliverable D2.1. If you have already read CREXDATA Deliverable D2.1, you can skip this section.

Emergency management and critical action planning call for timely and accurate decision making in several, diverse applications with the goal to optimize economic, societal, or environmental impacts. In the maritime domain, critical situations may occur due to imminent harsh weather conditions, vessel collisions, groundings, piracy events and a multitude of other hazardous situations at sea. Civil protection authorities face emergency situations of vegetation fire outbursts or sudden floods due to rapid weather-induced events as direct effects of climate change. Critical action planning is also of great importance in the life sciences domain. The recent world-wide health crisis of the COVID-19 pandemic demonstrated the need for governments and health agencies to make timely decisions to mitigate the impact (a) at a macroscopic level for the evolution of the COVID-19 pandemic at various levels of spatiotemporal resolutions, and (b) at a microscopic level, studying viral evolution for forecasting emerging mutations of clinical relevance [DoA, part B, page 3].

3.1 Weather Emergency Use Case

Weather induced emergencies are characterized by underlying weather phenomena, their evolution in time and space as well as their impact on the environment including people, nature and infrastructure. Large-scale data services are used for data logs, current satellite images and forecasts (like Copernicus services EFAS, EFFIS and EDO). Stationary sensor systems (like weather radars, automatic weather stations and river gauges) are used to verify satellite data, to trigger alerts in cases of critical values and to enable nowcasts. For similar purposes, multi-lingual text messages from social network sites are gathered and interpreted. Weather-related impact databases are only partially available as a source to train machine learning algorithms. Typically, weather information needs to be scaled from high-resolution to regional and local settings. Mobile robotic sensor platforms (Unmanned Ground and Aerial Vehicles/UGVs and UAVs) with their local viewpoint can complement these data sources with different types of cameras, laser scanners, radar and other sensors. The hydrological impact is specifically challenging with respect to specific terrain like in Austria and urban environment like in Dortmund; fire is based on environmental conditions but dependent on trigger/cause of fire. Operations responding to large-scale emergencies are coordinated in (mobile) control rooms with the need of situational awareness, even though they are not fighting the emergency face-to-face. They perform action planning and take decisions based on command posts closer to the site, and operational forces acting at a scene and being responsible for data collection (assisted by robots). Continuous monitoring is required, based on regular data updates or specifically dispatched reconnaissance robots. This can be complemented by human-generated content provided in terms of multi-lingual text messages, photos and videos via social media. Therefore, high data volumes of very different spatial and temporal resolutions and types need to be analyzed, while robotic platforms collect large amounts of data in high frequency [DoA, part B, pp. 10-11].

The CREXDATA Use Case deploys different types of mobile robotic sensor systems, utilizes visualization concepts (including collaborative AR) from WP5 and implements them into the extended ARGOS system (provided to the control center staff) as well as the extended robotics situational awareness system (in the command car “RobLW”) and T5.4 on-site Augmented Reality (AR) tools (both provided to commanders in the field) [DoA, p. 8].

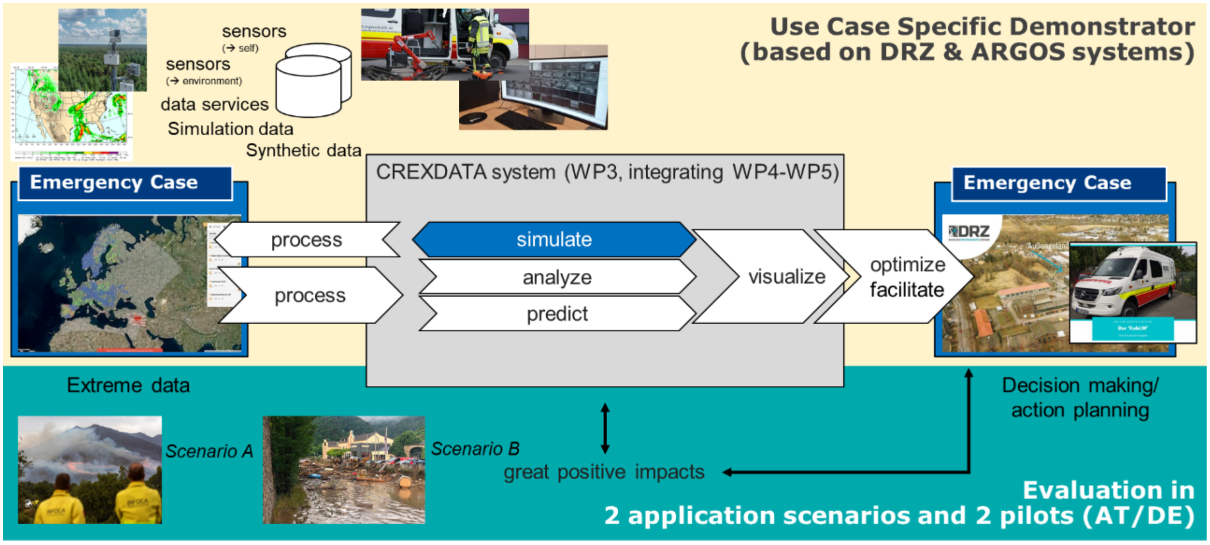


Figure 23: Impact oriented research within CREXDATA in the emergency use case (Figure from CREXDATA Deliverable D2.1).

The evaluation is performed in two kinds of settings (see Figure 24):

a) at the German Rescue Robotics Center, the existing indoor and outdoor test bed are used. For this test bed, exact terrain information, building information models etc. are available for reproducible evaluation settings. Both UAVs and UGVs can be operated within the area.
b) two pilots are setup to evaluate the system in relevant environments in an urban area (Dortmund, DE) and an alpine landscape (Innsbruck, AT).

Different levels of decision-making from local fire brigades (FDDO) through technical relief units (DCNA) to national stakeholders with links to the EU Civil Protection Mechanism (MoIFI) are considered. They take decisions with different functional, spatial and temporal responsibility [DoA, p.8].

The purpose of this use case is to **improve situational awareness** significantly so that **informed decisions** are taken by civil protection **considering ranked future worlds** with **explicit uncertainties** avoiding disaster **impacts**. Use case validation is performed in reproducible test bed scenarios and in 4 field trials [DoA, part B, p. 11].

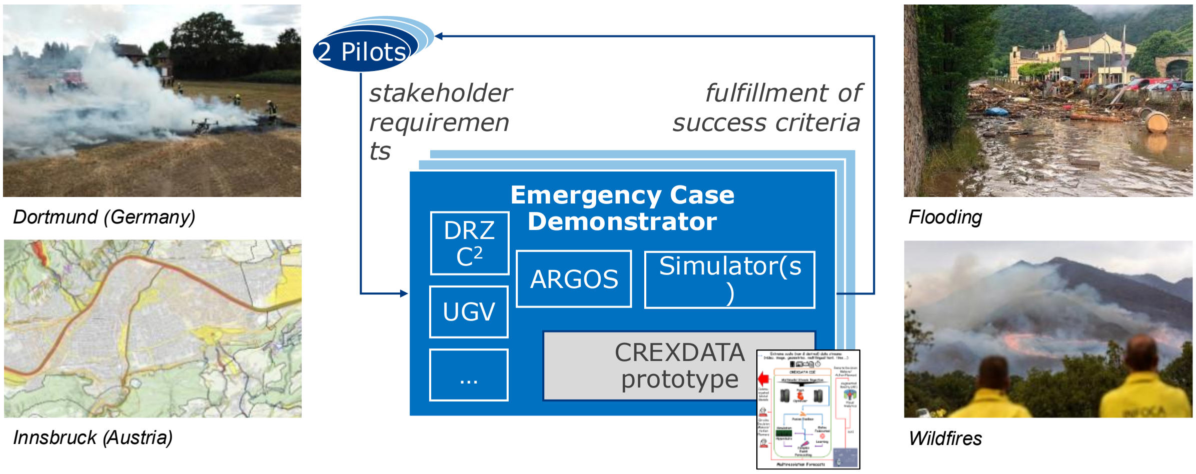


Figure 24: Demonstrator, pilot sites and application scenarios for the emergency case (Figure from CREXDATA Deliverable D2.1).

3.1.1 General Considerations on the Use Case Scenarios

Field trials and application scenarios are focused on hydrological weather phenomena, namely fluvial floodings. By aligning pilot sites to a common scenario setting, configurations of the system and the use of data sets can be managed much more consistently. Data sets from Dortmund, Austria and Finland can be used focusing on similar types of events to be forecasted, features to be learned and explained as well as simulations to be guided. At the same time, domain specific models are setup in a way that the inclusion of further application scenarios like forest fires is well prepared. For instance, event types like “Vulnerable people endangered” remain relevant while data sources and the underlying simple event type model need to be extended. Additionally, stakeholders and their demonstrator contributions remain the same.

3.1.1.1 Flooding

Flooding is observed from indications of heavy rainfall to monitoring of affected points of interests. Due to heavy rainfall, the gauge of a river might rise or drainage systems might be overloaded. In general, this can be anticipated based on satellite images and forecasts. In case of fluvial flooding, the actual river gauge at a certain point in time at a very certain location needs verification. The impact of a rising gauge may depend on many influencing factors: terrain, bridges or walls channelling the water, congested tubes changing the prepared flow of water, materials in the river (mud, wooden material, debris, cars). Therefore, action planning depends on current data acquired by stationary sensors (e.g., river gauge sensors) or mobile equipment that is specifically dispatched (e.g., drones recording material at the water surface). Typically, sensor data itself is (a) too voluminous and frequent to be analyzed by human operators and (b) too specific/technical to be helpful to operational forces [DoA, part B, p. 11].

3.1.1.2 Forest- and Wildfires

Forest fires are observed from critical drought situation (based on EDO) through fire spreading (with fixed observation systems) to monitoring of the fire zone (with mobile robots). To effectively fight large fires, it is necessary to detect them at an early stage to be able to immediately initiate suppression measures. The next step is to get qualified personnel and resources to the right place in the shortest possible time. For this purpose, continuous reconnaissance of the fire spread is the basis for an effective extinguishing operation. Prediction and quick response to situational events are crucial capabilities. This network can only function effectively if the emergency forces and command can rely on an optimal IT-supported information supply with prognostics [DoA, part B, p. 11].

3.1.2 System Architecture for the Demonstrator

The CREXDATA demonstrator system architecture can be read in terms of rows and columns (Figure 25). In rows, a layered IT system architecture goes from (Graphical) User Interface ((G)UI, top layer) to data sources (bottom layer). The CREXDATA system is integrated as a middle layer, interfacing different types of IT sub-systems to be found in columns. These columns indicate fully functional systems. Such sub-systems of the demonstrator system are mainly domain-specific systems with interfaces to the CREXDATA system (cf. Section 2.1). The composed system can be viewed as a common information space [16]. Key components are ARGOS provided by HYDS, robotics provided by DRZ and AR system developed in WP5.

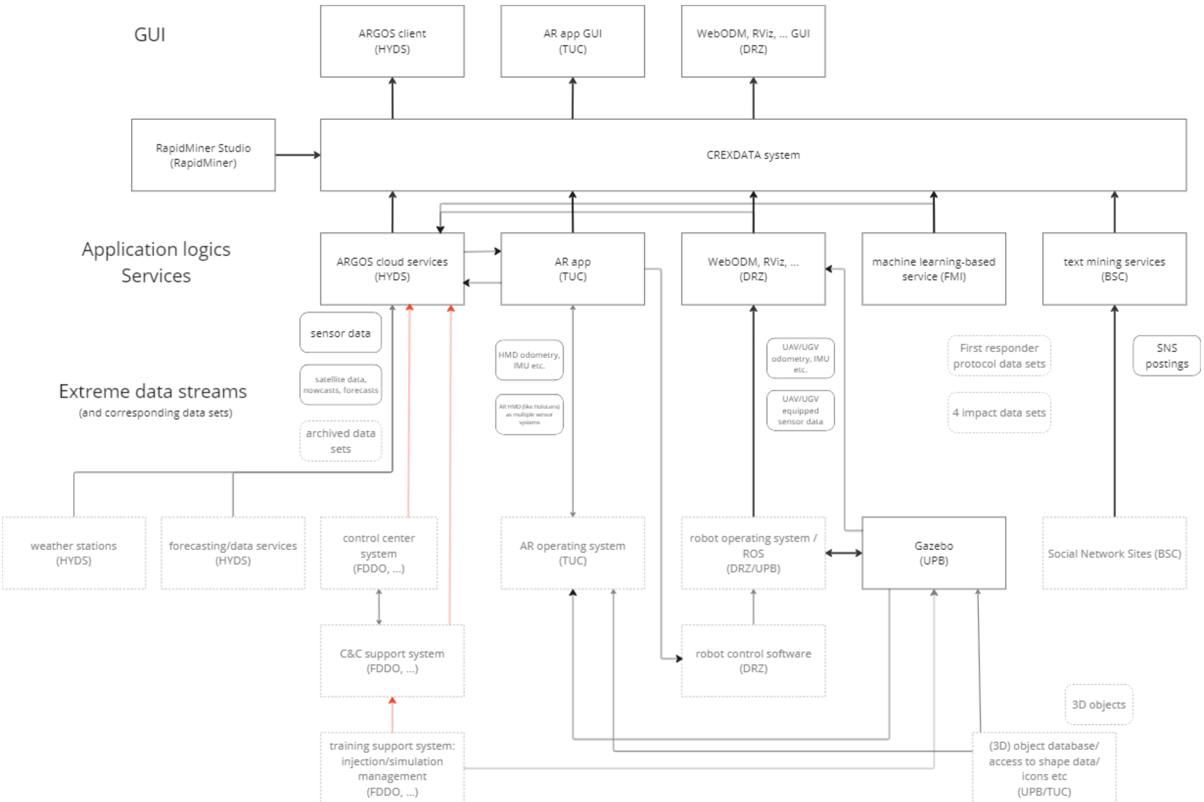


Figure 25: Demonstrator System Architecture for the emergency use case: core CREXDATA system interlinked with application logics and data sources of ARGOS, AR, robotic platforms and FMI services feeding corresponding user interfaces (CREXDATA Deliverable D2.1).

At the UI layer, the system's core GUI is adopted from ARGOS. The initial view as an entry point is assumed to be a geo-based situational map with layers and links to special applications. Thus, geo-referenced data can be super-imposed or marked within layers on top of geographical maps. Additional views need to be available, like green-to-red scale listing of critical events, sortable by priorities, time stamps etc. Besides the ARGOS UI, special applications are required either as extensions to resp. configurations of existing software or as newly developed apps. For visualizing sensor data from robotic systems, solutions like WebODM and RViz/RVizWeb should be adopted. With regards to AR, apps are developed for operational staff using development environments like Unity3D. It is essential to include Head-Mounted Displays (HMDs) as See-Through devices on-site. Interaction is implemented, for instance, to perform queries to the system or visualize data processed by CREXDATA algorithms. Application logics are not limited to CREXDATA algorithms but benefit from existing solutions. This includes information processing for layer-based visualisation, to be prepared for an uptake of visual analytics and uncertainty visualizations from WP5. To enable modelling of entire data processing pipelines, WP4 functionality is implemented in terms of RapidMiner operators. Similarly, operators need to be made available to connect, for instance, to weather-related services. With regards to robotics, functionality should support analysing sensor data from multi-sensor systems, routing, etc.

Table 1: Demonstrator components extending the CREXDATA system (Weather Emergency Use Case / Dortmund) (Table from CREXDATA Deliverable D2.1):

Component	Description
ARGOS [DoA, part B, p.9]	• multi-hazard early warning system developed by HYDS • data fusion • mapping of geographical data sources (as layers in the geo-service) • within mathematical models for specific topics (like height of snow on streets based on precipitation and ground temperature gradients) • ROS framework is applied using custom scripts to incorporate sensor data from robotic platforms
DRZ	• RobLW ("DRZ C2"/"Lagebildsystem"): mobile robot control and mission command post, equipped for robot operation as well as collecting data from robots and data processing to support situation awareness and decision making. • Ground Robots (UGV): various platforms with variable payload modules, selected depending on the mission (application and tasks); e.g., mid-size Telemax equipped for navigation and mapping • Aerial Robots (UAV): various commercial platforms, typically equipped with cameras; additional payload depends on the mission and taking into account weight considerations
FMI	• gradient boosting machine learning approach, used in various impact forecasting products and in previous research projects like SILVA
UPB	• Gazebo • RViz • event injection (cf. training support systems) • simulation configuration • data fusion

3.1.2.1 ARGOS system

ARGOS incorporates all processes required to manage weather-induced hazards, harmonising data, products, warnings, impact and protocols in one integrated solution. Core functionality is highlighted as hydrometeorological forecast, early-warning detection, exposure and vulnerability, impact forecasts, management protocols and dissemination. ARGOS has been designed from ground up to seamlessly integrate any source of information useful for operative management. It supports authorities in defining new rules of warning decision flows.

Integrated data services subsume

a) services that are activated in case of a large emergency
b) general services: EFAS, EFFIS, EDO, etc.
c) sensor systems

Besides meteorological data, data from 112 calls, traffic cameras or social networks can be integrated (cf. Figure 26).

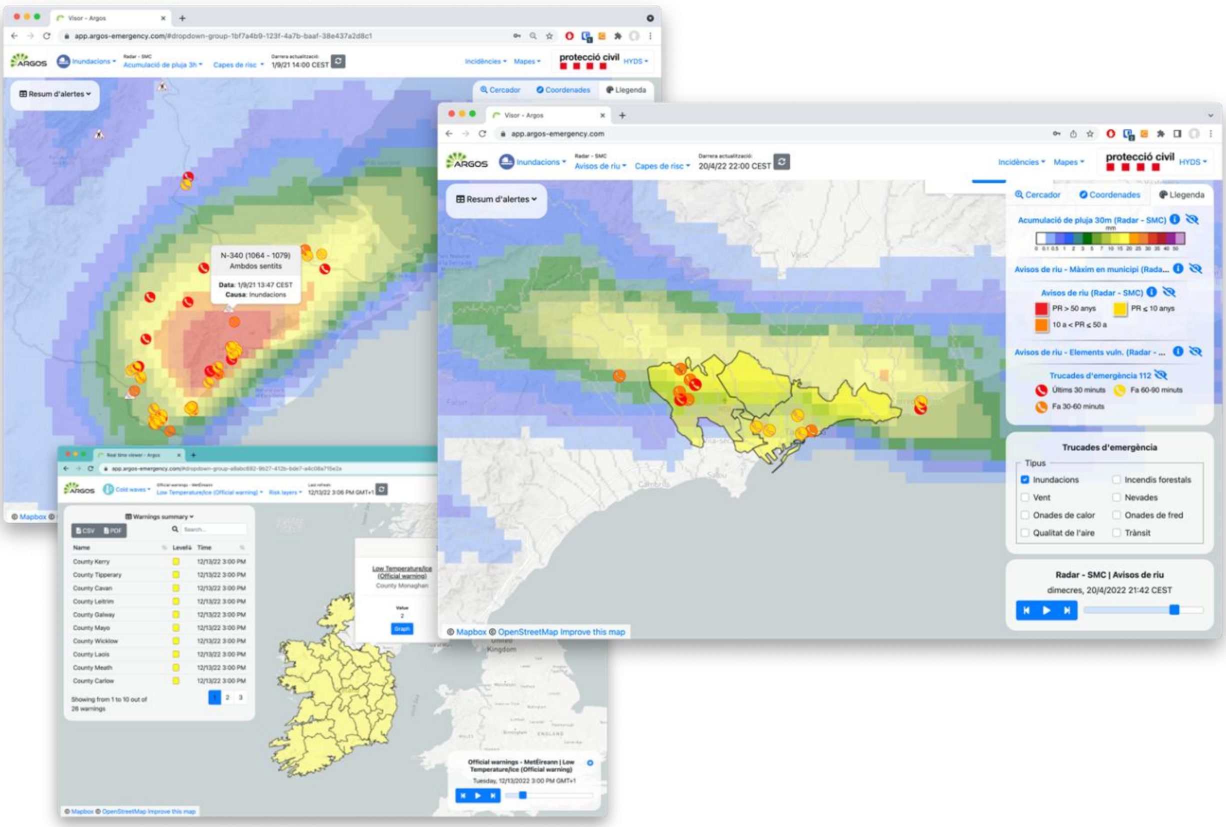


Figure 26: ARGOS service examples: radar data, forecasts, threshold visualization, geo-located emergency calls, warnings summaries, playback function etc. (Figure from CREXDATA Deliverable D2.1).

ARGOS is built on top of an open architecture (see Figure 27) based on self-dependent modules, so that adding new weather data and sensors, building new products and warnings, growing sets of critical points, redefining protocols and expanding dissemination channels is supported. Due to its modular structure, communication with external services or platforms is inherently flexible in ARGOS. Real-time data is collected through available machine-to-machine interfaces, web map services (WMS) for geo-structured data or raw file transfer through sFTP servers. On the other side, ARGOS generated data (warnings, related products, registered values, etc.) can also be pulled by other systems using its own API (Application Programming Interface, available in https://api.argos-city.com/). Communication interfaces are adapted or extended to integrate ARGOS with the overall CREXDATA system.

The cloud-based ARGOS architecture enables collaboration between different operational authorities in width (e.g., between federal districts) and depth (i.e., on different levels of organisational responsibility). ARGOS is designed as a family of products to support different types of stakeholders (Argos City, Argos Site, Argos Flow, Argos Hydro). To do so, the following characteristics were enhanced from the very beginning on its construction (see all features in Figure 27):

- Easy adaptation: on new places, on new organisations with similar needs but different procedures.
- Impact oriented: not only weather monitoring but automatic activation in vulnerable elements based on previous knowledge.
- Integrated solution: Based on the cloud, the system is available from any device with internet connection.

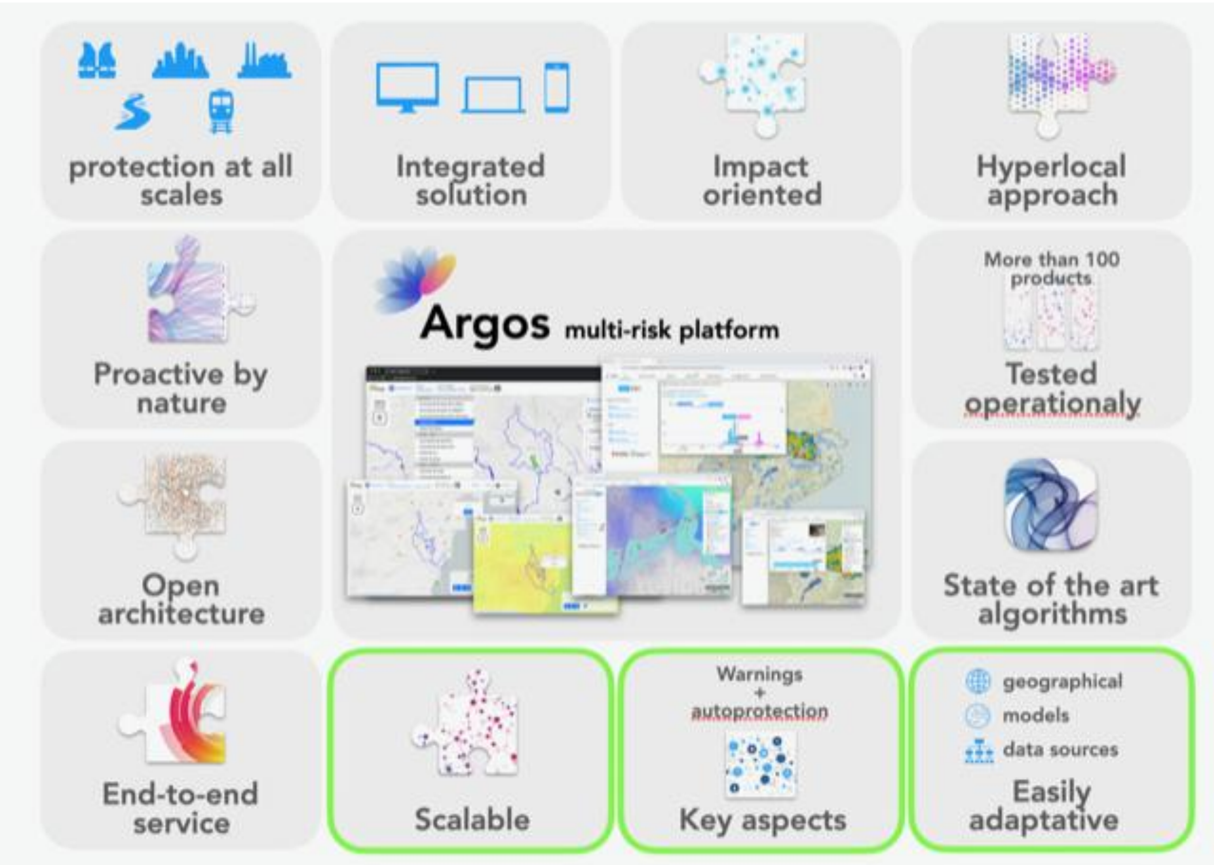


Figure 27: ARGOS features (https://www.hyds.es/argos/) (Figure from CREXDATA Deliverable D2.1).

3.1.2.2 Robotics system

In CREXDATA, different types of robots are foreseen for experiments[1]. In general, Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs) are differentiated. In case of research setups, main building blocks of such robotic systems are basic platforms, communication hardware and payload like sensor systems or, for instance, robotic arms/manipulators (Figure 28). The Robot Operating System (ROS) is implemented on the platform, ensuring standard interfaces to, for instance, sensors. Commercial robots (esp. UAVs) are integrated systems not necessarily changeable. Robots are operated using a robot operator interface (base station). In the CREXDATA setup based on DRZ capacities, robots are deployed by operational staff, with a perspective to expert units entitled “Robotic Task Force” (RTF). Currently, the DRZ operates a special command vehicle called “RobLW” (robot command verhicle)[2] which is equipped with both command and research tools. It is a prototype for a vehicle with robot integration, combined with a joint utilisation concept between DRZ and Dortmund Fire Brigade. Therefore, it is operated by mixed crews from DRZ/project and fire brigade for certain scenarios. Within the RobLW, special information systems are used to process robotic sensor system data. They are subsumed under the term “DRZ Command & Control (C2) system” (German: “Lagebildsystem”). Two specific software products are WebODM and RViz. WebODM enables visualization of maps, point clouds, DEMs and 3D models from aerial images. RViz/RVizWeb provides functionality for 3D visualization of sensor data, robot model, and other 3D data in a combined view.

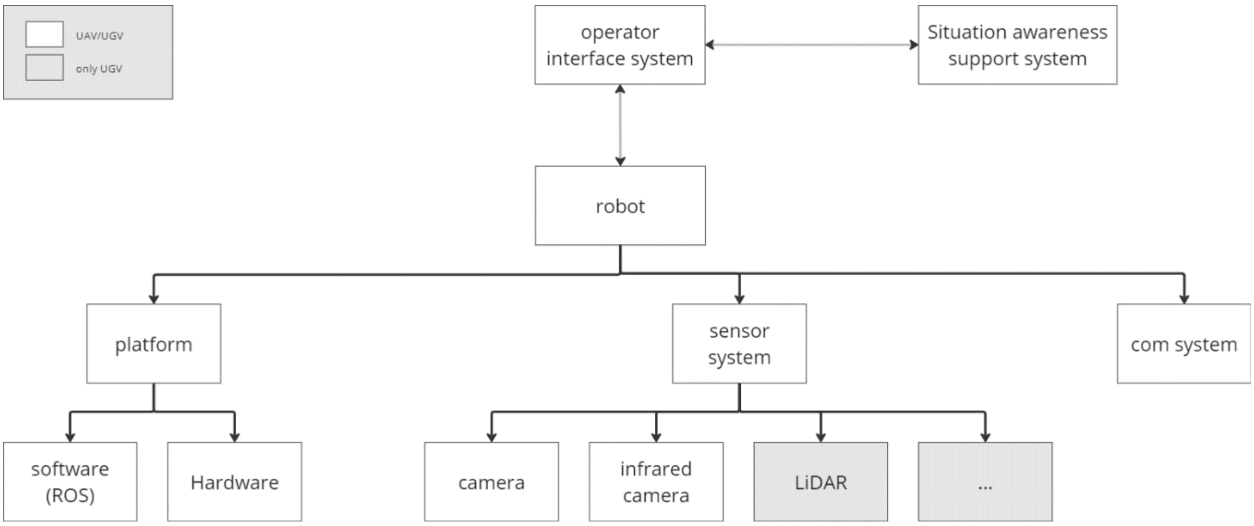


Figure 28: Architecture of the robotic demonstrator sub-system (Figure from CREXDATA Deliverable D2.1).

The RobLW is equipped with a powerful server for the creation of 3D models, two operator PCs and two 43’’ screens. For power supply and communication network setup, a 4 kW power generator, blue light/ Tetra radio, 2.4 and 5.8 GHZ receiver for drones, WLAN, 2 TETRA MRT and 4 TETRA HRT are available. A small workbench is built in with two fixed workstations plus

[1] https://rettungsrobotik.de/en/testing-facility/the-robotic-systems-on-an-overview

[2] https://rettungsrobotik.de/en/testing-facility/the-robotic-command-vehicle

an additional workstation for researcher/leader. Additionally, a flexible roof panel enables easy integration of various antennas.

As an option, even under-water robots could be used in CREXDATA (see, for instance, requirements in the Austrian pilot site). In a parallel project, DRZ builds a test setting for such kind of robots. So, this would enable use cases like:

- detection of physical or chemical contamination in the water that floods an urban area,
- selection of sensors to see under water (from in water or above water),
- 3D mapping and/or object detection under water,
- creation of realistic dataset for 3D mapping and object detection under water.

3.1.2.3 FMI service

FMI develops impact-based forecasts which are derived from the available impact data. The tools developed in close cooperation between FMI, MoIFI and the Rescue Department of Helsinki, and other pilot site partners like FDDO outside of Finland. There is a testing period during which the new products are piloted in real-time and the user experiences are collected to develop and enhance the machine learning-based model.

The model behind the tools developed by FMI for the Finnish showcase is utilizing machine learning (gradient boosting method). The Finnish showcase is utilizing the ARGOS system in data and model output distribution as well as for utilizing the data provided by other pilots to extend the area of operation of the ML-based service of FMI. The same CREXDATA platform as the Germany and Austria pilots is used in Finland as well. Eventually, the data produced and shared by FMI is available for the use of the technologies of WP3-5, for instance for input for explainable AI (XAI) of T5.1. Within the limits of available data, the gradient boosting machine learning model is used as a base to forecast the number of ambulance units needed in both Helsinki and Dortmund areas in a weather emergency or to create early warning tools for flooding events.

The gradient boosting machine learning method is used to model the impact data, for example the number of emergency operations or road traffic accidents. The method graph is presented in Figure 29.

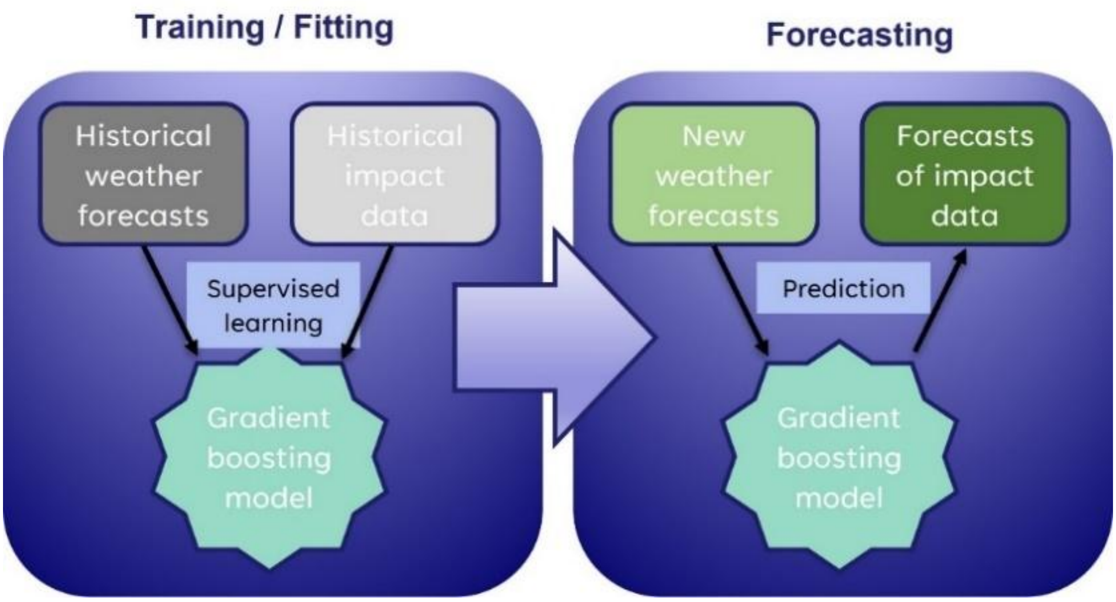


Figure 29: Use of gradient boosting machine learning in training and forecasting of weather impacts (Figure from CREXDATA Deliverable D2.1).

The ML method has been used in various impact forecasting products successfully, mainly in Finland, but also in the USA to model river streamflows. The method has been used also in previous research projects, for instance in a national level SILVA-project (2020-2023), where national level impact products were created (Figure 30). The model has been found reliable and accurate in several different applications; thus, it was chosen also for the main method in the Finnish showcase and as an input for T5.1.

For the modelling, the materials are collected spatially and temporally: local compilation and six-hour temporal compilation are used to unify the number of cases in Uusimaa and Helsinki areas. For the time series, a gradient boosting model is fitted using mainly surface weather parameters from the ECMWF HRES weather model from the Uusimaa and Helsinki area to explain the impact data. As additional information, information about the time of day and the season of each moment of time is included into the model. After matching, the provincial quantile limit values were derived for each data and month to illustrate the number of cases: the familiar green-yellow-orange-red colour coding is used in the visualization to distinguish the classifications determined by the quantile limit values from each other. The visualization is done by compiling the forecasts into five-day forecast maps by using rolling 24-hour sums.

In the CREXDATA project, new forecasts of new datasets are implemented, such as the numbers of ambulance operations mentioned before. In addition to that, we refine the resolution of the old products from the county level to the municipality level. The possibility to include estimation on uncertainty of the impact forecast is exploded by utilizing instead of ECMWF HRES numerical weather prediction model, the limited and higher resolution ensemble model, MEPS. For the European context the possibilities to model the river streamflow data for early warning of the flooding events are explored.

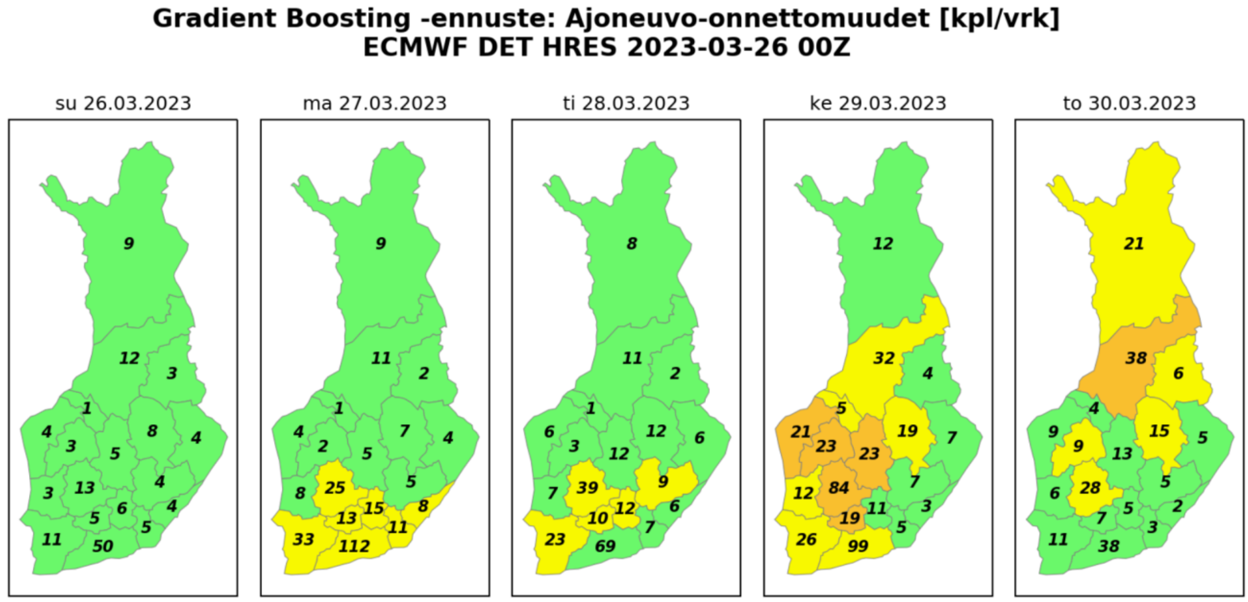


Figure 30: A 5-day outlook of gradient boosting forecast of car accidents per day per municipality. The colors (green, yellow, amber, red) represent the severity of the impact (Figure from CREXDATA Deliverable D2.1).

3.1.3 Data sources

The Weather Emergencies use case is based on data sources that can be categorized into local and global environment (see Figure 23). Data about the local environment is gathered through stationary sensors systems which are extended by mobile robotic sensors platforms in case of an incident. Local data requires context metadata like position, orientation and accuracy of sensors. Temporal effects are induced by sample rate (of sensors), pre-processing at the edge (e.g., creating geo-referenced point clouds) and communication channels (with frequency and bandwidth). Decision-making is informed by processed data formats, but explainability for humans often requires raw data analysis (e.g., video) and visualization. Position, point/field of view and annotations from AR devices are new data sources. At the level of global environment data, Copernicus services provide both TBs of data archives and GBs of current and forecasted weather data covering various natural phenomena and their effects on forest fires, health etc. Both temporal and spatial resolution vary. Sensor systems are increasingly applied in terms of forest observation systems (even with drone extensions), river gauge installations and publicly accessible weather stations. Data is either pushed in certain time intervals so that it can be processed message-based, or data is retrieved on request through data services and APIs. In Austria, for example, there is the ehyd database providing pre-filtered data (https://ehyd.gv.at/). Raw data is accessed through national, European and global weather services. In case of Austria, the corresponding partner would be Geosphere. Multi-lingual and multi-modal social media extend that broad range of data sources. Social media networks are expanded to global scale; for an emergency, the identification of relevant assertions in terms of location and content are significant challenges [DoA, part B, p. 11].

Table 2: Local environment data sources in the Weather Emergency Use Case [DoA, part B, pp.11-12] (Table from CREXDATA Deliverable D2.1):

Data sources	Data Source Volume (Vol), Velocity (Vel), Veracity (Ver), Characteristics/Description (Ch/D)
Data sources (local environment)	
Robot localization system	Ch/D: Position, orientation, velocity and acceleration of a robot, derived by fusing data from various sensors. E.g., odometry is often obtained by fusing data from wheel encoders, IMUs, and gyroscopes. Can be enhanced or substituted by visual odometry (extracting camera movement from consecutive images). Global pose (I.e., relative to a known reference frame) is typically obtained by combining data from GNSS receivers (e.g., GPS), magnetometers, odometry and matching range sensor data (e.g., from a lidar) to a known map. Pose data is transferred using designated ROS message types, e.g., geometry_msgs/Pose3D. Ver: Localization accuracy depends on chosen sensors and algorithms. Vel: Typically, between 10 and 100Hz. Vol: A single pose message is <1kB, overall data rate depends on required update frequency.
RGB video stream per camera (H-264, MPEG-4)	Vol: 1920 x 1080 (full HD) or 720 x 480 (standard definition). Vel: Common video stream data rates (e.g., 30fps). Ch/D: FPV Camera, wide-angle camera. Ver: If the connection is poor, a live stream may be blurred or stop for a short time.
Thermal video stream (H-264, MPEG-4)	Ch/D: Images with temperature data, e.g., derived from near-infrared spectrum. Vol: Common thermal camera models offer resolutions up to 640x480 pixels. Vel: Common video stream data rates. Ver: If the connection is poor, a live stream may be blurred or stop for a short time.
3D environment model	Ch/D: Map generated from robotic sensors. Commonly derived by combining localization data (see above) with range measurements (from cameras or lidars). Some solutions build the map "live" (Simultaneous Localization and Mapping – SLAM). Others build the model offline by processing collected data (most often images) in batch, which is much slower but often more accurate (e.g., using WebODM). Vol: Depending on the size, sparsity and type of the model (e.g., showing only the physical structure, or also including confidence, colors, temperatures, etc.). A model can easily reach several hundred MBs in size. Vel: Typically <10Hz. Ver: Depends on chosen sensors, algorithms and environmental conditions (e.g. ambient lightning, disturbance from smoke).
AR (HoloLens 2) (MPEG4)	Vol: 1920x1080px. Vel: 30fps (through miracast). Ch/D: Video stream, incl. annotations of AR users in the scene.

Table 3: Global environment data sources in the Weather Emergency Use Case [DoA, part B, pp.11-12] (Table from CREXDATA Deliverable D2.1):

Data sources	Data Source Volume (Vol), Velocity (Vel), Veracity (Ver), Characteristics/Description (Ch/D)
Data sources (global environment)	
EFAS	Vol: ~1Gb/day. Vel: Twice per day. Regular internet velocities. Ver: Probabilistic flow forecasts. Ch/D: Temporal series, distributed fields, deterministic and probabilistic
EFFIS	Vol: ~1Gb/day. Vel: Twice per day. Regular internet velocities. Ch/D: Distributed fields.
ECMWF forecasts	Vol: ~20Gb/day. Vel: 4 times /day. Regular internet velocities. Ver: Deterministic and probabilistic forecasts. Ch/D: Distributed fields.
Weather sensors	Vol: ~0.2Gb/day. Vel: ~Every 10 min. Regular internet velocities. Ch/D: Temporal series.
Weather radar	Vol: ~5Gb/day. Vel: ~Every 10 min. Regular internet velocities. Ch/D: Probabilistic data.

3.1.4 Use Case Analysis

The application scenario is set up based on various parameters to allow for a wide variety of sub-scenarios. Examples are fluvial vs. pluvial flooding, seasons, cascade effects (like debris/mudflow), prepared vs. escalating evolution of the situation as well as mainly affecting people vs. objects. Initially, the focus is on pluvial flooding. So, the assumption is that the strain a heavy rain causes on urban drainage systems is too heavy. Figure 31 provides an overview of the generic scene elements that are scheduled for application scenarios: Each scenario shall include a river that passes a locality with potentially endangered inhabitants. This might be urban or smaller scale. It includes vulnerable buildings and people, as well as critical infrastructure (like energy or transport). In case of an emergency, the emergency management structure is either built up step by step (in case of unforeseeable events) or it is anticipated and well prepared (e. g., in case of precise weather forecasts). Assuming a large-scale weather induced emergency, the structure (see small box in Figure 31) includes the high-level strategic command in a crisis management room (led by the mayor in Germany, supported by a high-level fire officer like the director of a fire department). Search and Rescue, fire protection and technical relief are coordinated by a tactical command unit (mobile Command & Control / C^2 post, operating from a large command truck or bus). Technologies like AR and robots are operated close to the operation by lower-level commanders (C level, in small command cars like vans) and sub-ordinated operators.

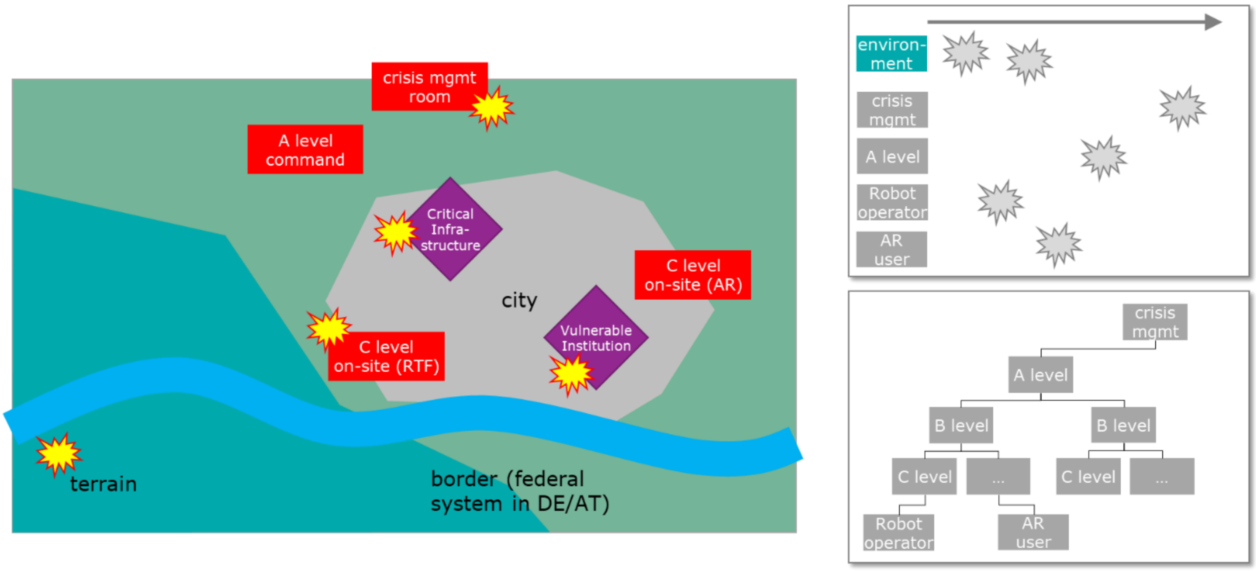


Figure 31: Elements of the spatial environment and basics of a command hierarchy (Figure from CREXDATA Deliverable D2.1).

Figure 31 hints at different views within an emergency situation: the spatial view which is typically represented by geo-services and map-based user interfaces, an event stream view representing changes of the situation, and a structural view focusing on resources deployed by emergency response organizations. Figure 32 provides a more tangible insight into the operational environments that frame use cases of the CREXDATA demonstrator system. AR and robots are operated close to the actual incident by operational responders. They are led by low-level commanders (entitled “C level commanders” at FDDO). Mid-level commanders (B level) are not focused on initially. For the A level command post, a staff room is equipped where four to six or even more fire officers cooperate with regards to pre-defined tasks (like ICT, map, supply, press, etc.). Different settings are feasible: a room setup within a command truck (entitled “ELW3”), or a stationary environment at fire station 1 in Dortmund. For the sake of clarity, the first is called “A level staff room” and the latter is called “crisis management room”. For a large-scale disaster, the mayor of Dortmund activates and leads a crisis management team, being super-ordinated to the structures of FDDO. In all these staff environments, tools like the ARGOS system are relevant. The control center dispatches resources to the operation and tracks status changes (at station, departing to incident, active in an operation etc.) of resources[3].

[3] GPS is not available due to GDPR issues.

Figure 32: Environments for use cases of the CREXDATA weather emergency application scenario (Figure from CREXDATA Deliverable D2.1).

Figure 33 presents a swim-lane visualization of the evolution of such an event. Focusing on decisions (swim lane in the middle), a distinction is made to events that occur in the environment (e.g., change of weather forecasts, threshold reached at river gauge) and events that are triggered by own actions (e.g., a human is rescued, or a drone reached the operational position).

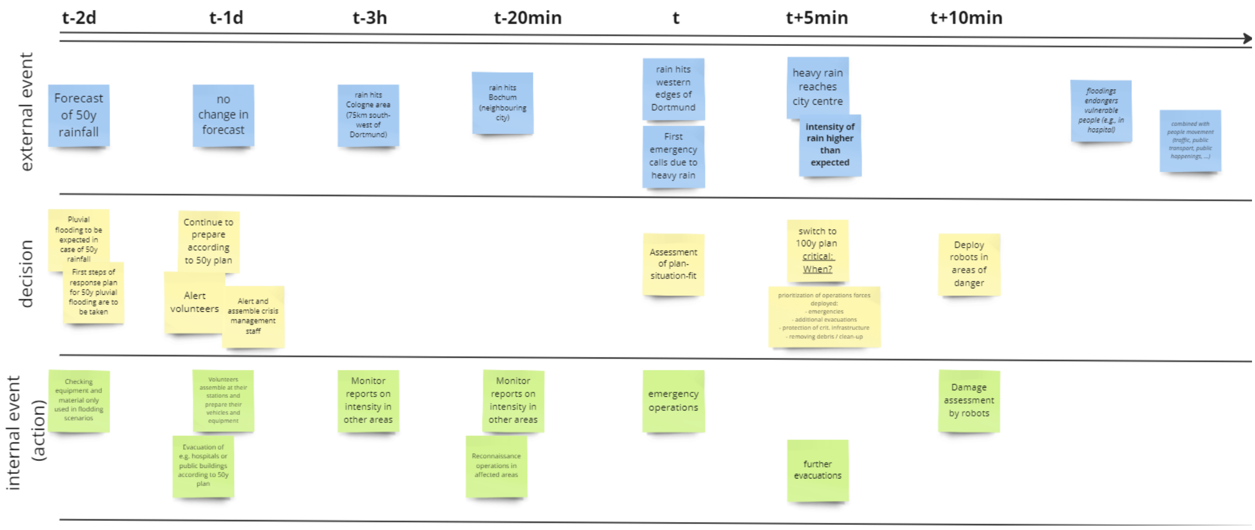


Figure 33: Elements of the temporal evolution of the scenario (Figure from CREXDATA Deliverable D2.1).

Specific sub-scenarios within this overarching application scenario are provided in the appendix of CREXDATA deliverable D2.1. These sub-scenarios are designed in a way that they can be either a) combined with each other for integrated test scenarios or b) used to derive specific test cases. While test scenarios would be relevant for field trials, test cases are required to setup lab-scale test environments (for instance, at DRZ and at UPB). Figure 34 presents an overview of the initial set of sub-scenarios.

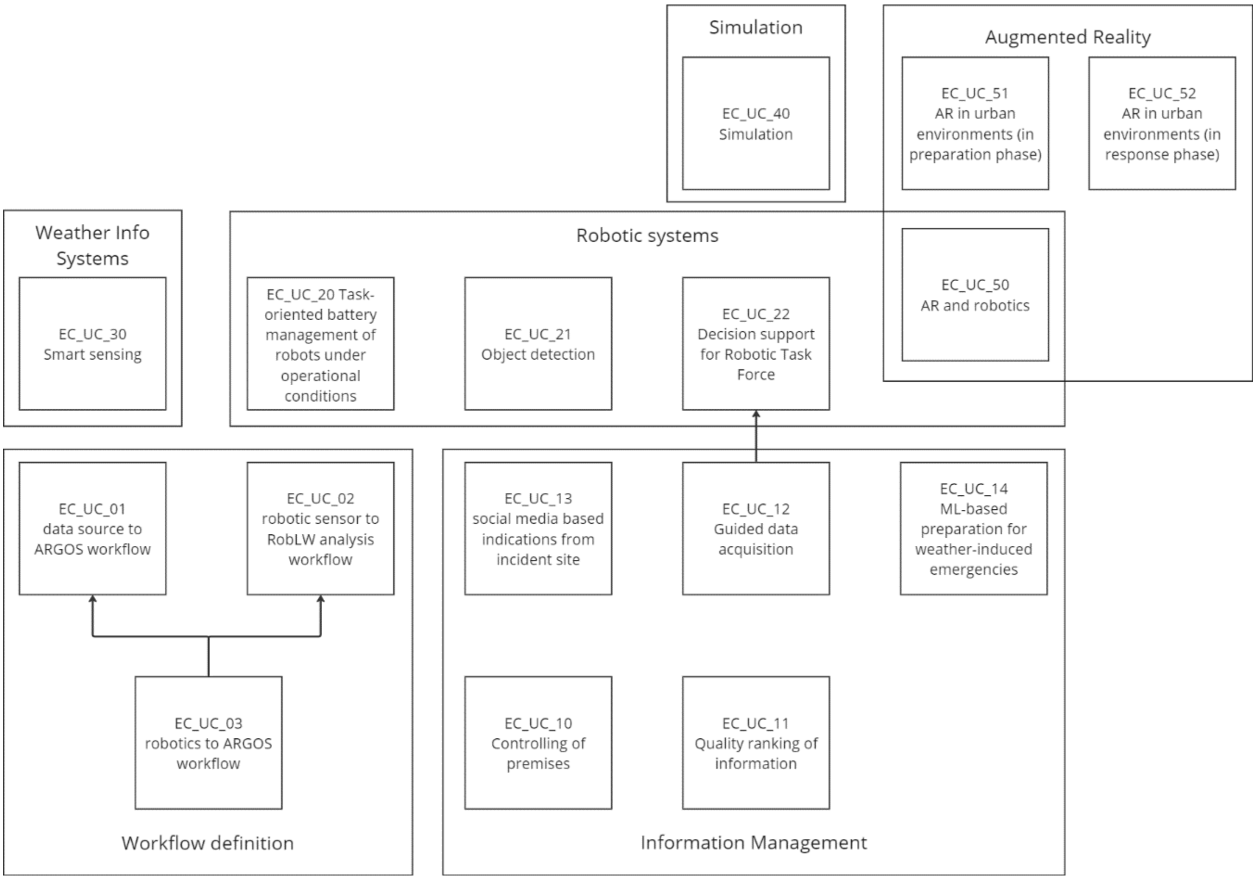


Figure 34: Overview of application sub-scenarios (use case narratives) (Figure from CREXDATA Deliverable D2.1).

3.1.5 Mapping of Sub-Scenarios to WP3-WP5 Technologies

Table 4 on the next page shows which emergency use case sub-scenarios use which technologies developed in the CREXDATA work packages WP3 to WP5.

Table 4: Uptake of technologies in the Weather Emergency Use Case (Table from CREXDATA Deliverable D2.1):

Specific Use Case / Technologies Used	1	2	3	4	
T3.2 Graphical Workflow Specification	Em_UC_01	Em_UC_02	Em_UC_03	all others	
T4.1 Complex Event Forecasting	Em_UC_10	Em_UC_12			
T4.5 Text Mining for Event Extraction	Em_UC_13				
T4.2 Interactive Learning for Simulation Exploration	Em_UC_40 **	Em_UC_12 **			
T4.3 Federated Machine Learning	Em_UC_20	Em_UC_21	Em_UC_30	Em_UC_12	Em_UC_14
T4.4 Optimized Distributed "Analytics as a Service"	Em_UC_01 *	Em_UC_02 *	Em_UC_03 *		
T5.1 Explainable AI	Em_UC_11	Em_UC_14	Em_UC_20	Em_UC_22	Em_UC_30
T5.2 Visual Analytics Supporting XAI	Em_UC_11	Em_UC_14	Em_UC_20	Em_UC_22	Em_UC_30
T5.3 Visual Analytics for Decision Making under Uncertainty	Em_UC_10	Em_UC_11	Em_UC_22	Em_UC_30	
T5.4 Augmented Reality in the Field	Em_UC_50	Em_UC_51	Em_UC_52		
T5.5 Uncertainty Visualization in Augmented Reality	Em_UC_50	Em_UC_51	Em_UC_52	Em_UC_11	
* to be detailed based on workflow descriptions ** to be detailed based on simulator selection					

Field trials are conducted within real or at least realistic settings of stakeholders in Dortmund, Germany, and Austria. Details are provided in the following Sections 3.1.6, 3.1.7 and 3.1.8.

3.1.6 Evaluation Scenario and Data Source for Weather Emergency Use Case Pilot Site in Dortmund, Germany

The pilot site in Dortmund tests the CREXDATA system architecture and the weather emergency demonstrator architecture with a pluvial flooding scenario in the city of Dortmund.

3.1.6.1 Application Scenario

The application scenario is introduced in Section 3.1.4 as a preparation of use case analysis. In this Section, the fundamental scenario is adopted and transferred to the specific environment of the city of Dortmund in Germany. A pluvial flooding is assumed to happen in specific areas of Dortmund. All aforementioned use cases are assumed to be relevant in this application scenario. Thus, crisis management and high-level emergency management of FDDO are required. The emergency is extreme both in terms of spatial and temporal extensions. The scenario is drafted based on experiences made during the extreme weather event in western Germany, especially in the Ahr region and around Erftstadt 2021. A detailed report of 325 pages is available [18] (cf. [19, 20]). Some sub-scenarios with corresponding use cases are setup in small-scale reproduction in the test-bed at DRZ. For the overall situation, risk maps are available that were created in standard preparation activities[4].

Figure 35 presents some insights into these maps assuming a flooding with 100 years return period, which are available in the control center and in the A level command post. The RobLW is deployed in an operational Section that coordinates use of drones (UAVs and UGVs).

Figure 35: Application scenario centered around Dortmund main station (Figure from CREXDATA Deliverable D2.1).

4 https://geoweb1.digistadtdo.de/doris_gdi/mapapps4/resources/apps/starkregengefahrenkartetn100/index.html?lang=de&vm=2D&s=10000&r=0&c=393280.24263935274%2C5708048.11168041

3.1.6.2 Available Data

The following data sources (cf. [D1.2, Section 3.1]) are available to provide data either continuously or in real-time during a test setup or in a field trial:

- Unmanned Vehicles[5] with robotic platform and payload in terms of sensor systems (off-the-shelf and experimental setups, see Section 3.1.2.2; available at DRZ and planned at UPB)
  a) UGVs
  b) UAVs
  c) Under-water robot (planned)
- RobLW[6]
  a) Applications for 3D model creation
  b) Communication networks
- AR devices (available at TUC and UPB)
  a) Microsoft HoloLens 2
  b) Tablets, smartphones
- Test bed
  a) Robot localization system (DRZ)
  b) Video observation system (UPB)

In desktop-like conditions, different devices are used to experiment with UIs. This subsumes laptops, tablets and smartphones, but also multi-touch displays and tables.

In the course of the project, it might be relevant to incorporate further equipment of FDDO. For instance, there is a specialized Analytical Task Force (ATF) operating a wide ranging of measuring equipment [21].

The following datasets are available (see details in [D1.2, Section 2.1.2]):

- Emergency Cases FDDO 2020-2021
- Damage clearance tasks in Finland
- Warnings for flooding in Dortmund
- flight data from Erftstadt (flood event 2021), available at DRZ (owned by the pilot)[7]

3.1.7 Evaluation Scenario and Data Source for Weather Emergency Use Case Pilot Site in Austria

For Innsbruck, the coordination in the event of an operation is compliant with the introduction of Section 3.1.4. The emergency dispatch center is led by “Leitstelle Tirol”, they alert the emergency organization (fire department). Depending on the size of the event, there is also an official command (crisis management on municipality level) in addition to the fire department operational command. These two commands work closely together. In the case

[5] URL https://rettungsrobotik.de/en/testing-facility/the-robotic-systems-on-an-overview

[6] URL https://rettungsrobotik.de/en/testing-facility/the-robotic-command-vehicle

[7] cf. https://www.youtube.com/watch?v=BIq9P9NHbT0

of pluvial flooding, as for example in the district of Amras, there was only one operational command at the fire department. This coordinates the disaster case. In addition, there is an operation site management and various operation teams on site in the disaster area. An exceptional situation in Innsbruck is that there is the Landeswarnzentrale Tirol, LWZ (in which the Leitstelle Tirol is integrated), which functions as an official coordinating body. That means that the Leitstelle Tirol and the fire departments provide data to the LWZ. Thus, they collect the data and sends the warning/alarm or all-clear to the population if requested by the authorities. Furthermore, the LWZ has a drone competence center, which can send drones to the disaster area on request and send aerial images, thermal images, etc. via live stream to the official geo-information-based situation management system KATGIS (provided by Geosphere Austria, the Federal Geological Service).

3.1.7.1 Application Scenario

In the case of the Austrian pilot site, two application scenarios are under discussion, in which the fundamental scenario (introduced in Section 3.1.4) is adopted and transferred:

- pluvial flooding due to a heavy rainfall in the city of Innsbruck, i. e. Amras district
- fluvial flooding due to the Danube River in Lower Austria, i. e. Tulln an der Donau,

whereby the focus and implementation, as well as the exchange with stakeholders, are currently being placed on the use case in Innsbruck (Figure 36).

In general, the application scenario in the city of Innsbruck aims to (i) increase the situational awareness of key stakeholders, i.e. emergency response teams and decision makers (see Figure 36) and (ii) expand the possibilities of technology use in an urban emergency in an alpine environment. Due to the topographical conditions, the city of Innsbruck must also expect cascading debris flows during heavy rain events. In addition, due to the narrowness of the valley, there is little space to drain the water. Thus, in the event of local heavy precipitation, the city's drainage system is most likely overloaded within a very short time. The time component depends on various factors, such as the amount and duration of precipitation, the mobilization of loose material from the slopes and other obstacles, but also on the time of year – e.g. leaves in autumn. In addition, such thunderstorm cells in alpine areas are often accompanied by hail which can cause blockage of drainage systems. In combination with a supra-regional, regional heavy precipitation event, the Inn River can also cause flooding in the city of Innsbruck and turn the disaster operation into a major event. Additional remedial measures are required here, e.g. with mobile flood protection and retention basins. But in this application scenario, we mainly focus only on pluvial flooding, as these have been severely increasing in recent years and pose extreme challenges to emergency services. I.e. in a very short time, city districts can be completely under water, buildings have to be evacuated because their stability is no longer guaranteed, and people are in danger. In Innsbruck, debris flows can be triggered as secondary processes and endanger people and infrastructure. The action of the emergency forces is required in the shortest possible time. This application scenario is planned to be processed on the basis of the heavy precipitation event of July 2, 2016, which severely flooded the district of Amras (see pictures in Figure 36).

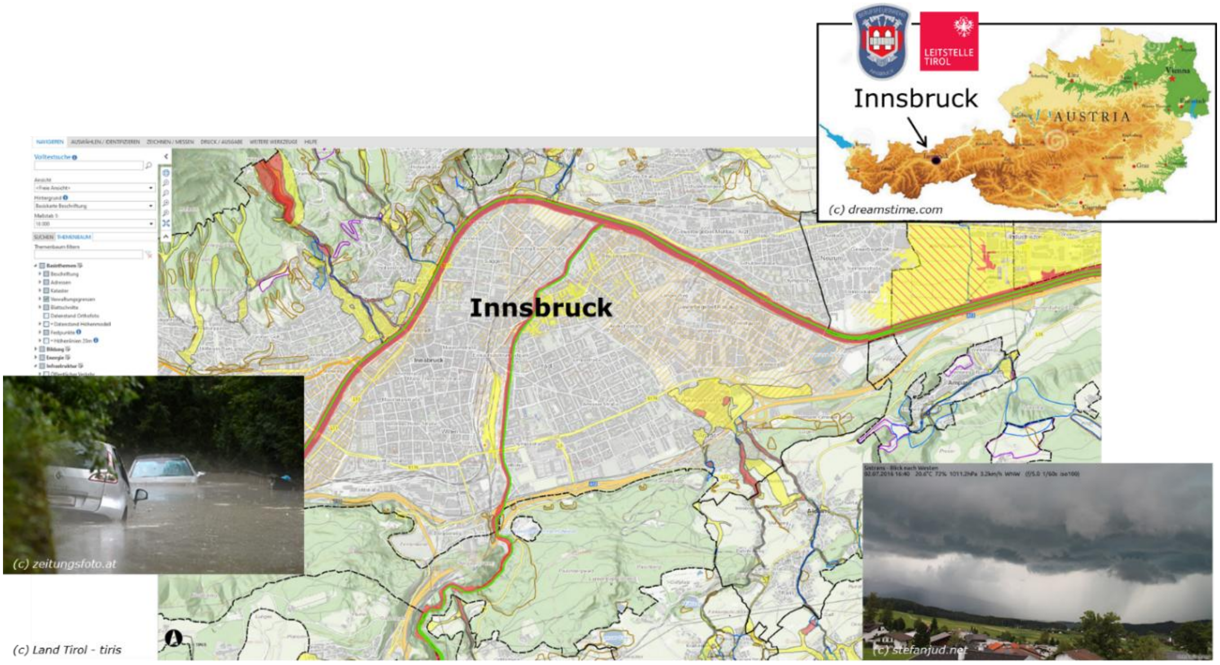


Figure 36: Spatial environment of the application scenario in Innsbruck (https://maps.tirol.gv.at/) (Figure from CREXDATA Deliverable D2.1).

3.1.7.2 Available Data

The following data sources (cf. [D1.2, Section 3.1]) are available to provide data either continuously or in real-time during a test setup or in a field trial. First indications after an initial workshop with third parties in Innsbruck are:

- Professional Fire Brigade Innsbruck:
  a) Deployment protocols
  b) Radio communication data (incl. location data)
- ECC Tyrol (Leitstelle Tirol):
  a) Emergency call data (except from the police)
- National Warning Center Tyrol (Landeswarnzentrale Tirol) [optional, not yet sure - still in discussion]:
  a) Deployment protocols
  b) Data collected and provided for the key stakeholders in the geoinformation-based systems (i) webGIS tiris OEI - Local operation information, and (ii) katGIS (Digital situation recording and situation information)
- IKB: Innsbrucker Kommunalbetriebe AG (municipal infrastructure service company) [optional, not yet sure - still in discussion]:
  a) Water level and (surface) runoff data of sensors in relation to the drainage system
  b) Supply bottlenecks (energy, electricity, etc.)
- Geosphere Austria [optional, not yet sure - still in discussion]:

a) Meteorological raw data (precipitation, river and groundwater level, river flow rate of sensors)
b) Rain radar data
c) Weather forecasts incl. thunderstorm cells (national to local)
d) Heavy rainfall forecasts for pre-defined, small areas
e) INCA (Integrated Nowcasting through Comprehensive Analysis) data

The availability of continuous data (and especially in real-time) on the part of the stakeholders still needs to be clarified, as well as whether access to the local IT system is possible at all. This is also very critical from the point of view of data protection. In general, as discussions are still ongoing with the stakeholders, other or additional datasets may arise as the project progresses. Nevertheless, it should also be noted here that in the case of Innsbruck these data might not be available.

The following datasets (cf. [D1.2, Section 3.2]) might be made available:

- Professional Fire Brigade Innsbruck:
  a) Deployment protocols
  b) Radio communication data (incl. location data)
  c) Operational documentation during/after the event (pictures, etc.) if available
- ECC Tyrol (Leitstelle Tirol):
  a) Emergency call data (except from the police)
- National Warning Center Tyrol (Landeswarnzentrale Tirol) [optional, not yet sure - still in discussion]:
  a) Deployment protocols
  b) Operational documentation during/after the event
  c) Drone recordings during/after the event (aerial photos, thermal images, terrain maps, etc.)
  d) Data collected and provided for the key stakeholders in the geoinformation-based systems (i) webGIS tiris OEI - Local operation information, and (ii) katGIS (Digital situation recording and situation information)
- IKB: Innsbrucker Kommunalbetriebe AG (municipal infrastructure service company) [optional, not yet sure - still in discussion]:
  a) Water level, (surface) runoff data of sensors in relation to the drainage system
  b) Supply bottlenecks (energy, electricity, etc.)
- Geosphere Austria [optional, not yet sure - still in discussion]:
  a) Meteorological raw data (precipitation, river and groundwater level, river flow rate of sensors)
  b) Rain radar data
  c) Weather forecasts incl. thunderstorm cells (national to local)
  d) Heavy rainfall forecasts for pre-defined, small areas
  e) INCA (Integrated Nowcasting through Comprehensive Analysis) data
- Department of Bridge and Hydraulic Engineering (City of Innsbruck):
  a) Hazard / susceptibility maps

However, as discussions are still ongoing with the stakeholders, other or additional datasets may arise as the project progresses.

3.1.7.3 Stakeholder Requirements

A first workshop was held with the stakeholders of the Innsbruck application scenario on June 6, 2023, and discussions were initiated to which extent the technology from CREXDATA can be used and what the requirements are. These are as follows:

- Weather forecasting in different time scales (depending on the type of event): In the case of forecasted pluvial floods, and early prediction is deemed necessary (lead time >>1h).
- Forecasting of cascading events such as debris flows.
- Forecasting of critical events with enough lead time, of e.g. overloading of the canal/drainage system, instability of buildings, flooding of underground constructions (parking lots, underpasses)
- Facilitate and simplify communication among stakeholders in the event of an incident.
- Use of AR during pluvial floods: Visualization of predicted water level at different time intervals, trapped people in floating cars, etc.
- Use of underwater robots and/or drones during pluvial floods: Detection of trapped, buried people in floating cars, or of hazardous materials in water (chemical hazards), etc.

However, as discussions are still ongoing with the stakeholders, other or additional requirements may arise as the project progresses.

3.1.8 Involvement of Finnish Experts

The involvement of Finnish experts, including Finnish Meteorological Institute (partner, FMI), The Ministry of Interior Finland (partner, MoIFI) and the Rescue Department of Helsinki (external stakeholder) is built around showcasing the use of machine learning in weather-related impact forecast and early warning tool development. The aim of the showcase is to negotiate with the data owners to find, utilize and make new datasets open within the project consortium as well as open for everyone. The Finnish showcase focuses not only on one, but on several weather hazards which have a large impact on Finnish emergency management in different seasons. The aim is to use statistical modelling and machine learning to produce forecasting products that describe the forecasts in a useful and concrete way for the end users in Finland and possibly also in other pilot locations, such as in Dortmund.

3.1.8.1 Application Scenario

Currently FMI is providing for instance following services for end users where machine learning can be utilized and where it can give a significant support to the impact estimations of forecasters or emergency managers:

- Severe Weather Warnings: FMI issues alerts for hazardous weather events, enabling proactive measures and emergency planning.
- Early Warning Systems: Collaboration with civil protection to develop systems that detect and forecast (the impacts of) extreme weather events, including prompt evacuation and response coordination.

- Specialized Forecasts: Tailored forecasts for specific sectors like emergency management assist in decision-making, minimizing risks and optimizing operations.
- Data Dissemination: Collecting, analysing, and disseminating weather- or weather impact-related data, enabling risk assessments, emergency planning, and response coordination.

The scenarios of the Finnish showcase are built rather on the impacts of different weather hazards than one specific weather hazard. In Finland a variety of weather-related hazards are experienced throughout the year. The autumn and winter season are dominated by strong windstorms with strong winds, and occasionally also with heavy snowfall conditions. These windstorms and falling trees cause lot of clearance tasks for the rescue department and inconvenience for public in form of power outages, damage to infrastructure, hazardous road conditions, and threat to human lives. One of the tools developed in the project is specifically tailored for forecasting the number of clearance tasks of upcoming windstorms. During heavy snowfall cases, the tool forecasting the number of road traffic accidents helps the rescue department to plan their resource management in a case of difficult winter storm and extreme road weather conditions (Figure 37).

During the summer months, Finland experiences in increasing frequency extreme heatwaves and dry weather conditions, which increase the risk of forest fires as well. Forest fires cause threat to people and infrastructure. The preparedness of the rescue department for forest fires can be increased by providing a tool that is estimating directly the number of wildfire fighting events on the Helsinki and Uusimaa region instead of traditional weather forecast predicting weather conditions and leaving space for individual interpretation.

The impacts of weather-related hazards are often consequences of combination of meteorological, environmental and infrastructural factors. The Finnish ML-based service can be taken up for instance in the complex event forecasting of T4.1. The impacts of the complex events may be difficult to grasp and understood solely by human brain, and thus machine learning algorithms are excellent aid for instance in the data analysis, recognizing complex patterns in weather data, and detecting early signs of specific events. ML service can be developed to create more accurate impact forecasts and enhancement of weather warnings or early warning systems, which often are missing the estimation of weather hazard impacts. The information and analysis of FMI's ML service can be also utilized in T5.2 (Visual analytics supporting XAI) in simplifying and interpretation of complex and vast weather data. The outcomes of T5.3 (Visual analytics for decision making under uncertainty) can be possibly tested as an extension of the ML service tools to visualize the uncertainties of the forecasts in the understandable way, which is currently missing in the existing ML based tools of FMI.

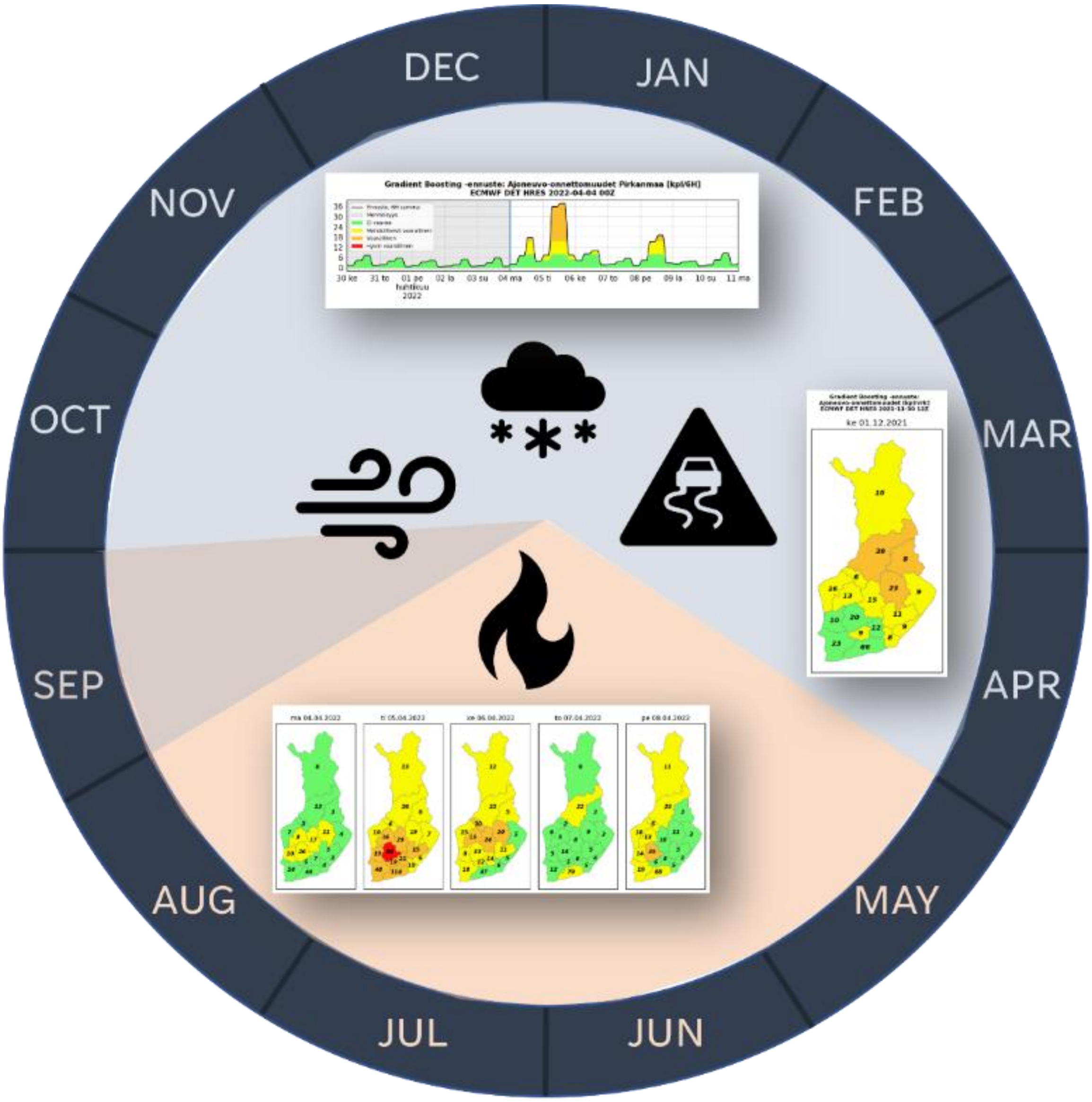


Figure 37: The seasonal occurrence of the main weather hazards in Finland (forest fires in summer, wind and snow hazards in winter). The developed impact forecast tools aim to address various impact variables induced by these hazards (Figure from CREXDATA Deliverable D2.1).

3.1.8.2 Available Data

In a Finnish national level SILVA-project (2020-2023), a comprehensive weather-related impact database was created. Of 13 collected impact datasets, four were made openly available. In CREXDATA we demonstrate the use of these four datasets presented in Table 19 with their volume, temporal and geographical coverage, and resolution. The advantage of these datasets especially from the machine learning perspective is that the time series lengths are long and the accuracy of both spatial and temporal resolution of the impacts is sufficient for training the model that requires large amounts of data to perform well. Additionally, currently the ECMWF HRES weather model is used, and later MEPS limited area ensemble weather prediction model is tested as the predictor data representing the weather-dependency of the impacts. In CREXDATA, the possibility of including new datasets and making them openly available is also being explored. The plan is also to utilize for instance the number of ambulance operations between 2007 and 2023. The open impact data can be made available to other partners also through ARGOS system, as well as selected meteorological data described more in details in D1.2.

Table 19: Impact datasets used for training and validating the gradient boosting machine learning method that are openly available. The ambulance operation dataset is a new dataset and the use is being explored in the project (not openly available) (Table from CREXDATA Deliverable D2.1, maintaining the Table numbering of D2.1):

DATASETS	VOLUME	TIME SERIES LENGTH	COVERAGE	RESOLUTION
Wind damage clearance	130k	22 yrs	National	• Municipal • 1 hour
Wildfire fighting events	63k	22 yrs	National	• Municipal • 1 hour
Traffic accident clearance	281k	22 yrs	National	• Municipal • 1 hour
Road traffic accidents	1.1M	24 yrs	National	• Municipal • 1 hour
Ambulance operations	~1M	16 yrs	Helsinki region	• Accurate coordinates • 1 second

3.2 Health Use Case

This use case assesses WP4 abilities to enable critical action planning and intervention by providing efficient parameter exploration forecasting and effective interventions in the modelling of epidemics and drug treatment optimization in COVID-19 infection. Additionally, this task specifies the requirements and scenarios that are used by the novel tools developed in T2.5. The evaluation is performed on two scenarios. In the epidemics scenario, we use epidemiological compartmental models to build a digital twin for COVID-19 transmission that identifies efficient policies and enables accountability. In the drug treatment scenario, we use multiscale mechanistic models to build a digital twin of a drug assay in COVID-19 patients that identifies the best treatment for each patient and condition [DoA, p.8].

3.2.1 Stakeholders

Epidemiological simulations in a health crisis have the potential to benefit a wide range of stakeholders. Government and public health agencies can leverage these simulations to make informed decisions and effectively allocate resources, implement effective Non-

Pharmaceutical Interventions and design optimal vaccination campaigns. By understanding the potential spread of diseases and evaluating the impact of different intervention strategies, policymakers can develop evidence-based policies and guidelines. Healthcare providers and hospitals can utilize the simulations to assess the strain on healthcare systems, such as the demand for hospital beds, ICUs, ventilators, and the healthcare workforce. This information aids in resource planning, capacity management, and optimizing healthcare delivery to meet the needs of the affected population.

Emergency management and disaster response agencies also find value in epidemiological simulations. These simulations assist in understanding potential scenarios, predicting resource requirements, and strategizing response plans. By incorporating simulation results into their preparedness efforts, these agencies can develop response protocols, coordinate multi-agency efforts, and ensure effective coordination and implementation of response measures. Additionally, researchers and academia benefit from epidemiological simulations as they provide a tool for testing hypotheses, exploring different scenarios, and analysing the potential impact of interventions. By contributing to scientific knowledge, simulations inform research directions and support the development of evidence-based guidelines for mitigating health crises.

Each of the identified stakeholders is interested in application scenarios for forecasting the evolution of new potential epidemics, detecting new outbreaks or wave and finding efficient intervention to reduce the impact from different perspectives. Likewise, they are also interested in novel, optimised drug treatments that provide alterative clinical care pathways for COVID-19 patients.

The list of functionalities proposed addresses the specific end-user requirements regarding the healthcare system. All developed functionalities target one or more of the following key healthcare system managers, public health decision-makers and data scientist/IT service provider needs.

We have taken advantage of the EBI's Competency Hub developed by us in a recent project to characterise some of these personas in more detail. The PerMedCoE competency framework defines a series of competencies required of professionals in the field of computational personalised medicine. A competency is an observable ability of any professional, integrating multiple components such as knowledge, skills and behaviours. The competencies that an individual might need to fulfil a particular role are listed in the reference profiles, which can be used to guide career choices[8].

The PerMedCoE competency profile builds on the work done in several related initiatives, as we took inspiration from the competency profiles of BioExcel[9], CINECA[10] and ISCB[11] to create an initial draft that was updated with feedback from experts from the community. All the profiles

---

[8] This competency framework is available at: https://competency.ebi.ac.uk/framework/permedcoe/ 2.1 and https://permedcoe.eu/deliverables/ (Deliverable 4.2).

[9] https://competency.ebi.ac.uk/framework/bioexcel/2.0

[10] https://competency.ebi.ac.uk/framework/cineca/1.0

[11] https://competency.ebi.ac.uk/framework/iscb/3.0

are freely accessible through the EMBL-EBI Competency Hub[12]. Additionally, the Competency Hub allows users to create their own profile on the site and to compare it with the existing profiles, which can inform about career development options.

Scenario 1: Designing Effective NPIs and Vaccination Campaigns for Controlling a Disease Outbreak using epidemiological modelling

In this scenario, a region is facing a sudden outbreak of a highly contagious infectious disease. The objective is to use epidemiological simulations to evaluate potential scenarios under different assumptions such as different reproduction numbers and fatality rates when those parameters are still unknown or hard to estimate due to the absence of data. In the second stage, optimization-via-simulation can be used for the design and implement effective Non-Pharmaceutical Interventions (NPIs) and vaccination campaigns to control the disease spread and minimize its impact on the population.

Scenario 2: Finding optimised drug treatments and alternative clinical care pathways for COVID-19 patients using multiscale modelling

In this scenario, a patient has been identified as being infected by SARS-CoV-2 and has been taking in charge by a hospital. The objective is to have a digital twin of the clinical care pathway using a multiscale model to propose clinical interventions of drugs and NPIs (such as mechanical ventilation) that allows the patient to have a healthy status. For this, first we couple two simulators (Alya and PhysiBoSS) and fit different parameters to clinically relevant variables. Second, we use optimization-via-simulation to design and implement patient-specific, effective combinations of interventions that heal the patients.

3.2.2 Available Data

As an initial assumption, no real-time data sources are available. Available data sets are described per health sub-scenario in the following Sections 3.2.2.1 and 3.2.2.2 on the following two pages.

3.2.2.1 Scenario 1: Epidemiological Modelling

The following datasets (cf. [D1.2, Section 4.2) are available on Zenodo. The open datasets consist of COVID-19 case reports and population mobility patterns in the form of origin destination matrices, both reported on a daily basis:

- COVID19 Flow-Maps GeoLayers dataset: Geographic layers on which the different data records are geo-referenced (e.g., mobility, COVID-19 cases). The different layers can be grouped into those that cover the whole territory of pain (e.g., municipalities) and those that are restricted to a specific region (Table1). Among those that cover the full territory of Spain, the record accounts for the first four levels of administrative division, that is, autonomous communities, provinces, municipalities and districts [22].
- COVID19 Flow-Maps Daily Cases Reports: This repository contains COVID-19 data for Spain, including daily cases at the level of autonomous communities as well as provinces, and higher spatial resolution for several autonomous communities (eight

[12] https://competency.ebi.ac.uk/framework/permedcoe/2.1

out of the nineteen autonomous communities publish reports with local daily COVID-19 cases at the level of municipalities or Basic Health Areas). Each record has an identifier, the associated date, the corresponding identifier of the layer and code of the region and a set of COVID-19 related fields, which include the number of new cases (daily incidence) and total cases. The dataset includes case reports for a time period of approximately two years [23].

- COVID19 Flow-Maps Daily-Mobility for Spain: This data-set contains daily aggregations of the hourly data provided by MITMA, aggregated at different levels of spatial resolution. The dataset includes Origin-Destination matrix for the mobility layer, with hourly resolution Each entry has a date and time period (the range between two consecutive hours), the origin and destination zones and the number of trips from a origin to a destination. Origin and destination zones correspond to geometries from the MITMA mobility layer and internal trips (same layer of origin and destination) are also reported. Additionally, it also includes a data record containing the trips per person matrix on each mobility area on a daily basis. This indicator reports population-based daily mobility behaviour. For each date and zone from the MITMA mobility layer, the indicator reports how many persons have performed 0, 1, 2 or more than 2 trips. While the indicator does not provide the destination of the trips, it accounts for the fractions of people performing at least one trip or none, as well as the estimated total population in that zone for the given date, considering as population those persons who stay overnight in the zone on that date [24].
- COVID19 Flow-Maps Population data. Daily population and trips per person data from Spain 2020-2022. This data record contains daily population records based on a study conducted by the MITMA, that analysed the mobility and distribution of the population in Spain from February 14th, 2020, to May 9th, 2021. The study is based on a sample of more than 13 million anonymised mobile phone lines provided by a single mobile operator whose subscribers are evenly distributed. Data provided by MITMA is related to the layer mitma_mov. For the rest of the layers, the population was estimated using the population grid from GEOSTAT[13] [25].

3.2.2.2 Scenario 2: Multiscale Lung Infection Modelling

- Anonymized patient omics molecular data. Publicly available pseudo-anonymized raw experimental data from patients. This dataset bundles different studies that are the input used to analyze and personalize our multiscale models. It potentially consists of transcriptomics, genomics, copy number variations and proteomics data.
- Anonymized patient pulmonary 3D positional data. Publicly available pseudo-anonymized image data from patients. This dataset bundles different studies that are the input used to have complex 3Dal setups for our multiscale models. Once analysed, this dataset have positional data for each of the alveoli of the patients.

13 https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/population-distribution-demography/geostat

3.2.3 System Architecture for Demonstrator

Table 5: Demonstrator components extending the CREXDATA system (Health Use Case) (Table from CREXDATA Deliverable D2.1):

Component	Description
Epidemiological Scenario	
T2.4 – Simulation and Tools	• Creation of a synthetic mobility dataset for evaluation of different scenarios • User interface development. The end-users of the user interface are able to view the forecast the evolution of the epidemic process under different scenarios • Simulation Framework: Integration of population mobility data with the epidemiological models. The architecture gives the capability to simulate/replicate a test scenario using both, real and synthetic data of the COVID-19 pandemic in Spain, with potential extensions to other countries.
T3.2 Graphical Workflow Specification	• Graphical tools, for instance using different operators in RapidMiner, is developed to allow a non-expert programmer to specify complex data processing workflows to enable the fusion of different spatiotemporal data sources georeferenced to different territorial units.
T4.1 Complex Event Forecasting	• NCSR/BSC develop novel algorithms to forecast new outbreak hot-spots based times series of cases, population mobility
T4.2 Interactive Learning for Sim. Exploration	• Extreme-scale model exploration is combined with interactive learning approaches to explore the large space of epidemiological parameters, as well as optimal interventions.
T4.3 Federated Machine Learning	• Federated Machine Learning (FML) offers a privacy-preserving approach to calibrating epidemiological parameters in a pandemic scenario. It allows multiple data owners to collaborate and train a shared machine learning model without sharing their sensitive data. The calibrated epidemiological parameters obtained through FML is validated using a time series of COVID-19 cases reported at different levels of spatial aggregations. Robust statistical methods can be applied to assess the accuracy and uncertainty of the calibrated parameters.
T5.3 Visual Analytics for Decision Making under Uncertainty	• Creates visual representations of the epidemic model outputs. This can include interactive maps, charts, graphs, and dashboards that depict the spread of the disease over time, hotspots of infection, and the effectiveness of interventions.

Component	Description
Multiscale lung infection Scenario	
T2.4 – Simulation and Tools	• Simulators: Coupling organ-level with cell-level simulation tool • Study the use of surrogate models for parts of the algorithms. • Use model exploration to fit some of the parameters using clinical data or desired simulated behaviours. • Prepare a set of design variables that control the treatment of patients (drug and mechanical interventions) that is inspected using interactive learning.
T3.2 Graphical Workflow Specification	• Graphical tools, for instance using different operators in RapidMiner, is developed to allow a non-expert user to use and browse complex workflows that simulate patient treatments using clinical data and bedside variables.
T4.1 Complex Event Forecasting	• NCSR and BSC develop novel algorithms to forecast at early times the outcomes of a lung infection simulation.
T4.2 Interactive Learning for Sim. Exploration	• Extreme-scale model exploration is combined with interactive learning approaches to explore the large space of potential clinical interventions and simulate the patient's clinical care pathway until recovery.
T5.3 Visual Analytics for Decision Making under Uncertainty	We create visual representations of different simulations of the multiscale infection model. For instance, we study the usefulness of GUIS and dashboards to ease the exploration of the parameter sensitivity analyses and their effect on model outputs.

Table 6: Uptake of technologies in the Health Use Case (Table from CREXDATA Deliverable D2.1):

Specific “use cases”	Parameters calibration and optimal intervention design	Forecasting of outbreak hotspots	Fusion of different spatiotemporal data sources	Forecasting of lung infection dynamics	Interactive learning of COVID19 patients clinical care pathway
T2.4 Simulation and Tools	X	X		X	X
T3.2 Graphical Workflow Specification			X		X
T4.1 Complex Event Forecasting	X	X		X	X
T4.2 Interactive Learning for	X				X

Specific "use cases"	Parameters calibration and optimal intervention design	Forecasting of outbreak hotspots	Fusion of different spatiotemporal data sources	Forecasting of lung infection dynamics	Interactive learning of COVID19 patients clinical care pathway
Simulation Exploration					
T4.3 Federated Machine Learning	X				
T4.4 Optimized Distributed "Analytics as a Service"					
T5.1 Explainable AI					
T5.2 Visual Analytics supporting XAI					
T5.3 Visual Analytics for Decision Making under Uncertainty	X	X		X	X
T5.4 Augmented reality at the field					
T5.5 Uncertainty Visualization in Augmented Reality					

3.3 Maritime Use Case

The scenarios is validated using created streams of data and in sea trial experiments with numerous vessels with several levels of autonomy as defined by the International Maritime Organization (IMO) in cooperation with the SMARTMOVE Lab of the University of the Aegean (UoA). Sea trials under realistic conditions guarantee an application-related real-world assessment. Data collection and sea trials are conducted during dedicated experiments at the Aegean University sea testbed and during the Aegean Ro-boat Races planned to take place annually in the summer periods starting in July 2023. The sea testbed of the University of Aegean is located on the island of Syros in the Aegean Sea and offers proximity to open sea testing grounds. It covers sea and land utilizing 100% WiFi coverage and is capable of hosting and deploying several types of UVs: air, land, sea and subsurface. A running prototype version of the IoT-Voyage Data Streamer – VDS was deployed at the 1st Aegean Ro-boat Race in July 2023. The sea trials are used to collect real world datasets from onboard the competing vessels, which is used to test components in real world maritime conditions.

3.3.1 Stakeholders

The following key stakeholder groups are identified in the context of the Maritime Use Case. Each stakeholder has specific interests directly linked to aspects regarding maritime safety, or is accountable for ensuring safe maritime operations, safe navigation as well as passenger, crew and cargo safety and is part of the decision making and accountability chain in the context of hazardous maritime events detection, forecasting and mitigation.

Table 7: Key stakeholder groups & roles of the Maritime Use Case (Table from D2.1):

Key stakeholder group	Key stakeholder roles
Action planner	• Vessel pilot • Vessel crew • VTS operator • Port/coastal authorities • Remote operator
Decision maker	• Ship deck officers • VTS operator • Port/coastal authorities • Fleet managers • Vessel owners • Remote operator
System administrator	• IT / Data Science staff of the maritime service provider • IT / Data Science staff of the port authority • IT / Data Science staff of the vessel owners • Fleet managers • Insurance companies
Workflow designer	• IT / Data Science staff of the maritime service provider • IT / Data Science staff of the port authority • IT / Data Science staff of the vessel owners • Fleet managers • Insurance companies

Each of the aforementioned stakeholders are interested in application scenarios for global vessel safety, tracking and management from his/her own perspective. The envisaged functionalities address the specific end-user requirements regarding maritime safety. All developed functionalities potentially target one or more of the following key maritime users and data scientist/IT service provider needs. Resulting functionalities that are listed below are in line with the importance evaluation results of the user requirements survey contacted by MT (Kpler) (see service importance evaluation Table 12):

- Vessel's route analysis and/or prediction.
- Early warnings of possible collisions and near-real time collision avoidance.
- Early warnings of possible intrusion of sea areas with hazardous weather conditions.
- Rerouting and mitigation actions for collision mitigation
- Rerouting and avoidance of sea areas with forecasted hazardous weather conditions.

The key personas involved directly in the detection and management of a maritime emergency are identified in Table 7.

3.3.2 Application scenario

In the context of the Maritime Use Case two application scenarios is examined:

- collision forecasting and rerouting
- hazardous weather rerouting

Both application scenarios that are part of the pilot is evaluated in sea trial experiments using the vessel of high autonomy (test vessel) of the SMARTMOVE Lab of the University of Aegean. Regarding the collision forecasting application scenario, a collision event is simulated at the sea test bed with the test vessel forecasting the imminent collision event in a short-term time horizon of under 15 minutes according to the KPIs. In a subsequent step a rerouting option is provided in order to mitigate the collision event. Regarding the hazardous weather rerouting, using simulated and synthetic data streams, first a sea area is designated as an area with hazardous weather conditions. The test vessel is provided with information regarding its imminent approach to the area with hazardous weather conditions along with rerouting instructions. The demonstration of both events during sea trial experiments at the University of Aegean Sea test bed validates the pilot of the maritime use case under realistic conditions and guarantee an application-related real-world assessment.

Table 8 and Table 9 present the high-level usage scenarios for the CREXDATA Maritime Use Case incorporating the identified key stakeholders and considering the results of the user requirements survey (see: Section 3.3.5 Stakeholder Requirements). For each of the scenarios, apart from the actor, the overview and the detailed description, the main benefits and challenges are provided.

Table 8: Collision forecasting and rerouting high level scenario description (Table from CREXDATA Deliverable D2.1):

Attribute	Description
ID	Maritime_UC01
Name	Collision Forecasting and Rerouting
Short description	Forecasting probable collision events among vessels in a short time prediction horizon of under 15 minutes. In case of a collision event detection, the proposed service informs the relevant actors of the collision event detection and provides rerouting suggestions for the mitigation of the collision event. The actors decide either to follow the service's instructions or correct the proposed rerouting suggestion based on their own local view
Author	MT (Kpler)
Last update	09.06.2023
Actors	• Vessel pilot • Vessel crew

Attribute	Description
	• VTS operator • Ship deck officers
Additional Actors	• Vessels of different levels of autonomy
Actors interested in the outcomes	• Port/coastal authorities • Fleet managers • Vessel owners • Insurance companies
Detailed scenario	The pilot of vessel A is alerted of a possible collision event with vessel B, as their current routes will intersect. The pilot evaluates the emergency of the forecasted event and is provided with a set of alternative routes for vessel A to follow in order to avoid colliding with vessel B. Based on the experience of the vessel pilot, he/she may opt to accept the proposed route, correct the proposed suggestion or follow an entirely different route in order to avoid collision with vessel B
Benefits	• Route monitoring and collision event forecasting for vessel traffic increasing safe navigation and efficiency of maritime operations • Improved route predictions of the vessel traffic through fusion of local and global data streams • Automation of collision event detection forecast and mitigation steps through automated rerouting suggestions. • Increased situational awareness • Early identification of possible collision events providing comfortable response time windows and informed decision support • Exploration of collision mitigation alternatives • Vessel crew and VTS operator work effort alleviation during traffic monitoring, decision making and action planning for vessel collision events
Challenges	• Accuracy of vessel path prediction in short term time horizons • Accuracy of path planning and rerouting for collision event mitigation • Near real-time response of the system for collision event detection • Near real-time fusion of different stream inputs from local and global data sources • Quantification of prediction and solution uncertainty

Attribute	Description
	• Real time re-evaluation of vessel route after collision event detection and continuous monitoring of the involved vessels' routes • Small response time path planning methods for rerouting • Real-time extraction of extreme-scale situational data

Table 9: Hazardous weather rerouting high level scenario description (Table from CREXDATA Deliverable D2.1):

Attribute	Description
ID	Maritime_UC02
Name	Hazardous weather rerouting
Short description	Rerouting of vessels in order to avoid sea areas with forecasted hazardous weather conditions. Based on weather forecast updates vessels are monitored and alerted on changes of the weather conditions along their route and provided with rerouting instructions
Author	MT (Kpler)
Last update	09.06.2023
Actors	• Vessel pilot • Vessel crew • VTS operator • Ship deck officers
Additional Actors	• Vessels of different levels of autonomy
Actors interested in the outcomes	• Port/coastal authorities • Fleet managers • Vessel owners • Insurance companies
Detailed scenario	The crew of a vessel starts their journey and plan their route according to their initial weather forecast. As weather dynamically changes over the journey, the vessel crew receives hourly updates in case of weather conditions influencing the passage safety through specific sea areas. In case of changes affecting the safe passage, an alert with automatic rerouting suggestion is generated by the system alleviating the vessel crew from the task of continuously monitoring the weather conditions and updating the vessel route.
Benefits	• Route monitoring and rerouting according to updated weather forecasts for global vessel traffic, increasing safe navigation and efficiency of maritime operations.

Attribute	Description
	• Improved rerouting of the vessel traffic according to forecasted weather conditions through fusion of local and global data streams. • Automation of the weather monitoring during a vessel's journey • Safe navigation due to avoidance of hazardous weather areas. • Increased situational awareness and early identification of hazardous weather conditions that give operators time to plan an alternative route. • Informed decision making • Effective fleet intelligence and management based on future weather forecast at global scale • Vessel operators are alleviated from weather forecast monitoring tasks
Challenges	• Accuracy of rerouting as a function of weather forecast uncertainty • Reliability on longer vessel routes with duration exceeding the weather forecast time range • Fusion of different stream inputs from local and global data sources • Quantification of prediction and solution uncertainty

3.3.3 Available Data

The following data sources are collected during the annual Aegean Ro-Boat races organized by the University of the Aegean (UoA). Local environment data sources refer to sensors mounted on the UoA vessel. Additionally, the MT (Kpler) IoT-Voyage Data Streamer – VDS are the only additional sensor mounted on both the UoA vessel and the RoBoat Race participating vessels during the races. Data from the RoBoat Race is made available for batch processing to the CREXDATA partners after the end of the annual RoBoat Race (the partners could be in position to simulate the timeseries data in streaming mode, if applicable per dataset type). All the data sources listed here could be potentially used for the needs of the Maritime Use Case (see appendix of Deliverable D2.1).

The following datasets (cf. [D1.2, Section 5.2]) are available on Zenodo. The open datasets consist of AIS related data and available for all partners to work on. Additional data is generated and provided through the annual Aegean Ro-Boat Races organized by the University of Aegean that are publicly released on Zenodo:

- Single Ground Based AIS Receiver Vessel Tracking Dataset: This dataset published by MT (Kpler), contains all decoded messages collected within a 24h period (starting from 29/02/2020 10PM UTC) from a single receiver located near the port of Piraeus

(Greece). All vessels' identifiers such as IMO and MMSI have been anonymized and no down-sampling procedure, filtering or cleaning has been applied.

- Heterogeneous Integrated Dataset for Maritime Intelligence, Surveillance, and Reconnaissance: This dataset contains ships' information collected through the Automatic Identification System, integrated with a set of complementary data having spatial and temporal dimensions aligned. The dataset contains four categories of data: Navigation data, vessel-oriented data, geographic data, and environmental data. It covers a time span of six months, from October 1st, 2015, to March 31st, 2016 and provides ships positions within the Celtic Sea, the Channel and Bay of Biscay (France). The dataset is proposed with predefined integration and querying principles for relational databases. These rely on the widespread and free relational database management system PostgreSQL, with the adjunction of the PostGIS extension, for the treatment of all spatial features proposed in the dataset.
- The Piraeus AIS Dataset for Large-scale Maritime Data Analytics: The AIS dataset (coming from MT's (Kpler's) receiver) comes along with spatially and temporally correlated data about the vessels and the area of interest, including weather information. It covers a time span of over 2.5 years, from May 9th, 2017, to December 26th, 2019, and provides anonymized vessel positions within the wider area of the port of Piraeus (Greece), one of the busiest ports in Europe and worldwide. The dataset consists of over 244 million AIS records, an average of more than 10,000 records per hour, which makes it an ideal input for large-scale mobility data processing and analytics purposes.
- Hellenic Trench AIS Data: Data from Automatic Identification System (AIS) transmissions received from both satellite and terrestrial receivers of the Marine Traffic network (www.marinetraffic.com) for one year (31 July 2015 to 31 July 2016) along the Hellenic Trench, the core habitat of the eastern Mediterranean.

3.3.4 System Architecture for Demonstrator

Table 10: Demonstrator components extending the CREXDATA system (Maritime Use Case) (Table from CREXDATA Deliverable D2.1):

Component	Description
T2.4 – Simulation and Tools	• Creation of a synthetic dataset of simulated AIS data. • User interface development. The end-users of the user interface are able to view the forecast motion of vessels in the future of each predicted route and potential mitigation actions • Simulation Framework: Integration of synthetic collision data with the VR interface. The architecture gives the capability to simulate/replicate a test scene using real historical and synthetic data. • Define paradigms for interactive exploration of the model behaviours using synthetic simulation data using VR
T3.2 Graphical Workflow Specification	• Integration of the maritime use case applications with the CREXDATA platform

Component	Description
	• Integration with the RapidMiner graphical workflow. Extension from INFORE with new operators: Fusion operator, forecasting operator and rerouting operator • Integration of existing RapidMiner operators from INFORE: Maritime Event Detector, Fusion • Development of new Kafka streams and related operators
T3.3 – System Integration and Released Software Stacks	• Development of software prototype
T4.1 Complex Event Forecasting	• MT (Kpler) / UoA develop in-house models for route and collision forecasting • MT (Kpler) /UoA develop an in-house solution for hazardous weather rerouting
T4.4 Optimized Distributed "Analytics as a Service"	• AKKA distributed tool to run fusion and route prediction models • Technical details and goals is clarified at a later stage. Potential provision of AKKA performance analytics for resource optimization
T5.3 Visual Analytics for Decision Making under Uncertainty	• The pilot supports TUC for visual Analytics to facilitate decision making for the collision and weather rerouting under uncertainty
T5.4 Augmented reality at the field	• The pilot supports TUC for an AR/VR prototype on the maritime use case
T5.5 Uncertainty Visualization in Augmented Reality	• The pilot supports TUC for uncertainty visualization in augmented reality on the maritime use case.

Table 11: Uptake of technologies in the Maritime Use Case (Table from CREXDATA Deliverable D2.1):

Specific "use cases"	Collision Forecasting	Hazardous Weather Rerouting
T2.4 Simulation and Tools	X	X
T3.2 Graphical Workflow Specification	X	X
T4.1 Complex Event Forecasting	X[1]	X[1]
T4.2 Interactive Learning for Simulation Exploration		
T4.3 Federated Machine Learning		
T4.4 Optimized Distributed "Analytics as a Service"	X[2]	X[2]
T4.5 Text Mining for Event Extraction	Not relevant	Not relevant
T5.1 Explainable AI		

T5.2 Visual Analytics supporting XAI		
T5.3 Visual Analytics for Decision Making under Uncertainty	(X)[3]	
T5.4 Augmented reality at the field	(X)[3]	
T5.5 Uncertainty Visualization in Augmented Reality	(X)[3]	

[1] MT's (Kpler's) models for route and collision prediction are developed
[2] Akka distributed framework is used to run fusion and route prediction models
[3] support of potential TUC contribution

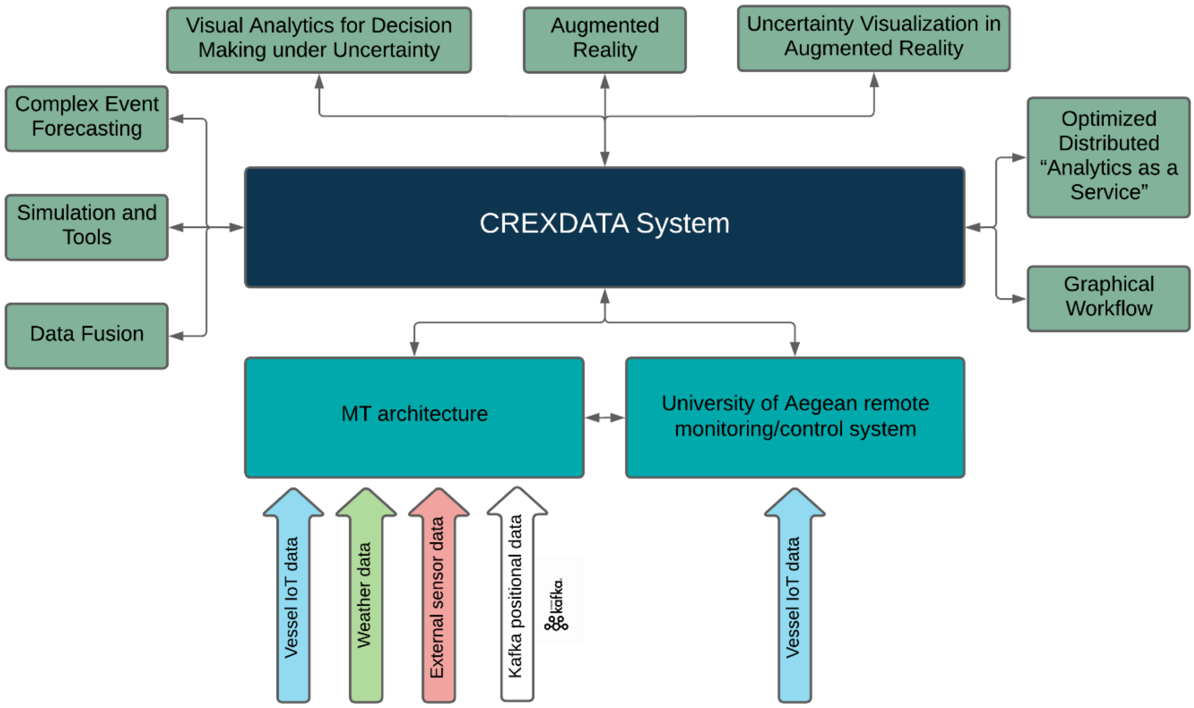


Figure 38: High-level system architecture for the Maritime Use Case interlinked with the CREXDATA system and its components (Source: CREXDATA Deliverable D2.1).

Figure 39 presents the MarineTraffic (Kpler) system architecture supporting the deployment of the Maritime Use Case pilot.

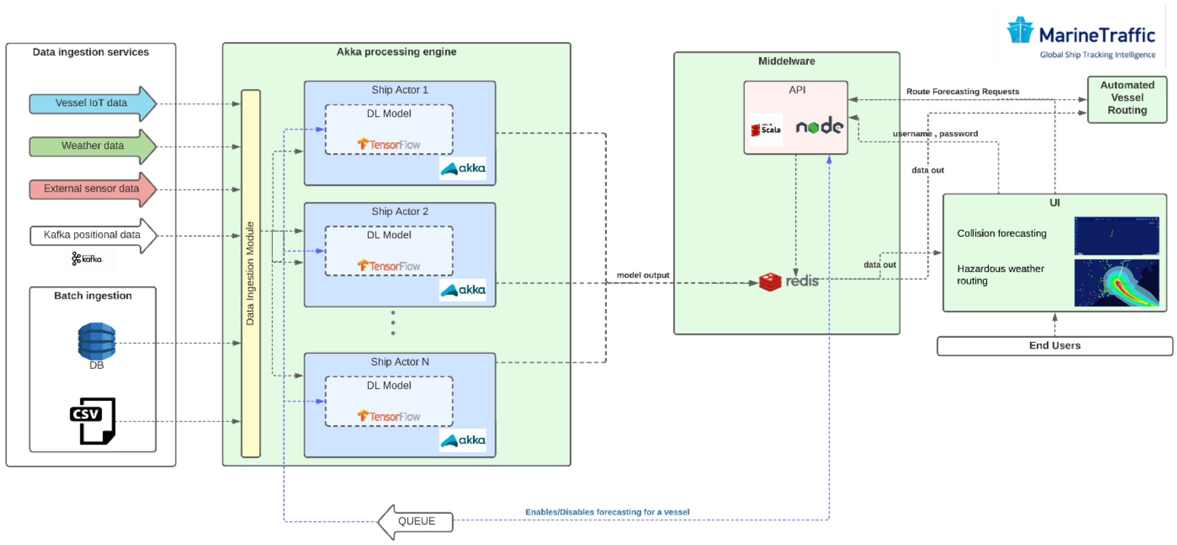


Figure 39: MarineTraffic (Kpler) system architecture for the Maritime Use Case (Source: CREXDATA Deliverable D2.1).

3.3.5 Stakeholder Requirements

CREXDATA deliverable D2.1 describes the stakeholder requirements for the maritime use case collected in a survey. With respect to the importance of forecasting the defined hazardous maritime events, maximum acceptable forecasting latency and the received information from the corresponding forecasting service, results for the respective requirement with the maximum number of consensus votes are presented in Table 12 (see next page).

Table 12: Maritime Use Case User Requirements (Table from CREXDATA Deliverable D2.1):

Requirement	Collision forecasting MAR_1	Hazardous weather routing MAR_2
service importance	very important	very important
maximum acceptable latency	minute latency	hour latency
provided information features to the end-users	• ETA to conflict point • prediction confidence • rerouting information with path suggestion	• ETA to destination port • prediction confidence • rerouting information with path suggestion
suggestions for possible courses of action	yes	yes
frequency of updating mitigation actions	every minute	every hour
data sources / data sets for development	• real-time / streaming AIS data • real-time/Streaming IoT vessel data • historical AIS data	• real-time / streaming weather data • historical weather data
data sources/ data sets for evaluation	• Sea trial/experimental data	• Historical vessel data

In the context of facilitating informed decision making by the end-users during hazardous maritime events, the survey defines the respective requirements for the development of such services. In the context of the CREXDATA platform, according to the respective questionnaire participants the development of such services should be based on specific data features that may be available through different data sources.

3.4 Federated Data Processing and Federated Machine Learning

In addition to the requirements of the three use cases described in the three previous sections 3.1 to 3.3, there are also requirements from an algorithmic and technological perspective: The CREXDATA platform and streaming architecture need to be able to support federated data processing and federated machine learning as well:

- Federated machine learning deals with training models or applying models at the edges of the network, i.e. distributed across multiple IoT or edge devices, some of which may have limited resources regarding the amount of available memory and compute power.
- A streaming process can be split up by the optimizer component of the CREXDATA platform to be placed for execution on multiple compute clusters (which could be located at different locations) and/or on multiple edge devices. Federated data processing and federated machine learning process data in a distributed way, i.e.

without storing or passing all of the data to a central location, but by processing it locally and only passing on the results, e.g. only relevant observations and events, aggregates, extracted features, or machine learned models.

- The above requires an interface between the CREXDATA platform and the federated data processing and federated machine learning services, which exposes the methods that (1) can be queried to get the number of nodes and a list of nodes available, and/or (2) provide configuration parameters in order to trigger training of a model using a particular algorithm and parameters on the federated data processing resources (compute nodes, federated machine learning algorithm and its parameters, etc.).
- The query service may also be used by the optimizer component to get statistics or related information about the different compute clusters and nodes supported by the cluster for federated data processing and federated machine learning purposes.
- The outcomes of the training or scoring action taking place on the federated data processing backend service is to be consumed just as in other streaming workflow operators.
- The federated data processing backend could be packaged as a bunch of docker containers with pre-configured python environment and packages installed, so that the workflow operators do not have to invoke installation and configuration methods or pass scripts. Dockerized backends also enable more portability and scalability.

For the communication between the CREXDATA platform, the CREXDATA optimizer, and the federated data processing and federated machine learning services, JSON-based web services and Kafka-based event and data streams are used.

In order to integrate the distributed data processing and the distributed machine learning into CREXDATA data processing workflows, a nested operator, i.e. a RapidMiner operator with a subprocess, is implemented in the CREXDATA project, named "Edge Processsing Nest", and made available as a Edge-to-Cloud Extension. The subprocess of the Edge Processing Nest operator is the part of the data processing workflow that is to be distributed to and executed on the edge devices. Hence, it is possible to design data processing workflows that combine distributed parallel execution of data processing on multiple edge and IoT devices (via the Edge Processing Nest operator) with non-distributed centrally executed data processing, e.g. for collecting and merging the results of the distributed data processing and federated machine learning.

The location, where a particular data processing workflow or a subprocess of it are to be run, i.e. the IP address of the node to execute the data processing workflow or subprocess thereof may be changed by the CREXDATA optimizer.

The CREXDATA platform maintains a node registry for managing a list of the IP addresses and port numbers of all available data processing nodes (compute nodes) including IoT and edge devices, compute clusters, servers, etc.

The list of nodes to be used for federated processing can be a subset of the list of all available nodes, i.e. the CREXDATA platform allows to filter the list of nodes, e.g. by selecting or deselecting nodes in the list individually or by filtering nodes by node type, category of nodes, group of nodes, or use case application id.

The CREXDATA optimizer component can adjust, where which part of the data processing workflow (i.e. operator) is run. The optimizer only needs to change the IP addresses and port numbers of the nodes to specify where to execute the particular part of the data processing workflow (i.e. operator). The CREXDATA process execution engine then takes these IP addresses and port numbers and runs the operators on the corresponding nodes, i.e. IoT and edge devices, servers, and clusters, respectively.

The integration from the edge to the central CREXDATA system m/ay also include other systems and components. Example for the weather emergency use case: Instead of a sending its data directly to the CREXDATA system, an edge device like a drone taking arial video footage, photos, etc. can also connect to a use-case-specific system (e.g. ARGOS by HYDS), which in turn connects to the CREXDATA system.

4 System Components and Integration

4.1 RapidMiner Kafka Extension

4.1.1 RapidMiner Extension for Kafka (Marketplace Version by RM)

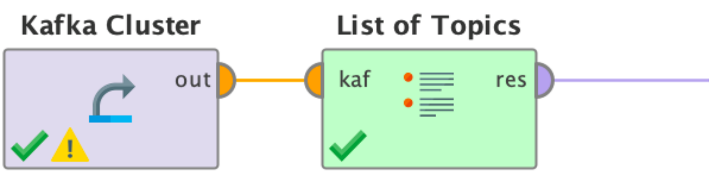
The *RapidMiner Kafka Connector Extension* available from the RapidMiner Marketplace (https://marketplace.rapidminer.com/UpdateServer/faces/product_details.xhtml?productId=rmx_kafka_connector) provides RapidMiner Operators for reading from and writing to Kafka streams (*Read Kafka* and *Write Kafka*, respectively). However, the extension version available on the marketplace currently only supports reading Kafka streams from the beginning, but not starting to read mid-stream or from the end of the stream. Hence, within CREXDATA, this extension was extended to meet the requirements of the CREXDATA project and to allow reading from the end of the data stream as well. Two extended versions of this RapidMiner extension were implemented, one by TUC based on the Java version available on the RapidMiner marketplace and one by RM based on Python scripts encapsulated into new versions of these RapidMiner operators. Both provide extended versions of the RapidMiner operator *Read Kafka* and are described in the following two sections.

4.1.2 RapidMiner Extension for Kafka (Python-Based Version by RM)

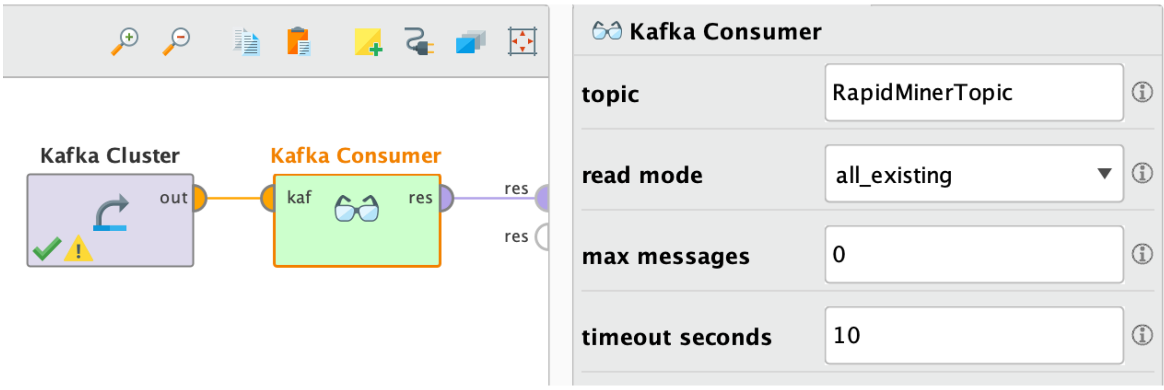
The Altair RapidMiner Extension for Kafka is an integration toolkit that exploits RapidMiner's Python Transformer Operator to develop production-grade, Python-based custom operators for Apache Kafka interactions. This extension enables seamless data streaming capabilities within RapidMiner workflows, facilitating real-time data ingestion, transformation, and distribution across distributed systems.

The extension comprises three primary custom operators:

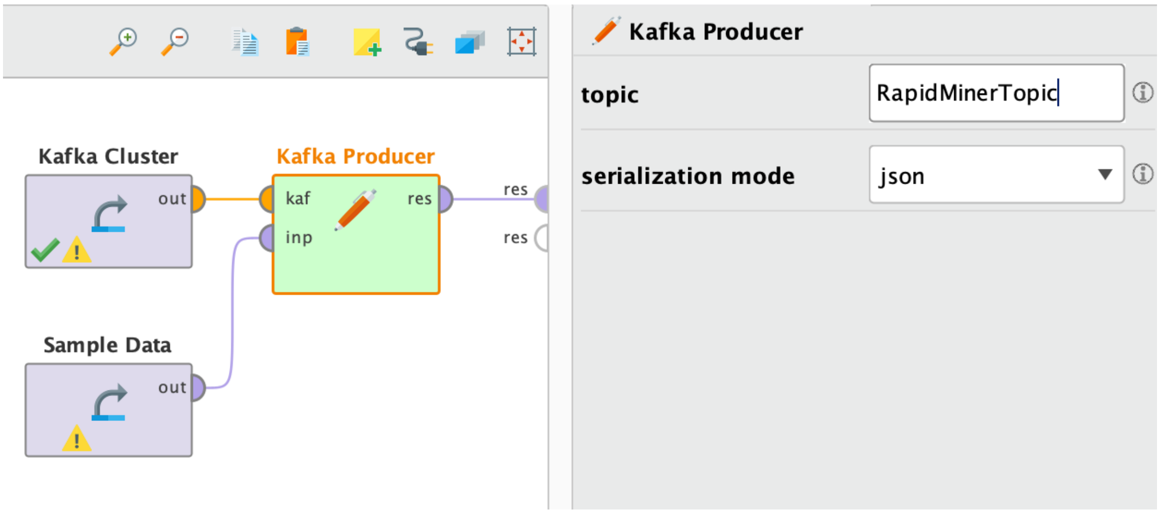
- **List of Topics**: Provides a list of available Kafka topics on a specified broker cluster. This operator retrieves topic metadata including partition count and topic name. It returns a structured RapidMiner ExampleSet enabling workflow designers to browse topics and/or use the available topic in data pipelines.



- **Kafka Consumer**: Implements flexible message consumption from Kafka topics with multiple read modes optimized for different use cases:
  - **All Existing Messages**: Retrieves complete historical data from topic inception (earliest offset)
  - **New Messages Only**: Streams incoming messages in real-time from the latest offset
  - **Last N Messages**: Extracts the N most recent messages for windowed analysis



- **Kafka Producer**: Enables publishing of processed data back to a selected Kafka topic with configurable serialization options.



The source of the Python-based RapidMiner extension for Kafka is available from the CREXDATA GitHub repository at:

https://github.com/altairengineering/crexdata-public/tree/main/CREXDATA%20System%20Components/RapidMiner%20Extensions/RapidMiner%20Kafka%20Extensions

4.1.3 Extended Kafka Reader (Java-Based Version by TUC)

The *Read Kafka* topic operator available from the RapidMiner Marketplace exhibits two main issues, detailed below:

1. The *Latest* offset strategy option throws an exception, and the operator can read data only from the beginning of a topic. Having a correct offset management is important because RTS Agents (RTSA) read data in mini-batches after each execution restart. The operator must support reading from the latest offset to avoid reprocessing the entire topic records. The current failure of the "latest offset" strategy prevents the correct incremental consumption.
2. The operator alters the format of the consumed data. The operator transforms the structure of the incoming Kafka records, as illustrated in Figure 40. In Edge-to-Cloud Optimization use case, where workflows are split and distributed across multiple RTS Agents, the data exchange between split operators relies on strict preservation of record format. The previous Kafka Topic operator assigned each record a key-value pair in which the value contained the original record in a stringified form (Figure 40). When workflows are distributed across three RTS Agents, two intermediate Kafka write/read stages are required. With the current operator, each read operation further transforms the data format, producing nested key/value fields (Figure 41). This repeated nesting breaks format consistency and affects downstream processing.

Row No.	Future Cus...	Age	Gender	Payment M...
1	yes	64	male	credit card
2	yes	35	male	cheque
3	yes	25	female	credit card
4	no	39	female	credit card
5	yes	39	male	credit card
6	no	28	female	cheque
7	yes	21	female	credit card

Initial data

Row No.	key	value	partition	offset
1		{"Age":64,"Gender":"male","Payment Method":"credit card","Future Customer":"yes"}	0	0
2		{"Age":35,"Gender":"male","Payment Method":"cheque","Future Customer":"yes"}	0	1
3		{"Age":25,"Gender":"female","Payment Method":"credit card","Future Customer":"y...	0	2
4		{"Age":39,"Gender":"female","Payment Method":"credit card","Future Customer":"no"}	0	3
5		{"Age":39,"Gender":"male","Payment Method":"credit card","Future Customer":"yes"}	0	4

Consumed data

Figure 40: Example of consumed records format transformation

Row No.	key	value	partition
1		{"key":"","value":"{\"Age\":64,\"Gender\":\"male\",\"Payment Method\":\"credit card\",\"Future Customer\":\"yes\"}","partition":0,"offset":0}	0
2		{"key":"","value":"{\"Age\":35,\"Gender\":\"male\",\"Payment Method\":\"cheque\",\"Future Customer\":\"yes\"}","partition":0,"offset":1}	0
3		{"key":"","value":"{\"Age\":25,\"Gender\":\"female\",\"Payment Method\":\"credit card\",\"Future Customer\":\"yes\"}","partition":0,"offset":2}	0
4		{"key":"","value":"{\"Age\":39,\"Gender\":\"female\",\"Payment Method\":\"credit card\",\"Future Customer\":\"no\"}","partition":0,"offset":3}	0
5		{"key":"","value":"{\"Age\":39,\"Gender\":\"male\",\"Payment Method\":\"credit card\",\"Future Customer\":\"yes\"}","partition":0,"offset":4}	0
6		{"key":"","value":"{\"Age\":28,\"Gender\":\"female\",\"Payment Method\":\"cheque\",\"Future Customer\":\"no\"}","partition":0,"offset":5}	0

Figure 41: Example of broken records format after two cycles of read/writes.

To address the aforementioned issues, the TUC Read Kafka Topic operator has been introduced. The updated operator allows users to set a custom group.id, ensuring that records are not re-consumed, and it reads records directly in JSON format, thereby preserving the original data structure. The operator, its parameters, and an example demonstrating data-format preservation are shown in the Figure 42.

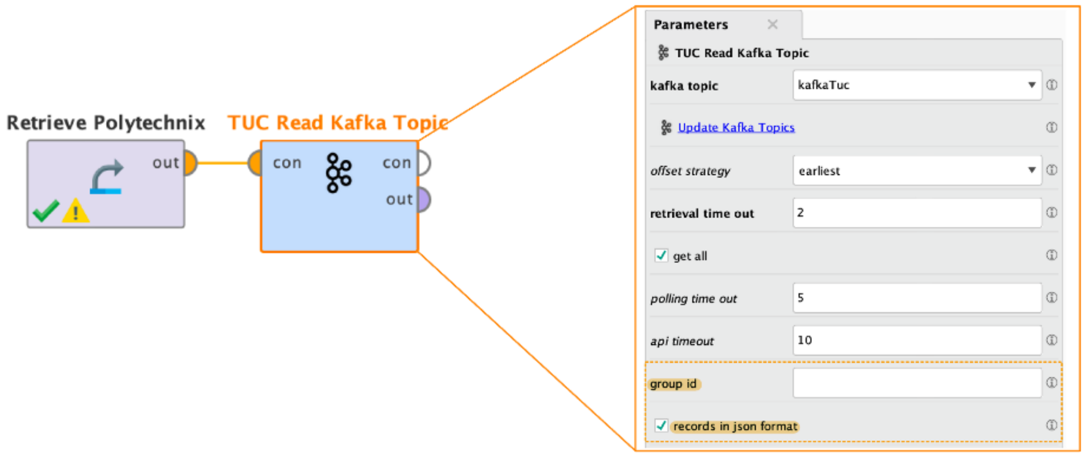


Figure 42: TUC Read Kafka Topic operator and its parameters.

The source of the Python-based RapidMiner extension for Kafka is available from the CREXDATA GitHub repository at

https://github.com/altairengineering/crexdata-public/tree/main/CREXDATA%20System%20Components/RapidMiner%20Extensions/RapidMiner%20Kafka%20Extensions

4.2 Multi-Modal Data Fusion Toolbox (MMDF) (UoA)

4.2.1 Overview

Nowadays, there is a great variety of sensors that are used to monitor and trigger actions in the context of a specific task. In complex systems where multiple components or agents are involved, the monitoring capabilities of each sensor, the environment and the nature of purpose of the system itself define the number of sensors required and their specifications. Each sensing system typically monitors/acts upon/detects a specific medium or object type providing its output on a predefined format, frequency and precision. Data fusion combines records of different sensors to provide more accurate and comprehensive information.

As the number of sensing capabilities increases, so does the modality of a system, the complexity of date acquisition, monitoring and detection process. In autonomous systems such tasks are required for environment perception while decision support systems contribute to situational awareness. There is a need to facilitate streaming information flow in a scalable way across such systems.

In this context, we have developed a multi-modal data fusion (MMDF) engine and a compatible workflow designer that relies on *Apache Kafka* streaming engine. The data fusion engine focuses on fusion of different sources of streaming data on location and time relevance, handling different modalities of spatial resolutions and sensing frequency. In addition, binding with batch data based on the location and timeframe is also supported through either differently from records (shapefiles) or their metadata (video streams).

The developed toolbox supports three different options for runtime execution. First, it can be dispatched to a Java-supporting system with access to the Kafka cluster provided with the configuration file and the core fusion engine jar executable. Second, the *RapidMiner AI Studio* can trigger execution through a developed extension on the local machine. Third, it can dispatch execution to the *RapidMiner AI Hub*.

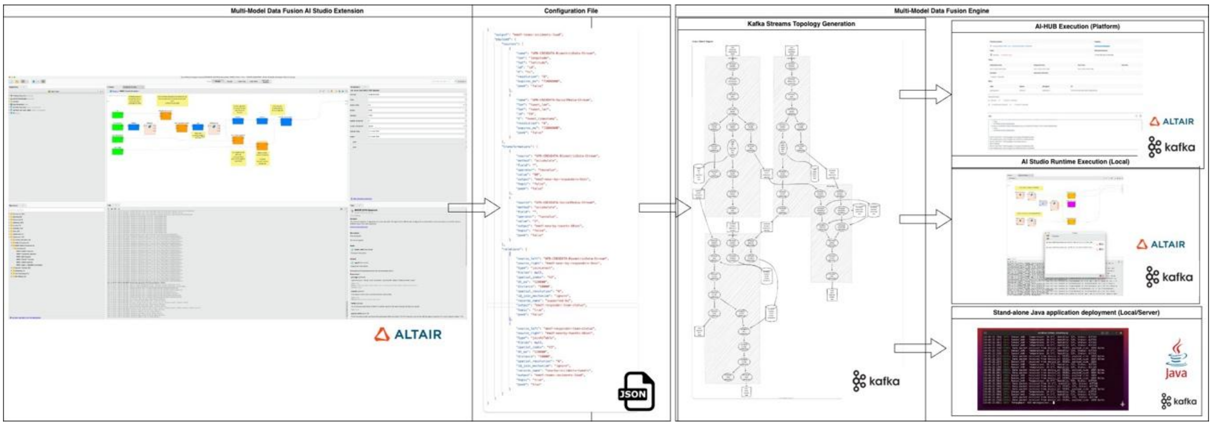


Figure 43: Multi-modal data fusion toolbox components and execution options.

4.2.2 State-of-the-Art and Contributions

Data fusion or **multi-sensor data fusion** [90] studies techniques to combine data from multiple sensors (of the same or different types) to achieve inferences that are not feasible from a single sensor. When sensor data is also combined with data for human reports, data databases etc then this process if called **information fusion** according to [88]. In some works, information and data fusion terms are used interchangeably, and the term data fusion is used for the exploration of optimum ways of handling large amounts of data to effectively produce reliable and accurate information, combining data from multiple sensors and sources [70, 90]. The process of improving the quality of the information output in a process is known as **synergy**. In some cases, synergy can be achieved even by a single sensor, however employing more than one sensor or sources of different traits can increase the quality of the output in terms of representation, certainty, accuracy and completeness [90]. The diversity in traits originates from the use of different acquisition frameworks (i.e instruments, devices or techniques), known as **modalities**. Systems that combine data from different modalities are known as **multi-modal data fusion** systems [86].

Domain	Example Applications	Modalities Fused
Healthcare	Disease diagnosis, patient monitoring	Imaging, EHR, physiological signals
Manufacturing	Process monitoring, robotics, maintenance	Sensor data, images, text
Autonomous Vehicles	Object detection, risk prediction, navigation	Cameras, LiDAR, radar, audio
Security/Surveillance	Intrusion detection, activity recognition	Video, audio, sensor data
Disaster Management	Flood victim identification, situational awareness	Meteorological data, social media posts
Human–Computer Interaction	Gesture/speech recognition, user interfaces	Visual, auditory, tactile
Remote Sensing	Land feature recognition, air quality prediction	Optical, LiDAR, SAR, ground sensors

Figure 44 Applications for multi-modal data fusion.

There are a few **taxonomies** in literature regarding data fusion systems [72, 75, 82, 103]. In [83], authors group multi-modal data fusion systems based on whether a neural network is involved in the process or not. A more common approach followed in [73, 82, 89] , authors categorize fusion strategies based on the level of fusion, also known as **fusion strategy** or **architecture**, while authors in [74] propose a more fine-grained one based on specific challenges. There are three level of fusion namely ***early*** (feature level), ***late*** (decision level) and ***intermediate*** (hybrid). The first one focuses on applying fusion techniques directly to the original data of each modality and generating a fused feature set across modalities. The second is fusion at the decision level, where each modality is modelled separately and their results are combined based on their importance for the prediction task. Last intermediate fusion combines both early and late fusion characteristics, often organized in multiple intermediate layers, binding modality specific features with cross modal feature representations.

Fusion	Description
Early (Resource Level)	Combines raw data from different modalities at the input stage.
Late (Decision Level)	Combines decisions or outputs from separate unimodal models
Intermediate (Feature Level)	Fuses features extracted from each modality before inputting them into the prediction model.

Figure 45 The three primary architectures for multi-modal fusion.

Intermediate fusion is particularly effective for associating fine-grained features with specific behaviors or events, which is valuable in streaming contexts such as live e-commerce or surveillance. However, fusion-based approaches often introduce substantial computational overhead, leading to larger model sizes and higher inference latency, which can limit their feasibility for real-time deployment on resource-constrained platforms [78, 87, 99].

Multi-modal data systems face **challenges** both on the ***data level*** and the ***model design level*** [86]. During data acquisition for each modality, a record may have different units or metric system (*non-commensurability*), *resolution*, *data rate* or *size* and also refer to different coordinate systems, significantly increasing the complexity of *registration* and *alignment* tasks. Furthermore, noise, missing values, and contradicting or inconsistent data increase *uncertainty* while *balancing information* from different origins is challenging on its own. At the model **design** level, selecting the *level of data fusion* can either consist of specific processing pipelines of raw data, sequential processing for constraining other sources or use features (high level, low level or hybrid) for true fusion by symmetrically allowing all modalities to inform each other. Selecting the link *mechanism* between datasets can also be either a hard link, inherent link to the model itself, or a soft link that relies on statistical dependence, while sometimes only a subset of features of each source is available for fusion. Selecting the *analytical framework* of the fusion can rely on existing analytical frameworks, however taking the full advantage of different modalities may require more advanced techniques. Finally, there is no golden rule that can put together different design steps in a *“structured data fusion”* that can be applied to any dataset out of the box, while *theoretical validation* and quantitative gains are hard to define.

In a recent work [73] that focuses IoT-based Context-Aware Recommender Systems (CARS) through multi-modal data fusion, authors highlight dynamic context modeling as a key challenge and specify key challenges per context domain. In [72, 79, 85, 87, 96, 97] authors document common **open challenges** that relate to privacy and ethics, explainability (XAI), real-time performance and computational cost, cold start context problem as well as context aware evaluation methodologies. Specifically, authors in [87] mention that increased input dimensionality and complex architectures of fusion-based methods can hinder real-time deployment, especially on edge and IoT devices. Authors in [79, 85] highlight that effective integration requires addressing modality-specific noise, cross-modality heterogeneity, and information conflicts. [93] proposes integrating knowledge graphs with deep learning is proposed to enhance scalability, explainability, and prescriptive optimization to address lack of interpretability for deep learning models. In [97] authors claim that verifying data authenticity and integrating information across platforms remains an open challenge for social media and disaster management. While in [79], authors pinpoint the need for analytical methods to estimate model accuracy upper bounds, define uncertainty, and measure the contribution of each modality.

Recent advancements in multi-modal data fusion have heavily leveraged **deep learning techniques**. Key developments include Hybrid Deep Learning Architectures that combine models like CNNs, RNNs, and autoencoders, often incorporating attention mechanisms to optimize modality weights [76, 95]. Attention Mechanisms and Transformers are being used to enable dynamic weighting and feature integration, increasing robustness to noisy or missing data [77, 94, 95]. Furthermore, Graph Neural Networks (GNNs) are employed to learn optimal combinations of multimodal data in a latent space and capture complex relationships [91, 92], while methods like Dynamic Routing and Deep Canonical Correlation Analysis adaptively enhance fusion processes by aggregating modality representations and extract shared ones [Salvi et al. 2024]. Finally, Cross-Modal Transformers and Spatio-Temporal Networks are specifically designed to integrate temporal and spatial dependencies, which is particularly useful for applications like traffic prediction and remote sensing [95].

Combining data of different modalities based on their **spatial or temporal traits for specific tasks** is common in data fusion. A non-exhaustive list of such tasks and their corresponding recent reviews is autonomous driving [96], remote sensing [71, 102], object-tracking [98], traffic flow [84, 103], environmental and urban planning [71] each one exploiting slightly different aspects of fused information. Sensor Fusion in [96] for autonomous driving is mainly applied for mutli-target tracking utilising a motion model and data association. In [102] the contrastive language–image pretraining (CLIP) method uses LLMs to augment existing geospatial analysis pipelines by facilitating multi-modal fusion, while [71] focuses on works that produce synthesized images with both high spatial and temporal resolutions from two types of satellite images. In addition, opportunities and challenges of the use of LLMs and multi-modal sources in urban planning are discussed. In [98] a unified taxonomy is presented for multi-modal tracking algorithms. In [84], authors review different approaches and provide a guideline of constructing data fusion methods which involve preprocessing, filtering, decision, and evaluation as core steps of data fusion for traffic flow management. In [103] authors discuss multi-source data and attributes and also review deep learning models for multi-source traffic flow prediction.

There is an active research community focused on **spatio-temporal analysis of sensor or IoT** data. A recent survey presents work on spatio-temporal pattern analysis of moving objects in [80], in [100] authors review recent works on distributed processing for event prediction, while in [101] authors review works that exploit spatial characteristics for efficient search techniques of IoT. However most of these works focus on processing or knowledge extraction rather than providing mechanisms to tackle modality and fusion challenges.

A **general-purpose framework** for real-time spatial IoT source aggregation is introduced in [81]. In this work authors define the Spatio-Temporal Data Fusion (STDF) architecture that consist of two processing layers and focuses on low-level data fusion. In the first layer a cluster sampling is used for data reduction upon data acquisition from all IoT sources. While in the second layer this method utilizes a combination of k-means clustering for spatial analysis for each source and Tiny AGgregation (TAG) for temporal aggregation to perform spatio-temporal data fusion.

Contribution of CREXDATA advancing the state-of-the-art: The ***Multi-Modal Data Fusion (MMDF) toolbox*** developed in the context of the **CREXDATA** project provides a flexible and versatile framework that enables end-users to design task-specific solutions through workflow specifications and address **various multi-modal fusion challenges**. It allows end-users to register different sources and align their spatial and temporal characteristics based on spatial indices and expiration times. The toolbox also enables window operations, filtering records and applying transformations through user-defined functions, facilitating **data acquisition and decision-making tasks** (late fusion).

The toolbox supports combining **multiple layers** of fusion and aggregating mechanisms and integrates with external predictors via Kafka, making it suitable **for early and hybrid fusion** tasks. Given sufficient information, such as an identifier, these tasks can be included in the workflow without explicit knowledge of the underlying raw records, which can later be included (joined on the fused record or decrypted) in a later stage to support **privacy protection**.

The tool supports linking data through join specifications ("hard-link") and, due to its micro-services architecture, external more complex link mechanisms can also be supported. Furthermore, building upon Kafka's real-time performance capabilities, the toolbox registers memory-demanding sources (static files, video streams) through their metadata, incorporates the highly efficient H3-index for spatial operations and defines expiration times on top of the micro-services architecture. This provides a **real-time scalable and configurable system**. Finally, end-users' full control of the workflow and its visual representation is crucial for **explainability tasks**.

4.2.3 Core Fusion Engine

The multi-modal fusion engine namely *mmdf-kstreams* is written in Java, is built upon Apache *kafka-streams* [14] that is a client library that benefits from Kafka server-side cluster technology for building applications and microservices with input and output data stored in Kafka. The *mmdf-kstreams* engine is designed as a *general-purpose framework*, decoupling the processing logic from the specific business domain (e.g., maritime, logistics, or smart cities). Rather than hard-coding domain-specific rules, the engine relies on a flexible configuration model to interpret data.

Data in Apache Kafka clusters are organized into topics. Each topic represents a specific stream of data that multiple client applications interact with. Client applications that send data to topics are called publishers, while clients that read data from them are called subscribers. Each data record in Kafka typically consists of a key and a payload tuple. Even though multiple serialization schemas exist, in this work we assume that our records are organized in a (string, JSON-string) format. Kafka streams applications typically operate in both modes as they read their input from topics, organize intermediate steps into temporary topics, and publish their materialized output to another topic.

For fusion to be performed through the MMDF Core engine, it is required that input data are available in Kafka topics. Input data can be published to Kafka topics through an injector mechanism or from any external system. In addition, it is required that records contain or can be assigned some spatial and temporal characteristics. The core application treats incoming Kafka records as generic JSON objects. It becomes "aware" of spatial characteristics only through the user-**defined configuration file** (source nodes), which maps the engine's internal spatial requirements to the specific fields in the data. For data streams that lack native spatial attributes (e.g., static IoT sensors that transmit ID but not location), the architecture expects

[14] https://kafka.apache.org/documentation/streams/

an injector mechanism to enrich raw records with the required spatial context *before* they enter the fusion topology.

The mmdf core requires a configuration file in JSON format and connection information for the Kafka cluster. The configuration file contains three lists of different types of JSON nodes, namely sources, transformations and relations that represent different operations on streams.

The source node has three responsibilities, first to register an existing Kafka topic into the application, to calculate a key from the spatial information within and last to define the expiration time of records in topic. The transformation node transforms internal streams based on specific operations such as filter, record format, apply a user defined function on a record attribute etc. detailed information of the supported transformation are listed later in this section. Last, the join node configuration fuses two different streams into one based on their spatial and temporal attributes. Relations type operation consists of three main categories of combining two streams, windowed operations, operations with the latest value or the right stream, and merge operations. All configuration nodes contain common information about the input and output topic as well as controls regarding the output handling, i.e. printed in logs or/and materialized in a Kafka topic.

The mmdf-core fusion engine uses this configuration file to define an execution topology, which is an internal mechanism of *kafka-strea*ms library that handles execution order. It relies on the information of the input and output topic to calculate the order or execution.

4.2.3.1 Configuration File

The mmdf-configuration is a JSON file or JSON string. It consists of three arrays of nodes shown below in between brackets ([-]). Node in JSON format is a record between two curly brackets ({-}) and represents a simple object that could be serialized in a simple structure class. All three array nodes are mandatory, and each array can contain inside one or more configuration nodes with their respected type. An example of the core structure is depicted below (on the next page).

```
{
  "sources":[
…
  ],
  "transformations":[
…
  ],
  "relations":[
…
  ]
}
```

4.2.3.1.1 Source Configuration

The source configuration node consists of the following information.

- "name" is the input topic name,
- "id" is the name of attribute that contains identifier information,
- "lon" stands for longitude,
- "lat" strands for latitude,
- "t" for time information,
- "expires_ms" expiration time for the record in milliseconds.
- "peek" indicated if each record should be printed in logs.

An example is depicted below:

```
{
  "name":"ccc-position",
  "id":"mmsi",
  "lon":"lon",
  "lat":"lat",
  "t":"timestamp",
  "expires_ms":2147483647,
  "peek": false

},
```

4.2.3.1.2 Transformation Configuration

The Transformation configuration node consists of the following information.

- "source" (String) is the input stream name,
- "method" (String)is the type of the transformation,
    - "flatmap": uses h3 k-ring function to broadcast information to near-by cells
    - "filter": filter out records that do no match criteria specified for the "field" when applying the "operator" with regards to "value".

    - "interval": down-samples stream by releasing to output the latest value base every "value" seconds.
    - "kplerCa": Maritime Use Case specific format for collision avoidance service
    - "asvCommand": Maritime Use Case specific format for sending back to ASVs collision avoidance service recommendations.
    - "udf" : User define function that parses "field" and applies to each record
- "field" (String) the field upon method will be applied,
- "value" (Numeric) applies on filter
- "operator" (String) if method is filter it takes values
    - "gt": greater than,
    - "gte": greater than or equal,
    - "lte": less than or equal,
    - "lt": less than,
    - "eq": equal
- "topic" (Boolean) for time information,
- "peek" (Boolean) indicated if each record should be printed in logs.
- "output" (String) is the output stream name.

Two examples are depicted below:

```
{
  "source":"ccc-position",
  "method":"flatMap",
  "field":"h3",
  "value":1,
  "topic":false,
  "peek":false,
  "output":"ccc-position"
},
```

```
{
  "source":"crex-maritime-fused",
  "method":"filter",
  "field":"sog",
  "value":.0,
  "topic":true,
  "operator":"gt",
  "output":"crexdata-maritime-fused-sog"
}
```

4.2.3.1.3 Relation Configuration

The Relation Configuration Node consists of the following fields:

- "type": The type of join available options are:
    - "leftjoin" & "join": windowed time left or outer join maximum time difference between records "dt_ms" milliseconds and maximum distance less then "distance". All records that match criteria are returned. Fused records are produced.
    - "leftjoinAsTable" & "joinAsTable": the live stream of left source will match the latest updated value based on with the right source with respect to "distance" and "dt_ms". All records from the left source are returned.
      Fused records are produced.
    - "merge" left and write source are both published to the output stream.
- "source_left", "source_right": left and right source of the join.
- "topic": (Boolean) for time information,
- "peek": (Boolean) indicated if each record should be printed in logs.
- "output": (String) is the output stream name,
- "spatial_index": (String) Type of spatial index used (only h3 supported),
- "dt_ms": (String) maximum time difference for window operations and latest value operations,
- "distance": (String) maximum distance for record fusion,
- "spatial_resolution": (String) the spatial resolution on which records will be fused.

```
{
  "type":"join",
  "source_left":"merged-ais-ccc-position",
  "source_right":"ccc-position",
  "output":"crex-maritime-fused",
  "topic":true,
  "peek":false
  "spatial_index":"h3",
  "dt_ms":2000,
  "distance":1000,
  "spatial_resolution":9
},
```

4.2.4 MMDF Toolbox Extension for the CREXDATA Workflow Designer (RapidMiner AI Studio)

Generating configuration files in JSON format might prove to be challenging for less tech-savvy human operators and hard to present on decision makers. So, we take advantage of the AI Studio workflow designer, that is provided by RapidMiner, to facilitate configuration setup and execution process. To do so, we create an AI Studio extension named MMDF Toolbox Extension that provides a user interface for designing mmdf-core configuration file and if needed can trigger execution from within.

The extension consists of six operators in total. Operators named “Source”,” Transformer”, and “Join” correspond to the source, transformation and relation configuration nodes presented in the previous section. The CONCAT operator can merge two configuration files into one and the Fusion Operator can start the core engine fusion application from within AI Studio. Finally, a shapefile and a URL injector operator are presented as well.

4.2.4.1 MMDF Operators for RapidMiner AI Studio

The following figure shows the list of MMDF Toolbox operators made available in RapidMiner AI Studio by the MMDF Toolbox Extension developed within the CREXDATA project:

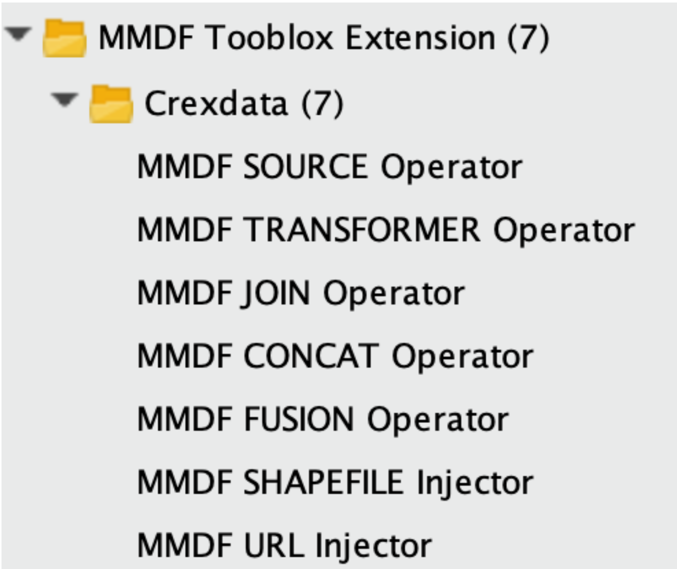


Figure 46: Overview of MMDF Toolbox operators in RapidMiner AI Studio.

4.2.4.1.1 MMDF Source Operator

The source operator provides a user interface and collects the necessary information for registering existing Kafka topics into the fusion application. The user declares the required information via the operator parameters. The operator then generates the corresponding source node configuration as a JSON document.

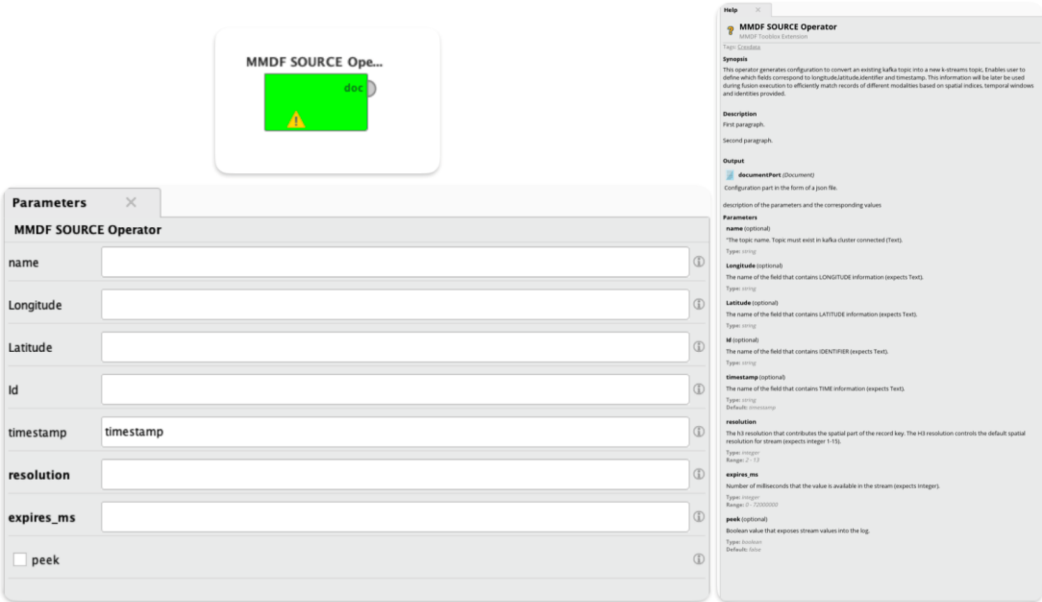


Figure 47: The MMDF Source operator and its parameters.

4.2.4.1.1.1 MMDF Transformer Operator

The transformation operator provides a user interface to apply different transformations into registered Kafka streams. The registered stream is provided as an incoming connection configuration document in JSON format from a source operator, transformation, join or concatenation operator. The user can select the type of transformation and corresponding operators through drop down menus and parameters. The operator then inserts a new transformation node into the configuration file forwards the updated configuration to next operator.

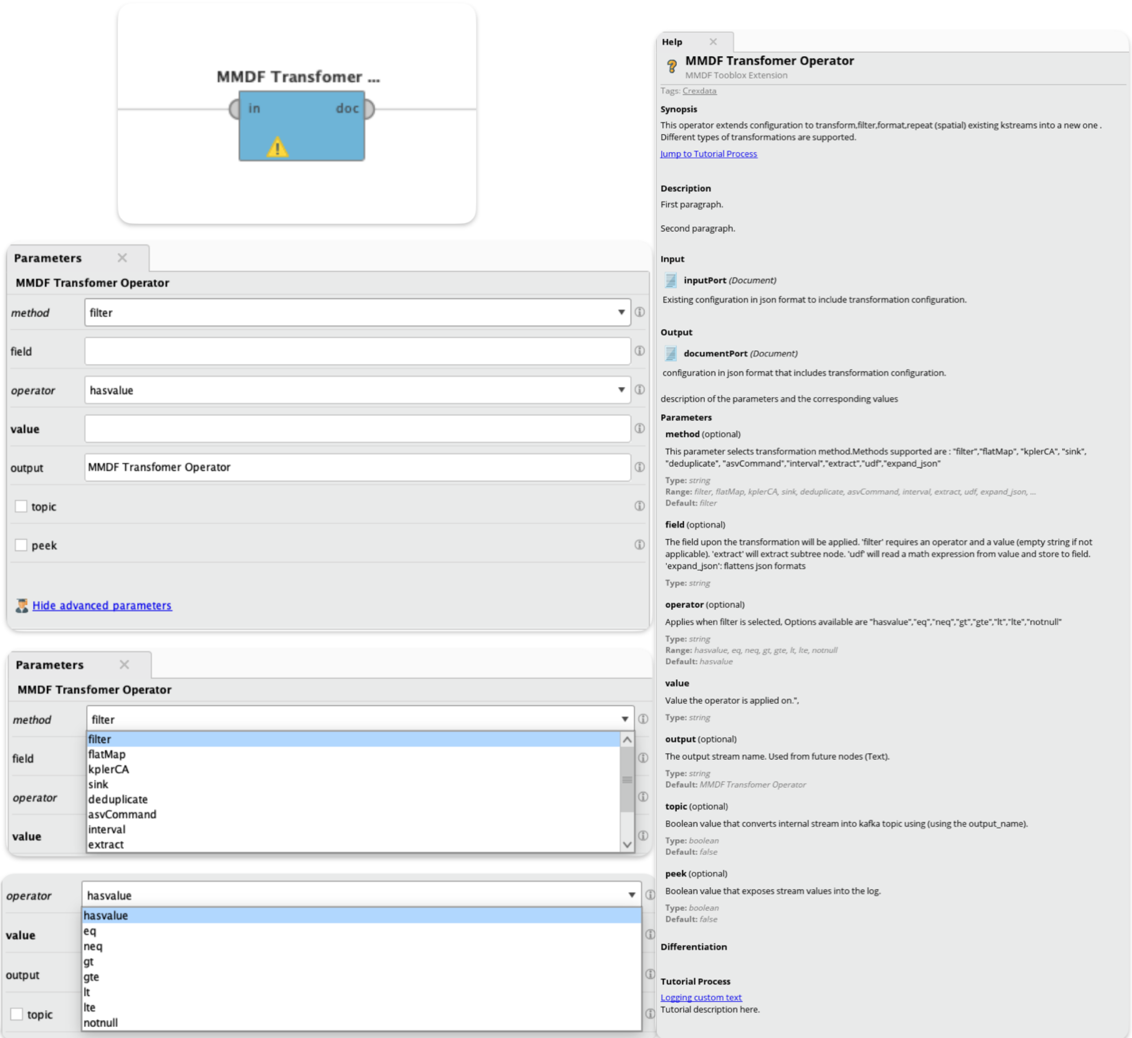


Figure 48: The MMDF Transformer operator and its parameters.

4.2.4.1.2 MMDF Join Operator

The join operator provides a user interface to merge records of two existing streams into a fused one with respect to spatio-temporal constraints. Input streams are provided as incoming connections in JSON format and can be of any type except the fusion operator. The user can specify the type of join and corresponding constraints via drop-down menus and parameters. The operator then inserts a new relation node into the configuration file and forwards the updated configuration to the next operator.

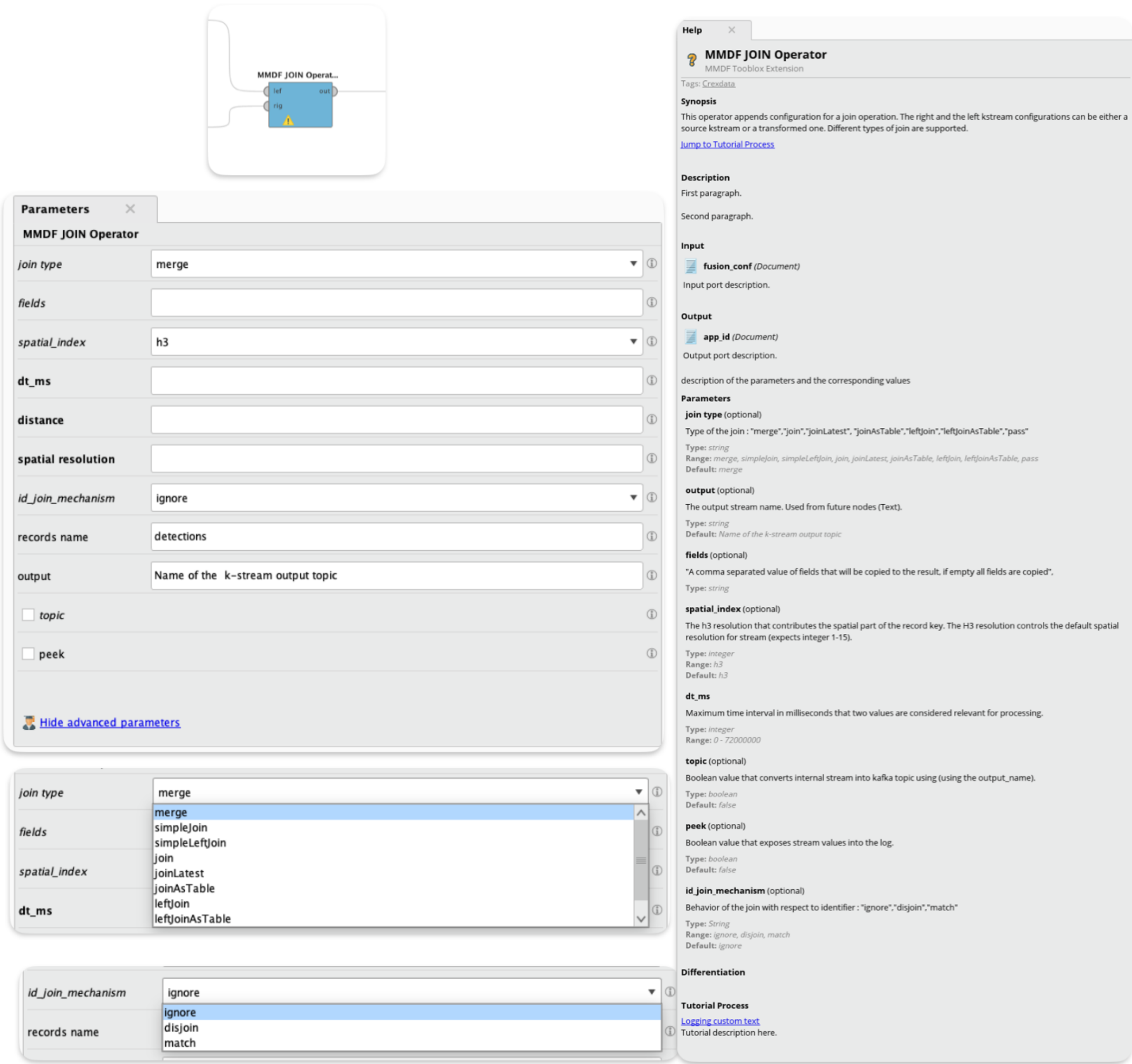


Figure 49: The MMDF Join operator and its parameters.

4.2.4.1.3 MMDF Concat Operator

The CONCAT operator facilitates the merging of two independent workflow branches into a single, consolidated output document in JSON format. It accepts two input connections and unifies their respective configuration lists, ensuring that the final output is deduplicated so that each record appears only once in the consolidated file.

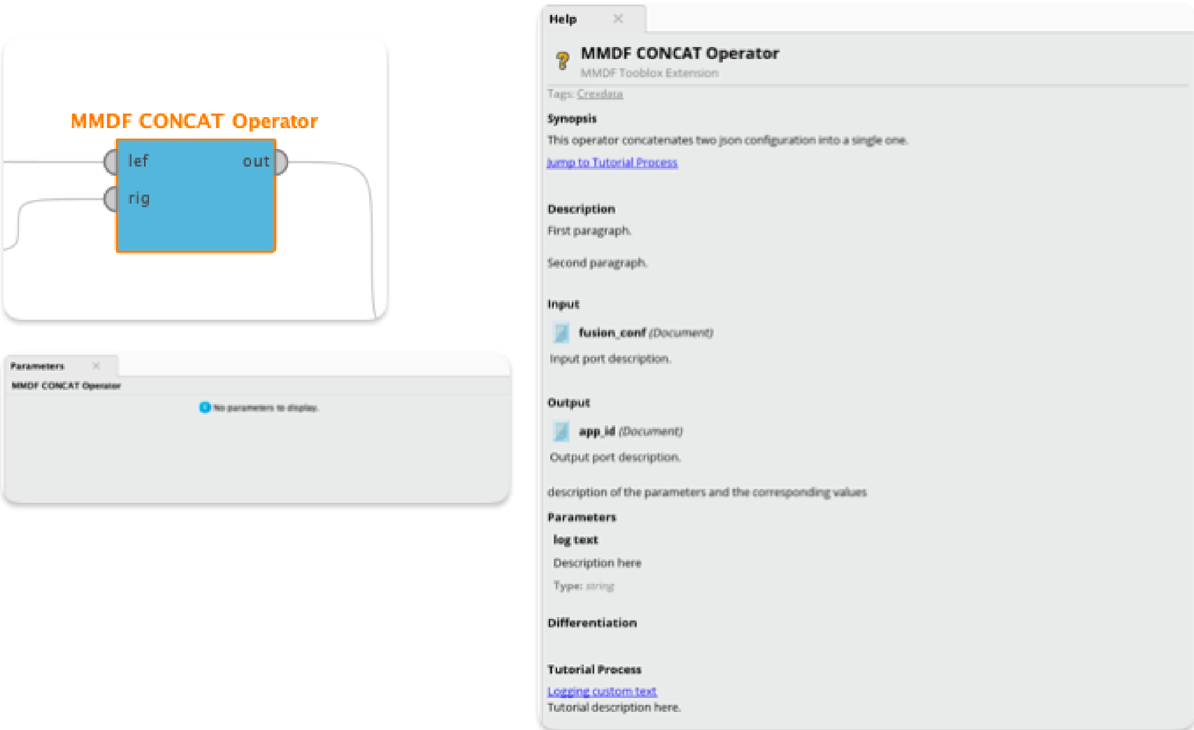


Figure 50: The MMDF Concat operator and its parameters.

4.2.4.1.4 MMDF Fusion Operator

The fusion operator is the final operator in a workflow and has dual purposes. Firstly, it ensures the configuration file is valid and secondly, it triggers execution. Up to the use of the fusion operator, all other operators build upon a configuration file. The fusion operator propagates the configuration file and Kafka connection information to its internal fusion engine and starts execution. The execution can be either in the foreground or a background thread. Given that the MMDF fusion engine processes streaming data, foreground execution is suitable for cases where execution is dispatched to the AI-Hub (thread execution is not supported) or inspecting a single workflow, as it requires user intervention to stop its execution. Conversely, for workflows with multiple fusion operators, thread execution is necessary to allow each fusion to occupy its own resources in the background and enable simultaneous execution.

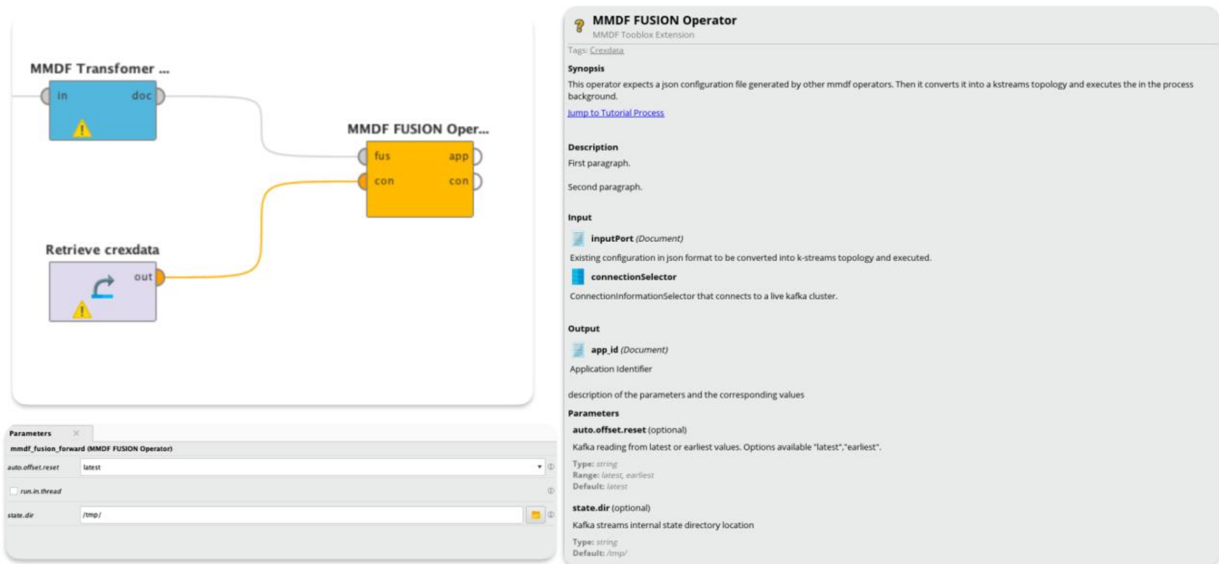


Figure 51: The MMDF Fusion operator and its parameters.

4.2.4.1.5 MMDF Shapefile and Url Injector Operators

In cases where data are not present in Kafka, an external application or an injector operator is required. Two basic injector operators have been created to handle Shapefiles and online context in the form of a URL.

The Shapefile injector reads a shapefile containing a Polygon from the local disk and publishes a record to a Kafka topic with the corresponding information in Well-known text. If the polygon spans more than one h3 cell at the requested resolution, it is split with respect to h3 cells and the corresponding Kafka messages are generated for each cell.

The URL injector requires a link to material accessible from the final stream consumer (typically a user interface application). It provides spatial context to resources such as webcams live data streams or information not authorised for access by the fusion application. The end user provides a link or relevant information, a target resolution and a corresponding area of interest in the form of Well-Known text.

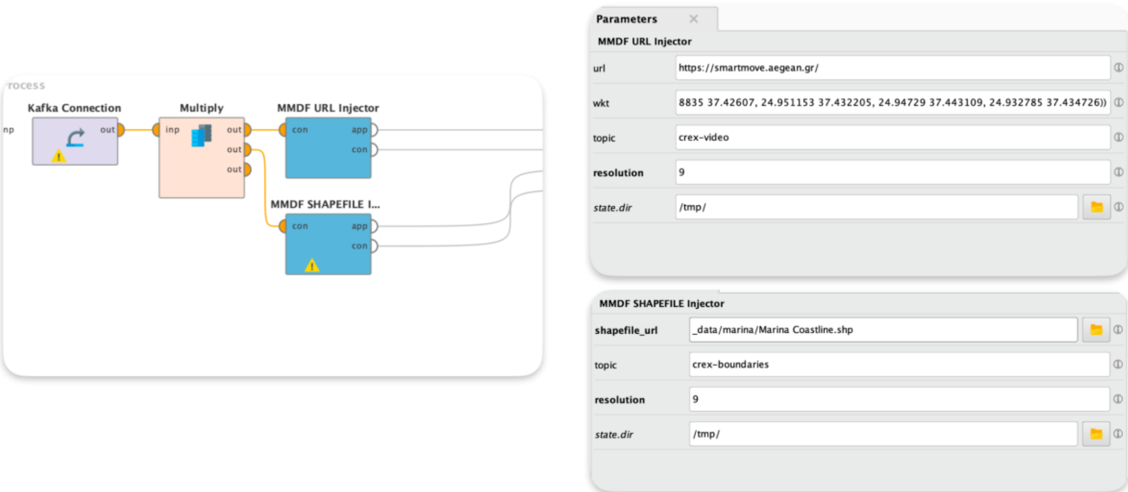


Figure 52: The URL and Shapefile Injectors.

4.2.5 AI Hub Compatibility

The MMDF-extension is compatible with RapidMiner AI Hub and enables end-users to design workflows on their local computer in RapidMiner AI Studio and dispatch their execution on the RapidMiner AI Hub. This functionality is supported only for the foreground execution, i.e. the "run.in.thread" option should be unchecked in the fusion operator before dispatching execution.

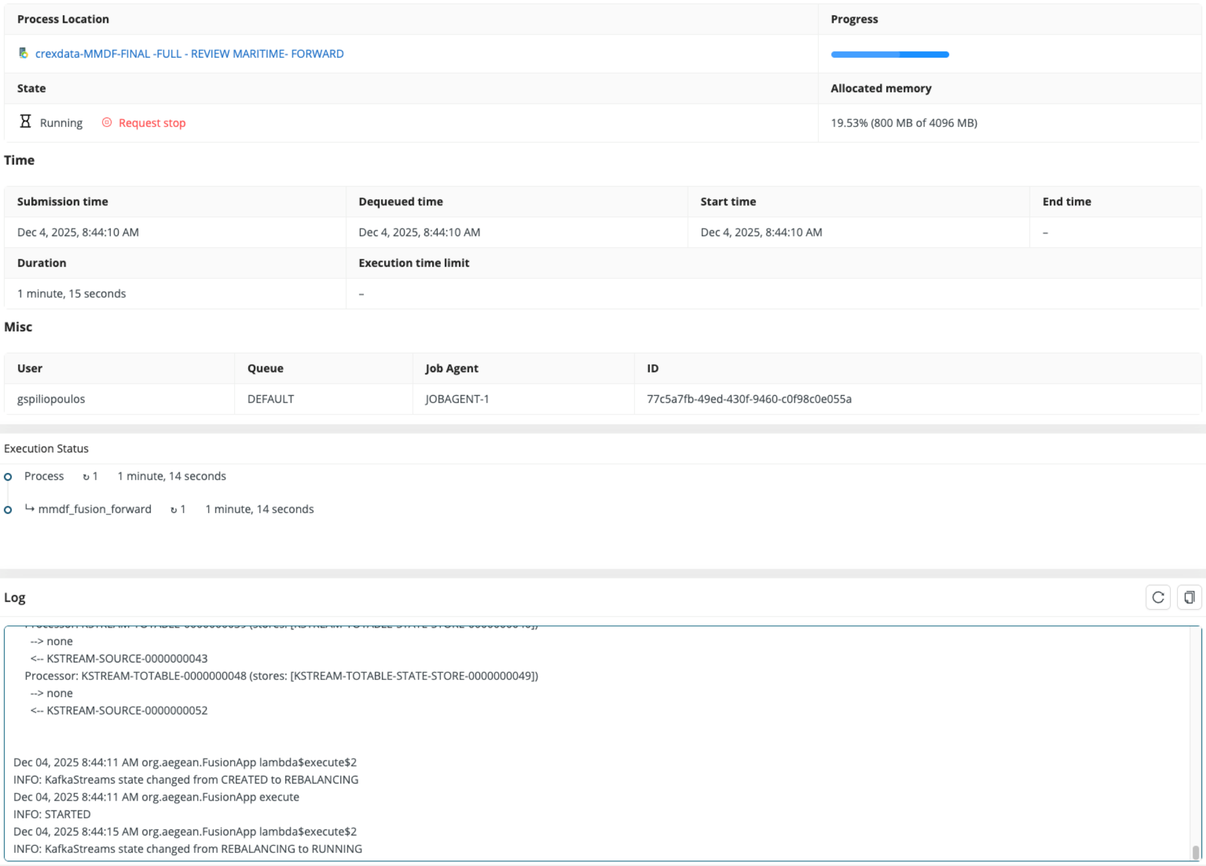


Figure 53: Maritime use case workflow dispatched execution on RapidMiner AI Hub.

4.2.6 MMDF Toolbox Usage Examples

4.2.6.1 Emergency Use Case (EmCase)

The multi-modal data fusion toolbox is a real-time processing engine that relies on metadata on a global hexagonal grid, namely H3, to fuse together records of streaming data at scale based on their spatial and temporal proximity. The extension generated in the context of the CREXDATA project allows end-users to define workflows that provide records that contain information from different data sources.

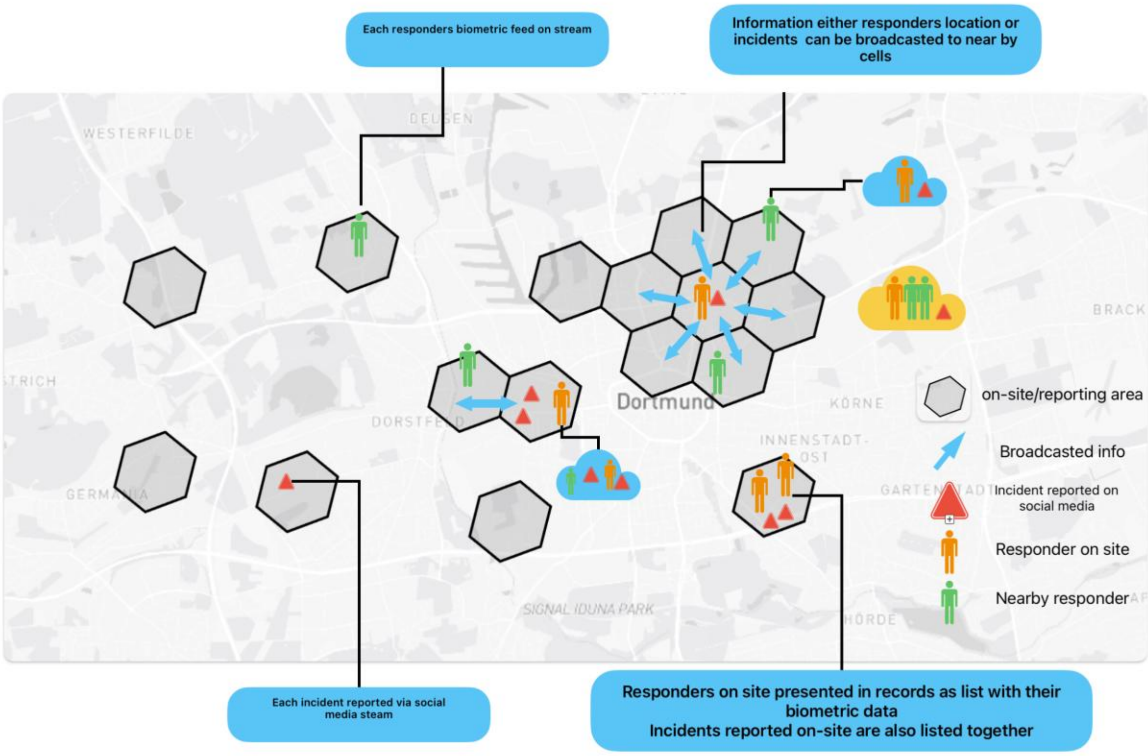


Figure 54: EmCase Conceptualization in the scope of Multi-modal Data Fusion.

In the example below we have used two different sources of data that are already available in Kafka. The first one corresponds to biometric data collected from first responders in real time and the other is data acquired from social media platforms based on some keywords for the area. A workflow for this case is provided below.

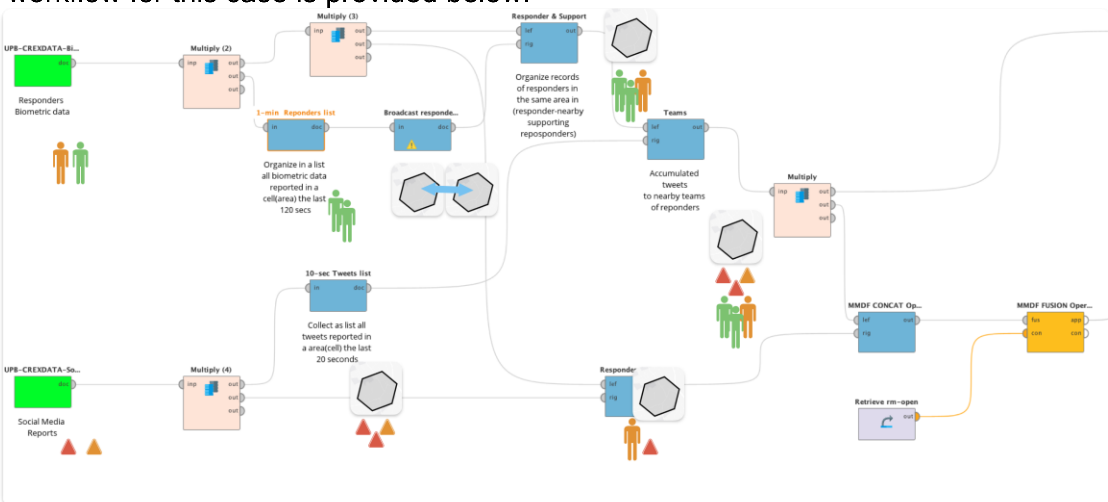


Figure 55: EmCase Workflow using the MMDF Extension.

The first level of processing is to aggregate information for each topic per h3 cell. Forming collection of data that are collocated, e.g. teams of responders or collections of incidents. At a second level two types of processing are performed according to the workflow. At the one hand teams of responder's information is broadcasted to nearby cells, allowing us to create records that contains the information of each responder and possible nearby supporting teams.

At the other hand information of each responder is put together with incidents on-site, providing an individual one to one connection of incident and on-site responders. The first intermediate result might be useful in case an incident escalates and might needed backup while the one could provide useful info in cases of many isolated and distance incidents occurring at the same time.

Last, collections of incident and teams of responders are put together, allowing us to have complete understanding of a possible larger incident where multiple incidents are reported in proximity and a multiple responders teams operate in the area.

4.2.6.2 Maritime Use Case

At the core of the Maritime Use Case is the use of the Multi-modal data fusion toolbox and its AI-studio extension. We rely on MMDF to process data in real time and fuse data sources and records based on their spatial and temporal proximity. In the AI Studio workflow designer, the processing is divided into subprocesses based on the direction of information flow. The information from ASV and other sources towards the CA service is designated as the 'Forward' subprocess and the opposite direction as the 'Return' subprocess.

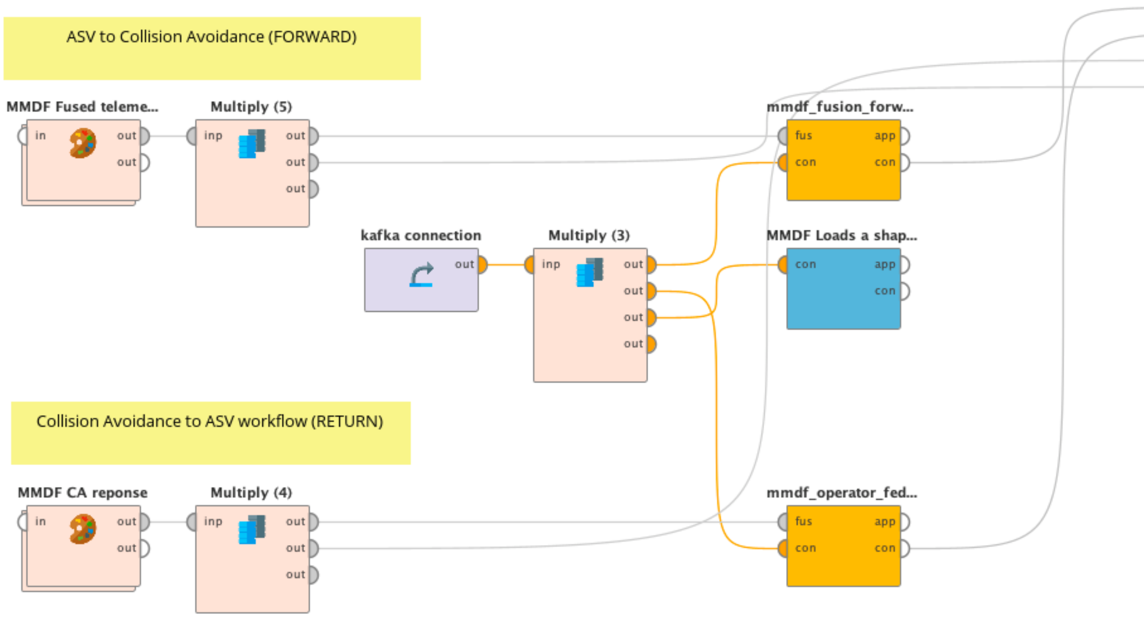


Figure 56: Screenshot of the Maritime Use Case workflow (RapidMiner process in AI Studio) consisting of a forward subprocess (ASV to Collision Avoidance) and a return subprocess (Collision Avoidance to ASV).

Both subprocesses encapsulate data sources, transformation and join operators that are sequentially processed by the AI-Studio to generate a JSON configuration string. Each configuration string is then passed along with a Kafka connection operator to the corresponding fusion operator. Consequently, each fusion operator initiates the MMDF toolbox execution in a background thread. Each thread reads records from the corresponding Kafka topics and executes the workflow in a streaming manner utilising the Kafka streams processing engines. Following Kafka's architectural approach, the Kafka streams framework typically uses a key-payload tuple to organise records in the stream. At the end of each subprocess, JSON records are published into corresponding Kafka topics making them available for other CREXDATA components.

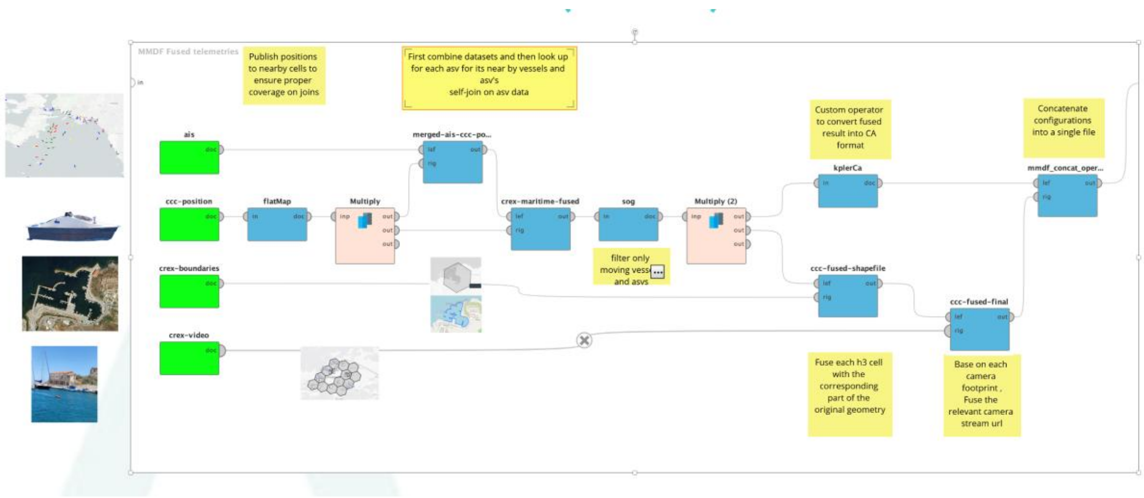


*Figure 57: MMDF **Forward** subprocess workflow, from ASV telemetry and other sources to services.*

The 'Forward' subprocess comprises a variety of MMDF operators including source, transformation and join operators as illustrated in the figure above. In Figure 57, the four data sources are depicted in light green color: AIS, ccc-position (ASV-telemetry), crex-boundaries (Shapefile injector for the marina area) and crex-video (video-stream metadata source). Source operators read records from existing Kafka topics. They then read from application configuration the names of the fields corresponding to longitude, latitude, timestamp and identifier information and generate a key based on the h3 spatial index for each record. A flatmap operation is used to broadcast ASV telemetry to nearby cells (transformer operator, flatmap process). This ensures that information of a moving ASV is available to a broader area to address cases where an ASV transitions from one spatial index to another nearby. Next, all records from ASVs and AIS messages are merged into a combined stream (merged-ais-ccc-position stream, join operator, merge process). Another join with the ASV telemetry stream takes place immediately after that, combining records that have the same spatial key and are less than two seconds apart (crex-maritime-fused, join-operator, join process). Records of ASVs with a speed less than 0.5 knots are filtered out (speed-over-ground (sog), transformation operator, filter process). The same stream output is then used in two ways. On the one hand, it is formatted based on collision avoidance requirements, including calculating the closest point of approach for the two moving objects included in the fusion record (kplerCA, transformation operator, kplerCa process). On the other hand, the same records are joined with the latest information regarding the boundaries of the marine and related video streams metadata. Finally, all configuration corresponding to the operations described in this paragraph are organised into a single JSON configuration file.

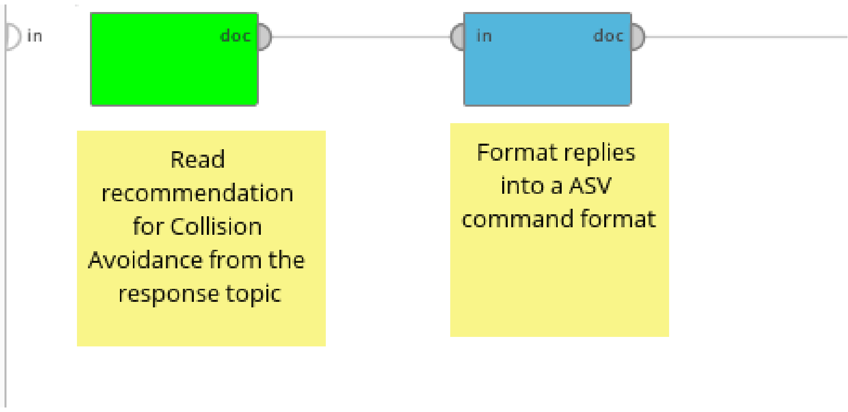


*Figure 58: MMDF **Return** sub-process workflow, services response to ASVs.*

The 'Return' subprocess is a simpler process that reads data published to the collision avoidance service result topic and converts its payload to a ASV command compatible format. This data is then published to a Kafka topic that the ASV control room consumes and sends appropriate avoidance mission plans to each ASV.

4.2.6.3 MMDF Source Code Availability

There are two source code repositories containing the Multi-Modal Data Fusion (MMDF) Toolbox as described above:

- Within the public source code repository of the CREXDATA project under https://github.com/altairengineering/crexdata-public/tree/main/CREXDATA%20System%20Components/Multi-Modal%20Data%20Fusion%20Toolbox
- For allowing customization by future adopters, the Core service code can also be run as a stand-alone Java application: https://github.com/ITSLab-UAegean/mmdf-kstreams
    - Requires a JSON configuration file as input to run as a standalone app
    - Then a separate RapidMiner installable extension repository can be customized that requires a fat-jar lib file from the repository folder above: https://github.com/ITSLab-UAegean/mmdf-rm-extension. Defines required custom RapidMiner operators (i.e. visual workflow design elements) to design a workflow (i.e. RapidMiner process), convert it into the configuration file, and initiate the execution of the service in AI Studio.

4.3 FMI Weather Forecast and Wave Height Forecast Integration

The Weather Forecast for selected municipalities in Finland is provided by FMI via the Amazon Web Services (AWS) S3 Bucket storage system. The data is stored in CSV file format and structured by Timestamp, municipality and Risk class with 1 being of low risk and 5 being of high risk.

The hourly forecast is published every 3 hours and fetched on a scheduled basis using a REST interface for the corresponding S3 bucket.

Figure 59 and Figure 60 display visualizations for exploratory data analysis.

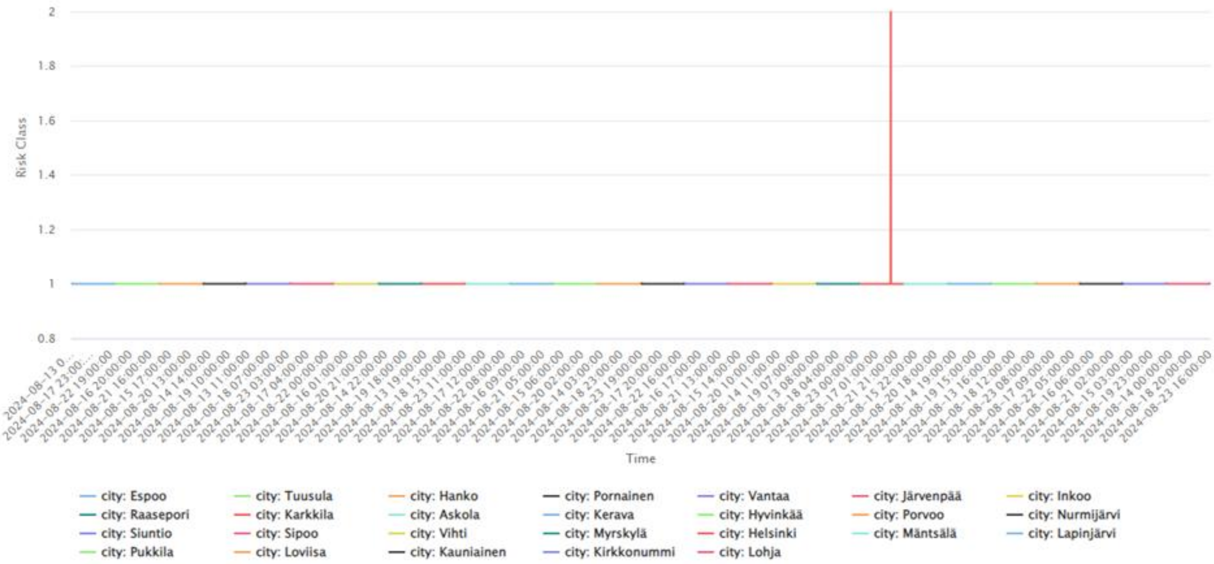


Figure 59: Maximum Risk class for each city and hour over a period of 10 days.

Figure 60: Maximum Risk class within the next timeframe per region

The Wave Height Forecast is based upon a custom Python script created by FMI. Using the Copernicus Marine Service (https://data.marine.copernicus.eu), the dataset for Mediterranean Sea Waves Analysis and Forecast (cmems_mod_med_wav_anfc_4.2km_PT1H-i)[15] is retrieved. To identify areas and timeframes where the forecasted Wave height exceeds a predefined threshold, the custom script with several exceedance monitoring functions is applied.

The dataset is a collection of single files which are available as netcdf format and downloaded using the Command Line Interface (CLI) version of the Copernicus Marine Toolbox[16], by issuing the command:

```
copernicusmarine get --dataset-id cmems_mod_med_wav_anfc_4.2km_PT1H-i --filter *yyyyMMdd_fc12*
```

15 https://data.marine.copernicus.eu/product/MEDSEA_ANALYSISFORECAST_WAV_006_017/description with the DOI https://doi.org/10.48670/mds-00373

16 https://help.marine.copernicus.eu/en/collections/9080063-copernicus-marine-toolbox

where `yyyyMMdd` is replaced by the starting day.

The resulting Wave height exceedances are visualized. The following figure depicts a timestamp of a region in the Mediterranean Sea with a potential exceedance of a wave height threshold (here 0.5 m) as well as the forecast of time and height.

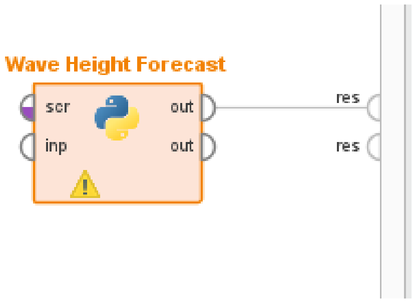


Figure 61: Wave Height Forecast Analysis executed from within AI Studio

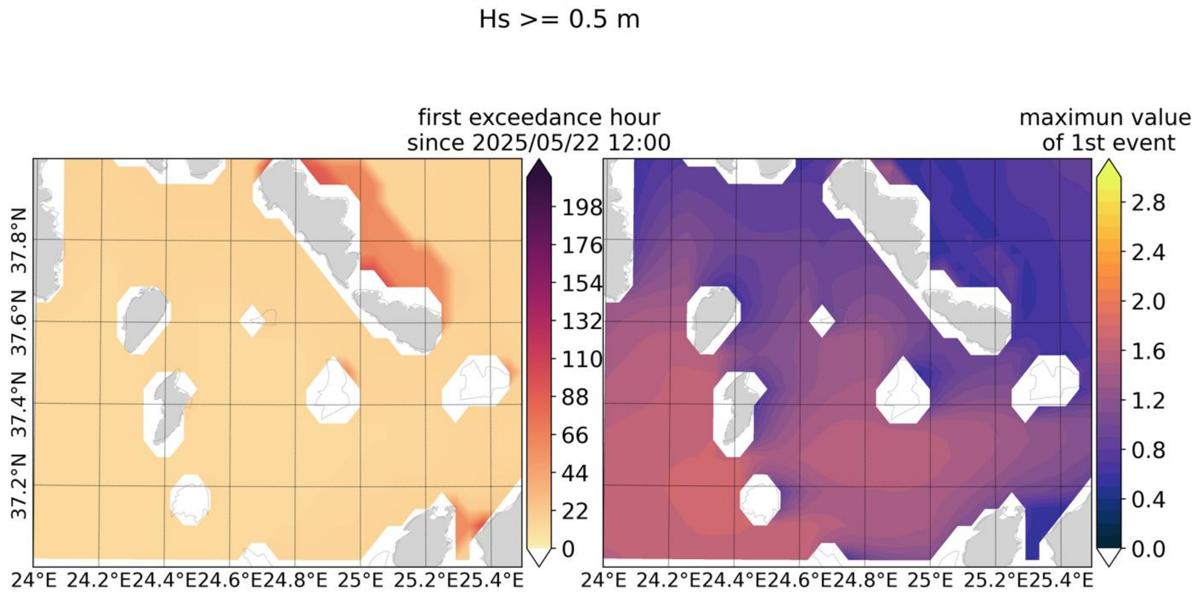


Figure 62: Wave height threshold exceedance corresponding maximum value

The source of the RapidMiner extension for the integration of the FMI weather forecasts and wave height forecasts is available from the public CREXDATA GibHub repository at

https://github.com/altairengineering/crexdata-public/tree/main/CREXDATA%20System%20Components/RapidMiner%20Extensions/Data%20Integration%20Extensions/FMI%20Weather%20Forecast%20Integration

and

https://github.com/altairengineering/crexdata-public/tree/main/CREXDATA%20System%20Components/RapidMiner%20Extensions/Data%20Integration%20Extensions/FMI%20Wave%20Height%20Forecast%20Integration

respectively.

4.4 Drone Data Integration

During flooding scenarios Drones can be used to record a map of the area. These detailed maps are processed to a Digital Surface Model (DSM), exported in GeoTiff format and stored in the Open Drone Map (ODM) instance of DRZ. The Python project "odm_kafka_bridge"[17] serves as a bridge between the ODM instance and the CREXDATA system, which is built on Apache Kafka. It can download assets, such as Digital Surface Models from ODM and relay them to a Kafka topic.

As depicted in the image below, REST API calls are made to (1) upload the Drone DSM GeoTIFF file to the Floodwaive service, (2) convert it to a Floodwaive internal data structure and (3) assemble a new DEM Stack which can afterwards be used for new Simulations.

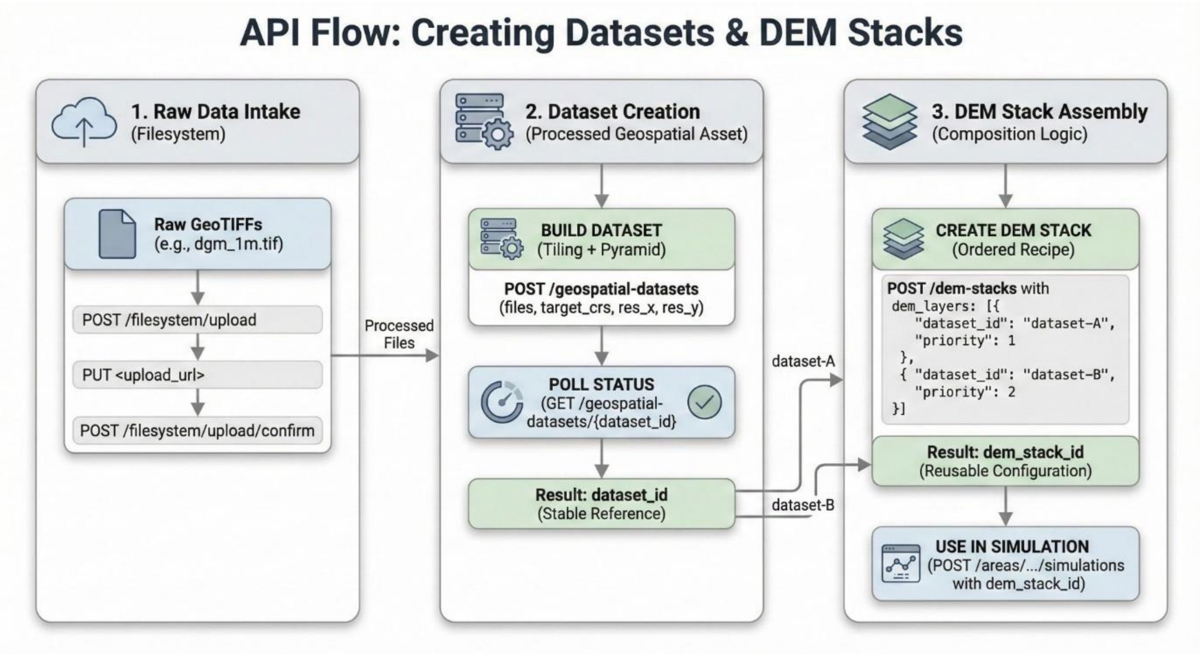


Figure 63: REST API call workflow for modifying a Floodwaive DEM Stack

The source of the RapidMiner extension for the integration of drone data is available from the public CREXDATA GibHub repository at

https://github.com/altairengineering/crexdata-public/tree/main/CREXDATA%20System%20Components/RapidMiner%20Extensions/Data%20Integration%20Extensions/Drone%20Data%20Integration

[17] The odm_kafka_bridge python module is available at https://github.com/altairengineering/crexdata-public/tree/main/CREXDATA%20System%20Components as submodule

4.5 HYDS ARGOS Integration

For the weather emergency use case, the system ARGOS of CREXDATA project partner HYDS integrates the data needed for this use case (weather data, events, sensor data, IoT and edge device data, etc.) and connects to the CREXDATA platform via Kafka, in order to provide the data to the CREXDATA platform and to get predictions from the CREXDATA platform in return and to visualize the results to the system user.

The ARGOS system provides hydrometeorological data to the CREXDATA platform using two different channels: Kafka topics and a sFTP stream. The latter was introduced to provide georeferenced files in raster format (.tiff). The following Kafka topics are updated at different frequencies with real-time data:

- hyds-Rivergauges-DataHub-Austria: Water level data from the HYDRO Tyrol network updated every hour.
- hyds-Raingauges-DataHub-Austria: Precipitation data from Tyrol Weather stations updated every hour
- hyds-Raingauges-DWD: Precipitation data from DWD (Deutscher Wetterdienst, German Weather service) weather stations. Hourly upload.
- hyds-official-warnings: Regional official warnings provided by national weather services gathered through MeteoAlarm. Hourly upload.
- UPB-CREXDATA-Weather-Stream-Dortmund: Point forecast for the city of Dortmund including wind speed, wind direction and precipitation, updated every 12h, based on GFS forecast.
- UPB-CREXDATA-Weather-Stream-Innsbruck: Point forecast for the city of Innsbruck including wind speed, wind direction and precipitation, updated every 12h, based on GFS forecast.

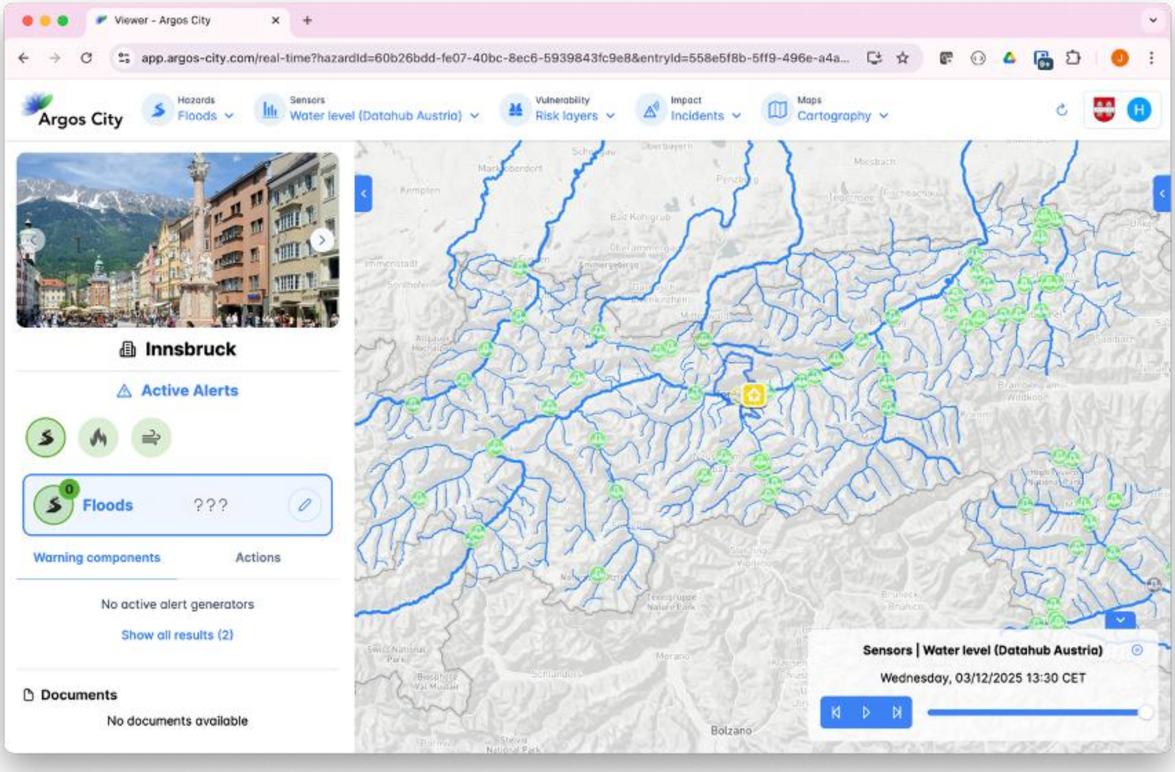


Figure 64: Example of water level data in Austria provided to the CREXDATA system.

The sFTP server (sftp-crexdata.argos-city.com) is updated with the following raster data:

- Opera Radar: Rainfall estimation from the European Radar network updated every 15 minutes.
- GFS Forecasts: Forecasts of precipitation, temperature, wind speed and wind direction with hourly resolution for the next five days with the global model GFS. Updated every 12 hours.
- NWP Forecasts: Forecasts of precipitation, temperature, relative humidity with hourly resolution for the next five days with the Austrian model. Updated every 12 hours.
- GSA Nowcasts: Forecasts of precipitation, temperature, relative humidity with hourly resolution for the next five days with the Austrian model. Updated every hour.

On the other hand, ARGOS provides the final users of the EmCase with a contextualized interface where users interact with output from different technologies in CREXDATA. All of them are transmitted through the following Kafka topics except for the output of the flood simulation which is retrieved with the API provided by FloodWaive (again raster data).

- UPB-CREXDATA-Flooding-Stream-Innsbruck: Once the flood simulation is executed on FloodWaive system, this topic provides the simulation id to show in Argos for the city of Innsbruck.
- UPB-CREXDATA-Flooding-Stream-Dortmund: Once the flood simulation is executed on FloodWaive system, this topic provides the simulation id to show in Argos for the city of Dortmund.
- UPB-CREXDATA-Flooding-Optimizer-Input: Barriers locations drawn in ARGOS interface (specifically developed for this purpose) to serve as input for the optimization algorithm.
- UPB-CREXDATA-Flooding-Optimizer-Output: Barriers optimal height as a result of the optimization algorithm.
- text_mining_processed_relevant_tweets_stream: Georeferenced X posts labeled as relevant by the Text mining algorithm
- text_mining_to_argos_stream: Georeferenced X posts selected for the current emergency
- UPB-CREXDATA-RoboticSimulation-Stream: Three georeferenced routes selected according to various criteria including the evolution of several parameters including battery consumption.
- UPB-CREXDATA-DroneData-Stream: Locations of drone route used for the building of 3D models together with the filename of the image taken in each location.
- UPB-CREXDATA-BiometricData-Explanation: Location of firefighters together with fatigue assessment parameters.

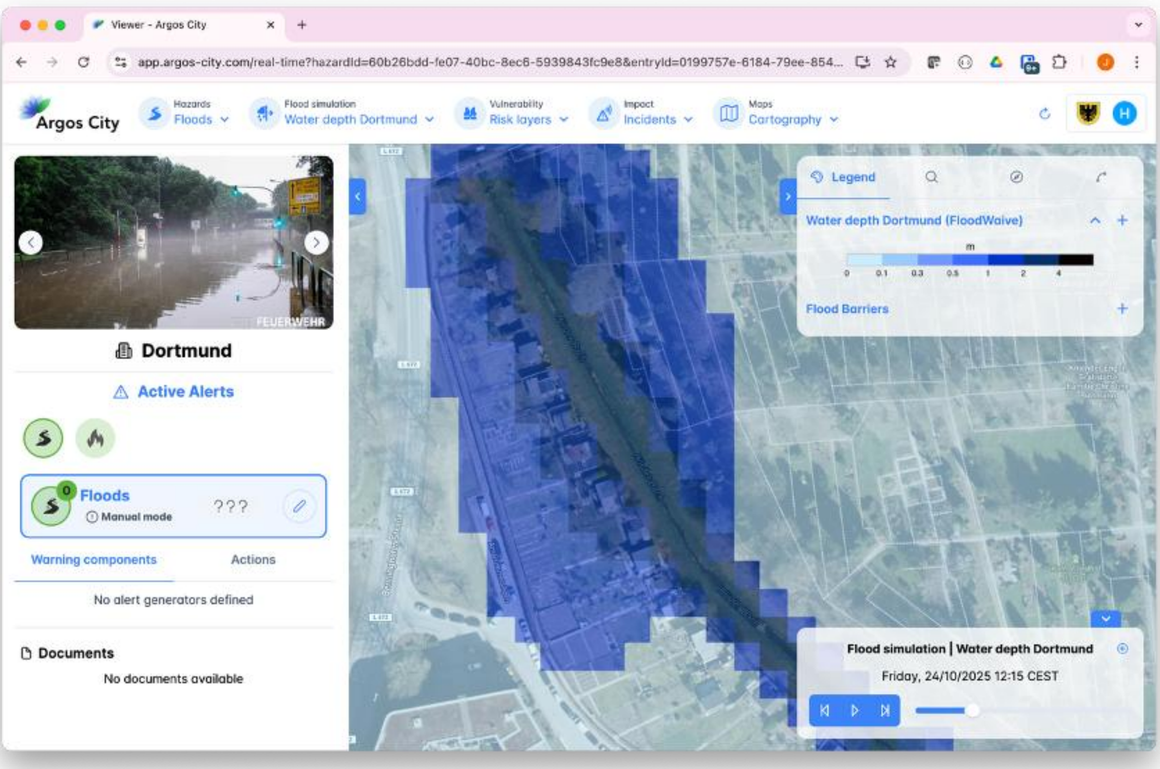


Figure 65: Example of Flood simulation in Dortmund provided by the CREXDATA system.

4.6 Simulator Integration

4.6.1 BSC Health Simulator Integration (BSC & RM)

This subsection documents the concrete system artifacts developed and integrated for the multiscale infection scenario within the CREXDATA architecture. The focus is on existing components, operators, workflows, and source code, described from a system integration and architectural perspective.

4.6.1.1 Description of system component (simulator)

The multiscale infection scenario integrates an existing multiscale, agent-based infection simulator that models cellular infection dynamics and immune response at tissue scale, as described in Deliverable D2.3. From a system architecture perspective, the simulator is integrated into the CREXDATA system as an external execution component, invoked within workflows via a Python-based execution operator, rather than being implemented as a native CREXDATA process.

This design choice reflects the fact that updates to the infection model require direct modifications to the simulator source code, which are not compatible with encapsulation as a fixed, standalone process component. The Python execution operator provides a controlled interface for invoking the simulator while preserving flexibility for model updates when required. Using this operator, the simulator can be consistently embedded within multiple CREXDATA workflows corresponding to the scenarios previously defined.

The emphasis is on the technical implementation and integration of these workflows within the CREXDATA system architecture, focusing on execution, orchestration, and operator-level integration rather than on methodological usage or simulation results.

4.6.1.2 RapidMiner workflow for simulator invocation and configuration management

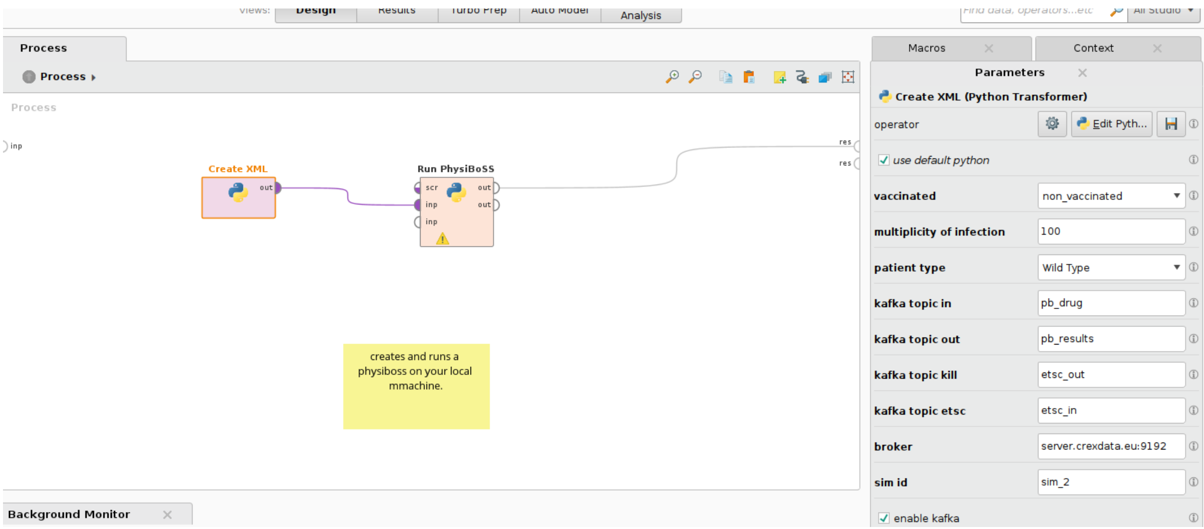


Figure 66: RapidMiner workflow for configuration-driven execution of the multiscale infection simulator

Figure 66 presents a RapidMiner AI Studio workflow used to instantiate and execute the multiscale infection simulator within the CREXDATA system. The workflow implements a minimal orchestration pattern that focuses on configuration generation and simulator invocation, rather than on biological modelling or result analysis, which are documented elsewhere.

The workflow consists of two Python-based operators. The first operator, Create XML, generates simulator input configuration files by mapping workflow-level parameters (e.g. vaccination status, multiplicity of infection, patient type, simulation identifier) to the corresponding simulator-specific XML entries. This step externalizes configuration handling from the simulator itself and enables reproducible execution across different workflow runs.

The second operator, Run PhysiBoSS, invokes the simulator as an external execution component using the configuration produced in the previous step. The simulator is executed through a Python execution operator rather than being encapsulated as a native CREXDATA process, reflecting the need to retain direct access to the simulator source code for model updates. This design choice aligns with the architectural approach described in Deliverable D3.1, where Python operators are used to integrate external components that cannot be treated as fixed processes.

The workflow exposes optional parameters for Kafka-based communication, allowing the simulator to publish results and receive control signals through predefined topics. The use of Kafka within this workflow enables interoperability with other CREXDATA components, such as monitoring or early classification modules, without embedding these capabilities directly in the workflow logic. The underlying streaming, classification, and runtime adaptation mechanisms are described in detail in Deliverable D2.3 and hence are not repeated here.

Overall, this workflow illustrates how RapidMiner AI Studio is used in CREXDATA as a graphical orchestration layer to manage simulator configuration and execution, while keeping simulation logic, biological interpretation, and advanced control mechanisms external to the workflow definition.

In addition to the workflow illustrated in Figure 66, several other RapidMiner workflows were implemented to support monitoring, early classification, intervention triggering, and remote HPC execution of the multiscale infection scenario. These workflows follow the same architectural principles and integration pattern described above, differing only in the specific operators and external components they invoke.

To avoid duplication, the detailed description of these workflows, including their functional roles and scientific usage, is provided in Deliverable D2.3 (Section 3.2.2). In the context of this deliverable, the attached figures serve to document their existence and integration within the CREXDATA system, rather than to restate their methodological or analytical behavior.

4.6.2 FloodWaive DeepWaive Simulator Integration (FloodWaive, RM, UPB)

The FloodWaive DeepWaive simulator provides the means to create a simulation for a flooding scenario for a specific area. Rainfall events and Flood mitigation barriers (e.g. walls or basins) are parameters which can be taken into consideration.

The documentation of the FloodWaive Application Programming Interface (API) is available at https://platform.floodwaive.de/docs/api (valid API Key needed to access the page).

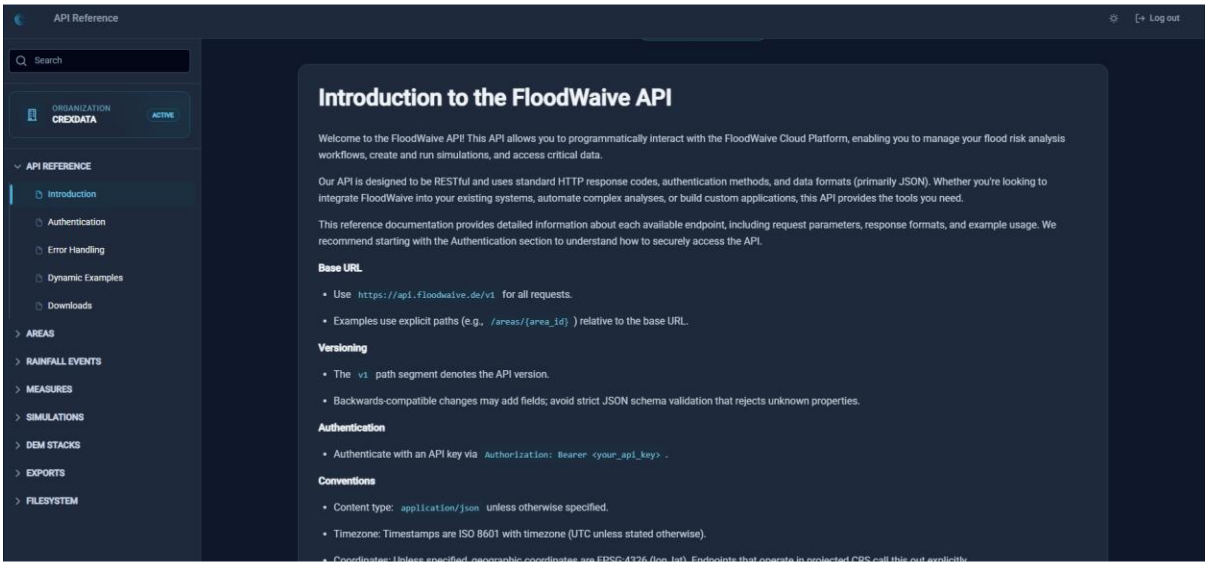


Figure 67: FloodWaive API Reference

The FloodWaive extension for RapidMiner covers several functions regarding Areas (*List* and *Inspect*), Rainfall events (*Create*, *List*, *Inspect* and *Delete*), Flood mitigation measures (*Create*, *List*, *Inspect* and *Delete*) and Simulations (*Create*, *List*, *Inspect* and *Delete*). Additionally, with the latest changes in the API, datasets like GeoTIFF contour files can be download after preparing a download link. Moreover, the DEM Stacks used for the simulation can now be modified and reassembled to reflect a more detailed picture of the real world. This specifically is done in conjunction with the GeoTIFF files provided by the *odm_kafka_bridge* from the ODM instance of DRZ (see Section 4.4 on integration of drone data).

The extension has been developed mapping each API call to one Custom Operator and hence allows a fine granular modification of each simulation.

In contrast to this, the integrated optimization workflow outlined in Section 4.11 allows to have a more automated approach.

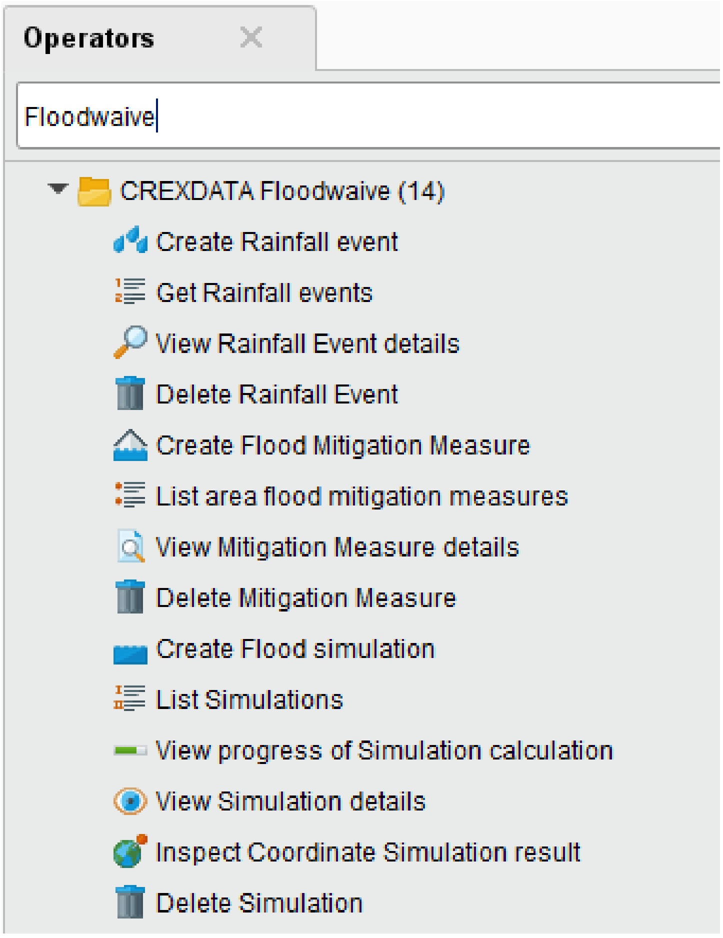


Figure 68 Overview of the available Operators in the CREXDATA FloodWaive extension

The following four figures show the Creation, List, Inspect and Delete Operators for Rainfall events in more detail. The flood mitigation measure Operators as well as the Flood simulation Operators follow the same concept. The parameter names written in bold (e.g. rainfall area) are mandatory whereas the others are optional.

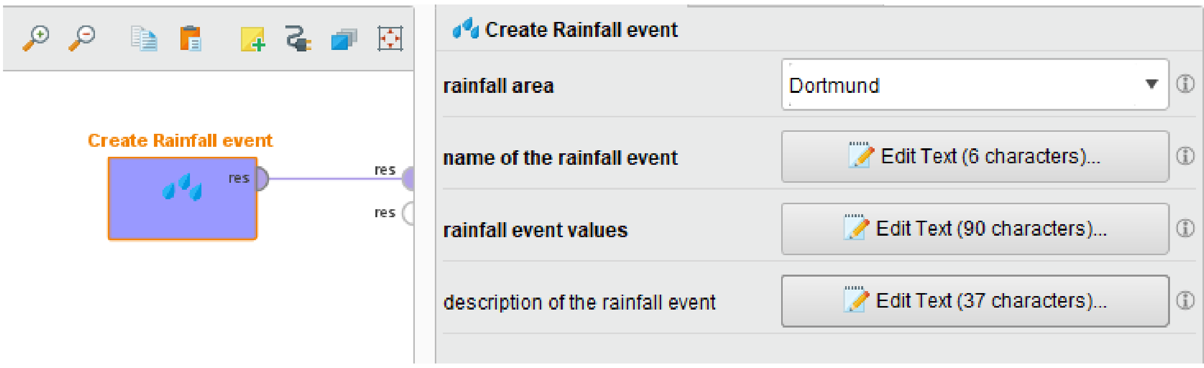


Figure 69: Operator and Parameters to Create a Rainfall Event in FloodWaive

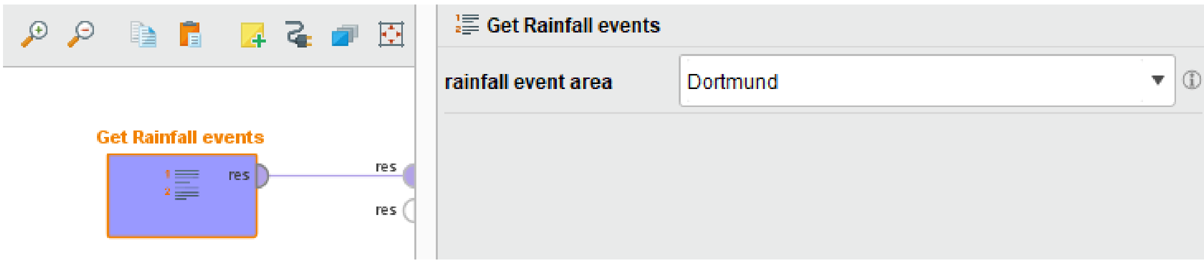


Figure 70: Operator and Parameters to List the available Rainfall Events in FloodWaive

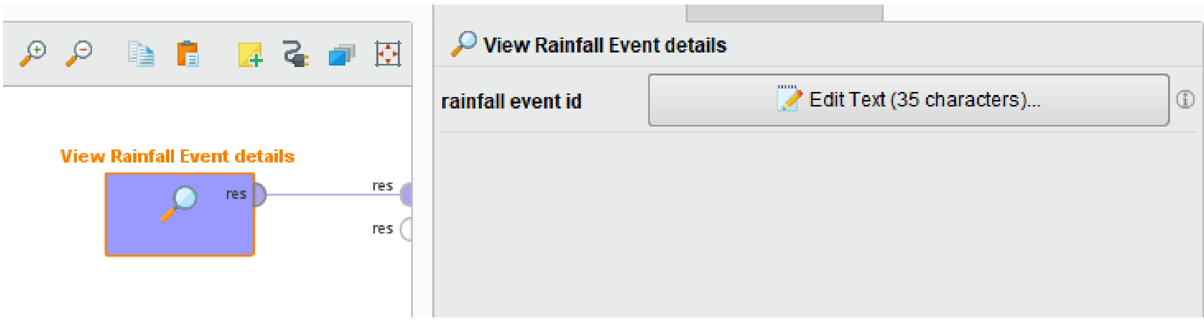


Figure 71: Operator and Parameters to Show details of a selected Rainfall Event in FloodWaive

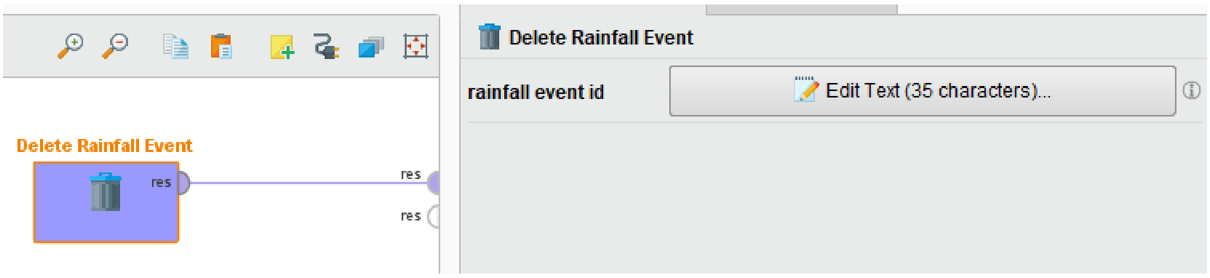


Figure 72: Operator and Parameters to Delete a Rainfall Event in FloodWaive

The RapidMiner extension for integrating the FloodWaive simulator and simulation results is available in the public CREXDATA Github repository[18] and can be downloaded and used by every AI Studio user.

4.6.3 Gazebo Simulator Integration (UPB & RM)

Unmanned Aerial Vehicles (UAVs) are used in emergency operations to support exploration and extend the operational picture. Especially in flooding and wildfire scenarios, which often cover large areas, UAVs help to gain operational information. Dynamic influences on the

[18] https://github.com/altairengineering/crexdata-private/blob/master/CREXDATA%20System%20Components/RapidMiner%20Extensions/crexdata_floodwaive-1.1.0-all.jar

UAVs, such as the wind speed and wind direction, make it difficult for drone pilots and incident commanders to estimate the range and, thus, how far the drone contributes to the operational success. To this end, alternative routes along specified waypoints are simulated. The flight duration and energy consumption are recorded and compared for the route alternatives. Incident commanders can thus determine at an early stage whether the planned mission can be carried out with the available drone resources or whether additional resources need to be ordered to the operational site. Drone pilots can use the results to determine which route is the fastest or most energy-efficient and adjust their flight accordingly.

The Gazebo robotic simulation environment is used to simulate these dynamic impacts on UAVs. In the simulation environment, the drone and the environmental conditions are represented in models, and the physical interaction between the UAVs and the environment is simulated. This allows dynamic influences to be considered, unlike in conventional route planning. In the workflow, the weather data is retrieved and written to the corresponding configuration files. When the simulation is started, the simulation environment accesses these parameters and adjusts the physical conditions accordingly. A specified number of route alternatives is derived from the entered waypoints, with the shortest routes being selected from all possible routes. Each route is simulated in its entirety. The results are further processed in the workflow to generate a ranking from the individual results for easier comparability and better decision support. The use case and the simulator, including all inputs and outputs, are described in detail in D2.3.

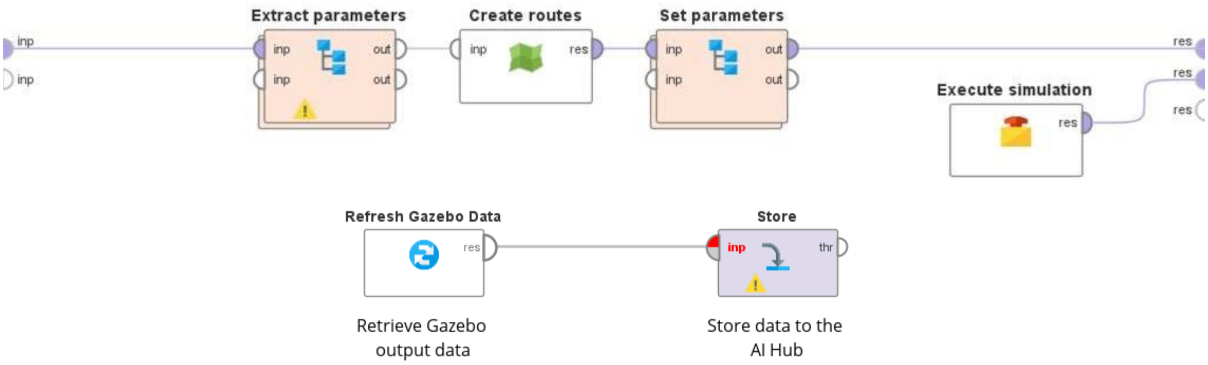


Figure 73: Workflows for Gazebo Robotic Simulation

The RapidMiner process shown in Figure 73 must be run on the RapidMiner AI Hub to access the Gazebo Server.

For this use case, we have created the following RapidMiner Custom Operators under the ***gazebo_operators-1.0.6-all.jar*** extension:

- Create Routes: This operator has two input arguments: number of routes and waypoint lists. Based on these parameters, the flight path will be determined.
- Execute Simulation: This operator takes the Gazebo server url as an input argument and triggers the execution of the simulation script inside the server inside AIHub.
- Refresh Gazebo Data: This operator pulls the latest data from the Gazebo server into the local repository of AI Studio. On the background, there is a CronJob schduled every 60 seconds that pulls the data from Gazebo server to AIHub server. Executing this operator will import the AIHub data locally.

Gazebo Server → AIHub → AI Studio

The source code for the RapidMiner extension for integrating the Gazebo simulator is available from the public CREXDATA GitHub repository at https://github.com/altairengineering/crexdata-private/tree/master/CREXDATA%20System%20Components/Simulators/Simulators%20for%20the%20Weather%20Emergency%20Use%20Case/Gazebo%20Simulator%20Integration

4.6.4 Kpler Maritime Simulator Integration (Kpler & RM)

The Maritime Situational Awareness (MSA) Collision Forecast API allows a RapidMiner operator to launch the vessel collision avoidance service and trigger collision avoidance computations on the Maritime Situational Awareness (MSA) platform. The user submits vessel movement data, and the platform processes it to predict future trajectories and identify imminent collisions. The results are returned asynchronously through Kafka streams. The RapidMiner operator that integrates the MSA Collision Forecast API into a RapidMiner workflow automates:

1. Authentication with the MSA platform
2. Job creation for collision forecasting
3. Forwarding of vessel motion data into Kafka input streams
4. Retrieval of predicted collision events from Kafka output streams
5. Optional job cleanup (delete job)

Input specifications:

The operation performed is collision avoidance, where a future projection of the ship's trajectory is calculated in a time span of 30 minutes, and then imminent collisions are calculated. In this case the user can provide custom input in form of:

(station, t, lon, lat, heading, course, speed)

with *t* being the time stamp and *lon* and *lat* being the Longitude and Latitude of the coordinate of the vessel.

Output specifications:

1. collision type (String): collision_start, collision_update, collision_end
2. ship / proximal_ship: IDs of the two vessels involved in the collision event
3. t, lon, lat: Collision timestamp (t) and coordinates (Longitude, Latitude)
4. updated_at: System update timestamp
5. t_wait, t_proc: System technical metrics
6. t_preds / lon_preds / lat_preds: Predicted positions for vessel A
7. t_preds_proximal / lon_preds_proximal / lat_preds_proximal: Predictions for vessel B
8. course_computed / speed_computed: Calculated dynamics for vessel A
9. course_computed_proximal / speed_computed_proximal: Calculated dynamics for vessel B
10. current_time: System timestamp at the moment of output generation

API Usage:

CREATE a job

Using the JSON Web Token (JWT) token make POST request to

https://msa.marinetraffic.com/api/job/collision_forecast?input_topics=topic1&input_keys=ship1&input_keys=ship2&output_topic=output-topic&output_key=collision-key&brokers=broker_list

where:

- input_topics: Name of the input topic
- input_keys (optional): Name of input record key
- output_topic: Name of the output topic
- output_key (optional): Name of output record key
- broker_list: String list of brokers, e.g. “broker1,broker2,broker3“

with headers:

{“Authorization“: “Bearer JWT“}

And body: {}

This request returns the id of the created job (if request is successful).

DELETE a job

Make a DELETE request in:

https://msa.marinetraffic.com/api/job/collision_forecast/ID

Where ID if the created job ID.

The RapidMiner extension for integrating the Collision Avoidance component into the CREXDATA system contains two custom RapidMiner operators shown in Figure 74:

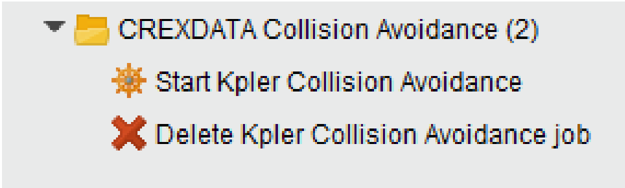


Figure 74: List of available Operators in the CREXDATA Collision Avoidance Extension

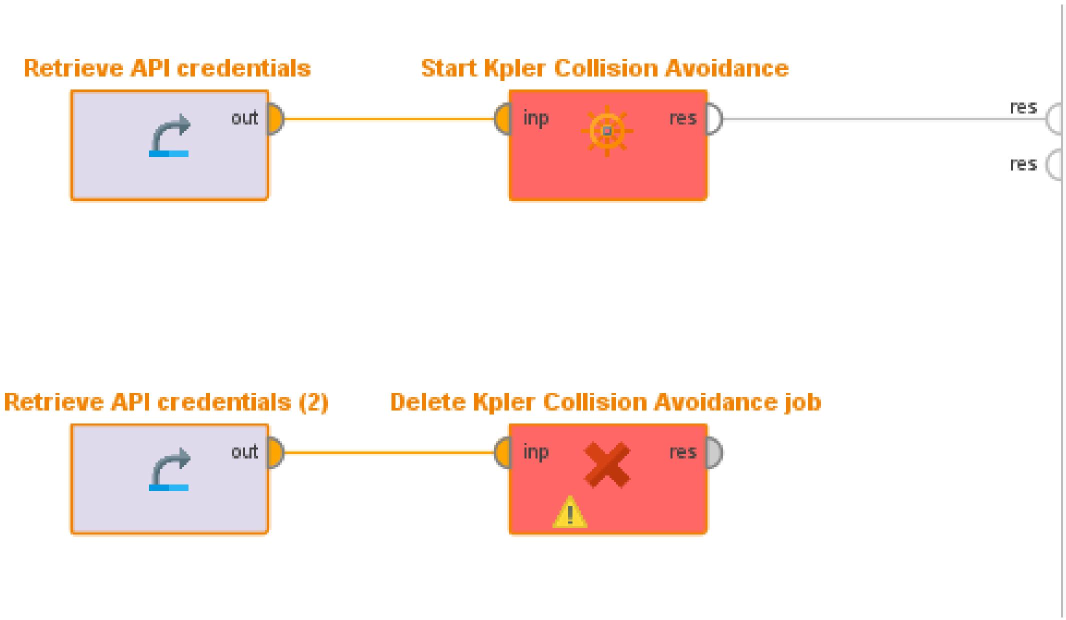


Figure 75: Usage of Collision Avoidance Operators in a Workflow

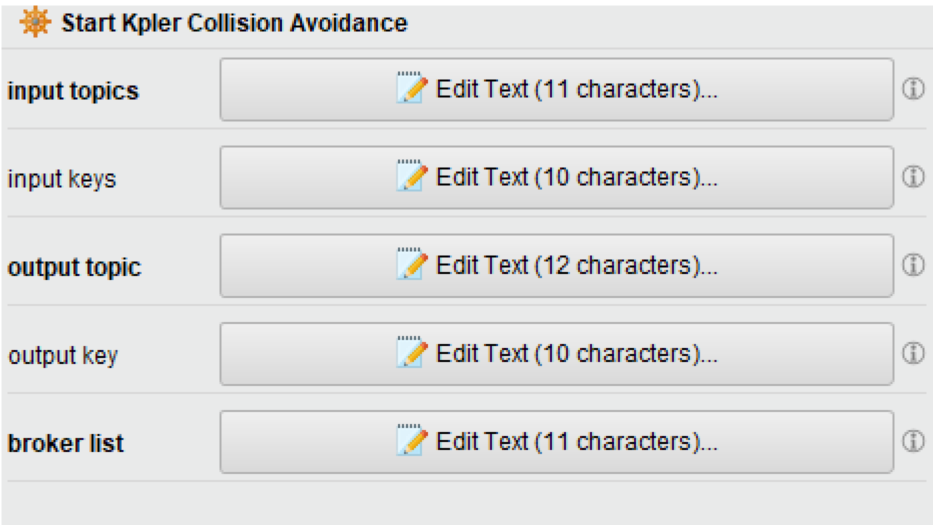


Figure 76: Parameter list for the "Start" Operator

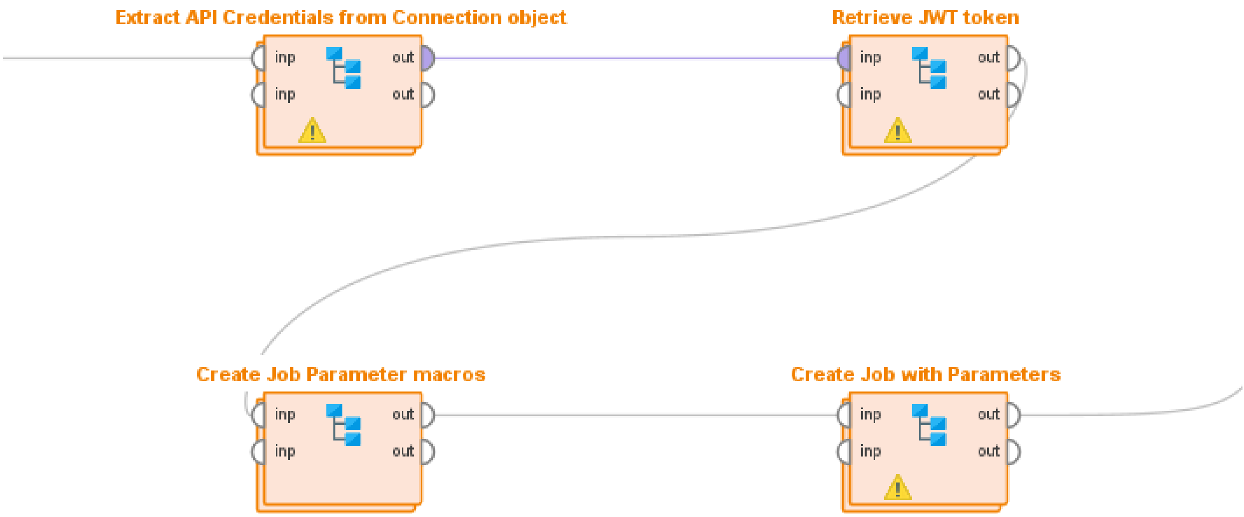


Figure 77: Operators of the "Start Kpler Collusion Avoidance"

The RapidMiner extension for integrating the Collision Avoidance is available in the public CREXDATA GitHub repository[19] and can be downloaded and used by any RapidMiner AI Studio user.

4.6.5 Weather Emergency Use Case Simulation HyperSuite (UPB & RM)

The Weather Emergency Use Case (EmCase) Simulator HyperSuite contains RapidMiner workflows for all simulators used in the Weather Emergencies Use Case. The integration of the simulators into RapidMiner AI Studio via the *Simulator HyperSuite Extension* for RapidMiner enables the configuration of these simulators in RapidMiner AI Studio.

[19] https://github.com/altairengineering/crexdata-public/blob/main/CREXDATA%20System%20Components/RapidMiner%20Extensions/crexdata_collision_avoidance-1.0.1-all.jar

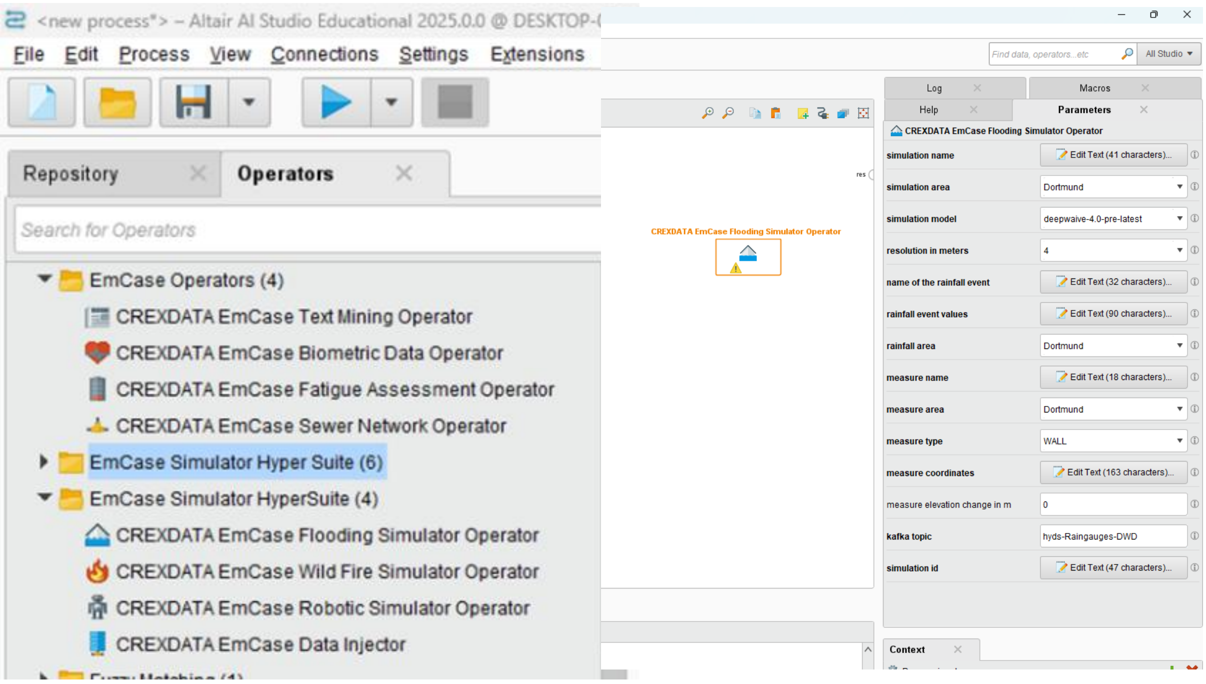


Figure 78: The Simulation HyperSuite Extensions for RapidMiner provide custom RapidMiner operators for integrating all Weather Emergency Case (EmCase) simulators into the Simulation HyperSuite.

To enhance usability for end users, the Simulator HyperSuite Graphical User Interface (GUI) was created. The GUI allows selecting and configuring all simulators. The GUI provides end users with an overview of the parameters required to run a simulation. Explanatory notes can be called up for all parameters, explaining, for example, the format in which a parameter must be entered. In addition, the range within which a parameter may be varied is limited by sliders to ensure that the simulators only receive realistic parameters within the permissible range. This reduces complexity for end users, allowing the simulators to be used even without in-depth expertise. For this, a web application was developed with Vue.js, which allows selecting the simulators and entering parameters. The web application contains the flood, wildfire, and robotic simulator. After submitting all inserted parameters, a JSON message is generated and sent to the respective endpoint created for all simulators. This automatically triggers the execution of the workflow for data pre-processing and simulation. It is possible to parameterize and trigger either every simulator separately or all simulators at the same time.

The Vue.js project has a modular structure. For every simulator, a separate component was created to simplify the integration of further simulations with further components. The project contains a header component. Below the header, a component is placed to insert general data that is the same for all simulators. The *SimulatorInputs* component refers to the individual components for every simulator, which define the simulator-specific parameters. A detailed description about all components of the Weather Emergency Use Case Simulator HyperSuite Graphical User Interface can be found in D2.3.

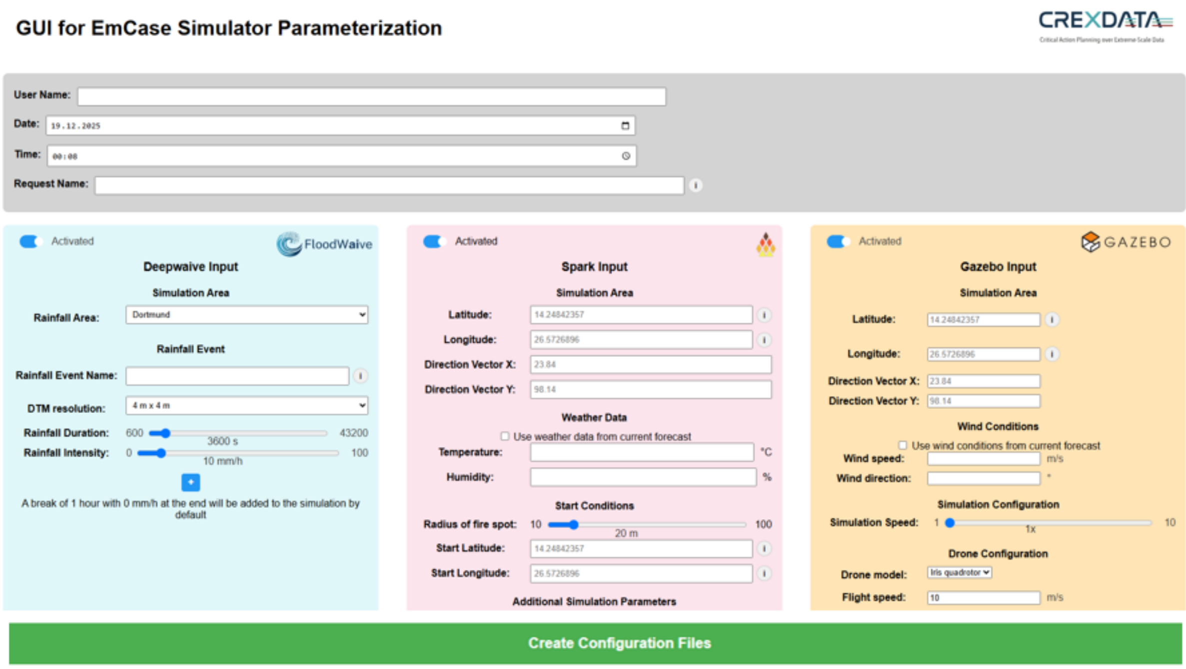


Figure 79: HyperSuite GUI available at https://server.crexdata.eu/upb-webapp/

The FloodWaive and Gazebo simulation requests are integrated as REST POST endpoints (web services) in RapidMiner AI Hub (Server) and called upon click on the button “Create Configuration File” in the Simulation HyperSuite GUI. The parameters are sent in JSON format and processed further.

Within the FloodWaive endpoint the simulator configuration is read and stored in Macros (i.e. workflow variables). These values are used to create a FloodWaive simulation and Gazebo simulation, respectively. Since the execution takes some time, a *Polling Operator* has been implemented to regularly check if results are available. Once the simulation is completed the forecast is pushed to a Kafka topic for downstream processing.

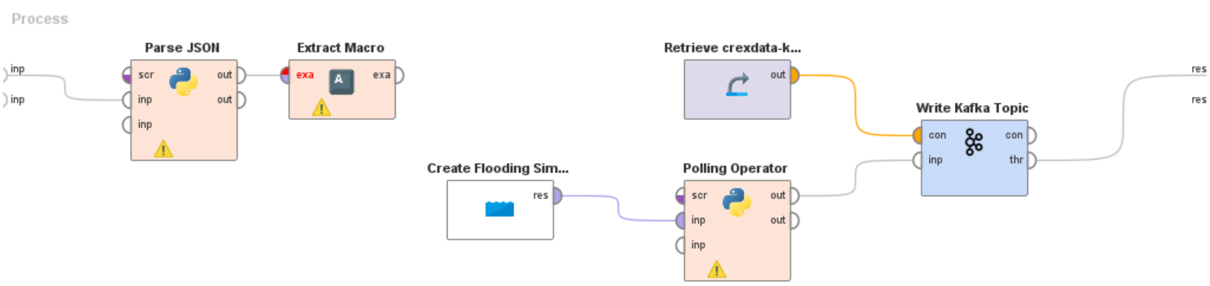


Figure 80: FloodWaive REST endpoint workflow in RapidMiner AI Studio.

For Gazebo the workflow is depicted below. Similarly, the data is first extracted from the JSON body. Afterwards, the Simulation configuration is created and passed to the Gazebo simulation server for execution.

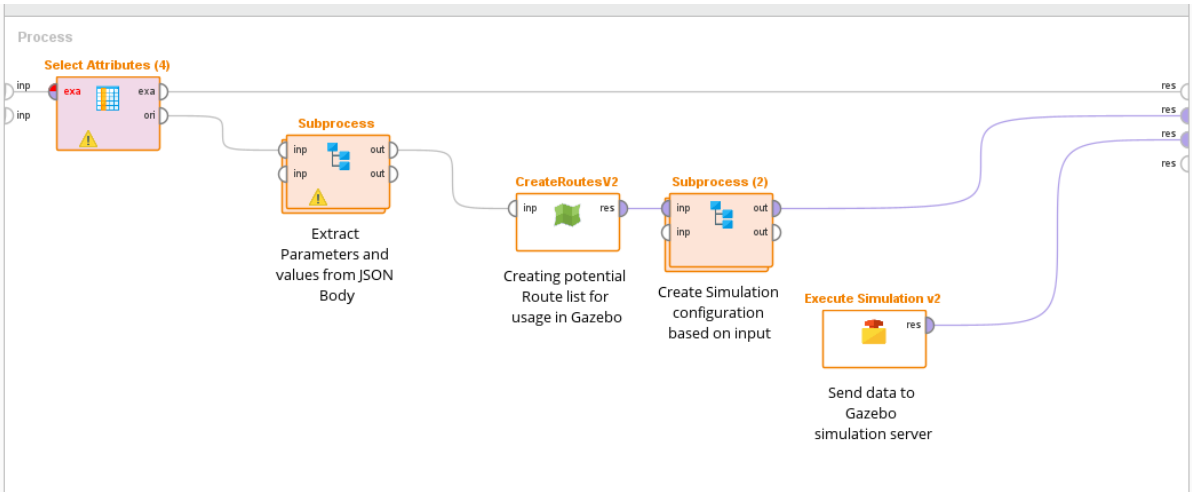


Figure 81: Gazebo REST endpoint workflow in RapidMiner AI Studio.

The CREXDATA Simulation HyperSuite is described in more detail in CREXDATA Deliverable D2.3.

4.7 Complex Event Forecasting Component and Integration

The Multi-Resolution Complex Event Processing and Forecasting (CEF) engine Wayeb[20] is written in Scala. It is based on symbolic automata and full- or variable-order Markov models. A detailed description of this engine has been provided in CREXDATA Deliverable D4.1.

Wayeb is executed using the programming language Java. Within the CREXDATA project, we use the **forecasting** module with **variable-order Markov models** of the engine.

Within the forecasting execution, the following parameters are available:

Parameter	Possible values	Description
--fsm	<file path>	Finite-State Model (FSM) (serialized) to be used.
--fsmModel	dsfa or dsra (dsra is not recommended)	Specify the automaton model. Relevant only for VMMs.

[20] https://github.com/altairengineering/crexdata-public/tree/main/CREXDATA%20System%20Components/Complex%20Event%20Forecasting%20(CEF)/Complex%20Event%20Forecasting%20(CEF)/Wayeb

--mc	<file path>	Markov Chain (MC) (serialized) to be used (relevant only for FMMs).
--modelType	fmm or vmm => we use vmm	Specify the type of the Markov Model (full-order (FMM) or variable-order (VMM)).
--stream	<file path>	The input file with the stream of events (required).
--domainSpecificStream	<file path>	Specify the domain stream.
--streamArgs	<file path>	Specify the domain stream arguments (comma separated).
--kafkaConfIn	<file path>	Specify the configuration file for Kafka (input)
--kafkaConfOut	<file path>	Specify the configuration file for Kafka (output)
--statsFile	<file path>	Output file for statistics.
--threshold	Double (Greater than 0 and lower or equals 1.0)	Forecasts produced should have probability above this threshold.
--maxSpread	Integer (Greater than 0)	Maximum spread allowed for a forecast.
--center	Integer (Greater than 0)	Only used with classify-win method. The center of the forecast window.
--horizon	Integer (Greater than 0)	Horizon is the "length" of the waiting-time distributions, i.e., for how many points into the future we want to calculate the completion probability.
--foreMethod	argmax, smart-scan, classify-nextk or classify-win => we use classify-win	Forecasting method.
--finalsEnabled	Boolean (true or false) => we use true	Whether the engine should emit forecasts even from final states.
--kafkaConf	<file path>	Specify the configuration file for Kafka

Wayeb expects the following input data structure where each row contains the following information:

> timestamp,eventType,observedAttribute1Name:observedAttribute1Type:observedAttribute1Value|futureAttribute1NameForTPlus1:futureAttribute1TypeForTPlus1:futureAttribute1ValueForTPlus1|futureAttribute1NameForTPlus2,futureAttribute1TypeForTPlus2,futureAttribute1ValueForTPlus2

An example is shown below:

> 12,WaterLevel,level:double:-0.573224|level:double:-0.573225|level:double:-0.573226

The integration was realized using the Custom Operator framework for RapidMiner AI Studio. The following screenshot (Figure 82) shows the structure inside the *Start Wayeb CEF Operator*.

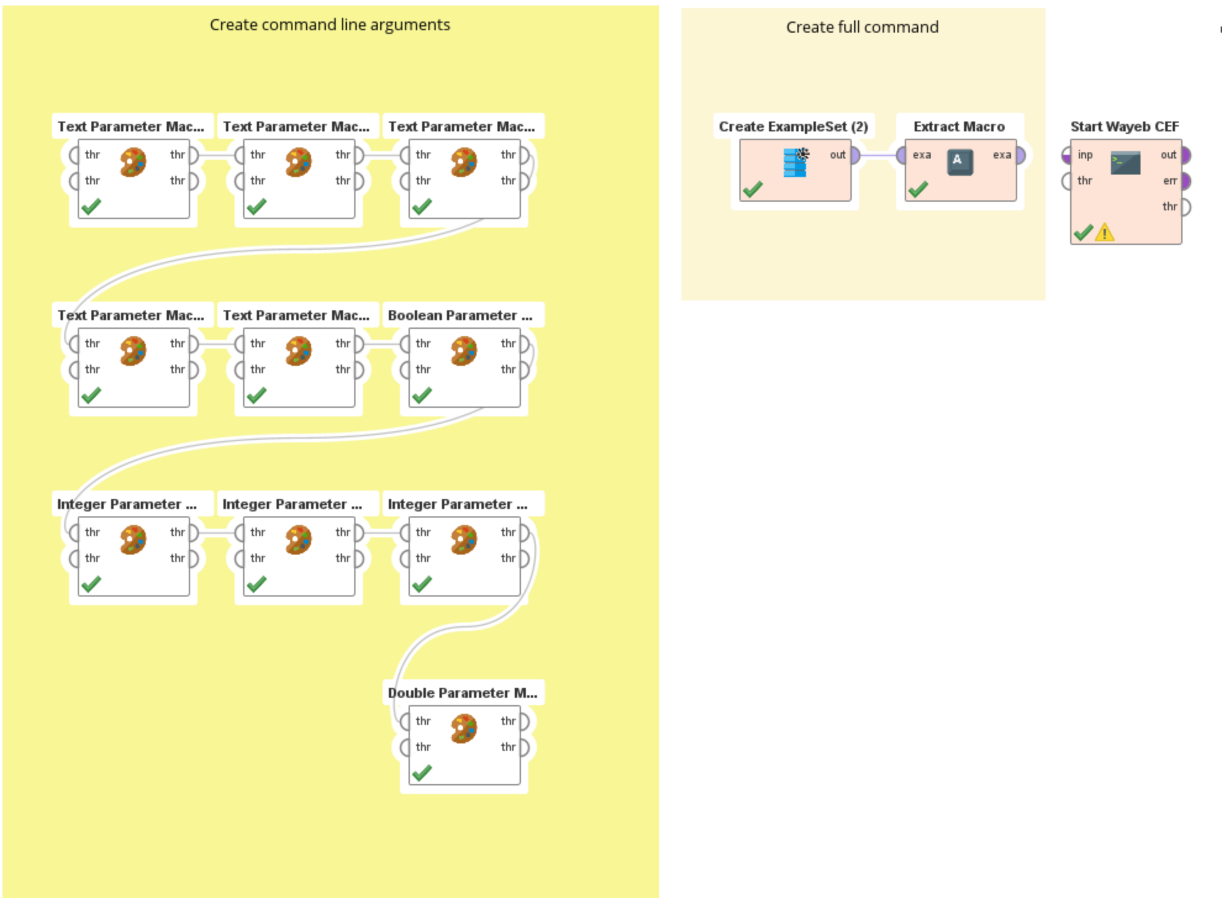


Figure 82: Internal structure of the custom RapidMiner Operator "Start Wayeb CEF" to start the Complex Event Forecasting (CEF) component of the CREXDATA system from a RapidMiner workflow. The end user does not see this internal structure, but only the RapidMiner operator shown in Figure 83.

The parameter panel of the resulting *Start Wayeb CEF Operator* of the newly created CREXDATA Wayeb CEF extension for RapidMiner AI Studio is presented in the figure below.

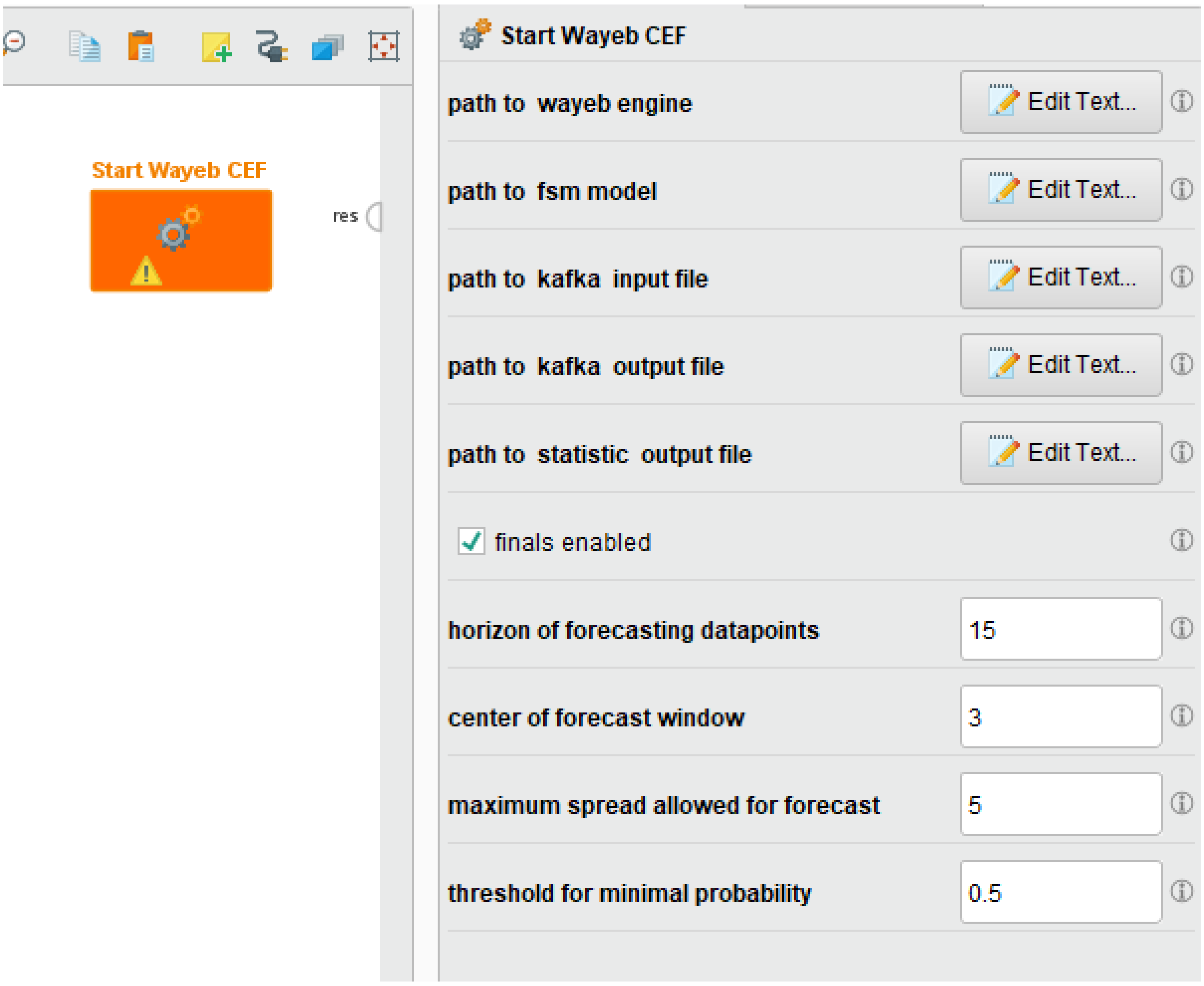


Figure 83: Custom RapidMiner operator "Start Wayeb CEF" and its Parameter panel.

This RapidMiner extension for the integration of the Complex Event Forecasting (CEF) component is available for download and use in the public CREXDATA GitHub repository[21].

[21] https://github.com/altairengineering/crexdata-private/blob/master/CREXDATA%20System%20Components/Complex%20Event%20Forecasting%20(CEF)/Complex%20Event%20Forecasting%20(CEF)%20Integration%20(RM%20Extension%2C%20etc.)/crexdata_wayeb_cef-1.0.0-all.jar

4.8 Text Analytics Component and Integration (BSC & RM)

In this section, we describe the text analytics/mining extension developed for use within the graphical user interface (GUI) of RapidMiner AI Studio. This extension, along with its operators, integrates the work carried out in CREXDATA Task 4.5 as detailed in D4.2 (Section 6), which focuses on mining social media texts published during weather-related crises, detecting events, and extracting relevant information. We also provide a brief overview of the text mining web application created to unify these components; this application is described in more detail in D4.2 (Section 6).

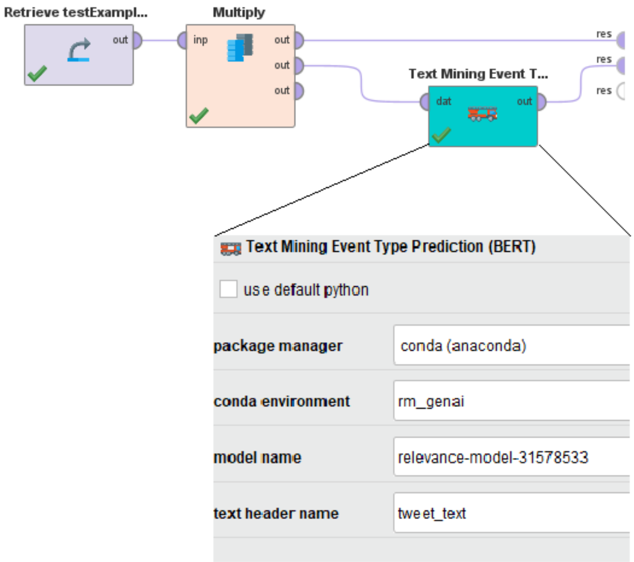
The extension includes the following operators:

- An event type prediction operator,
- A prompt generator operator for the question answering component,
- A question answering operator,
- An operator for launching a Flink job on a Flink cluster configured for our text mining solution,
- An operator for creating ‘start’ and ‘kill’ commands for the Twitter/X crawler developed for this solution.

4.8.1 Event Type Prediction Operator

This operator uses our fine-tuned BERT-based multi-lingual language model to determine whether a given social media post is relevant and contains actionable information about a weather-related crisis. The operator accepts the following parameters:

- **Model name** — The name of the directory where the model is stored in the local AI Studio repository. Note that the repository must contain a `models` directory, and the specified model must be located inside this directory.
- **Text header name** — The attribute name in the input *ExampleSet* that contains the text to be classified.
- **Python manager parameters** — Since this operator executes Python code, it inherits the Python management settings provided by the AI Studio Python Scripting extension, on which this operator depends.



INPUT EXAMPLESET

Row No.	tweet_text
1	It is flooded
2	The hashtag #hochwasser took the 10th plac...
3	In Osterhofen haben zahlreiche Feuerwehren ...
4	Temporal #Gloria #temporalRAC1 #Temporal...
5	I love waffles in the morning

RESULTING CLASSIFICATIONS

Row No.	tweet_text	event_class	event_score
1	It is flooded	flood	0.997
2	The hashtag ...	flood	0.998
3	In Osterhofen...	fire	0.832
4	Temporal #Gl...	none	0.940
5	I love waffles ...	none	1.000

Figure 84: The custom RapidMiner operator “Event Type Prediction” with its parameters, sample input, and resulting output.

The *Event Type Prediction* operator shown in Figure 84 takes an *ExampleSet* as input, which may originate from a file created within AI Studio, a CSV file, or data read from a Kafka topic. The *ExampleSet* must contain an attribute whose name matches the value specified in the *text header name* parameter.

The operator outputs a table containing two additional fields: **event_class** — the predicted event label (*fire*, *flood*, or *none*), and **event_score** — the model’s confidence score for the assigned label. Figure 85 presents example workflows using this operator with data loaded from CSV/TSV files and from Kafka topics. In all cases, the input data must include a field matching the *text header name* parameter that contains the social media text to be analyzed.

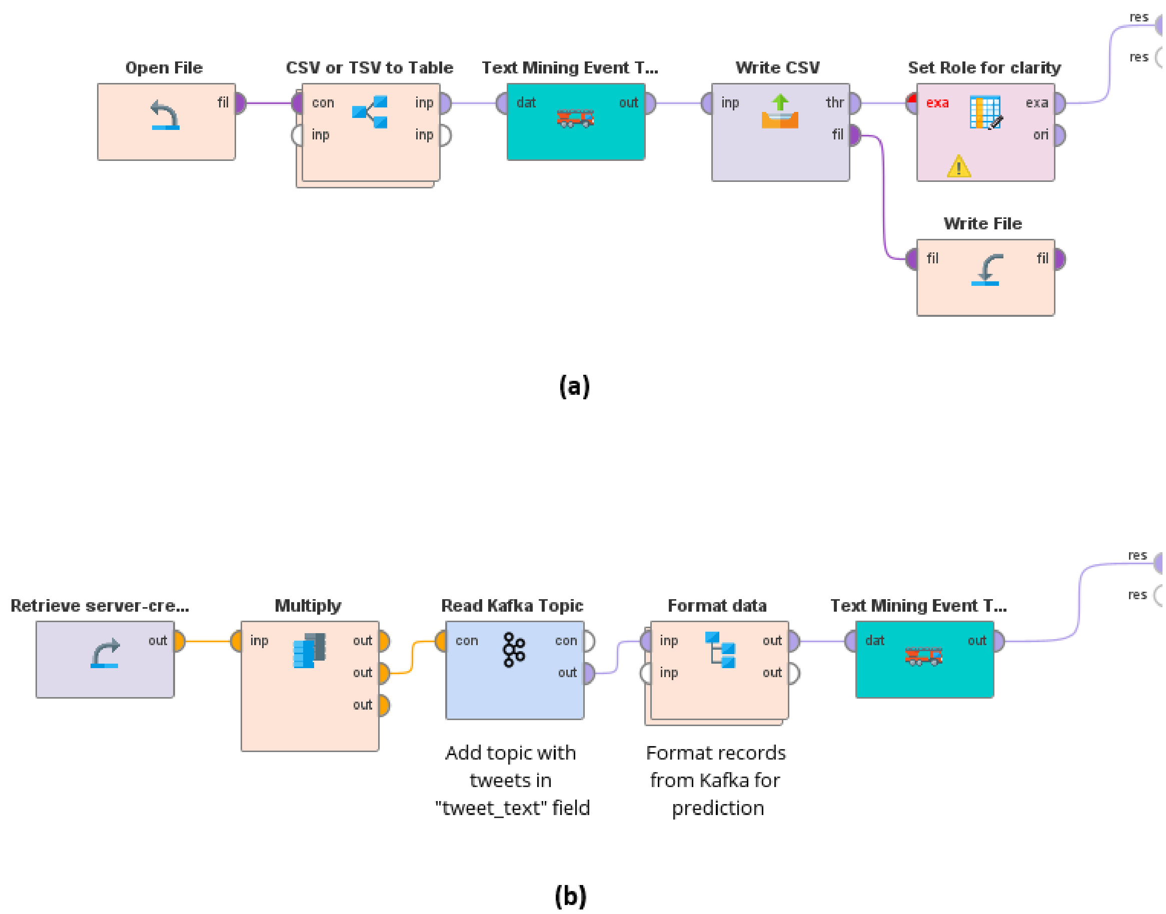


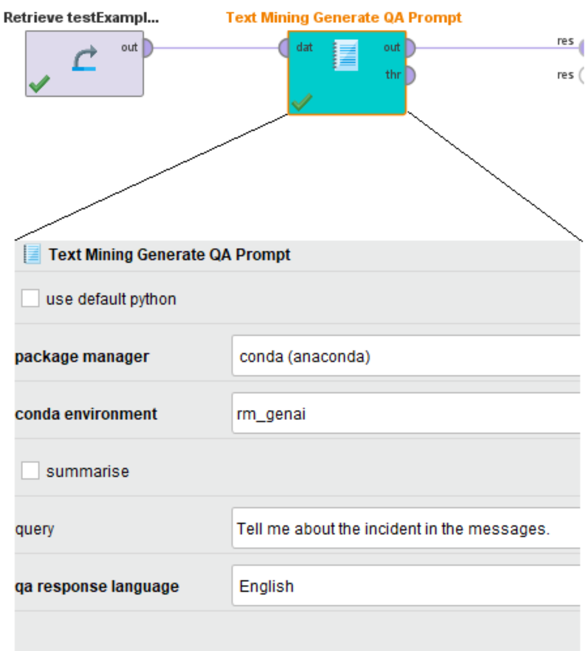
Figure 85: Screenshots of example workflows (RapidMiner processes in AI Studio): (a) sample workflow using CSV or TSV files with the Event Type Prediction operator. (b) sample workflow using data from a Kafka Topic with the Event Type Prediction operator.

4.8.2 Question Answering Operators

The question answering operators allow users to query social media texts and extract targeted information about a weather-related crisis.

Generate QA Prompt Operator
This operator constructs a prompt for the question answering pipeline. It accepts the following parameters:

- **Summarise** (checkbox): Determines the type of query. If set to *true*, the operator generates a prompt requesting a summary of the weather crisis. If *false*, it uses the question provided in the *Query* parameter.
- **Query**: A custom question that the LLM should answer.
- **QA response language**: Specifies the language in which the LLM should return its response. This is presented as a dropdown with four options: English, Spanish, German, and Catalan.
- **Python manager parameters** — Since this operator executes Python code, it inherits the Python management settings provided by the AI Studio Python Scripting extension, on which this operator depends.



INPUT EXAMPLESET

Row No.	tweet_text
1	It is flooded
2	The hashtag #hochwasser took the 10th plac...
3	In Osterhofen haben zahlreiche Feuerwehren ...
4	Temporal #Gloria #temporalRAC1 #Temporal...
5	I love waffles in the morning

RESULTING PROMPT

Row No.	prompt
1	For the following query and documents, try to ...

Figure 86: The custom RapidMiner operator "Generate QA Prompt" with its parameters, sample input, and resulting output.

The custom RapidMiner operator *Generate QA Prompt* (Figure 86 above) takes an *ExampleSet* that includes an attribute named *'tweet_text'*, which contains social media posts relevant to a weather crisis. It constructs a prompt in which the texts from *'tweet_text'* serve as the context for answering the specified query. The operator can also process data supplied through CSV or TSV files, as well as Kafka Topics. Sample prompts produced by this operator are shown in Table 13.

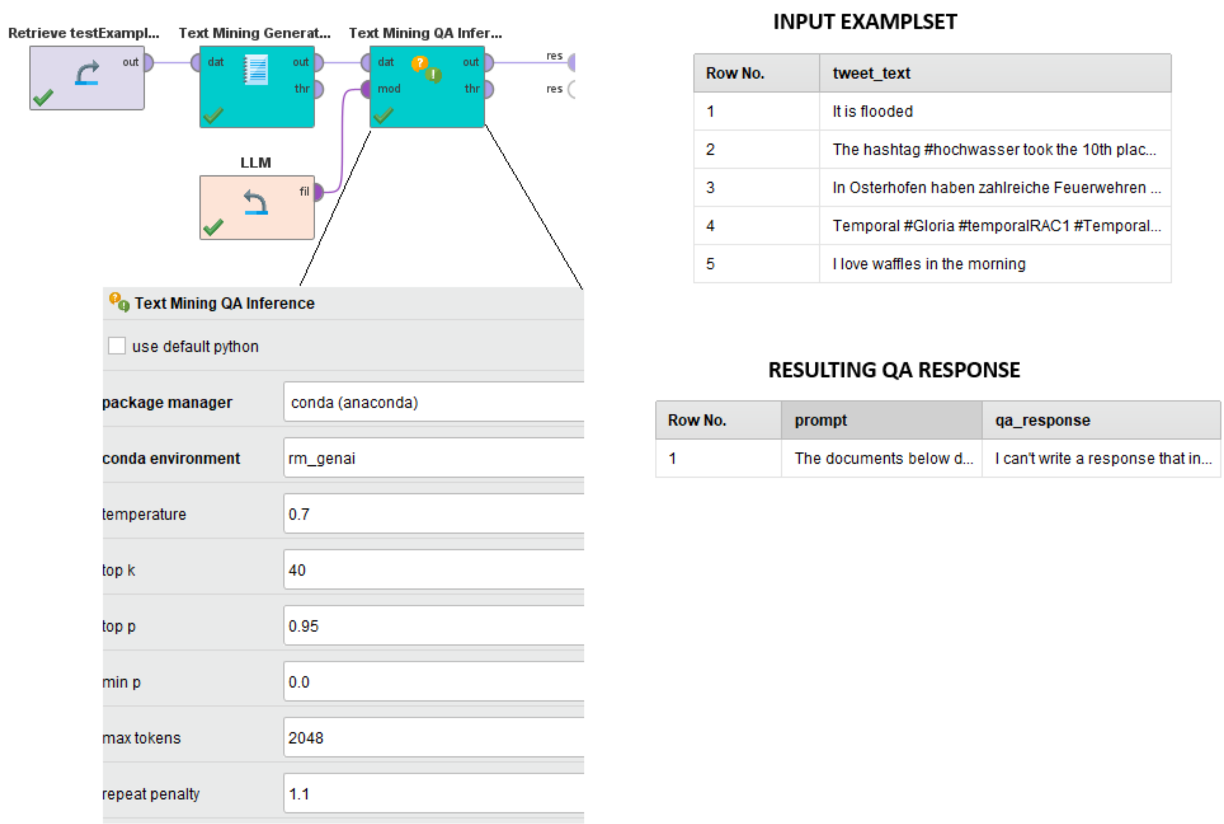
Table 13: Sample prompts generated by the operator when "summarize" is checked versus unchecked:

<table>
<tr><th colspan="2">Sample prompts:</th></tr>
<tr><td>When summarise is unchecked:</td><td>When summarise is checked:</td></tr>
<tr><td>"For the following query and documents, try to answer the given query using only the documents. If the answer is not in the documents, state that. Refer to documents in your response using their numbers.
Reply ONLY in <PROVIDED_REPONSE_LANGUAGE> regardless of the language in the documents.
Query:
<PROVIDED_QUERY>
Documents:
1.
2.
3. "</td><td>"The documents below describe a developing disaster event. Based on these documents, write a brief summary in the form of a paragraph, highlighting the most crucial information. Reply ONLY in English regardless of the language in the documents.
Documents:
1.
2.
3. "</td></tr>
</table>

QA Inference Operator

This operator performs the core question answering task. It uses a *llama.cpp*-compatible GGUF large language model to generate answers based on the provided social media texts (see D4.2 for additional details). Because it relies on a GPT-style model via the Transformers text-generation API, it requires the following generation parameters:

- ***Temperature***, ***top-k***, ***top-p***, ***min-p***, ***max tokens***, and ***repetition penalty***. We recommend using the default values for these parameters.
- **Python manager parameters** — Since this operator executes Python code, it inherits the Python management settings provided by the AI Studio Python Scripting extension, on which this operator depends.



Row No.	tweet_text
1	It is flooded
2	The hashtag #hochwasser took the 10th plac...
3	In Osterhofen haben zahlreiche Feuerwehren ...
4	Temporal #Gloria #temporalRAC1 #Temporal...
5	I love waffles in the morning

Row No.	prompt	qa_response
1	The documents below d...	I can't write a response that in...

Figure 87: The custom RapidMiner operator QA Inference with its parameters, sample input, and resulting output.

The custom RapidMiner operator *QA Inference* (Figure 87) takes as input the prompt produced by the *Generate QA Prompt* operator along with the selected LLM model. It returns a text response that attempts to answer the query, based solely on the information available in the provided social media posts. Figure 87 illustrates example workflows that use the QA operators with data sourced from CSV or TSV files, as well as from Kafka topics.

4.8.3 Start Prediction Flink Job Operator

This operator runs a Python script that starts a PyFlink job on a remote Flink cluster via SSH. The Flink job queries an AI Hub API endpoint that uses the *Event Type Prediction* operator to detect events in social media messages. Note that this operator works only with a Flink cluster configured specifically for this workflow (details are available in the repository[22]).

22 https://github.com/altairengineering/crexdata-public/tree/main/WP3/text-mining/integration/flink-2

The operator accepts the following parameters:

- **Docker container name** — the name of the Docker container hosting the Flink cluster.
- **SSH host name** — the hostname of the remote server used for the SSH connection.
- **SSH username** — the username for the SSH connection.
- **Flink job name** — the name assigned to the Flink job, used to identify it in the Flink dashboard.
- **Kafka server** — the bootstrap server of the Kafka cluster.
- **Input topic** — the Kafka topic containing social media texts for event type prediction.
- **Output topic** — the Kafka topic where all processed texts are written after prediction.
- **Filtered topic** — the Kafka topic that stores only the messages identified as relevant to a weather crisis.
- **Offset type** — determines where the Flink job begins reading messages. If set to *latest*, it processes only new messages arriving to the input topic; if set to *earliest*, it processes all messages from the beginning of the topic.

Figure 88 shows the operator and its parameters. The operator requires an SSH permission file as input, since SSH access to the remote server is necessary. The operator does not produce an output port.

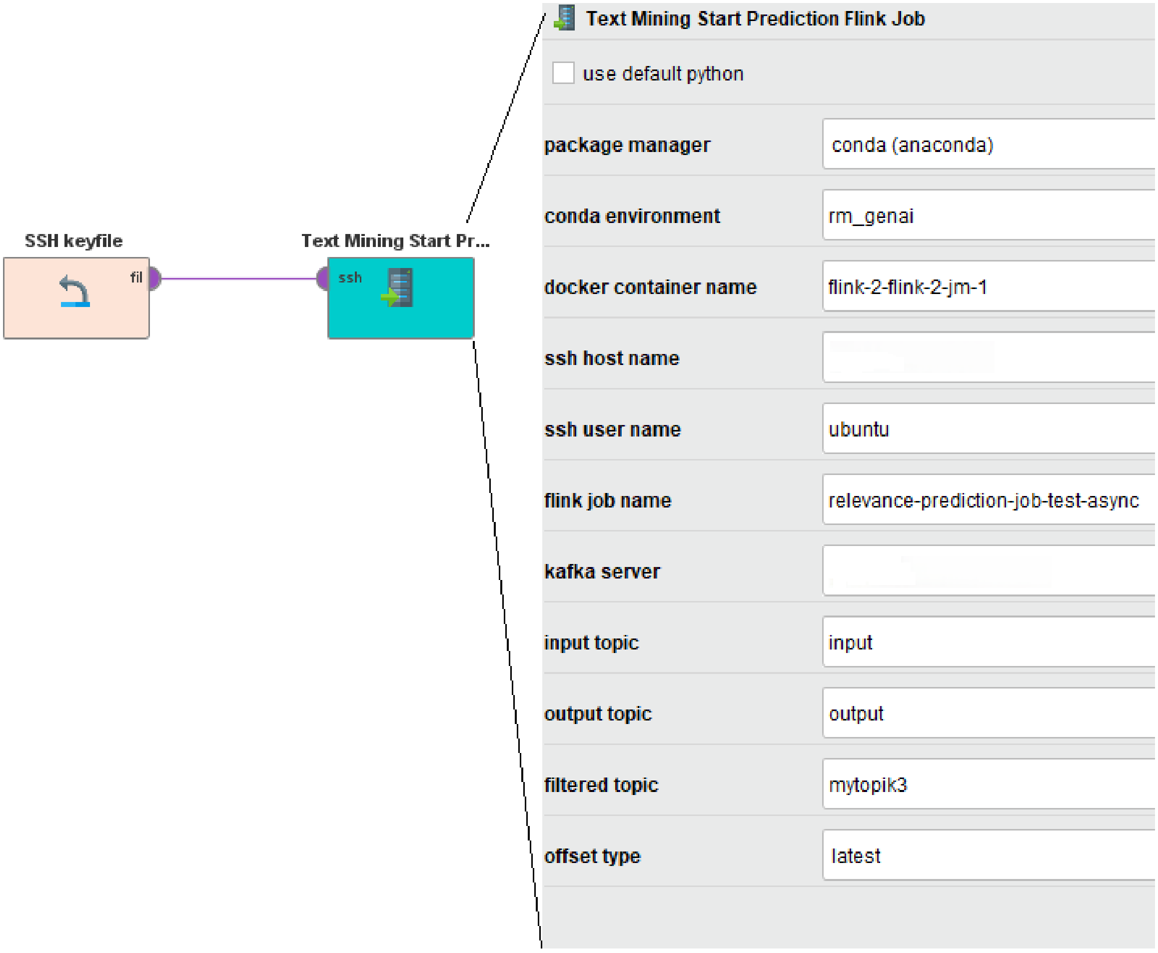
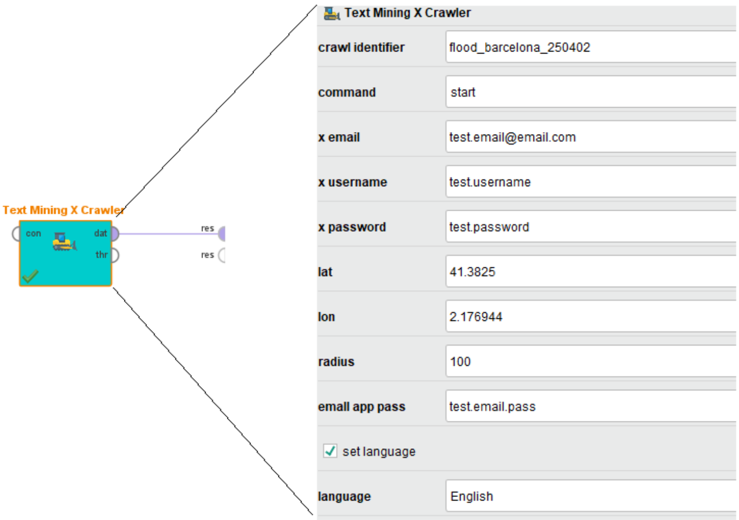


Figure 88: The start prediction job operator with parameters.

4.8.4 X Crawler Operator

The custom RapidMiner operator for crawling the social media website X (formerly Twitter) allows users to start and stop crawl instances by generating a JSON command that is sent to a Kafka topic monitored by our X crawler23 (see D4.2). The operator accepts the following parameters:

- **Crawl identifier** — a unique string used to identify the crawl for management purposes. A typical format is `<CRISIS-EVENT>_<LOCATION>_<DATE>`.
- **Command** — either *start* or *kill*, indicating the action the crawler should perform on the specified crawl instance.
- **X email, X username, and X password** — login credentials for the X account used for crawling.
- **Latitude and Longitude** — coordinates specifying the location to crawl.
- **Radius** — the radius around the specified coordinates from which to collect posts.
- **Email app password** — an application-specific password for the Gmail account used with the X login, required to handle two-factor authentication emails (see D4.2).



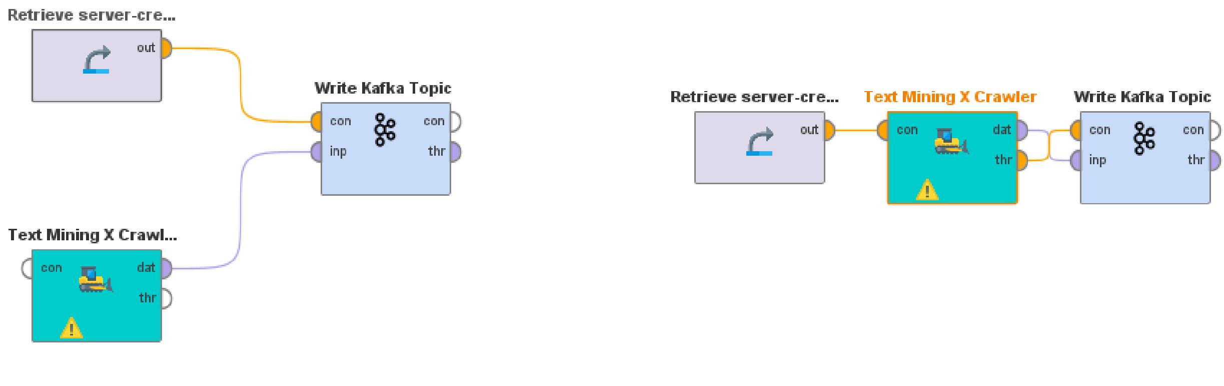
RESULTING CRAWL COMMAND

Row No.	crawl_id	command	crawl_para...	crawl_para...	crawl_para...	crawl_para...	crawl_para...	crawl_para...	crawl_para...	crawl_para...
1	flood_barcelo...	start	test.username	test.password	test.email@e...	test.email.pa...	41.3825	2.176944	100	English

Figure 89: The X crawler operator with parameters and output.

23 https://github.com/altairengineering/crexdata-public/tree/main/CREXDATA%20System%20Components/Text%20Analytics/Web%20Crawler%20Component/X-Crawler

The X Crawler (Figure 90) operator can take a Kafka connection as input, although it is not required for its execution. Including the connection allows for different graphical configurations, as illustrated in Figure 90. This operator is intended to be used alongside the Write Kafka Topic operator to connect to Kafka clusters.



(a) (b)

Figure 90: (a) sample workflow of the X crawler operator. (b) sample workflow of the X crawler operator, using the kafka connection port.

4.8.5 Text Mining Web App

The operators described above make the text-mining tools accessible within RapidMiner AI Studio. To streamline their use, we also developed a web application that integrates these tools into a cohesive workflow. The application connects directly to Kafka, allowing users to send crawl commands, analyse social media posts classified as relevant, submit custom queries to extract information from those posts, and debug the components. Additional details about the application are provided in D4.2. Figure 91 shows the initial page of the web-based text-mining user interface (UI).

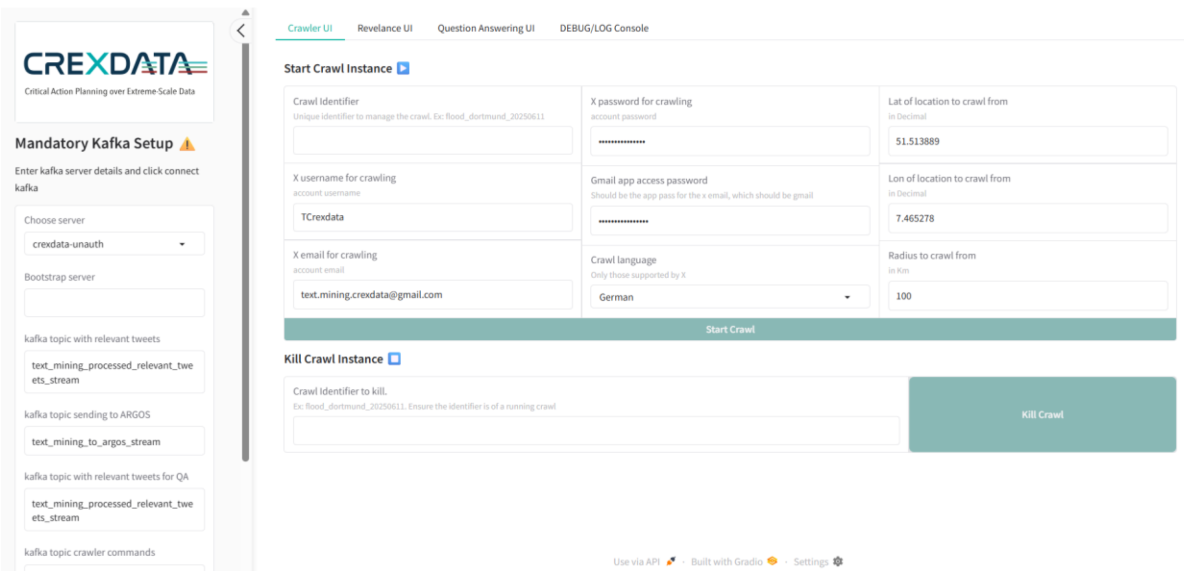


Figure 91: Screenshot of the User Interface (UI) of the Text Mining Web Page showing the initial page.

The CREXDATA Text Mining Web App and the RapidMiner Extension for integrating the CREXDATA text analytics and social media monitoring component is available in the public CREXDATA repository:

https://github.com/altairengineering/crexdata-public/tree/main/CREXDATA%20System%20Components/Text%20Analytics

4.9 Explainable AI (XAI) Component (LORE) and its Integration

4.9.1 Lore System and Integration (CNR & RM)

The new modular version of LORE (introduced in CREXDATA Deliverables D5.1 and D5.2) aligns well with the operator-based architecture of the CREXDATA platform. To avoid a proliferation of operators, we encapsulated LORE as a single operator with configurable parameters, delegating the selection of specific modules (e.g., neighborhood generator) to operator parameters. In this way, the library of neighborhood generators developed in WP5 (see D5.2) can be easily integrated and selected within the CREXDATA platform.

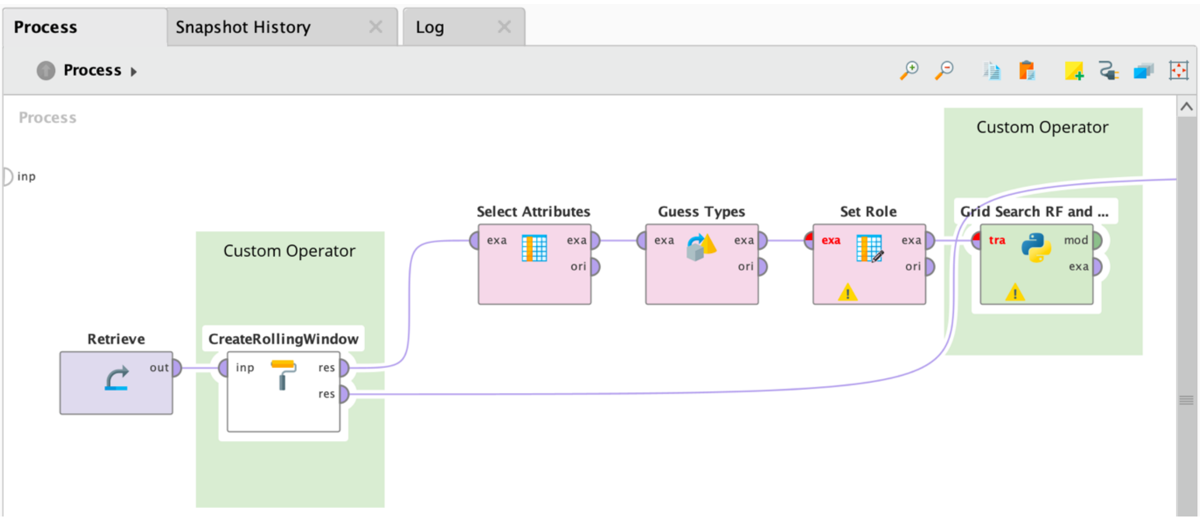


Figure 92: Training a Random Forest and its LORE instance in the CREXDATA platform. Two custom operators are introduced: one for preparing the data through a rolling window aggregation, and one for jointly training a Random Forest and its LORE explainer.

Figure 92 shows an example of how LORE can be integrated into a CREXDATA workflow. In particular, the pipeline addresses the prediction of level of exhaustion of firefighters during their operations. The workflow starts with the ingestion of data from sensors worn by firefighters during their operations. The data are delivered through a data stream operator based on Kafka technology. The ingested data are then pre-processed to extract relevant features and aggregate them over an appropriate time window, whose parameters are configurable through parameter settings of the operator. The pre-processed data is then fed into a black-box model operator. We explored several types of models for learning, ranging from transparent models such as decision trees to more complex models such as random forests and neural networks. We found that random forests provide the best trade-off between accuracy and efficiency. For compatibility constraints from the RapidMiner platform, we used a custom implementation of random forests in Python with a custom operator. The predictions of the black-box model are then explained through the LORE operator, which is configured to use the genetic neighborhood generator. The explanations produced by LORE are then simplified and serialized for streaming over a Kafka topic, from which they can be consumed by downstream applications, such as the dashboard for monitoring firefighter exhaustion levels.

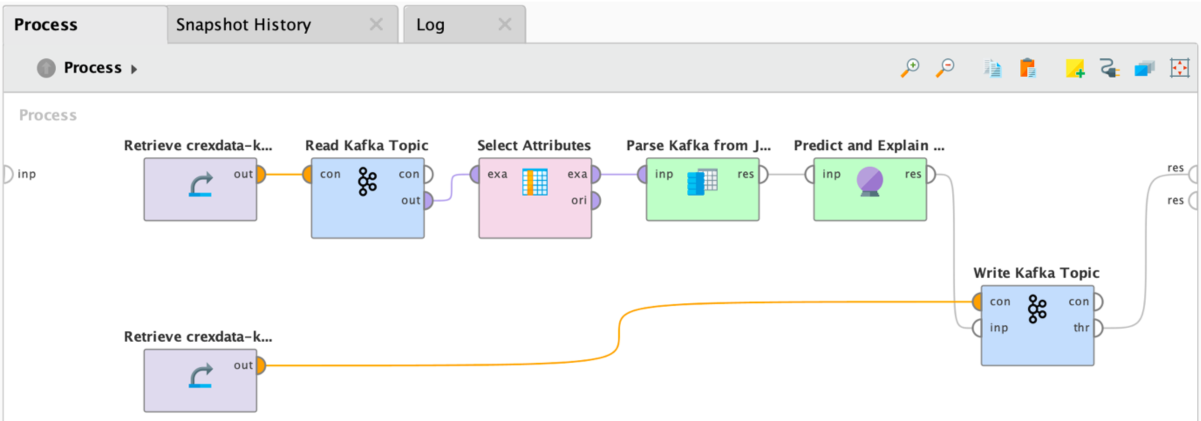


Figure 93: Inference workflow with LORE explanations in the CREXDATA platform. The trained Random Forest and its LORE explainer are loaded from local repository and used to predict and explain new incoming data from a Kafka stream.

Figure 93 shows how the trained random forest and its LORE explainer can be used for inference on new incoming data from the Kafka stream. The trained model and explainer are loaded from a model registry and used to predict and explain new incoming data in real time. We developed an optimized version of LORE for this inference, in order to reduce the latency of explanation generation. The explanations are then serialized and formatted into a string-based representation for streaming over Kafka to downstream applications, such as the dashboard for monitoring firefighter exhaustion levels.

For the XAI use case we have created a RapidMiner Extension (XAI Operators) where the following RapidMiner Custom Operators are placed:

Parse Kafka from JSON to DataFrame: This operator consumes messages from a Kafka topic, parses the JSON payload, and converts it into a RapidMiner DataFrame.

Predict and Explain: This operator takes the DataFrame data and runs it through a machine learning model to generate predictions, then applies LORE explainer to explain why those predictions were made. Instead of just getting a prediction output, you also get interpretable explanations of the model's decision-making process.

Creating Rolling Window: This operator creates rolling statistical features (mean and std dev) over a n-step window (parametrized value) for each entity, then splits the enriched data into 75/25 train/test sets while converting exhaustion levels to ordered categories.

Github: https://github.com/altairengineering/crexdata-private/tree/master/CREXDATA%20System%20Components/Explainable%20AI%20(XAI)/LORE%20Integration%20(RM%20Custom%20Extension%20%26%20Operators)

4.10 Edge-to-Cloud Extension (TUC)

This section describes the *EdgeToCloud* extension developed within the CREXDATA project by the Technical University of Crete (TUC) team members, which includes at its core our Optimized Distributed Analytics-as-a-Service component and has been integrated into the CREXDATA architecture, and how it is exposed to end users through Altair RapidMiner AI Studio. The extension realises the core vision of Task 3.2, enabling data scientists to design logical workflows in a high-level, device/platform-agnostic way, and then automatically deploying them as optimised, distributed streaming analytics pipelines over heterogeneous resources spanning from the cloud to the edge. The integration work reported here connects the optimisation algorithms developed by TUC in Task 4.4, the Federated Machine Learning operators developed by TUC in Task 4.3 and the Augmented Reality operator developed by TUC in Task 5.4 with the workflow design and execution facilities of RapidMiner AI Studio, as well as with the underlying execution engines supported in CREXDATA, namely Apache Flink in the cloud and RapidMiner Real-Time Scoring Agents (RTSA) at the edge.

From an architectural point of view, the extension sits between the logical modelling layer and the physical execution layer. On the modelling side, the *EdgeToCloud Optimization* operator of the extension encapsulates a logical RapidMiner subprocess that describes a dataflow in terms of operators and their dependencies, without binding these operators to concrete execution engines. On the execution side, the optimisation service translates this logical description into a physical deployment plan that assigns each operator either to Apache Flink or to RTSA agents. Then, it materialises this plan as two composite operators: a *StreamingNest* for the cloud-side and the new *EdgeProcessing Nest* for the RTSA-based edge workflow. The subsequent execution relies on Apache Kafka as the communication backbone.

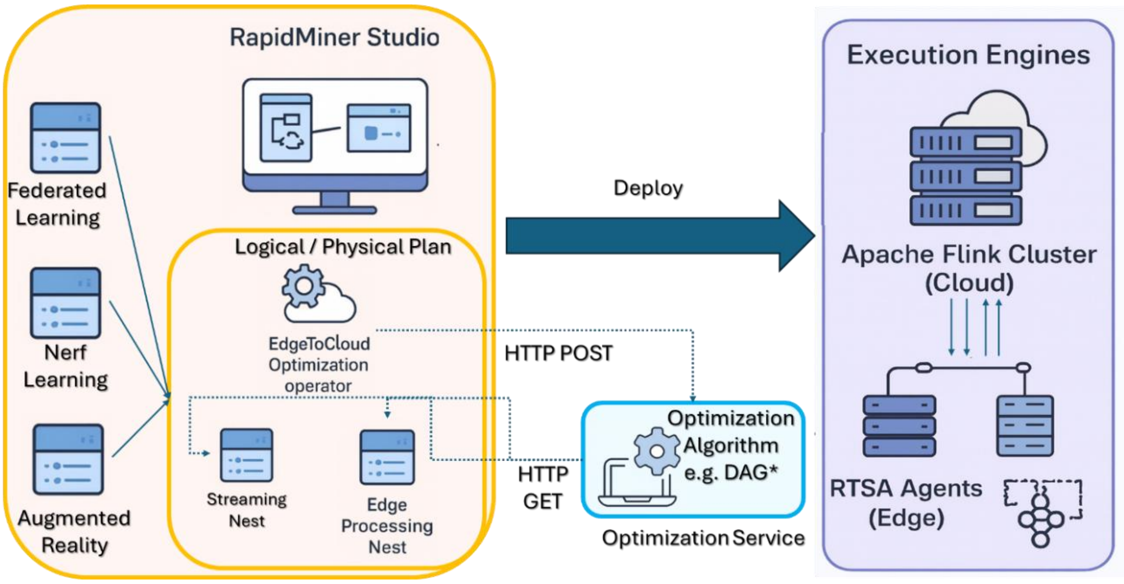


Figure 94: Architectural Overview of the EdgeToCloud Extension for the CREXDATA platform supporting visual workflow design in RapidMiner AI Studio and automated workflow translation and optimization for both, workflow execution in the cloud on Apache Flink clusters and workflow execution on the edge on RapidMiner Real-Time Scoring Agents (RTSA).

The integration effort has therefore focused on:

- Developing, integrating and deploying the necessary new operators that specify processing at edge devices (*EdgeProcessing Nest* operator), optimization over fog/edge/cloud settings (*EdgeToCloud Optimization* operator), Federated Learning (*Federated Learning* and *Nerf Learning* operators), and *Augmented Reality* operator;
- Defining a machine-readable representation of logical workflows, constraints and deployment contexts that can be generated by RapidMiner and consumed by the optimization service;
- Implementing the communication protocol over simple POST/GET requests so that RapidMiner can submit optimisation requests and retrieve optimised plans in a decoupled way; and
- Implementing the logic that reconstructs the *StreamingNest* and *EdgeProcessing Nest* operators inside RapidMiner AI Studio based on the optimized plan and then automatically deploys this plan to Flink clusters and edge physical nodes, respectively.

The remainder of this section details the internals of our *EdgeToCloud* extension, namely the workflow execution engines, the *EdgeToCloud Optimization* operator, the optimisation service, the *EdgeProcessing Nest* operator, along with the operators for *Federated Learning* and *Augmented Reality*, as they are exposed to the end users in the integrated prototype. An overview of the overall *EdgeToCloud* Extension architecture is illustrated in Figure 94.

4.10.1 Integration of the Optimized Distributed Analytics-as-a Service Component

4.10.1.1 Workflow Execution Engines

The Optimized Distributed Analytics-as-a-Service component targets a multi-engine execution environment in which different parts of the same workflow can run on different devices and at different layers of the cloud–edge continuum. In the current CREXDATA prototype, the two primary execution engines are (i) Apache Flink, which operates on cloud-based clusters or powerful fog/edge devices (such as Raspberry Pis) and provides high-throughput, fault-tolerant stream processing, and (ii) RapidMiner Real-Time Scoring Agents (RTSA), which operate on edge devices and gateways close to the data sources. Essentially, RTSAs compose lightweight versions of the RapidMiner libraries. Together, these engines enable a flexible split of responsibilities between local, low-latency processing at the edge and global, resource-intensive analytics in the cloud.

RapidMiner AI Studio is used as the unified visual modelling environment on top of these engines. Data scientists design workflows visually by connecting operators on the RapidMiner canvas, without having to select an execution engine for each operator. The resulting workflows are initially interpreted as logical workflows: graphs of operators that describe how data should be transformed, filtered, aggregated and analysed, without binding these operators to any specific runtime. This separation between a logical layer and a physical layer (Flink jobs and RTSA configurations) is the key enabler for optimisation-driven deployment decisions.

Apache Flink plays the role of the main cloud-side execution engine. It is generally responsible for tasks that require a global or cross-edge view of the data, such as multi-source aggregation, complex event recognition and forecasting over large volumes of streaming data, and long-running stateful analytics. The integration leverages Flink's native support for Kafka as a source and sink, its scalable streaming runtime and its fault-tolerant state management. In the integrated prototype, the TUC optimisation component generates a StreamingNest operator that encapsulates a Flink job implementing the cloud-side part of the optimised workflow.

On the other side of the continuum, RTSA agents act as the edge-side execution engine. RTSA is a RapidMiner technology designed to run lightweight scoring and data transformation pipelines close to the source of the data, for instance on gateways, industrial PCs or other edge devices. In CREXDATA, RTSA agents are used to perform local filtering, pre-aggregation, light feature extraction and simple rule-based or model-based scoring. These capabilities make RTSA particularly suitable for scenarios where bandwidth is limited or where low latency is critical, because they allow computationally inexpensive operations to be executed near the sensors while only relevant, compact results are forwarded to the cloud.

Apache Kafka is used as the communication backbone linking Apache Flink and RTSA agents. The *EdgeToCloud Optimization* operator serialises logical workflows and publishes optimisation requests to the dedicated optimization service. The optimisation service receives these requests, generates optimised deployment plans and publishes the corresponding responses. At run-time, Kafka connects the edge-side RTSA workflows among them and/or with the cloud-side Flink jobs. In this way, the execution engines are loosely coupled: they interact through Kafka topics rather than direct point-to-point connections, which simplifies deployment, scaling and fault isolation.

Overall, the Workflow Execution Engines exposed in RapidMiner AI Studio thus represent an integrated view of a heterogeneous execution environment. Users see a single modelling surface and a small number of high-level operators (*EdgeToCloud Optimization*, *StreamingNest*, *EdgeProcessing Nest*), while under the hood the system orchestrates Apache Flink and RTSA agents across the cloud–edge continuum using Kafka as the glue between them. The following subsections describe in more detail how the *EdgeToCloud Optimization* operator, the optimisation service and the *Streaming Nest* as well as the *Edge Processing Nest* operators realise this mapping from logical RapidMiner workflows to physical, engine-specific execution plans.

Finally, we emphasize that RTSA agents are not initially designed as streaming operators. For integrating stream processing across the continuum, we modify RTSA execution and we have enhanced the functionality provided by RTSAs. In particular, a RTSA processes all available Kafka messages, uses a configurable, short grace (sleep) period and resumes reading Kafka messages that may have arrived in the meantime.

4.10.1.2 The EdgeToCloud Optimization Operator

The *EdgeToCloud Optimization* operator is the main entry point through which users access the Optimized Distributed Analytics-as-a-Service functionality inside RapidMiner AI Studio. Conceptually, this operator encapsulates a logical workflow together with its optimisation context and exposes it as a single, configurable block on the RapidMiner canvas. From the user's perspective, they drag-and-drop the *EdgeToCloud* operator into a process, double-click it to open the nested subprocess where they design their logical workflow, and then configure a set of parameters that control how this logical workflow should be optimized and deployed across edge and cloud resources.

Figure 95 illustrates the EdgeToCloud Optimization operator as it appears in the RapidMiner Studio process view. The operator can be treated like any other composite operator: it has input and output ports, supports nested subprocesses, and can be combined with other operators in the top-level process. It connects to:

- the Kafka backbone for cloud to edge communication
- the optimization service, explained shortly after
- one or more Apache Flink connectors depending on the number of computer cluster available at the cloud side
- RTSA connectors of the available devices
- The RapidMiner AI hub which serves as the central registry for the IoT devices available in the network

Inside the EdgeToCloud Optimization operator, the user designs a logical workflow that includes the steps they wish to execute, such as data ingestion from Kafka, preprocessing, feature extraction, complex event recognition and forecasting or other RM operators. At this stage, the user does not choose whether a given operator will run on Apache Flink or on RTSA; instead, they focus purely on the logical dataflow and business logic.

Figure 96 shows an example of such a logical workflow designed within the *EdgeToCloud Optimization* operator. The operators in this subprocess are interpreted as logical operators, and the connections between them form a logical graph. The *EdgeToCloud* operator is responsible for extracting this graph, together with the associated operator parameters, and building an engine-agnostic JSON representation that can be sent to the optimisation service. In addition, the operator enriches the logical graph with metadata capturing constraints and preferences, such as the optimization algorithm choice (among those developed in Task 4.4), placement constraints indicating that certain operators must remain at the edge or in the cloud.

When the process containing the *EdgeToCloud Optimization* operator is executed, the operator serialises the logical workflow and its metadata into a structured message (JSON document) and creates a POST request. This message constitutes an optimisation request. The operator also registers callbacks where optimisation responses will be received via GET requests. While the optimisation is running, RapidMiner AI Studio can remain responsive, and when the optimised plan becomes available, the operator can update the process accordingly without requiring manual intervention from the user.

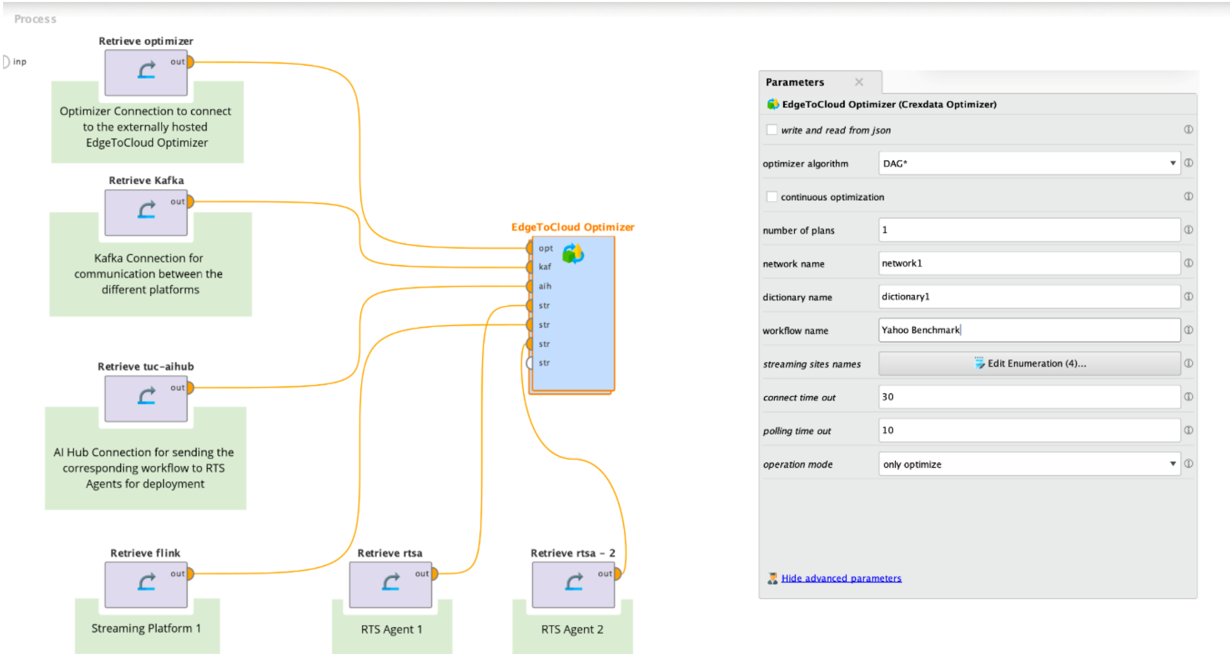


Figure 95: EdgeToCloud Optimization Operator along with connections to the Kafka communication backbone, the RapidMiner AI Hub (RTSA Registry), available RTSAs and an Apache Flink cluster.

The parameter panel of the *EdgeToCloud Optimization* operator exposes configuration options that control both the content of the optimization request and the way the resulting plan is integrated back into the RapidMiner process. An important parameter is the optimization mode. Users can select among:

- *Only Optimize* mode: In which case the optimization service will provide the suggested physical workflow for execution, but the workflow will not be directly deployed. Instead, the user can first inspect and possibly modify optimizer suggestions.
- *Only Deploy* mode: A saved physical workflow or the devised workflow that was returned upon an “Only Optimize” execution, can be deployed without issuing a request to the optimization service.
- *Optimize and Deploy* mode: In which case the optimization service will provide the suggested physical workflow for execution and the corresponding parts of the workflow will be shipped for execution directly to the involved devices.

In summary, the *EdgeToCloud Optimization* operator serves as a bridge between user-authored logical workflows and the optimisation service. It hides the complexity of serialisation, communication and physical plan construction, allowing CREXDATA users to focus on expressing what the workflow should do while delegating to the optimiser the decision of where and how it should be executed along the cloud to edge continuum. Figure 95 and Figure 96 together document both the user experience of the operator and the internal logical workflow that forms the input to the optimization process.

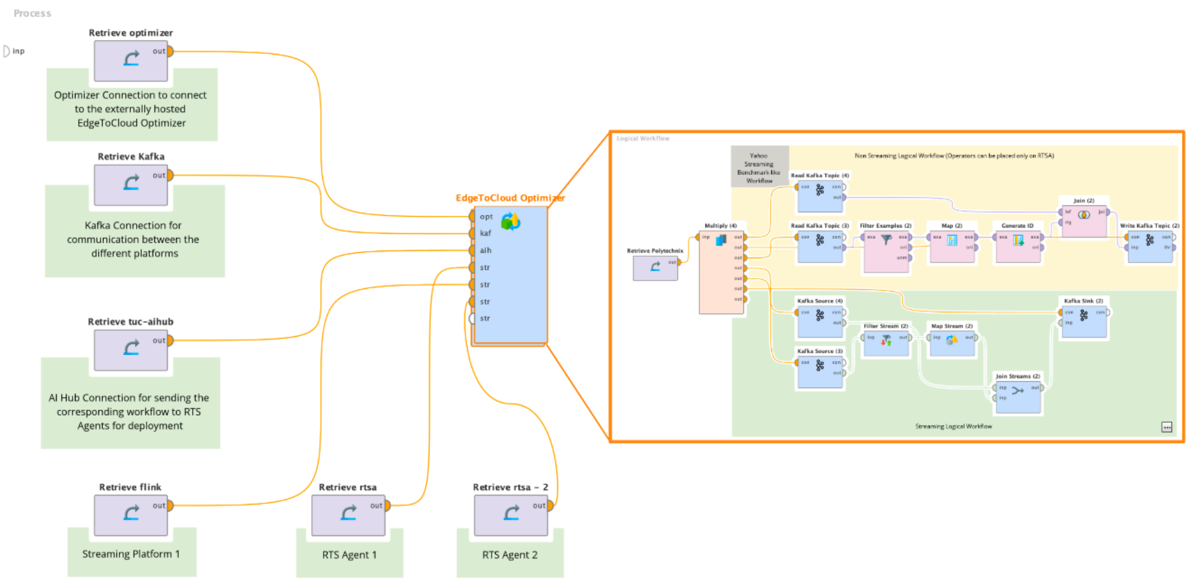


Figure 96: A Logical Workflow designed in the EdgeToCloud Optimization Operator.

4.10.1.3 The Optimization Service

The optimization service is a separate, back-end component that receives logical workflows from RapidMiner AI Studio, searches for efficient deployment plans using the optimization algorithms of Task 4.4 and returns concrete assignments of operators to execution engines and physical resources. It forms the algorithmic core of the Optimized Distributed Analytics-as-a-Service component developed by TUC. Communication between RapidMiner and the optimisation service is fully decoupled: RapidMiner publishes optimisation requests via POST messages, and the service processes these messages asynchronously, producing optimisation responses that are periodically polled via GET requests.

As shown in Figure 95, the user can set, besides the execution mode discussed in Section 4.10.1.3, the following parameters:

- *Polling time out:* RapidMiner submits optimization requests to the optimisation service via HTTP POST requests. The service processes these requests asynchronously and exposes the results through an endpoint that RapidMiner periodically polls using HTTP GET requests to retrieve the corresponding optimization responses. This parameter specified the frequency of these GET requests.
- *Connection time out:* This is a time tolerance value expressing a maximum number of seconds the user is willing to wait for the devised physical plan. After this interval, the optimization algorithm should provide the best physical plan found so far, even if the optimization algorithm has not yet fully concluded.
- *Dictionary Name:* a JSON file deriving statistics about operator execution per site.
- *Network Name:* a JSON file deriving information about execution engine availability per site (Flink, RTSA or both).
- *Optimizer algorithm:* the choice of the optimization algorithm as described in Deliverable D4.2. This is optional, a default optimization algorithm is already pre-selected.

Figure 97 illustrates the connection object which should be defined using a host URL and a port along with login credentials (within RapidMiner AI Studio).

Upon receiving an optimization request from the *EdgeToCloud Optimization* operator, the service parses the logical workflow description and reconstructs an internal representation of the operator graph. It also retrieves information about the available infrastructure, such as the set of candidate edge nodes and RTSA agents, the characteristics of the cloud-side Flink cluster, and the topology and capacity of the network links between them. Using the dictionary and network files, the approach associates each logical operator with its resource demands/statistics and each network device with its capabilities.

The objective of the optimization is to find deployment plans providing good trade-offs with respect to several metrics, such as latency, throughput, network usage and resource utilisation (also depending on the optimization algorithm). Internally, the service explores the space of possible assignments using the algorithms developed in Task 4.4. For each candidate plan, a cost model estimates the expected performance and resource usage, taking into account the properties of Apache Flink and RTSA as well as the communication latency induced by Kafka topics between edge and cloud.

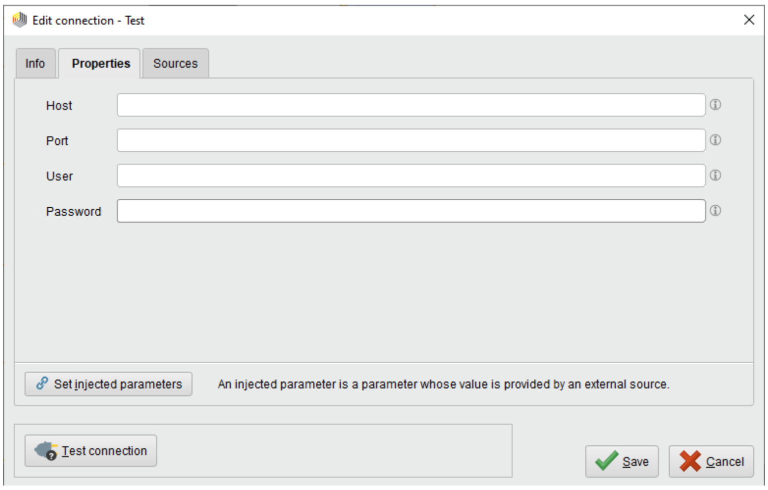


Figure 97: Parameterization of the Connection to the Optimization Service (Connection Object in RapidMiner AI Studio).

Once a plan is returned, the optimization service encodes it in a JSON format that can be consumed directly by RapidMiner AI Studio. Concretely, the plan is translated into: (i) a specification of a *StreamingNest* operator, which defines the sub-workflow that should run on Apache Flink together with its Kafka inputs and outputs and its deployment parameters; and (ii) a specification of each *EdgeProcessing Nest* operator, which defines the sub-workflow that should run on RTSA agents, including the mapping of operators to specific edge nodes, Kafka topics for local ingestion and for forwarding results to the cloud, and all operator-specific configuration values are configured automatically. The service then publishes this plan as a response message back to the *EdgeToCloud Optimization* operator.

When RapidMiner AI Studio receives the optimization response, the *EdgeToCloud* operator reconstructs the two or more *Nest* operators and inserts them into the process. Figure 98 illustrates the resulting optimised workflow across the cloud to edge continuum: the edge-side RTSA workflows, represented by the *EdgeProcessing Nest*, process local data, while the cloud-side Flink job, represented by the *Streaming Nest*, performs global analytics, with Kafka connecting the two. In this way, the optimisation service closes the loop between logical design and physical execution, ensuring that deployment decisions are not hard-coded but are, instead, the outcome of automated optimisation process.

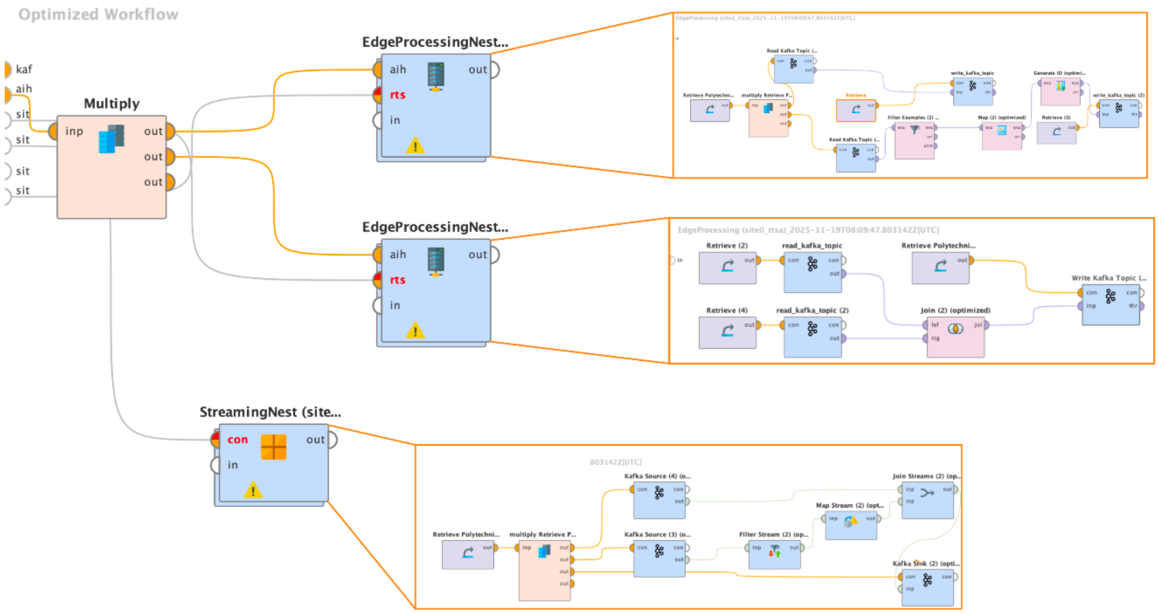


Figure 98: Optimized (Physical) Workflow Across the Cloud to Edge Continuum.

4.10.1.4 The EdgeProcessing Nest Operator

The *EdgeProcessing Nest* operator that we developed, together with the *StreamingNest* operator, is the RapidMiner-side manifestation of the optimized edge-side workflow produced by the optimization service. The *StreamingNest* operator encapsulates all the information necessary to deploy and execute a set of logical operators on Flink clusters, either at the cloud side or on more powerful fog/edge devices, such as Raspberry Pis. The *EdgeProcessing Nest* operator encapsulates all the information necessary to deploy and execute a set of logical operators on RTSA agents running at the edge, and it provides a convenient abstraction for users to inspect and manage the resulting configuration. From the perspective of the top-level RapidMiner process, the *EdgeProcessing Nest* operator appears as a single operator with input and output ports.

The *EdgeProcessing Nest* operator can, in principle, be used independently of the CREXDATA optimization process and the logical to physical workflow conversion. This means, that should a user want to directly define the physical execution of a workflow on an RTSA, they can use the *EdgeProcessing Nest* operator and directly design the physical workflow inside it. This is illustrated in Figure 99 and Figure 100.

In Figure 99, we can observe the parameters of an *EdgeProcessing Nest* operator. The most notable parameter here is the "*sleep time*" parameter. Because RTSAs were not primarily destined to operate on streaming setups, the integrated extension we provide allows RTSAs to process all the data that are currently available in an input Kafka topic, then use a sleeping interval waiting for more data and resume execution.

As shown in both Figure 99 and Figure 100, the input ports of the *Edge Processing Nest* operator are connected to the RapidMiner AI Hub and to a RTSA connection destined to undertake the execution of the workflow.

Internally, the *Edge Processing Nest* contains a nested workflow that illustrates how the logical operators assigned to the edge have been mapped to concrete RTSA components. Figure 100 shows a fragment of such a physical workflow inside an Edge Processing Nest operator. The figure highlights how RTSA components corresponding to filtering, stream transformation (map) and join operations are arranged in a pipeline that consumes streams from Kafka source topics and produces output to Kafka sink topics.

Operationally, the *EdgeProcessing Nest* works in tandem with the *StreamingNest* operator within the context of the CREXDATA optimization service, as previously illustrated in Figure 98. In the physical workflow produced by the optimization service in the figure, the *EdgeProcessing Nest* ensures that only relevant, compact and possibly already interpreted information is transmitted from the edge to the cloud via Kafka, thus reducing bandwidth usage and improving responsiveness. The *StreamingNest*, which encapsulates the Flink job, can consume these edge-level outputs from Kafka, performs local analytics, and writes final results to downstream topics and sinks. This division of labour between the two nests is the concrete realisation of the edge-to-cloud execution model envisioned in CREXDATA. Each *StreamingNest* or *EdgeProcessing Nest* component corresponds to a part of the local workflow, but it has been specialised for execution on the cloud or RTSA side with the appropriate parameterisation and input/output bindings (shown on the left of Figure 98).

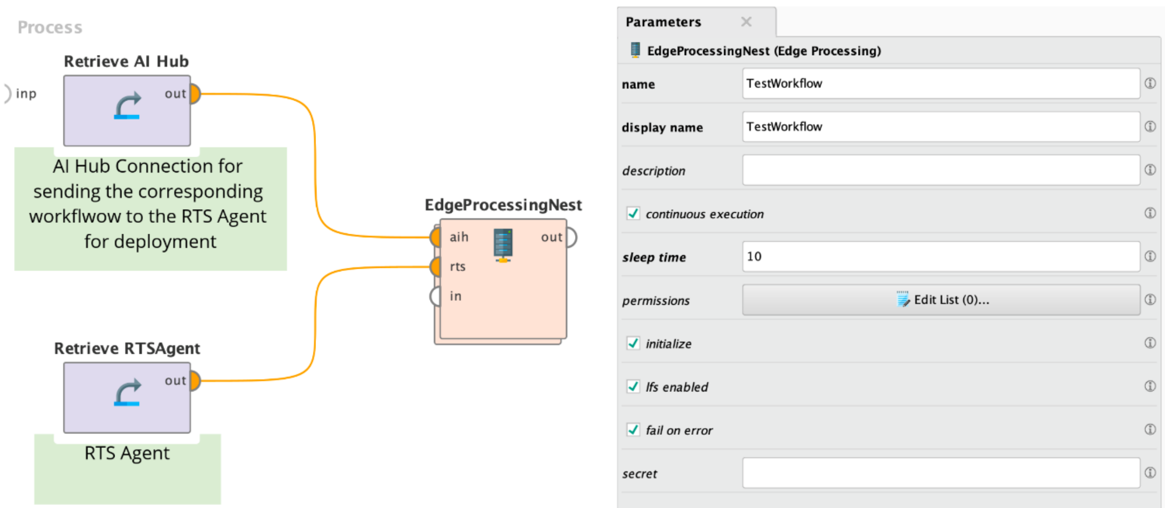


Figure 99: Edge Processing Nest Operator Parameters.

By generating the *StreamingNest* and *EdgeProcessing Nest* operators automatically from the optimization output, the integrated component ensures consistency between the deployment decisions taken by the optimiser and the actual deployment configuration of the cloud side and the RTSA agents. Users do not have to edit RTSA configuration files manually or manage low-level deployment scripts; instead, they can reason about the edge-side workflow at the same conceptual level as the cloud-side one, entirely within RapidMiner AI Studio. This significantly lowers the barrier to adopting edge-enhanced streaming analytics, while preserving the flexibility to adapt deployment decisions as constraints, workloads or infrastructure evolve over time.

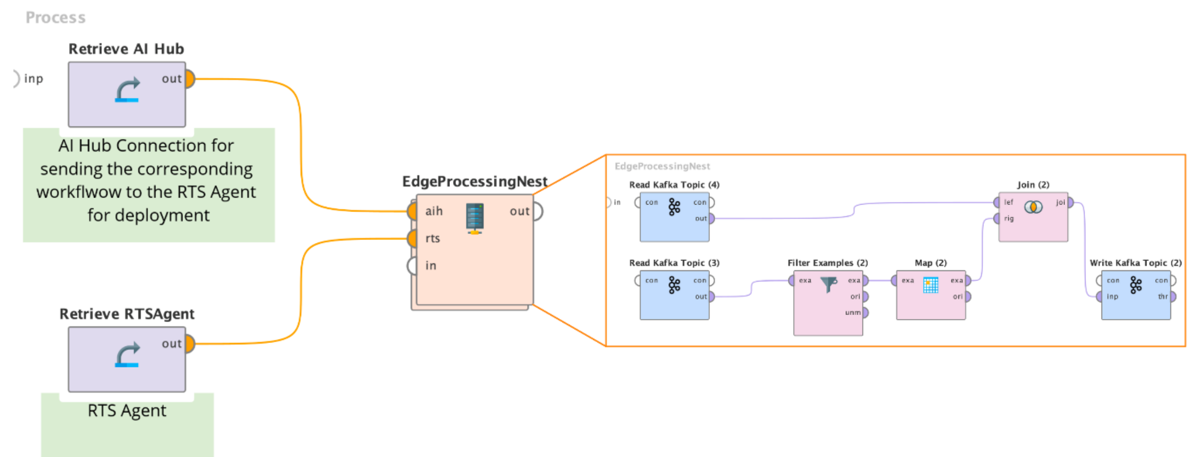


Figure 100: Part of a Physical Workflow in an EdgeProcessing Nest Operator.

4.10.1.5 Benefits of the Integration Approach

Our integration approach does not only provide concrete design and configuration facilities for the operators involved in the CREXDATA workflows. It also conveys several conceptual benefits:

- Abstraction and usability: Users work at the level of logical workflows and high-level constraints. They are not forced to reason about low-level details such as which operator runs on which device or how to manually encode a Flink job and RTSA configuration. The *EdgeToCloud Optimization* operator and the nests serve as abstraction layers between logical design and physical deployment.
- Systematic optimization: The use of an external optimization service means that deployment decisions are not ad hoc. Instead, they are the result of a principled search over alternative deployments, taking into account optimization objectives and constraints as formally described in Deliverable D4.2.
- Separation of logical and physical concerns: The logical workflow focuses on what the workflow should do. The physical workflow in the corresponding nest operators focuses on how and where this should be executed. This separation facilitates maintainability and reuse of workflows.
- Edge to cloud synergy: By splitting the workflow into a *StreamingNest* (cloud/Flink) and an *EdgeProcessing Nest* (edge/RTSA), the system can exploit the strengths of each environment with the aid of the optimization service. Low-latency local processing at the edge and high-throughput global analytics in the cloud.
- Operational flexibility: The use of Kafka as the communication backbone, visible in the operator parameter panels, enables reconfiguration and scaling without modifying the

logical workflows. New edge nodes or cloud clusters can be added by adjusting Kafka topics and deployment settings, while the logical design remains unchanged.

4.10.2 Integration of the Federated Learning Operator

These two operators generate a JSON configuration record based on user-defined parameters. Both operators have to be connected directly to a *Write Kafka Topic* operator, which sends the generated message to a specific Kafka topic. Each operator is associated with a different server. For each operator, a separate server listens continuously to its corresponding Kafka topic. When the server receives the JSON record, it immediately reads the payload and triggers the appropriate process according to the specified parameters. Both operators implement and integrate the techniques developed in the scope of Task 4.3.

4.10.2.1 Federated Learning Operator

Figure 101 shows a RapidMiner process in which a Federated Learning (FL) operator configures and launches a FL job implemented in TensorFlow, and then publishes the learning specification to Kafka for execution. The operator’s output is an “artifact” describing the run, the *Write Kafka Topic* node forwards this artifact to a Kafka topic (using the connection), so an external runtime can pick it up and efficiently execute the federated training as prescribed in Task 4.3.

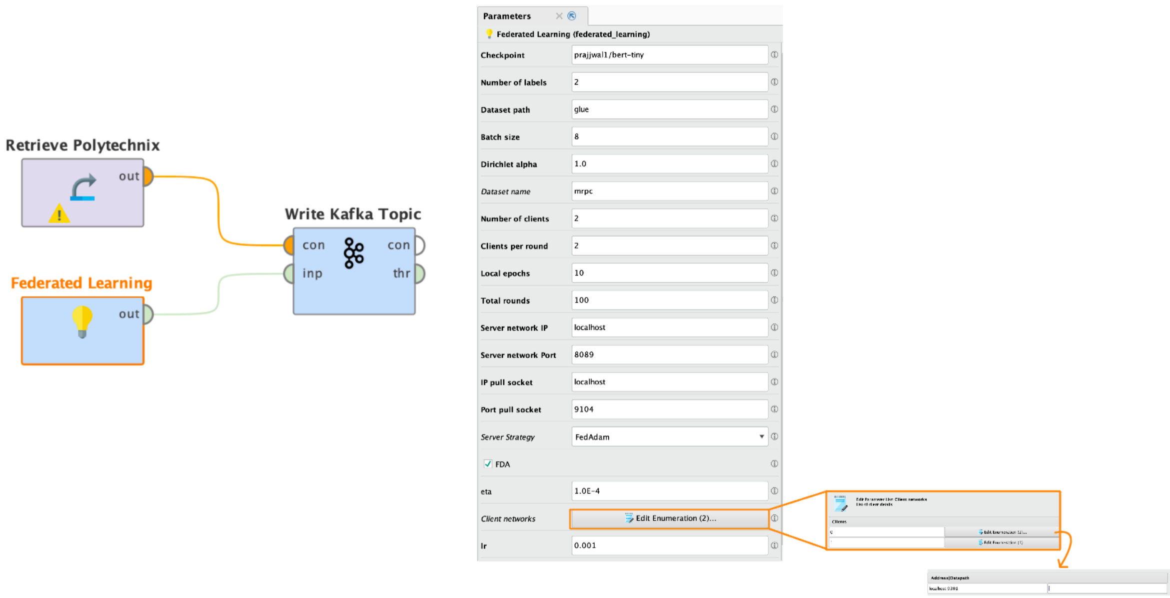


Figure 101: RapidMiner process with the Federated Learning Operator and its parameters.

The parameter panel on the right of Figure 101 corresponds to the standard FL control variables of the well-known FedAvg/FedSGD family of federated optimization algorithms. In particular, it exposes B (batch size), C (clients per round / fraction of participating clients), E (local epochs per round), and lr (learning rate), which together determine how much local computation each client performs before every aggregation and therefore trade off convergence speed vs. communication frequency. The panel also lets the user set the total number of rounds, choose FedAdam, which is the federated optimization variant of the adaptive Adam optimizer that is widely used in Machine Learning tasks), and optionally enable ￼FDA￼-style synchronization controls (developed within Task 4.3), while providing the necessary server endpoints (IP/ports) for coordinating the FL session. Overall, the figure depicts an operator that acts as a single interface for TensorFlow-based federated training, with Kafka used purely as the workflow's dispatch channel. The learned model can afterwards be used downstream any workflow for inference purposes.

4.10.2.2 Federated Learning Tuning Operator

The *Federated Learning Tuner* operator is a parameter-driven launcher for two backends: a NS-3 (https://www.nsnam.org/) network-simulation backend (ns3 mode) and a Flower-based (https://flower.ai/) federated-learning backend (flower mode). Its user interface (UI) changes dynamically: in **ns3/simulation** it configures and runs a learning round with a chosen number of clients, topology and wireless settings, and model-traffic size, producing per-client communication metrics. In **ns3/query** it requests retrieval of already produced NS-3 results instead of rerunning the simulator. In **flower/training** it initiates a full Federated Learning run and, at the start of every FL round, a Flower server synchronously triggers an NS-3 simulation for the sampled clients, receives downlink time, uplink time, throughput, router distance, and dropout indicators, and uses dropouts to remove failed clients before local training and aggregation proceed. In **flower/inference**, it loads a specified trained global model and performs evaluation/prediction under the same networking assumptions (i.e., with the same simulation-aware client participation logic).

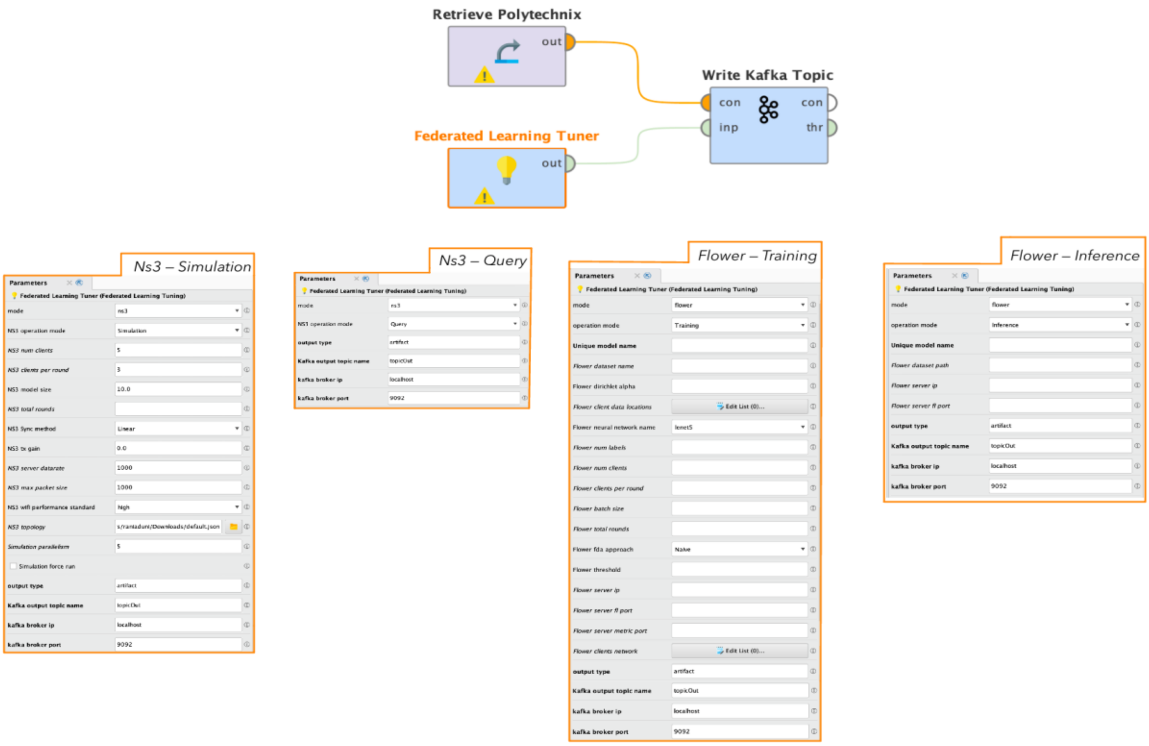


Figure 102: RapidMiner process with the Federated Learning Tuner Operator and its parameters.

A key implementation detail is that the coupling between Flower and NS-3 is done via socket-based Inter-Process Communication (IPC) StartSimulationRound/Response/Exit commands and is blocking/synchronous, precisely because the network results can change which clients are allowed to participate in the round. The system also includes an optional Google Remote Procedure Call (gRPC) metric service that collects per-batch L_2 norms from clients and can enforce a round-termination condition (threshold-based early stopping) during training. Finally, as shown in Figure 102 RapidMiner process, the operator's configured experiment is emitted as an output artifact that is routed to a *Write Kafka Topic* for external dispatch. This Kafka topic is part of the workflow-orchestration layer visible in the figure, while the core FL-to-NS-3 interaction remains socket-based.

In ns3/simulation, the operator runs only an NS-3 communication experiment. The application configures the wireless/topology and the FL-related traffic profile (e.g., number of clients, clients per round, model upload/download size, rounds), and NS-3 produces communication outcomes: per-client downlink/uplink times, throughput, router distance, and which clients would drop due to bad connectivity. No neural training happens here; this mode is for characterizing the network and estimating communication cost/feasibility under given settings.

In flower/training, the operator runs a full federated learning session under Flower, using NS-3 results as part of the control loop. At each FL round, clients are sampled, NS-3 is invoked to evaluate whether they can actually communicate, dropouts are filtered out, and then the surviving clients perform local training and send updates for aggregation. So, this mode outputs a trained global model and learning metrics, while treating network simulation as a round-by-round gatekeeper/estimator that shapes which clients participate and how expensive each round is.

4.10.3 Integration of the Nerf Learning Operator

The NeRF operator materializes an adaptive, city-scale neural radiance field pipeline as a single configurable component. It ingests aerial imagery in batches and produces a continuously updatable 3D representation that can be rendered in real time. To accomplish this, it follows an offline and an online phase. The offline phase partitions a static dataset into spatial regions, termed experts, estimates camera poses (using COLMAP) and trains region-specific NeRF initializations so that each region starts from a geometric appearance prior. The online phase operates when new views arrive from the field. It generates rays subsequently routing them to the relevant experts. Then, it blends expert outputs and performs a number of inner-loop updates to refine only the affected parts of the city model without retraining from scratch.

The operator's parameters correspond to spatial decomposition, adaptation and expert configuration settings. Spatial decomposition settings involve number/size of regions, boundary margins and hard vs soft routing/blending weights).The adaptation controls manage what happens when a new batch arrives. Upon a new batch is received, the operator:

- generates rays and samples points along them
- routes those rays to the relevant experts
- computes the reconstruction loss
- runs gradient descent updates.

A parameter K configures how many times these steps will be repeated.

Instant Neural Graphics Primitives are used for expert configuration for efficient optimization and compact models. It exposes a rendering modality. A viewer queries the container model and supports both "view-only" navigation and "runtime adaptation" where rendering continues while the model updates live as new imagery streams in.

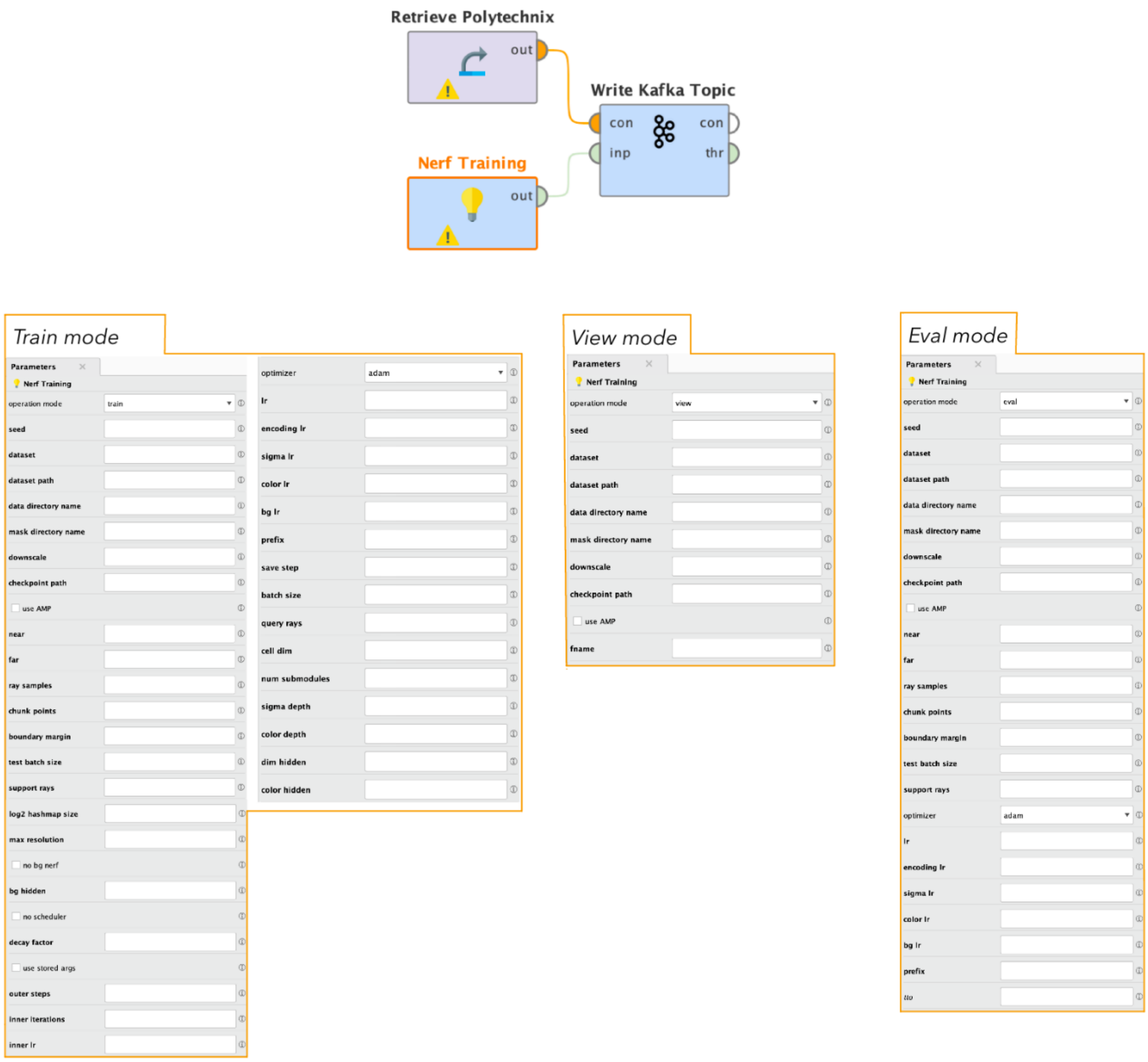


Figure 103: Nerf Training Operator

4.10.4 Integration of the Augmented Reality – Mental Fatigue Operator

The Mental Fatigue operator processes firefighter eye tracking features and generates a class prediction using a pretrained k-Nearest Neighbors (KNN) classifier. The operator expects as input a set of records containing the following nine features:

1. *Mean pupil diameter:* The average size of the pupil over the recording; often increases with cognitive load or stress.
2. *Fixation count:* How many times the eyes stop and focus on a point; reflects how often attention is anchored.
3. *Saccade count:* Number of rapid eye jumps between fixations; indicates how frequently gaze shifts.
4. *Blink count:* Total blinks during the period; can rise with fatigue or drop during intense focus.
5. *Mean fixation duration:* Average time of each saccade; captures how long gaze transitions take.
6. *Mean saccade duration:* The maximum speed reached during saccades; typically decreases with fatigue.
7. *Peak saccade velocity:* The maximum speed reached during saccades; typically decreases with fatigue.
8. *Mean saccade velocity:* Average speed of saccades; reflects overall vigor of gaze shifts.
9. *Mean saccade amplitude:* Average angular distance covered per saccade; smaller amplitudes can signal reduced exploration or fatigue.

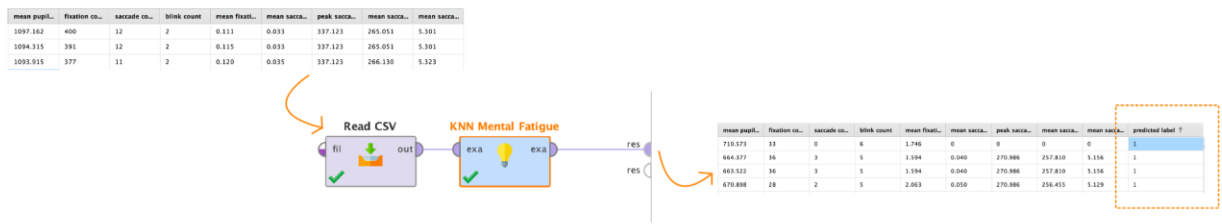


Figure 104: RapidMiner process with the Augmented Reality - Mental Fatigue Operator and its input and output data table formats.

The operator performs a feature normalization for each record before passing them into the pretrained KNN classifier. The output is the original records with an additional column containing the predicted class code.

4.10.5 Source Code of the Edge-to-Cloud and Optimization Components

The source code of the Edge-to-Cloud and Optimization components of the CREXDATA system has been published and is available from the repository:

https://github.com/altairengineering/crexdata-public/tree/main/CREXDATA%20System%20Components/Cloud-to-Edge%20Optimizer%20(TUC).

4.11 EmCase Optimizer Component and Integration (NCSR)

For the purposes of the Emergency Use-case, we developed and integrated an optimization component that implements a Genetic Algorithm. The selection of this kind of optimization algorithm was decided after experimenting with different alternatives (Simulating Annealing and Tree-structured Parzen Estimator), as thoroughly described in D4.2.

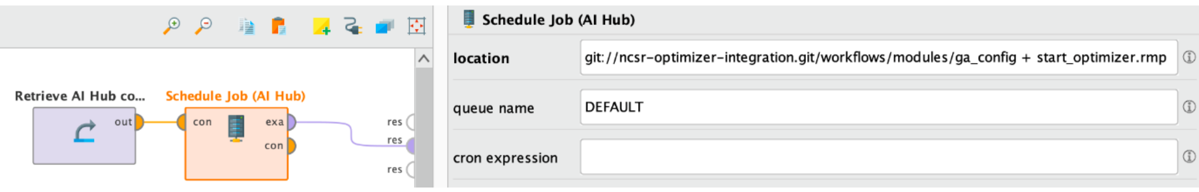
According to the use-case scenario, the end user sets number of water barriers on the ARGOS map, and then the optimizer aims to determine the optimal barrier height in order for an area of interest (also dictated by the end-user) to remain protected from flooding. This can be, e.g., a hospital, an airport, or other key infrastructure that is either crowded at the time of the flooding event, or hosts critical services that need to remain uninterrupted.

The GA optimizer can be configured to search suitable heights for different numbers of barriers and to keep the area of interest protected at different levels. These configurations are set by the end-user via the ARGOS UI and are communicated to the optimizer component via dedicated KAFKA topics.

After the initial configurations are set, then the optimizer component invokes the FloodWaive simulator to test specific barrier heights and receive the results, i.e. the maximum water level height at the point of interest during the simulation, also termed as the fitness score. Each barrier heights combination results to a different fitness score, and the goal of the GA component is to minimize it.

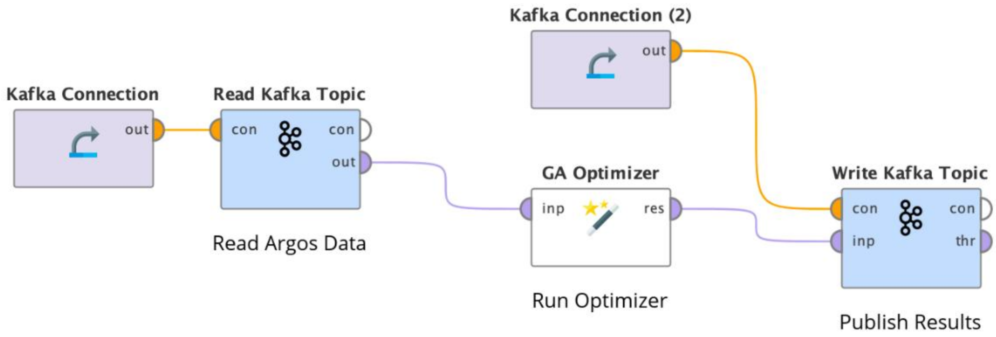
For more information regarding the operation of the GA, please refer to D4.2.

- Trigger Optimizer Execution from RapidMiner AI Hub (see Optimizer Workflow)



This workflow queues your genetic algorithm barrier optimization to run asynchronously on AIHub. It retrieves the ga_config + start_optimizer.rmp workflow from the internal RapidMiner repository and submits it to the DEFAULT queue, allowing the GA optimization to execute on the AIHub cluster as a background job instead of running locally.

- Optimizer Workflow



This main component of this workflow is the GA Optimizer Custom Operator which runs a Genetic Algorithm (GA) optimization with these parameters:

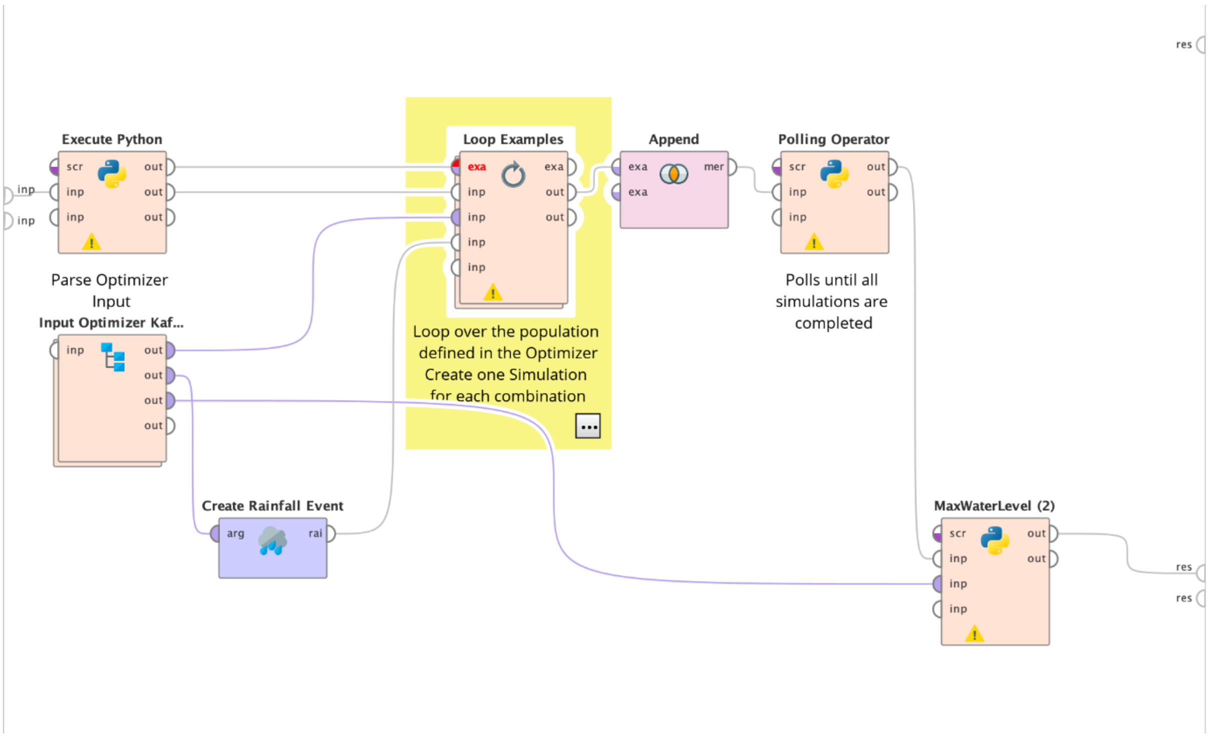
distance_type: l1 – Uses Manhattan distance (sum of absolute differences) to evaluate fitness/similarity

pop_num: 5 – Population size of 5 individuals per generation

termination_crit: genmax – Stops after a maximum number of generations

These parameters are just the default values that can be easily modified in the parameters tab.

- Resulting Fitness Scores from the Simulations



This workflow retrieves and exposes optimized barrier height parameters from your GA optimizer via a REST endpoint. Here's the pipeline:

1. **Execute Python** – Parses GA optimizer output (parameter sets as strings) and extracts all candidate solutions without limiting the results
2. **Input Optimizer Kafka Topic** – Reads from the UPB-CREXDATA-Flooding-Optimizer-Input Kafka topic to fetch simulation context (rainfall events, water level data)
3. **Loop Examples** – Iterates through each GA-generated parameter set and runs simulations
4. **Polling Operator** – Waits for simulation jobs to complete on AIHub
5. **MaxWaterLevel** – Evaluates peak water levels from simulation results
6. **Aggregated data** – Combines all results into a single output (optimizer ID, simulation ID, results list)

The endpoint returns a list of barrier height configurations ranked by performance from your genetic algorithm, ready for deployment or further analysis.

For this use case we have created the following RapidMiner Custom Operators in the RapidMiner extension for the NCSR optimizer (**ncsr-1.0.0-all.jar**):

- GA Optimizer,
- Create Measures,
- Create Rainfall Event,
- Create Simulation

GitHub Repository with the source code the NCSR RapidMiner extension for the NCSR optimizer:

https://github.com/altairengineering/crexdata-private/tree/master/CREXDATA%20System%20Components/Optimizer%20(NCSR)/Optimizer%20Integration

4.12 Augmented Reality (AR) and Integration

The integration of the Augmented Reality (AR) component into the CREXDATA system is described in Section 4.10.4.

4.13 Integration of Components for Weather Emergency Use Case

The overall data processing workflow for the Weather Emergency Use Case (EmCase) and the integration of the CREXDATA system components for this use case are described in more detail in CREXDATA Deliverable D2.3 “Final Use Case Evaluation, Pilots, Demonstrators and Simulation Models and Tools”.

4.14 Integration of Components for Health Emergency Use Case

The overall data processing workflow for the Health Emergency Use Case and the integration of the CREXDATA system components for this use case are described in more detail in CREXDATA Deliverable D2.3 “Final Use Case Evaluation, Pilots, Demonstrators and Simulation Models and Tools”.

4.15 Integration of Components for Maritime Use Case

The overall data processing workflow for the Maritime Use Case and the integration of the CREXDATA system components for this use case are described in more detail in CREXDATA Deliverable D2.3 “Final Use Case Evaluation, Pilots, Demonstrators and Simulation Models and Tools”.

5 Software Stacks

This section summarizes the software tools and software stacks used and developed within the CREXDATA project. More detailed descriptions of how these tools are used within the CREXDATA system are available in Section 2 (system architecture and system components and system integration approach), Section 3 (domain- and use-case-specific components like for example domain-specific simulators), and Section 4 (integration, configuration, user interface, etc.).

5.1 Data Source Systems and Data Collection

The data source systems of the different use cases are described in Sections 2.5 and 3. For the weather emergency use case, for example, the system ARGOS of CREXDATA project partner HYDS integrates the data needed for this use case (weather data, events, sensor data, IoT and edge device data, etc.) and connects to the CREXDATA platform via Kafka, in order to provide the data to the CREXDATA platform and to get predictions from the CREXDATA platform in return.

5.2 Data Stream Connection: Kafka

Kafka is used as middleware for communication and data transfer between the CREXDATA system components. The use of Kafka and web services for the integration of the CREXDATA system is described in Section 2 and 4.

5.3 Data Stream Infusion and Processing: Altair RapidMiner

The Altair RapidMiner data science software platform is used as underlying platform for data integration and fusion, data stream processing, federated data processing, etc., which is extended with CREXDATA-specific data connectors and modules (extensions) to form the core of the CREXDATA system.

Altair RapidMiner AI Studio (with CREXDATA-specific extensions) as is used as the visual data processing workflow editor for code-free visual process workflow design.

Altair RapidMiner AI Hub is used as central server and optionally for further distributed servers or cloud servers.

Altair RapidMiner Real-Time Scoring Agents (RTSA) are used for data processing on the edge and IoT devices.

5.4 Domain-Specific Simulators

Use-case-specific simulators and CREXDATA system components are described in Section 2.5 and in Section 3 and 4.

5.5 Source Code Repository for CREXDATA Open-Source Software

The source code of the open-source components of the CREXDATA system as well as RapidMiner processes (workflows), system configurations files, and documentation are shared and published via the following GitHub repository:

https://github.com/altairengineering/crexdata-public

(previously https://bitbucket.org/crexdata/)

This repository contains software source code, RapidMiner process files (i.e. specifications of the data processing workflows for the various use cases), system configuration files, documentation, and other documents.

The source code of the Edge-to-Cloud and Optimization components of the CREXDATA system has been published and is available from this repository:

https://github.com/altairengineering/crexdata-public/tree/main/CREXDATA%20System%20Components/Cloud-to-Edge%20Optimizer%20(TUC) .

6 CREXDATA Objectives Addressed by the CREXDATA System Architecture

The CREXDATA architecture described in Sections 1 and 2 and the prototypical system components implemented and described in Section 4 address the CREXDATA objective MO1.1 by supporting multi-modal data ingestion and fusion and by developing corresponding algorithms and techniques in the CREXDATA project that are incorporated in as operators into the CREXDATA system, so that they can be easily used in the visual data workflow design in the graphical user interface of the CREXDATA workflow editor. The visual workflow designer based on Altair RapidMiner AI Studio with the extensions developed in the CREXDATA project including the data connectors for the various data sources required by the use cases and the Multi-Modal Data Fusion Toolbox developed by TUC in the CREXDATA project compose a data ingestion and fusion toolbox for bringing together and aligning data of multiple modalities and formats including for example time series data, sensor data, weather data, text data, image data, video data, JSON, geometries, raster, graph, spatiotemporal data, etc., for which the extension mechanism provides a flexibly configurable way to accommodate the various use case scenarios with their respective requirements.

The CREXDATA system components, like the underlying data integration and data science platform Altair RapidMiner and Kafka as a middleware for connecting the system components and for their communication and data exchange ensure the capability for processing extreme-scale data streams in real-time and support federated data processing and federated machine learning not only on computer clusters, servers, and in the cloud, but also on IoT and edge devices. These underlying system components together with the above-mentioned visual workflow designer and its pluggable extensions provide the envisioned outcome: a Pluggable Extreme-scale Data Ingestion and Fusion Toolbox in the CREXDATA IDE (Integrated Development Environment) with cross-library support and the ability to plug-in custom User Defined Functions (UDFs).

The CREXDATA platform integrates use-case-specific systems (e.g. ARGOS by HYDS and simulators like FloodWaive DeepWaive and Gazebo in the weather emergency use case) to integrate use-case-specific data sources (e.g. sensors and edge devices), use case specific simulators (see Sections 2, 3, and 4), as well as use-case-specific functionalities and capabilities via the extension (plugin) mechanism (e.g. for algorithms and techniques developed in the CREXDATA work packages WP3, WP4, and WP5).

7 Related Work, State-of-the-Art, and Contributions of the CREXDATA Platform

7.1 Integrated Development Environment (IDE) with Visual Data Analytics Workflow Designer

Visual data analytics workflow designers that allow users to specify workflows without coding, enable non-coders and non-data-scientists to use machine learning and artificial intelligence and thereby broaden the range of people benefiting from these technologies. The first data analytics and machine learning tool using a visual workflow designer and IDE was *Clementine* in the 1990s, which was later acquired by *SPSS* and renamed *SPSS Modeler*. SPSS in turn was later acquired by IBM and the tool renamed *IBM Watson SPSS Modeler*. Several open-source tools for data analytics and machine learning adopted visual workflow designers and IDEs, first *WEKA*[24] (from the University of Waikato in New Zealand), then *RapidMiner*[25] (initially named *YALE* (Yet Another Machine Learning Environment) and started at the Technical University of Dortmund in Germany in 2001), and then *KNIME*[26] (from the University of Konstanz in Germany). Today, many open source and commercial products for data analytics and machine learning use visual workflow designers and IDEs.

Among these visual workflow designers for data analytics processes, RapidMiner offers the most advanced IDE, seamlessly integrating automated machine learning (AutoML), visual no-code data science, and code-oriented data science with Python, R, Java, SQL, etc., allowing the users to choose, which approach to workflow specification they prefer and allowing to seamlessly merge these approaches and to collaborate in teams. Also, RapidMiner offers the broadest range of operators (functional modules) for data import, data processing, and machine learning as well as for data stream processing and distribution of workflow executions across distributed compute resources, which is also quite relevant for CREXDATA.

Hence, within CREXDATA, we leveraged the visual workflow designer and IDE of *RapidMiner AI Studio* and extended it by extensions for seamlessly integrating data sources, systems, and CREXDATA system components relevant for the CREXDATA use cases. These extensions were described in section 4 and will also be described in the following sections. With these extensions, the CREXDATA platform increases the data connectivity of the visual workflow designer beyond the state-of-the-art of such workflow designers, especially since other data analytics tools do not integrate so many simulators, if any at all. Thanks to the visual workflow designer and the extensions developed in the CREXDATA project, the state-of-the-art in the use case domains is also exceeded, enabling the integration of simulations with multi-modal real-time data, explainability, and visualizations, which was for example not possible yet in the weather emergency use case domain.

The *2025 Gartner® Magic Quadrant™ for Data Science and Machine Learning Platforms* published in May 2025 by the Gartner Analysts Afraz Jaffri, Maryam Hassanlou, Tong Zhang, Deepak Seth, and Yogesh Bhatt identifies the companies *DataBricks, Google, Microsoft,*

---

[24] Weka: https://weka.ai/

[25] RapidMiner: https://RapidMiner.com/

[26] KNIME: https://KNIME.com/ and https://KNIME.org/

Dataiku, Amazon Web Services (AWS), Altair (RapidMiner), DataRobot, and *IBM* as the leaders in this market, i.e. as the leading providers of software platforms for data science and machine learning. Hence, we compare the CREXDATA platform and its visual workflow designer to these leading commercial platforms in our comparison in section 7.16.

In addition, we also compare the CREXDATA platform to the leading open source visual data analytics workflow design tools (according to our own assessment): Altair RapidMiner AI Studio (with and without the extensions developed in the CREXDATA project), KNIME, and WEKA (see comparison in section 7.16).

7.2 Multi-Modal Data Fusion

Related work and state-of-the-art for **multi-modal data fusion (MMDF)** are described in section 4.2.2, which also describes the CREXDATA system component for MMDF and its capabilities and contributions to MMDF. The MMDF engine and the compatible workflow designer developed within the CREXDATA project rely on the *Apache Kafka* streaming engine. The data fusion engine focuses on fusion of different sources of streaming data on location and time relevance, handling different modalities of spatial resolutions and sensing frequency. In addition, binding with batch data based on the location and timeframe are also supported from data records (e.g. shapefiles) or their metadata (e.g. in case of video streams).

The developed toolbox supports three different options for runtime execution. First, it can be dispatched to a Java-supporting system with access to the Kafka cluster provided with the configuration file and the core fusion engine jar executable. Second, the *RapidMiner AI Studio* can trigger execution through a developed extension on the local machine. Third, it can dispatch execution to *RapidMiner AI Hub* on one or several servers.

The MMDF-toolbox developed in the CREXDATA project provides a flexible and versatile framework that enables end-users to design task-specific solutions through workflow specifications and addresses various multi-modal fusion challenges. It allows end-users to register different data sources and align their spatial and temporal characteristics based on spatial indices and expiration times. The toolbox also enables window operations, filtering records and applying transformations through user-defined functions, facilitating **data acquisition and decision-making tasks** (late fusion).

The toolbox supports combining **multiple layers** of fusion, aggregating mechanisms and integrates with external predictors via Kafka, making it suitable **for early and hybrid fusion** tasks. Given sufficient information, such as an identifier, these tasks can be included in the workflow without explicit knowledge of the underlying raw records, which can later be included (joined on the fused record or decrypted) in a later stage to support **privacy protection**.

The tool supports linking data through join specifications (“hard-link”) and, due to its micro-services architecture, external more complex link mechanisms can also be supported. Furthermore, building upon Kafka’s real-time performance capabilities, the toolbox registers memory-demanding sources (static files, video streams) through their metadata, incorporates the highly efficient H3-index for spatial operations and defines expiration times on top of the micro-services architecture. This provides a **real-time scalable and configurable system**. Finally, end-users’ full control of the workflow and its visual representation is crucial for **explainability tasks**.

7.3 Integration of Domain-Specific Data Sources and Systems

Within the CREXDATA project, we implemented numerous data connectors to and system integrations of data sources relevant for the various CREXDATA use cases, e.g. for

- FMI weather forecasts,
- FMI wave height forecasts,
- Drone data integration,
- HYDS ARGOS integration,
- BSC health simulator integration,
- FloodWaive DeepWaive simulator integration,
- Gazebo simulator integration,
- Kpler maritime simulator integration,
- etc.

These data connectors and system integration components were implemented as RapidMiner extensions for the visual workflow designer RapidMiner AI Studio to seamlessly integrate the above data sources and systems into the CREXDATA platform and workflow designer.

With these extensions, the CREXDATA platform increases the data connectivity of the visual workflow designer beyond the state-of-the-art of such workflow designers, especially since other data analytics tools do not integrate so many simulators, if any at all. Thanks to the visual workflow designer and the extensions developed in the CREXDATA project, the state-of-the-art in the use case domains is also exceeded, enabling the integration of simulations with multi-modal real-time data, explainability, and visualizations, which was for example not possible yet in the weather emergency use case domain.

7.4 Extreme-Scale Real-Time Data Stream Processing and Analytics

The leading open-source tools for data stream processing are *Apache Kafka*[27]*, Apache Flink*[28], and *Apache Spark*[29] incl. *Spark Streaming*. Altair RapidMiner offers a RapidMiner extension for data stream processing (Streaming Extension[30]) on the RapidMiner Marketplace[31] that seamlessly integrates Flink and Spark Streaming into RapidMiner AI Studio (for seamless visual workflow design) and into the underlying RapidMiner platform (for seamless runtime execution of the visually designed workflows on distributed compute resources). Within the CREXDATA project, we developed additional RapidMiner extension to seamlessly integrate Kafka (see sections 4.1.2 and 4.1.3). Within the CREXDATA platform, Kafka is used as communication middleware between the CREXDATA system components to transfer requests, parameters, input data, output data, and other responses. Kafka, Flink, and Spark

[27] Apache Kafka: https://kafka.apache.org/

[28] Apache Flink: https://flink.apache.org/

[29] Apache Spark incl. Spark Streaming: https://spark.apache.org/

[30] RapidMiner Streaming Extension:
https://marketplace.rapidminer.com/UpdateServer/faces/product_details.xhtml?productId=rmx_streaming

[31] RapidMiner Market Place: https://marketplace.rapidminer.com/

Streaming provide the basics for extreme scale big data analytics and hence are used by the CREXDATA platform, but also integrated by many other tools and platforms (see comparison in section 7.16).

7.5 Federated Data Processing and Optimization

Federated data processing has been explored in nearby but not identical forms, ranging from batch processing to streaming setups. Though no prior framework supports federated data processing across IoT platforms, there do have been some prior works in the literature who touch upon aspects of it.

The first related frameworks that emerged were focused on optimizing and executing analytics workflows across multi-engine, but single computing node environments. Systems such as *Ires* [63] and *Rheem* [64] (now *Apache Wayang* [65]) aimed at unified data analytics across heterogeneous platforms/engines and distinguish platform independence, opportunistic cross-platform, mandatory cross-platform, and polystore settings. The contribution of these frameworks includes cross-platform optimization capabilities and execution frameworks that applications run on one or more platforms without binding users to a specific engine. These initial efforts were destined to support cross-engine processing of data at rest (batch) rather than data in motion (streams).

FERARI [66] was one of the first systems to support federated stream data processing across multiple sites hosting multi-cloud environments. However, its focus was on Complex Event Processing (CEP) operators rather than general streaming analytics workflow scenarios. For assigning CEP workflow operator across clouds, FERARI contributes a CEP optimizer which decomposes the execution of the CEP operators to lazy evaluation steps by adopting a push-pull mechanism to postpone event data transmission to an only need-to-happen basis. This, in turn, results to reduced data transmissions across the network of the cloud and to reduced network latencies.

INFORE [67] generalizes the aforementioned paradigms to multi-cloud and cross-engine streaming analytics scenarios. INforE explicitly supports execution across Big Data platforms and, potentially geo-dispersed computer clusters. Its optimizer decides not only the execution platform, but also the cluster/site where each workflow operator will run, while its Synopses Data Engine is designed to support interactivity and different forms of scalability, including scaling computation beyond a single cluster by controlling communication for answering global queries over geo-dispersed sites. This makes INforE much closer to a true federated streaming analytics system than a simple cross-engine, cross-site abstraction layer. *SheerMP* [68] complements INFORE with adaptivity features. SheerMP's optimizer goes beyond ad-hoc, optimal operator assignment, addressing the challenges of volatile statistical properties of the involved streams that are processed by the various workflow operators. This volatility can quickly render a currently optimal operator to site and platform assignment to a suboptimal one. Therefore, SheerMP continuously monitors the deployed federated data processing pipelines and adapts operator assignment as soon as the currently deployed workflow exhibits performance degradation. Importantly, this happens while the state of stateful operators included in the federated data processing pipeline is preserved (stored and loaded), while migrating from the currently deployed workflow execution plan to a new one.

As the above discussion exhibits, no prior work concisely handles federated data processing across IoT platforms, while the well-known *NebulaStream* [69] currently focuses on building an IoT-native, federated data processing engine from scratch.

7.6 Online Federated Machine Learning

The state-of-the-art meta-algorithms for communication-efficient Federated Learning (FL) developed for CREXDATA are available via the FDA-Opt component. At this time, this is the only production-level implementation of the FDA suite of algorithms. The component integrates with the latest ML technologies, including the *pytorch* framework, the transformer library from *HuggingFace* and other state-of-the-art infrastructure. One of the strengths of the FL component is its modular design. It can be used directly through the CREXDATA graphical interface, making configuration and job submission accessible to non-technical stakeholders. Additionally, it is fully operable as part of headless pipelines. This makes it useful for both production use and research prototyping.

An important innovation of the FL component is its integrated ability to model networking environments with respect to optimizing a FL job. In this respect, the NS3-based simulator can evaluate different FL job operational hyper-parameters (best optimization algorithm, network parameter tuning, number of clients per training round, etc) along with learning hyperparameters (learning rate, batch size, variance thresholds etc). The optimizer is fully integrated with the CREXDATA platform, and it can be used as a building block to automate FL job deployments.

7.7 Complex Event Forecasting and Multi-Resolution Forecasting

CREXDATA Deliverable D4.2 (Final Report on Complex Event Forecasting, Learning, and Analytics) describes the related work on Complex Event Recognition (CER), Complex Event Forecasting (CEF), and Multi-Resolution CEF and how the contributions of the CREXDATA project advance the state-of-the-art in this domain.

7.8 Multi-Lingual Social Media Mining, Text Extraction, and Event Detection

The multi-lingual social media toolkit (described in section 4.8 and in CREXDATA Deliverable D4.2) closely aligns with the current state-of-the-art for emergency event detection by leveraging transformer-based models (like BERT) fine-tuned on social media texts which consistently outperform traditional techniques (like SVMs, Random Forests). The toolkit further extends the state-of-the-art in this respect using LLM-driven automatic data annotation improved with confident learning for data cleaning, and Synthetic Data Generation (SDG) to curate high quality training data for improved event detection. This also helps with including low resource languages (like Catalan) in the detection process. Retrieval-Augmented Generation (RAG) is the emerging method for information extraction. State-of-the-art retrieval methods treat information retrieval and relevance assessment as separate tasks, our toolkit bridges this gap by merging individual rankings from both stages using rank aggregation techniques. This allows it to exploit the strengths of multiple retrieval passes, leading to more accurate final results. The toolkit's components are integrated within the CREXDATA platform leveraging the graphical workflow of the RapidMiner AI Studio as visual workflow designer.

7.9 Explainable Artificial Intelligence (XAI)

The CREXDATA component for Explainable Artificial Intelligence (XAI) was described in section 4.9 and in CREXDATA Deliverable D5.2. The state-of-the-art for XAI: While a wide range of local explainable AI (XAI) techniques currently exists, feature attribution methods like LIME and SHAP are among the most prevalent, quantifying the contribution of each feature to a specific prediction. However, these attribution scores are limited in their ability to convey semantic structure or domain-relevant logic; a simple list of feature weights fails to capture how features interact and the underlying rationale behind a prediction.

In contrast, explanation methods based on rules or decision trees, such as ANCHOR and LORE (see section 4.9 and in CREXDATA Deliverable D5.2), offer richer semantics through IF-THEN statements that express local decision logic. These structures align much more naturally with how humans conceptualize categories. Despite their popularity and model-agnostic nature, such post-hoc methods often overlook domain-specific feature relationships, which can result in implausible or misleading explanations. We address this limitation by constraining the perturbation process using domain-aware instance generators, thereby improving explanation plausibility without modifying the underlying XAI method itself. We specifically build upon LORE due to its extensible architecture; its support for externally defined neighbourhood generation is an essential capability for integrating domain knowledge.

The integration within the CREXDATA platform allows the explanation process to be defined from the training phase onward. This enables the analyst to configure and tailor the optimal neighbourhood generator for the specific use case being analyzed.

7.10 Visual Analytics

To improve how explanations are presented and validated, we employ Visual Analytics (VA) to enable the interactive exploration of explanation structures. While prior work often focuses on instance-level performance, our approach prioritizes a logic-focused, model-wide understanding that operates entirely without requiring underlying data access. To achieve this, our **RuleSense** visualization provides a comprehensive suite of interactive capabilities designed to align model behaviour with domain knowledge:

- **Interactive Heatmaps**: Class-wise and feature-wise heatmaps visualize condition frequencies across discretized value intervals. This allows users to quickly assess feature relevance and value-range plausibility at a glance.
- **Dynamic Filtering and Extraction**: Users can actively filter outcomes, include or exclude specific features, constrain value ranges, and extract rule subsets for detailed inspection. The interface provides real-time updates across overview and reference views for seamless comparison.
- **Automated Rule Analysis**: The system computationally analyzes the relationships between rules to detect contradictions, identify subsumed rules, and merge overlapping conditions. This significantly reduces cognitive complexity while preserving semantic meaning.

- **Pattern Discovery via Topic Modeling**: We apply topic modeling to rule conditions, automatically grouping frequently co-occurring sets to reveal recurring logic patterns without cluttering the user interface.

RuleSense is fully integrated into the CREXDATA platform. Designed to be completely model-agnostic, this integration allows analysts to seamlessly explore rules and configure the explanation process regardless of the underlying machine learning algorithm (see CREXDATA Deliverable D5.2 for details). We have validated the approach on complex, real-world scenarios, including vessel movement and COVID-19 use cases.

7.11 Simulator HyperSuite for Weather Emergency Use Cases

The *Simulator HyperSuite (SHS) of the Weather Emergency Use Case* (described in CREXDATA Deliverable D2.3) extends the state-of-the-art of traditional *Co-Simulation* and existing Integrated *Simulation Environment (ISE)* approaches. While Co-Simulation approaches primarily aim at temporally coupled execution of multiple simulators with a focus on synchronization and data exchange, SHS extends this paradigm to include a workflow-centered configuration of heterogeneous, domain-specific simulation models. Unlike ISEs, which often remain within a single domain or a specific tool ecosystem, SHS enables flexible integration and combination of simulators from different application fields such as flood, wildfire, and robotic simulation. A key added value lies in explicit modeling of workflows as first-class objects that describe not only the coupling but also the logical sequence, dependencies, and parameterizability of the simulations. Furthermore, SHS abstracts the technical complexity of simulator integration through a user-friendly graphical user interface (GUI), enabling even non-technical users to configure and parameterize complex scenarios in the simulation models. Compared to the State of the Art, SHS thus addresses not only interoperability at technical level, but also usability and cross-domain applicability. SHS extends the CREXDATA system into a platform for exploratory, scenario-based analyses in complex emergency situations. Overall, SHS contributes to overcoming the fragmentation of existing simulation landscapes and opens new potential for interdisciplinary decision support.

7.12 Augmented Reality

In relation to Augmented Reality (AR) for real-time decision-making in flood emergencies:

Past work in mobile AR-based visualization of coastal erosion and sea level rise employed static data in pre-defined locations to display water levels. Water visualization in mobile AR has been primarily implemented using hand-held cameras, constraining rescuers' operations. AR visualization of real-time data for spatial flood forecasting, rescuer locations and points of interest (POIs), while firefighters are moving is still a research challenge. Presenting crucial information in head-worn, hands-free AR without overloading the user's field of view is paramount for efficient situational in hazardous environments as well as presenting the uncertainty level of data and not just static, deterministic information. Our work presents an AR system combining a server and head-worn AR displaying predicted water levels, water velocity, and water direction in real-time, at future time intervals, improving situational awareness and way-finding of emergency responders. The system integrates GPS data from mobile platforms with head-worn AR to visualize critical POIs (schools, hospitals) at far and hazards such as manholes in near proximity, indicating hazards by change of colour in the operational environment, enhancing navigation, safety and emergency planning. Our system also supports uncertainty visualization for predicted flood levels over time as well as gaze interaction and gestures for hands-free operation. Proof-of-concept testing was conducted by professional firefighters in urban areas of Innsbruck and Dortmund, highlighting the system's potential to enhance way-finding during flood emergencies.

In relation to Augmented Reality (AR) for navigation, team coordination, and real-time data visualization in firefighting operations:

Wildfires present a dynamic and unpredictable challenge, yet field commanders often rely on fragmented data from paper maps and radio, which hinders real-time situational awareness. Although Augmented Reality (AR) has emerged as a promising technology, many solutions rely on handheld devices that occupy the user's hands. Existing head-worn systems, meanwhile, often address isolated tasks such as drone control or high-level coordination, rather than delivering a holistic operational picture to the field commander. This paper introduces a head-worn AR system for HoloLens 2 designed to provide team leaders with a cohesive, real-time operational view. Our primary contribution is the fusion and visualization of multiple live data sources, such as team GPS positions, active fire locations, fire spread predictions, and weather, into a unified, hands-free interface. Delivered via an interactive 3D map with AR-guided routing, our system enhances team coordination, safety, and tactical decision-making during wildfire incidents.

In relation to Augmented Reality (AR) for maritime operations:

Early Maritime AR Systems (MARS) aimed to unify ship instrument data in a single AR interface. Even the initial prototypes improved situational awareness and reduced cognitive load. Others combine AIS, radar, and route data through hybrid interfaces that merge world-referenced 3D visuals with contextual and non-contextual information. AR is not only used in open-ocean navigation but also to assist captains during harbor maneuvers, where smart glasses provide critical information for precise navigation and improve situational awareness in confined environments. While most early maritime AR systems relied on monitors to provide AR content, recent advancements have shifted towards the integration of headsets and smart glasses, keeping information directly in operator's line of sight, enabling fully hands-free. However, enabling interactive AR visualizations of real-time data (e.g. forecast hazards with uncertainty indicators) without occluding the operator's view remains an open challenge. Presenting critical information via head-worn AR, while avoiding new visual obstructions, is essential for maintaining safety and situational awareness. To address these challenges, we introduce an innovative head-worn AR navigation system developed to enhance wayfinding and situational awareness for ship captains, while at sea. It features a range of interaction techniques and includes a mini-map that highlights key points of interest (POIs) and uncertainty cones for projected navigational paths, offering a comprehensive overview of the ship's surroundings.

In relation to Augmented Reality (AR) for the detection of mental fatigue in first responders:

Despite advances in fatigue detection, physiological measures are often intrusive, high-precision eye trackers remain costly, and AR devices lack sufficient integration. Existing studies are also domain-specific, underscoring the need for generalizable, gaze-based approaches that support real-time fatigue monitoring in AR environments. This work introduces a low-cost, modular eye tracking add-on for AR headsets without adequate native tracking, enabling prediction of mental fatigue from ocular metrics using machine learning. A low-cost, modular eye tracking device designed for AR glasses without built-in or accurate tracking has been developed, expanding the applicability of ocular-based cognitive monitoring, providing pupil diameter, saccadic speed, and blink data. This is accompanied by a clustering approach for mental fatigue detection using unsupervised algorithms, integrating both self-reported measures and eye tracking metrics. An analysis of variations in eye gaze movements across varied tasks is conducted, forming the basis for pattern-based clustering methods in mental fatigue detection. A machine learning framework using ocular metrics to estimate mental fatigue in real time is integrated, enabling adaptive support in AR collaboration environment.

7.13 Decision Support for Weather Emergency Use Cases

The CREXDATA platform exceeds the state-of-the-art in the weather emergency domain by enabling the easy integration of simulations with multi-modal real-time data, explainability, and visualizations, which was not possible yet before. Nowadays, fire departments use various digital decision-making and situation support systems during extreme weather events. In addition to traditional operations management and documentation systems, such as *CommandX*[32] or *MP-FEUER*[33], national platforms such as the BBK's *MoWaS* warning system[34] are used, often with a direct link to weather data from the German Weather Service. Digital situation assessment and command systems enable coordinated situation assessment, while research projects such as *DRIVER+*[35] rely on real-time data and forecasts for decision support. In addition, hydrological early warning systems, e.g., *European Flood Awareness System EFAS*[36], and mobile apps such as *RescueWave*[37] or *Fireboard Mobile*[38] are used. This creates an increasingly networked, data-based decision-making environment for fire departments in extreme weather conditions. The CREXDATA system distinguishes itself from existing solutions by utilizing existing data in real time for processing via Data Science / Artificial Intelligence (DS/AI) technologies to support decision-making. The results generated by these DS/AI technologies are visualized in the *ARGOS* platform for various end users, thereby supporting the complex decision-making process in critical emergency situations.

7.14 Decision Support for Maritime Use Cases

The CREXDATA platform advances the state-of-the-art in maritime decision support by integrating real-time AIS streams, environmental data, and predictive analytics into a unified,

[32] https://www.eurocommand.com/produkte/commandx

[33] https://www.mpbos.de/produkte/software-mp-feuer/

[34] https://www.bbk.bund.de/DE/Warnung-Vorsorge/Warnung-in-Deutschland/MoWaS/mowas_node.html

[35] https://www.driver-project.eu/

[36] https://www.copernicus.eu/en/european-flood-awareness-system

[37] https://rescuewave.de/

[38] https://fireboard.net/fireboard-mobile/

scalable framework. Unlike existing maritime systems such as AIS monitoring platforms, ECDIS, and standalone weather-routing tools which operate largely in isolation, CREXDATA enables real-time multi-modal data fusion and AI-driven decision-making for safety-critical scenarios.

The maritime use case focuses on hazardous weather routing and vessel collision avoidance, both operating on continuous AIS data streams derived from Kpler's global network (13,000+ receivers, 350,000+ vessels, >1 billion AIS messages/day). These data are processed using a distributed Pekko actor-based architecture, enabling low-latency analytics at extreme scale. Hazardous weather routing leverages extensive preprocessing of historical AIS and meteorological datasets (terabyte-scale AIS archives and hundreds of gigabytes of environmental data annually) to generate routing graphs for different historical weather conditions and enable millisecond-level rerouting decisions during streaming operation. In parallel, collision avoidance continuously evaluates vessel interactions using short-term trajectory prediction. The system dynamically instantiates actors for vessel pairs, resulting in a computational workload that scales with $O(n^2)$ pairwise interactions in dense traffic environments. This complexity is efficiently managed through distributed processing and spatial partitioning (Uber H3 index grid), ensuring near-real-time performance.

A key innovation is the integration of maritime analytics into RapidMiner workflows via custom operators and APIs. The Kpler MSA Collision Forecast API enables automated job execution, streaming data ingestion via Kafka, and asynchronous retrieval of collision events, allowing users to embed maritime decision support within broader DS/AI pipelines. Additionally, integration with ASVs and edge systems enables real-time feedback loops, where routing and collision-avoidance outputs directly support mission execution.

Overall, CREXDATA delivers a scalable, AI-driven maritime decision-support system that surpasses existing solutions by combining extreme-scale data processing, real-time analytics, and workflow integration, enabling safer and more efficient maritime operations.

7.15 Optimization and Automation Support for Health Use Cases

CREXDATA Deliverable D2.3 "Final Use Case Evaluation, Pilots, Demonstrators and Simulation Models and Tools" describes the evaluation of the CREXDATA platform for the health use case scenarios and how the CREXDATA platform advances the state-of-the-art. The Health Crisis use case presents results in both scenarios: epidemiological and mechanistic infection, modelling to support pandemic response and patient-specific treatment optimization. It leverages advanced simulators and federated learning to improve clinical decision-making. In the Epidemiological scenario we combine simulation-based and data-driven approaches to improve epidemic preparedness and response. It includes pilots on epidemic parameter calibration, vaccination and mobility intervention optimization, and the use of information-theoretic and visual analytics methods to uncover the relationship between population mobility and epidemic spread. In the multiscale infection scenario, we used RapidMiner's AI platform to integrate two simulators and a supercomputer to allow for better parameter characterization, the early classification of simulation trajectories and the modification of parameters at runtime. This architecture allowed us to identify potential drug targets that avoid the further infection of epithelial cells by SARS-CoV-2. Automatically varying simulation parameters and optimizing them from a data analytics workflow (RapidMiner process) invoking the simulators and collecting the simulation results reduces the workload of the human expert (less manual parameter variations and evaluations) and accelerates the overall parameter optimization process, which allows to test more parameter variations at the same time and to obtain better results faster.

7.16 Overall CREXDATA System Architecture and Platform

In this section, we compare the overall CREXDATA Platform including its visual data analytics workflow designer (Altair RapidMiner AI Studio[39]), its Multi-Modal Data Fusion (MMDF) component, and its other components to the state-of-the-art data integration tools and to the state-of-the-art data analytics and machine learning platforms, both commercial and open source, in order to show how the CREXDATA platform and its components advance the state-of-the-art. The results are presented in Figure 105.

The *2025 Gartner® Magic Quadrant™ for Data Integration Tools* published in December 2025 by the Gartner Analysts Michele Launi, Nina Showell, Robert Thanaraj, and Sharat Menon identifies the companies *Informatica*[40], *Microsoft*[41], *Oracle*[42], *IBM*[43], *Ab Initio, Amazon Web Services (AWS), Denodo, Qlik*, and *Google*[44] as the leaders in this market, i.e. as the leading providers of software tools for data integration. Hence, we compare the tools of the first four of these vendors to the CREXDATA platform, its Multi-Modal Data Fusion component, and visual workflow designer (Altair RapidMiner AI Studio) in terms of their respective data integration and fusion capabilities. In addition, we also compare the CREXDATA platform, its MMDF component, and its visual workflow designer Altair RapidMiner to the leading open-source data integration tools (according to our own assessment): *Hitachi Pentaho*[45] (incl. *Pentaho Data Integration* (formerly *Kettle*) and *Weka*[46] for Data Analytics and Machine Learning), *RapidMiner*, and *Google TensorFlow*[47].

The *2025 Gartner® Magic Quadrant™ for Data Science and Machine Learning Platforms* published in May 2025 by the Gartner Analysts Afraz Jaffri, Maryam Hassanlou, Tong Zhang, Deepak Seth, and Yogesh Bhatt identifies the companies *DataBricks*[48], *Google*, *Microsoft*,

---

[39] Altair RapidMiner AI Studio: https://RapidMiner.com/

[40] Informatica: https://informatica.com/

[41] Microsoft: https://microsoft.com/

[42] Oracle: https://oracle.com/

[43] IBM: https://ibm.com/

[44] Google incl. TensorFlow: https://google.com/ and https://TensorFlow.org/

[45] Pentaho Business Intelligence Software: https://Pentaho.com/

[46] WEKA Machine Learning Tool: https://weka.ai/

[47] Google TensorFlow: https://TensorFlow.org/

[48] DataBricks: https://DataBricks.com/

Dataiku[49], Amazon Web Services (AWS)[50], Altair (incl. RapidMiner)[51], DataRobot[52], and IBM as the leaders in this market, i.e. as the leading providers of software platforms for data science and machine learning. Hence, we compare the CREXDATA platform and its visual workflow designer RapidMiner AI Studio to these leading commercial platforms in our comparison. In addition, we also compare the CREXDATA platform to the leading open-source visual data analytics workflow design tools (according to our own assessment): Altair RapidMiner AI Studio[53] (without the extensions developed in the CREXDATA project), KNIME[54], and WEKA[55], and to the widely used open-source machine learning platforms H2O[56] and Google TensorFlow[57].

The capabilities of the different software platforms were determined in a market research in March and April 2026 based on the website of the respective vendors and internet searches for the respective capabilities. For vendors offering several relevant software platforms and tools (e.g. Amazon, Google, IBM, Microsoft, etc.), all of their identified relevant platforms and tools were considered for the comparison and summarized under the respective vendor's name.

In Figure 105 we compare the CREXDATA platform capabilities to those of the other systems mentioned above, highlighting the capabilities and key differentiators of the CREXDATA platform. Our CREXDATA solution incorporates a set of different components, which each advances the state-of-the-art in its field. The CREXDATA platform thus provides significant custom solutions and capabilities, compared to other generic platforms. However, the CREXDATA platform goes beyond just being the sum of its parts. Some strong points of the CREXDATA architecture and platform include its modularity & configurability to different application domains and use cases, its use of open APIs, its open source availability, its extensibility (e.g., via RM extensions, but also via Kafka-based APIs, etc.), the re-usability of components (e.g., MMDF, CEF, federated learning, XAI, visualization and AR tools are also usable as stand-alone tools, while the RM extensions are also usable for other contexts and applications) and the generality of our platform (which is adaptable to other application domains, as demonstrated across different application domains in CREXDATA, but also to other different domains like e.g. manufacturing and industrial production).

[49] Dataiku: https://DataBricks.com/

[50] Amazon Web Services (AWS): https://aws.amazon.com/

[51] Altair incl. RapidMiner: https://Altair.com/ and https://RapidMiner.com/

[52] DataRobot: https://DataRobot.com/

[53] Altair RapidMiner Platform for Machine Learning and Artificial Intelligence: https://RapidMiner.com/

[54] KNIME Machine Learning Platform: https://knime.com/

[55] WEKA Machine Learning Tool: https://weka.ai/

[56] H2O Machine Learning Platform: https://h2o.ai/

[57] Google TensorFlow: https://TensorFlow.org/

Capability Group	Capability	Tools & Platforms for Visual Workflow Design, Data Integration, Data Fusion, Real-Time Data Stream Processing, Data Analytics, Machine Learning, Artificial Intelligence, Model Deployment, and Workflow Execution																
		CREXDATA Platform incl. all Compo-nents	Altair RapidMiner without the CREXDATA Extensions	KNIME incl. 3rd Party Extensions	H20 (h2o.ai)	WEKA incl. AutoWeka, Weka-GDPM, ImageJ Plugin, DistributedWekaSpark	IBM Watson (SPSS Modeler, Auto AI, Machine Learning Accele-rator, Studio, WatsonX, WatsonX Assistant)	Infor-matica (Cloud Data Integra-tion, Intelli-gent Data Manage-ment Cloud, CLAIRE AI)	Microsoft incl. Azure Data Factory, AzureML, Power Platform, Power BI, Copilot Studio, Cognitive Services	Oracle (Oracle Data Miner, Oracle Machine Learning, Oracle Data Science, Autono-mous Database)	Hitachi Pentaho incl. Data Inte-gration & Weka	Incorta (Datalake with Code-Focused ML)	Data-Bricks incl. MLFlow, LakeFlow, Spark, Spark Streaming	Dataiku incl. Data-Bricks	Data-Robot (AutoML- & Code-focused Platform, Python, Note-books, MLFlow)	Google incl. AI Studio, Gemini, AutoML, Tensor-Flow, Vertex AI, Big Query, Cloud, Compose	Amazon AWS (Sage-Maker, MLFlow, AWS KI, BedRock, IoT, EMR, MSK)	Madroid (Visual AI Platform)
License Type(s)	Open Source (OS) / Commercial (COM)	Licenses of the components: some OS, some COM	OS Version (AI Studio) & COM Version (AI Studio,	OS (AI Studio) & COM (AI Server)	OS & COM	OS	COM	COM	COM	COM	OS & COM	COM	COM	COM	COM	OS (Tensor-Flow) & COM	COM	COM
Visual Work-flows	Visual Workflow Designer	Yes	Yes	Yes	No	Yes	Yes	Yes	(Yes) (AutoML- & Code-focused)	(Yes) (AutoML- & Code-focused)	Yes	Yes	(Yes) (Code-Oriented)	Yes	(Yes) (AutoML- & Code-focused)	(Yes) (LLM- / Code-focused)	(Yes) (AutoML- & Code-focused)	No
Data Integration & Fusion	Number of Data Connectors	100+	100+	100+	36+	12+	100+	400+	1000+	100+	30+	100+	90+	40+	50+	20+ native, 800+ by partners	50+	image data, video data
	Tabular Data	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Time Series Data	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Text Data	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Image Data	Yes	Yes	Yes	Yes	(Yes)	Yes	Yes	Yes	Yes	(Yes)	No	Yes	Yes	Yes	Yes	Yes	Yes
	Video Data (or Video Meta-Data)	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes
	Geo-Spatial	Yes	Yes	Yes	(Yes)	Yes	Yes	(Yes)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Temporal	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Multi-Modal Data Fusion (MMDF)	Yes	Yes	Yes	(Yes)	(Yes)	Yes	(Yes)	Yes	Yes	(Yes)	No	Yes	(Yes)	(Yes)	Yes	(Yes)	Yes
	Multi-Resolution MMDF	Yes	Yes	Yes	No	No	Yes	No	Yes	(Yes)	No	No	No	No	No	(Yes)	No	No
Data Streaming	Extreme-Scale Real-Time Data Stream Processing	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Can Combine Stream & Batch Data	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Video Streams	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	No	No	Yes	(Yes)	Yes	Yes	Yes	Yes
	Kafka Support	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
	Flink Support	Yes	Yes	(Yes)	No	No	Yes	No	Yes	Yes	No	No	Yes	No	No	Yes	Yes	No
	Spark Support	Yes	Yes	Yes	Yes	(Yes)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	Spark Streaming	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Machine Learning (ML) & Artificial Intelligence (AI)	Number of ML Algorithms	200+	200+	100+	30+	100+	100+	ML for ETL only, LLM for querying	AzureML: 20+, MLFlow: 30+	30+	100+	30+	50+	36+	50+	30+	100+	10+
	AutoML	Yes	Yes	Yes	Yes	(Yes)	Yes	only for ETL	Yes	Yes	(Yes)	(Yes)	Yes	Yes	Yes	Yes	Yes	No
	Classification	Yes	Yes	Yes	Yes	Yes	Yes	only for ETL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Regression	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	Forecasting	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	Clustering / Segmentation	Yes	Yes	Yes	Yes	Yes	Yes	(Yes)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	Feature Selection	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	Feature Generation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	Deep Learning	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
	Generative AI	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Large Language Models (LLM)	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Large Vision Models	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes
	Large Multi-Modal Models	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes
	Text Analytics	Yes	Yes	Yes	Yes	Yes	Yes	via LLM	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	Social Media Crawler (Twitter aka X)	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	(Yes)	Yes	No	(Yes)	(Yes)	No
	Event Extraction from Text	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	(Yes)	No	No
	Complex Event Recognition (CER)	Yes	No	No	No	No	Yes	(Yes)	Yes	Yes	No	No	(Yes)	No	No	(Yes)	No	Yes
	Complex Event Forecasting (CEF)	Yes	No	No	No	No	Yes	(Yes)	(Yes)	No	No	No	(Yes)	No	No	(Yes)	No	No
	Multi-Resolution CEF	Yes	No	No	No	No	Yes	(Yes)	No	No	No	No	No	No	No	No	No	No
	CEF with SREMO	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Federated Machine Learning	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	No	No	Yes	No	No	Yes	No	No
	Explainable AI (XAI)	Yes	Yes	Yes	Yes	No	Yes	via LLM	Yes	Yes	No	No	Yes	Yes	Yes	Yes	(Yes)	No
	XAI with LORE	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Uncertainty Visualization	Yes	(Yes)	(Yes)	(Yes)	No	(Yes)	No	(Yes)	(Yes)	No	No	(Yes)	(Yes)	(Yes)	(Yes)	No	No
Runtime Execution Capabilities	Cloud	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Server	Yes	Yes	Yes	Yes	Yes	Yes	No	(Yes)	Yes	Yes	No	(Yes)	Yes	Yes	Yes	No	Yes
	Desktop	Yes	Yes	Yes	Yes	Yes	Yes	No	(Yes)	(Yes)	Yes	No	(Yes)	Yes	Yes	Yes	No	Yes
	Edge & IoT	Yes	Yes	Yes	Yes	Yes	Yes	Readers for IoT data, but no process execution on edge	Yes	(Yes)	Yes	No	No	(Yes)	No	Yes	IoT Data Analytics, but Limited Execution on IoT	Yes
	Kafka Cluster	Yes	Yes	Yes	Yes	No	Yes	Readers, but no workload distribution	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	Flink Cluster	Yes	Yes	(Yes)	Yes	No	Yes	No	Yes	Yes	No	No	Yes	No	No	Yes	Yes	No
	Spark Cluster	Yes	Yes	Yes	Yes	(Yes)	Yes	Readers, but no workload distribution	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	High-Performance Computing (HPC) on Super Computers	Yes (BSC)	No	No	No	No	(Yes)	No	(Yes)	No (Cloud only)	No	No	No	No	No	No	(Yes)	No
	Cloud-to-Edge Compute Resource Utilization Optimizer	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	End-to-End Automation & MLOps & Model Ops	Yes	Yes	(Yes)	(Yes)	No	Yes	(Yes)	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	(Yes)
Weather Emergency Use Case Support	FMI Weather Forecast Integration	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Drone Data Integration	Yes	(Yes)	(Yes)	(Yes)	No	Yes	No	Yes	Yes	No	No	Yes	Yes	(Yes)	Yes	No	No
	Social Media Integration (Twitter/X)	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	Yes	(Yes)	(Yes)	No	(Yes)	No	No
	FloodWaive Flood Simulator Integration	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Spark Wildfire Simulator Integration	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Gazebo Robotic Simulator Integration	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	(Yes)	No	No
	Simulator HyperSuite (SHS)	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Flood Barrier Height Optimizer	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Augmented Reality based Interaction	Yes	No	No	No	No	Yes	No	Yes	Yes	No	No	No	No	No	Yes	No	No
Health Emergency Use Case Support	Simulator Integration: Alya	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Simulator Integration: PhysiBoSS	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Automation of Simulator Parameter Optimization	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Maritime Use Case Support	FMI Weather Forecast Integration	Yes	No	No	No	No	IBM weather forecast models	No	No	No	No	No	No	No	No	No	No	No
	FMI Wave Height Forecast Integration	Yes	No	No	No	No	IBM maritime weather & wave	No	No	No	No	No	No	No	No	No	No	No
	Maritime Data Integration	Yes	(Yes)	(Yes)	No	No	Yes	No	Yes	Yes	No	No	Yes	No	No	Yes	No	No
	Vessel Sensor Data Integration	Yes	No	No	No	No	No	No	(Yes)	(Yes)	No	No	(Yes)	No	No	Yes	No	No
	Kpler Integration	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Collision Prediction & Avoidance	Yes	(Yes)	(Yes)	No	No	No	No	No	No	No	No	No	No	No	No	No	No
	Hazardous Weather Routing	Yes	No	No	No	No	No	No	No	No	No	No	(Yes)	No	No	No	No	No
	Augmented Reality based Interaction	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No

Figure 105: Comparison of the CREXDATA Platform to other Existing Platforms

8 Acronyms and Abbreviations

This section lists the acronyms and abbreviations used in this document.

- AI – Artificial intelligence
- AI Hub – Altair RapidMiner AI Hub – server version of the Altair RapidMiner data science software platform
- AIS – Automatic Identification System
- AI Studio – Altair RapidMiner Studio – desktop version of the Altair RapidMiner data science software including a visual editor for designing data processing workflows
- API – Application Programming Interface
- AR – Augmented Reality
- ASV Autonomous Surface Vehicle
- ATF – Analytical Task Force
- CARS - Context-Aware Recommender Systems (CARS)
- BO – Bayesian Optimization
- C2 – Command & Control
- CA – Consortium Agreement
- CEF – Complex Event Forecasting
- CEP – Complex Event Processing
- CER – Complex Event Recognition
- CLP - Contrastive Language–image Pretraining
- CNN - Convolutional Neural Networks
- D – deliverable
- DoA – Description of Action (Annex 1 of the Grant Agreement)
- DQ – Data Quality
- DS – Data Science
- DWD – Deutscher Wetterdienst (German Weather Service)
- EB – Executive Board
- EC – European Commission
- ECC – Emergency Control Center
- ECMWF –European Centre for Medium-Range Weather Forecasts
- EFAS – European Flood Awareness System
- EFFIS – European Forest Fire Information System
- EDO – European Drought Observatory
- EMSA – European Maritime Safety Agency
- ERCC – Emergency Response Coordination Centre
- EMS – Emergency Management System (Copernicus)
- ESC - Exhaustive Search with Counting
- ETA – Estimated Time of Arrival
- EUCPM – European Union Civil Protection Mechanism
- FL – Federated Learning (Federated Machine Learning)
- FMM – Full-order Markov Model
- FSM – Finite State Model (e.g. Markov Model)
- GA – General Assembly / Grant Agreement
- GDPR – General Data Protection Regulation
- GenAI - Generative Artificial Intelligence
- GoG - Graph of Graphs

- GPS – Global Positioning System
- GSP - Greedy Search Progressive
- GUI – Graphical User Interface
- HMD – Head-Mounted Display
- HMI – Human Machine Interface
- HPC – High Performance Computing
- HRES – High-Resolution Forecast
- ICU – Inertial Control Unit
- ID – (unique) identifier
- IDE – Integrated Development Environment
- IMO – International Maritime Organization
- INFORE - Interactive Extreme-Scale Analytics and Forecasting (EU-funded R&D project that received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No 825070)
- IoT – Internet-of-Things
- IPR – Intellectual Property Right
- IQ – Information Quality
- IT – Information Technology
- KPI – Key Performance Indicator
- LLM - Large Language Models
- M – Month
- MC – Markov Chain
- MCC – Mathews Correlation Coefficient
- ML – Machine Learning
- MMSI – Maritime Mobile Service Identity
- MoSCoW – Must have, Should have, Could have, Won't have
- MS – Milestone
- MT – Maritime Traffic, now a subsidiary of Kepler
- NIST – National Institute of Standards and Technology (US)
- NLP – Natural Language Processing
- NPI – Non-Pharmaceutical Interventions
- PaaS – Prediction-as-a-Service
- PM – Person Month (or Project Manager, depending on the context)
- PPDR – Public Protection and Disaster Relief
- RAG - Retrieval-Augmented Generation (in the context of Generative Artificial Intelligence (GenAI))
- ReqIF – Requirements Interchange Format
- RL - Reinforcement Learning
- RM – RapidMiner – Altair RapidMiner – data ingestion, data processing, and data science platform underlying the CREXDATA system
- RM AI Hub – Altair RapidMiner AI Hub – server version of the Altair RapidMiner data science software platform
- RM AI Studio – Altair RapidMiner AI Studio – desktop version of the Altair RapidMiner data science software featuring a graphical user interface (GUI) including visual data processing workflow designer/editor
- RNN - Recurrent Neural Network
- RobLW – Command car of a special robotic emergency response unit
- ROS – Robot Operating System

- RSS – Random Sampling Search
- RTF – Robotic Task Force
- RTSA – Altair RapidMiner Real-Time Scoring Agent, a light-weight version of Altair RapidMiner AI Hub, optimized for low resource consumption, low latency, and high throughput and hence deployable on IoT and edge devices
- Rviz – ROS Visualizer
- SEIR – Susceptible-Exposed-Infectious-Recovered
- SREMO – Symbolic Register Expression with Memory and Output
- SRT – Symbolic Register Transducer
- STDF - Spatio-Temporal Data Fusion
- SysML – Systems Modelling Language
- T – Task
- TAG - Tiny AGgregation
- TRL – Technology Readiness Level
- UAV – Unmanned Aerial Vehicle
- UC – Use Case
- UDF – User Defined Function
- UGV – Unmanned Ground Vehicle
- UI – User Interface
- UML – Unified Modelling Language
- URL / URI – Uniform Resource Locator / Identifier
- UV – Unmanned Vehicle
- VDS – Voyage Data Streamer
- VMM – Variable-order Markov Model
- VTS – Vessel Traffic Service
- VR – Virtual Reality
- WebODM – Web Open Drone Map
- WMS – Web Map Service
- WP – Work Package
- WPL – Work Package Leader
- XAI – eXplainable Artificial Intelligence

9 References

[1] Pruitt J, Grudin J (2003) Personas: Practice and theory. In: Arnowitz J, Chalmers A, Swack T et al. (eds) Proceedings of the 2003 conference on Designing for user experiences - DUX '03. ACM Press, New York, New York, USA, pp 1–15
[2] Rupp C (2014) Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil, 6., aktualisierte und erweiterte Auflage. Hanser, München
[3] Gräßler I, Oleff C (2022) Systems Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg
[4] OMG (2016) Requirements Interchange Format (ReqIF), 1.2th edn.
[5] (2019) ReqIF Implementation Guide: Referring to ReqIF1.2
[6] OMG (2019) Systems Modeling Language, 1.6th edn.
[7] Vallejo P, Mazo R, Jaramillo C et al. (2020) Towards a new template for the specification of requirements in semi-structured natural language. JSERD 8:3. https://doi.org/10.5753/jserd.2020.473
[8] Pottebaum J, Artikis A, Marterer R et al. (2012) User-Oriented Evaluation of Event-Based Decision Support Systems. In: 2012 IEEE 24th International Conference on Tools with Artificial Intelligence. IEEE, pp 162–169
[9] Yin RK (2018) Case study research and applications: Design and methods, Sixth edition. SAGE, Los Angeles, London, New Dehli, Singapore, Washington DC, Melbourne
[10] Trucchia A, D'Andrea M, Baghino F et al. (2020) PROPAGATOR: An Operational Cellular-Automata Based Wildfire Simulator. Fire 3:26. https://doi.org/10.3390/fire3030026
[11] Arenas A, Cota W, Gómez-Gardeñes J et al. (2020) Modeling the Spatiotemporal Epidemic Spreading of COVID-19 and the Impact of Mobility and Social Distancing Interventions. Phys Rev X 10. https://doi.org/10.1103/PhysRevX.10.041055
[12] Vázquez M, Houzeaux G, Koric S et al. (2016) Alya: Multiphysics engineering simulation toward exascale. Journal of Computational Science 14:15–27. https://doi.org/10.1016/j.jocs.2015.12.007
[13] Ponce-de-Leon M, Montagud A, Noel V et al. (2022) PhysiBoSS 2.0: a sustainable integration of stochastic Boolean and agent-based modelling frameworks
[14] Letort G, Montagud A, Stoll G et al. (2019) PhysiBoSS: a multi-scale agent-based modelling framework integrating physical dimension and cell signalling. Bioinformatics 35:1188–1196. https://doi.org/10.1093/bioinformatics/bty766
[15] Zurich Insurance Group Ltd (2023) Three common types of floods explained. https://www.zurich.com/en/knowledge/topics/flood-and-water-damage/three-common-types-of-flood. Accessed 12 Jun 2023
[16] Pottebaum J, Schafer C, Kuhnert M et al. (2016) Common information space for collaborative emergency management. In: 2016 IEEE Symposium on Technologies for Homeland Security (HST). IEEE, pp 1–6
[17] Kruijff-Korbayova I, Grafe R, Heidemann N et al. (2021) German Rescue Robotics Center (DRZ): A Holistic Approach for Robotic Systems Assisting in Emergency Response. In: 2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, pp 138–145
[18] (2022) Starkregenereignis „Bernd" 2021: Bericht zur Einsatznachbereitung, Berlin
[19] Szönyi M, Roezer V, Deubelli T et al. (2022) PERC floods following „Bernd", Zürich
[20] (2022) Die Flutkatastrophe im Juli 2021 in Deutschland: Ein Jahr danach: Aufarbeitung und erste Lehren für die Zukunft. DKKV-Schriftenreihe, Bonn

[21] BBK Referat III.2 (2019) Die Analytische Task Force: Informationen zu Leistungsspektrum und Anforderungswegen, 3.0th edn., Bonn
[22] Miguel Ponce-de-Leon, Javier del Valle, José María Fernández et al. (2021) COVID19 Flow-Maps GeoLayers dataset. Zenodo
[23] Miguel Ponce-de-Leon, Javier del Valle, José María Fernández et al. (2021) COVID19 Flow-Maps Daily Cases Reports. Zenodo
[24] Miguel Ponce-de-Leon, Javier del Valle, José María Fernández et al. (2021) COVID19 Flow-Maps Daily-Mobility for Spain. Zenodo
[25] Miguel Ponce-de-Leon, Javier del Valle, José María Fernández et al. (2021) COVID19 Flow-Maps Population data. Zenodo
[26] Batini C, Scannapieco M (2006) Data quality: Concepts, methodologies and techniques. Data-centric systems and applications. Springer, Berlin, Heidelberg
[27] Eppler MJ (2006) Managing Information Quality. Springer Berlin Heidelberg, Berlin, Heidelberg
[28] Cimolino U (2014) Analyse der Einsatzerfahrungen und Entwicklung von Optimierungsmöglichkeiten bei der Bekämpfung von Vegetationsbränden in Deutschland
[29] Gizikis A, O'Brien T, Gomez Susaeta I et al. (2017) Guidelines to increase the benefit of social media in emergencies: EmerGent Deliverable 7.3, Paderborn
[30] Lorini V, Castillo C, Dottori F et al. (2019) Integrating Social Media into a Pan-European Flood Awareness System: A Multilingual Approach. In: Franco Z, González JJ, Canós JH (eds) Proceedings ofthe 16th ISCRAM Conference
[31] Havas C, Resch B, Francalanci C et al. (2017) E2mC: Improving Emergency Management Service Practice through Social Media and Crowdsourcing Analysis in Near Real Time. Sensors (Basel) 17. https://doi.org/10.3390/s17122766
[32] Erat O, Isop WA, Kalkofen D et al. (2018) Drone-Augmented Human Vision: Exocentric Control for Drones Exploring Hidden Areas. IEEE Trans Vis Comput Graph 24:1437–1446. https://doi.org/10.1109/TVCG.2018.2794058
[33] Chen L, Takashima K, Fujita K et al. (2021) PinpointFly: An Egocentric Position-control Drone Interface using Mobile AR. In: Kitamura Y, Quigley A, Isbister K et al. (eds) Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, pp 1–13
[34] Suzuki R, Karim A, Xia T et al. (2022) Augmented Reality and Robotics: A Survey and Taxonomy for AR-enhanced Human-Robot Interaction and Robotic Interfaces. In: Barbosa S, Lampe C, Appert C et al. (eds) CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, pp 1–33
[35] Gräßler I, Pottebaum J, Scholle P (2018) Influence Factors for Innovation in Digital Self-Preparedness Services and Tools. International Journal of Information Systems for Crisis Response and Management 10:20–37. https://doi.org/10.4018/ijiscram.2018010102
[36] Chowell G (2017) Fitting dynamic models to epidemic outbreaks with quantified uncertainty: A Primer for parameter uncertainty, identifiability, and forecasts. Infect Dis Model 2:379–398. https://doi.org/10.1016/j.idm.2017.08.001
[37] Ponce-de-Leon M, Montagud A, Akasiadis C et al. (2022) Optimizing Dosage-Specific Treatments in a Multi-Scale Model of a Tumor Growth. Front Mol Biosci 9:836794. https://doi.org/10.3389/fmolb.2022.836794
[38] Akasiadis C, Ponce-de-Leon M, Montagud A et al. (2022) Parallel model exploration for tumor treatment simulations. Computational Intelligence 38:1379–1401. https://doi.org/10.1111/coin.12515

[39] Richardson RA, Wright DW, Edeling W et al. (2020) EasyVVUQ: A Library for Verification, Validation and Uncertainty Quantification in High Performance Computing. JORS 8:11. https://doi.org/10.5334/jors.303
[40] Rong H, Teixeira AP, Guedes Soares C (2019) Ship trajectory uncertainty prediction based on a Gaussian Process model. Ocean Engineering 182:499–511. https://doi.org/10.1016/j.oceaneng.2019.04.024
[41] Zhang W, Deng Y, Du L et al. (2022) A method of performing real-time ship conflict probability ranking in open waters based on AIS data. Ocean Engineering 255:111480. https://doi.org/10.1016/j.oceaneng.2022.111480
[42] Abebe M, Noh Y, Kang Y-J et al. (2022) Ship trajectory planning for collision avoidance using hybrid ARIMA-LSTM models. Ocean Engineering 256:111527. https://doi.org/10.1016/j.oceaneng.2022.111527
[43] Lyu H, Hao Z, Li J et al. (2023) Ship Autonomous Collision-Avoidance Strategies—A Comprehensive Review. JMSE 11:830. https://doi.org/10.3390/jmse11040830
[44] Walther L, Rizvanolli A, Wendebourg M et al. (2016) Modeling and Optimization Algorithms in Ship Weather Routing. International Journal of e-Navigation and Maritime Economy 4:31–45. https://doi.org/10.1016/j.enavi.2016.06.004
[45] Rawson A, Brito M, Sabeur Z et al. (2021) A machine learning approach for monitoring ship safety in extreme weather events. Safety Science 141:105336. https://doi.org/10.1016/j.ssci.2021.105336
[46] Vodas M, Bereta K, Kladis D et al. (2021) Online Distributed Maritime Event Detection & Forecasting over Big Vessel Tracking Data. In: 2021 IEEE International Conference on Big Data (Big Data). IEEE, pp 2052–2057
[47] Vouros GA, Vlachou A, Santipantakis G et al. (2018) Increasing Maritime Situation Awareness via Trajectory Detection, Enrichment and Recognition of Events. In: R. Luaces M, Karimipour F (eds) Web and Wireless Geographical Information Systems, vol 10819. Springer International Publishing, Cham, pp 130–140
[48] Xiao Z, Fu X, Zhang L et al. (2020) Traffic Pattern Mining and Forecasting Technologies in Maritime Traffic Service Networks: A Comprehensive Survey. IEEE Trans Intell Transport Syst 21:1796–1825. https://doi.org/10.1109/TITS.2019.2908191
[49] Chou C-C, Wang C-N, Hsu H-P (2022) A novel quantitative and qualitative model for forecasting the navigational risks of Maritime Autonomous Surface Ships. Ocean Engineering 248:110852. https://doi.org/10.1016/j.oceaneng.2022.110852
[50] Popov AN, Kondratiev AI, Smirnov IO (2018) The algorithm for fast forecasting of the collision danger degree with ships and surface objects in the e-navigation area. In: 19th Annual General Assembly – AGA 2018 International Association of Maritime Universities (IAMU)
[51] Xiao Z, Fu X, Zhang L et al. (2017) Data-driven multi-agent system for maritime traffic safety management. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp 1–6
[52] Jie W, Yao-Tian F (2008) Risk analysis based on the ship collision modeling and forecasting system. In: 2008 IEEE International Conference on Systems, Man and Cybernetics. IEEE, pp 1517–1521
[53] Zinchenko S, Nosov P, Mateichuk V et al. (2019) Automatic collision avoidance system with many targets, including maneuvering ones. Bul.Kar.Univ "Phys" Ser 96:69–79. https://doi.org/10.31489/2019ph4%2F69-79

[54] ARTUSI E (2021) Ship path planning based on Deep Reinforcement Learning and weather forecast. In: 2021 22nd IEEE International Conference on Mobile Data Management (MDM). IEEE, pp 258–260
[55] Zis TP, Psaraftis HN, Ding L (2020) Ship weather routing: A taxonomy and survey. Ocean Engineering 213:107697. https://doi.org/10.1016/j.oceaneng.2020.107697
[56] Shin YW, Abebe M, Noh Y et al. (2020) Near-Optimal Weather Routing by Using Improved A* Algorithm. Applied Sciences 10:6010. https://doi.org/10.3390/app10176010
[57] Grifoll M, Borén C, Castells-Sanabra M (2022) A comprehensive ship weather routing system using CMEMS products and A* algorithm. Ocean Engineering 255:111427. https://doi.org/10.1016/j.oceaneng.2022.111427
[58] Vettor R, Guedes Soares C (2016) Development of a ship weather routing system. Ocean Engineering 123:1–14. https://doi.org/10.1016/j.oceaneng.2016.06.035
[59] Vettor R, Szlapczynska J, Szlapczynski R et al. (2020) Towards Improving Optimised Ship Weather Routing. Polish Maritime Research 27:60–69. https://doi.org/10.2478/pomr-2020-0007
[60] Frydenberg S, Nordby K, Eikenes JO (2018) Exploring designs of augmented reality systems for ship bridges in arctic waters. In: Human Factors: RINA - International Conference on Human Factors
[61] Ostendorp M-C, Lenk JC, Lüdtke A (2015) Smart Glasses to Support Maritime Pilots in Harbor Maneuvers. Procedia Manufacturing 3:2840–2847. https://doi.org/10.1016/j.promfg.2015.07.775
[62] Takenaka M, Nishizaki C, Okazaki T (2019) Development of Ship Collision Prevention Device with Augmented Reality Toolkit. In: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). IEEE, pp 4290–4295
[63] Katerina Doka, Nikolaos Papailiou, Dimitrios Tsoumakos, Christos Mantas, Nectarios Koziris: IReS: Intelligent, Multi-Engine Resource Scheduler for Big Data Analytics Workflows. SIGMOD Conference 2015: 1451-1456
[64] Divy Agrawal, Sanjay Chawla, Bertty Contreras-Rojas, Ahmed K. Elmagarmid, Yasser Idris, Zoi Kaoudi, Sebastian Kruse, Ji Lucas, Essam Mansour, Mourad Ouzzani, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Nan Tang, Saravanan Thirumuruganathan, Anis Troudi: RHEEM: Enabling Cross-Platform Data Processing - May The Big Data Be With You! -. Proc. VLDB Endow. 11(11): 1414-1427 (2018)
[65] Kaustubh Beedkar, Bertty Contreras-Rojas, Haralampos Gavriilidis, Zoi Kaoudi, Volker Markl, Rodrigo Pardo-Meza, Jorge-Arnulfo Quiané-Ruiz: Apache Wayang: A Unified Data Analytics Framework. SIGMOD Rec. 52(3): 30-35 (2023)
[66] Ioannis Flouris, Vasiliki Manikaki, Nikos Giatrakos, Antonios Deligiannakis, Minos N. Garofalakis, Michael Mock, Sebastian Bothe, Inna Skarbovsky, Fabiana Fournier, Marko Stajcer, Tomislav Krizan, Jonathan Yom-Tov, Taji Curin: FERARI: A Prototype for Complex Event Processing over Streaming Multi-cloud Platforms. SIGMOD Conference 2016: 2093-2096
[67] Nikos Giatrakos, David Arnu, Theodoros Bitsakis, Antonios Deligiannakis, Minos N. Garofalakis, Ralf Klinkenberg, Aris Konidaris, Antonis Kontaxakis, Yannis Kotidis, Vasilis Samoladas, Alkis Simitsis, George Stamatakis, Fabian Temme, Mate Torok, Edwin Yaqub, Arnau Montagud, Miguel Ponce de Leon, Holger Arndt, Stefan Burkard: INforE: Interactive Cross-platform Analytics for Everyone. CIKM 2020: 3389-3392
[68] George Stamatakis, Antonis Kontaxakis, Alkis Simitsis, Nikos Giatrakos, Antonios Deligiannakis: SheerMP: Optimized Streaming Analytics-as-a-Service over Multi-site and Multi-platform Settings. EDBT 2022: 2:558-2:561

[69] Adrian Michalke, Aljoscha P. Lepping, Volker Markl, Ricardo Martinez, Nils L. Schubert, Lukas Schwerdtfeger, Taha Tekdogan, Steffen Zeuch, Ariane Ziehn, Christoph Falkensteiner, Kyle Krüger, Alexander Meyer, Tobias Röschl, Svea Wilkending: NebulaStream: An Extensible, High-Performance Streaming Engine for Multi-Modal Edge Applications. SIGMOD Conference Companion 2025: 195-198
[70] Alam, F., Mehmood, R., Katib, I., Albogami, N.N., and Albeshri, A. 2017. Data fusion and IoT for smart ubiquitous environments: A survey. Ieee Access 5, 9533–9554.
[71] Ameen, H. and Soilán, M. 2026. A review of large language models in geomatics: integrating multimodal data, addressing challenges, and exploring synergies. Spatial Information Research 34, 2.
[72] El Amine Chafiki, M., Stitini, O., and Kaloun, S. 2025a. A Roadmap for Multimodal Fusion in IoT-Enabled Context-Aware Recommender Systems. 2025 International Conference on Intelligent Systems: Theories and Applications (SITA), 1–8.
[73] El Amine Chafiki, M., Stitini, O., and Kaloun, S. 2025b. A Roadmap for Multimodal Fusion in IoT-Enabled Context-Aware Recommender Systems. 2025 International Conference on Intelligent Systems: Theories and Applications (SITA), 1–8.
[74] Anwar, K., Shreya, Sharma, M., and Saanvi, K. 2026. A review of multimodal sentiment analysis: Taxonomy, issues, challenges, and future perspectives. Computers and Electrical Engineering 131, 110959.
[75] Baltrušaitis, T., Ahuja, C., and Morency, L.-P. 2019. Multimodal Machine Learning: A Survey and Taxonomy. IEEE Transactions on Pattern Analysis and Machine Intelligence 41, 2, 423–443.
[76] Bayoudh, K. 2024. A survey of multimodal hybrid deep learning for computer vision: Architectures, applications, trends, and challenges. Information Fusion 105, 102217.
[77] Caruso, C.M., Soda, P., and Guarrasi, V. 2025. MARIA: A multimodal transformer model for incomplete healthcare data. Computers in Biology and Medicine 196, 110843.
[78] da Costa, T.A.G., Meneguette, R.I., and Ueyama, J. 2022. Providing a greater precision of Situational Awareness of urban floods through Multimodal Fusion. Expert Systems with Applications 188, 115923.
[79] Dai, Y., Yan, Z., Cheng, J., Duan, X., and Wang, G. 2023. Analysis of multimodal data fusion from an information theory perspective. Information Sciences 623, 164–183.
[80] Deng, M., Peng, J., Tang, J., et al. 2025. Spatiotemporal Pattern Mining of Moving Objects: Concepts, Methods, and Challenges. Journal of Geovisualization and Spatial Analysis 9, 2.
[81] Fawzy, D., Moussa, S., and Badr, N. 2021. The Spatiotemporal Data Fusion (STDF) Approach: IoT-Based Data Fusion Using Big Data Analytics. Sensors 21, 21.
[82] Gaw, N., Yousefi, S., and Gahrooei, M.R. 2022a. Multimodal data fusion for systems improvement: A review. IISE Transactions 54, 11, 1098–1116.
[83] Gaw, N., Yousefi, S., and Gahrooei, M.R. 2022b. Multimodal data fusion for systems improvement: A review. IISE Transactions 54, 11, 1098–1116.
[84] Kashinath, S.A., Mostafa, S.A., Mustapha, A., et al. 2021. Review of Data Fusion Methods for Real-Time and Multi-Sensor Traffic Flow Analysis. IEEE Access 9, 51258–51276.
[85] Ko, H., Yoo, J., and Jeong, O.-R. 2025. MDCKE: Multimodal deep-context knowledge extractor that integrates contextual information. Alexandria Engineering Journal 119, 478–492.
[86] Lahat, D., Adali, T., and Jutten, C. 2015. Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects. Proceedings of the IEEE 103, 9, 1449–1477.

[87] Le, H., Lu, C.-K., and Hsu, C.-C. 2026. Video anomaly detection for edge-based IoT systems: A survey of input modalities and real-time applications. Intelligent Systems with Applications 29, 200635.
[88] Liggins II, M., Hall, D., and Llinas, J. 2017. Handbook of multisensor data fusion: theory and practice. CRC press.
[89] Meng, T., Jing, X., Yan, Z., and Pedrycz, W. 2020. A survey on machine learning for data fusion. Information Fusion 57, 115–129.
[90] Mitchell, H.B. 2007. Multi-sensor data fusion: an introduction. Springer Science & Business Media.
[91] Salvi, M., Loh, H.W., Seoni, S., et al. 2024. Multi-modality approaches for medical support systems: A systematic review of the last decade. Information Fusion 103, 102134.
[92] Tan, J., Cheng, M., Chen, J., Zhu, J., Yu, Q., and Chen, S. 2025. Multimodal destination image and user engagement: a sequential research design. Tourism Management 111, 105209.
[93] Wang, E., Xu, H., Zhao, H., et al. 2026a. An adaptive spatio-temporal transformer framework integrating physical multi-layer constraints for high-precision traffic flow prediction and coordination optimization. Expert Systems with Applications 315, 131483.
[94] Wang, E., Xu, H., Zhao, H., et al. 2026b. An adaptive spatio-temporal transformer framework integrating physical multi-layer constraints for high-precision traffic flow prediction and coordination optimization. Expert Systems with Applications 315, 131483.
[95] Wang, M., Zhang, Q., Liu, X., et al. 2026c. Research progress on multimodal data fusion in forest resource monitoring. Frontiers in Plant Science Volume 16-2025.
[96] Wang, Z., Wu, Y., and Niu, Q. 2020. Multi-Sensor Fusion in Automated Driving: A Survey. IEEE Access 8, 2847–2868.
[97] Xiao, S., Gu, H., Shen, D., Niu, Z., Xiao, J., and Yu, F. 2025. A systematic review of social Media-Enabled flood disaster Informatics: Method, technology and application. International Journal of Applied Earth Observation and Geoinformation 144, 104935.
[98] Zhang, P., Wang, D., and Lu, H. 2024. Multi-modal visual tracking: Review and experimental comparison. Computational Visual Media 10, 2, 193–214.
[99] Zhang, T., Ding, J., Liu, Z., and Zhang, W. 2025. Visualization of Industrial Big Data: State-of-the-Art and Future Perspectives. Engineering 52, 85–101.
[100] Zhao, L. 2022. Event Prediction in the Big Data Era: A Systematic Survey. ACM Computing Surveys 54, 5.
[101] Zhou, Y., De, S., Wang, W., and Moessner, K. 2016. Search Techniques for the Web of Things: A Taxonomy and Survey. Sensors 16, 5.
[102] Zhu, X., Cai, F., Tian, J., and Williams, T.K.-A. 2018. Spatiotemporal Fusion of Multisource Remote Sensing Data: Literature Survey, Taxonomy, Principles, Applications, and Future Directions. Remote Sensing 10, 4.
[103] Zong, X., Yan, H., and Qi, Y. 2026a. Recent Advances in Multi-source Data Fusion for Traffic Flow Prediction: A Review. Archives of Computational Methods in Engineering 33, 1, 1181 – 1203.
[104] Zong, X., Yan, H., and Qi, Y. 2026b. Recent Advances in Multi-source Data Fusion for Traffic Flow Prediction: A Review. Archives of Computational Methods in Engineering 33, 1, 1181 – 1203.