EMERALDS

Project Title	Extreme-scale Urban Mobility Data Analytics as a Service
Project Acronym	EMERALDS
Grant Agreement No.	101093051
Start Date of Project	2023-01-01
Duration of Project	36 months
Project Website	https://emeralds-horizon.eu/

D2.1 – EMERALDS Reference Architecture

Work Package	**WP 2, Reference Architecture and Toolset Integration**
Lead Author (Org)	**George Tsakiris (KNT)**
Contributing Author(s) (Org)	**Anita Graser & Anahid Jalali (AIT), Georgios Theodoropoulos & Nikos Koutroumanis & Christos Doulkeridis & Yannis Theodoridis (UPRC), Ignacio Elicegui & Yerhard Lalangui (ATOS), Charlotte Fléchon (PTV), Mahmoud Sakr & Bahare Salehi (ULB), Stathis Antoniou & Foivos Galatoulas (INLE), Panagiotis Ilia & Argyris Papadopoulos (TUC), Argyrios Kyrgiazos (CARTO), Mattia Pretti & Luca Paone (SiSTeMa), Antonis Mygiakis (KNT)**
Due Date	**30.09.2023**
Date	**22.09.2023**
Version	**V1.0**

Dissemination Level

X PU: Public

This project has received funding from the Horizon Europe R&I programme under the GA No. 101093051

Versioning and contribution history

Version	Date	Author	Notes &/or Reason
0.1	08/03/2023	George Tsakiris (KNT)	ToC and V0.1
0.2	18/08/2023	All editors and contributors	Input in Sections 3, 4, 5
0.3	07/09/2023	George Tsakiris (KNT)	Initial Draft Version
0.4	11/09/2023	All editors and contributors	1st complete draft
0.7	22/09/2023	KNT Team	2nd complete draft
1.0	26/09/2023	KNT Team	Final Version

Quality Control (includes peer & quality reviewing)

Version	Date	Name (Organisation)	Role & Scope
0.5	15/09/2023	Ignacio Elicegui (ATOS)	First Reviewer Comments
0.6	15/09/2023	Dani Baldo, Argyris Kyrgiazos (CARTO)	Second Reviewer Comments
0.8	25/09/2023	Javier De La Torre, Dani Baldo (CARTO), Ignacio Elicegui (ATOS)	Final check and approval by the IRs
0.9	25/09/2023	Yannis Theodoridis (UPRC)	Scientific and Technical Manager Review and approval
1.0	29/09/2023	Ioanna Fergadiotou, Foivos Galatoulas (INLE)	Final review by the PC

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101093051

Table of Contents

List of Figures

List of Tables

Terminology

Terminology/Acronym	Description
2D/3D	2 Dimensions / 3 Dimensions
AI/ XAI	Artificial Intelligence/ Explainable AI
API	Application Programming Interface
BDVA	Big Data Value Association
CC	Computing Continuum
CI/CD	Continuous Integration/ Continuous Deployment
DevOps	Development and Operations
DoA	Description of Action
EC	European Commission
ECMWF	European Centre for Medium-Range Weather Forecasts
FCD	Floating Car Data
FL	Federated Learning
GDPR	General Data Protection Regulation
GPS	Global Positioning Systems
GPU	Graphics Processing Unit
IDS	Intrusion Detection System
IDSA	International Data Spaces Association
KPI	Key Performance Indicator
MAaaS	Mobility Analytics as a Service
MAE	Mean Absolute Error
MAIaaS	Mobility Artificial Intelligence as a Service
MAPE	Mean Absolute Percentage Error
MDA	Mobility Data Analytics
ML/AL	Machine Learning/ Active Learning
MLOps	Machine Learning Operations
MS	Milestone
MSE	Mean Square Error
NN/ GNN	Neural Networks/ Graph Neural Networks
OGC	Open Geospatial Consortium

OS	Operating System
OSM	Open Street Maps
PoC	Proof of Concept
PSNR	Peak Signal-to-Noise Ratio
RA	Reference Architecture
RBAC	Role Base Access Control
RIA	Research and Innovation action
RRMSE	Relative Root Mean Square Error
SecOps	Security Operations
SotA	State-of-the-Art
TEE	Trusted Execution Environment
TRL	Technology Readiness Level
UC	Use Case
UX	User Experience
VA	Visual Analytics
VPN	Virtual Private Network
WP	Work Package

Table 1 - Terminology

GA Matrix of alignment

Table 2 outlines the outputs of D2.1 mapped to the GA commitments as stated in the Description of Action (DoA) Annex 1 and Annex 2.

GA Components Title (and type)	GA Component Outline	Document Chapter(s)	Justification
Deliverable			
D2.1 EMERALDS Reference Architecture	EMERALDS Toolset reference architecture that utilises dedicated components from partners and open software, and distributed systems including edge/fog nodes and cloud nodes with the aim to deliver the EMERALDS services and their respective configuration towards establishing robust extreme mobility data analytics pipelines. The report will also include	**Chapters 2, 3, 4, 5, 6, 7**	In this document the EMERALDS Toolset Reference Architecture is described and analysed with respect to its constituents. Chapter 2 provides an overview of the architecture, guiding the reader from the ideation and conceptual architecture phases to the consolidated version taking into account the technical challenges, functional and non-functional requirements, and

	information on functional requirements, technical development aspects and definition of technical KPIs.		opted functionalities of the extreme scale urban mobility analytics tools developed within the frame of the project. Chapter 3 elaborates the design specification per sub-component of the toolset. Chapter 4 presents Visual Analytics and Dashboards architectural considerations. Chapter 5 outlines the MAIaaS platform. Chapter 6 identifies the specifications compliance with established BDA architectures. Chapter 7 draws the learnings from the reference architecture design.
Tasks			
Task 2.1 Reference Architecture and Containerization of Services	Task 2.1 will describe project's specifications and design a distributed reference architecture across the computing continuum for achieving and enabling Extreme Data Analytics at all project levels. The identified specifications will result in consolidating the technological requirements to be adopted in the toolset architecture with respect to criteria such as performance, accuracy, usability, intended analytics service (diagnostic, descriptive, predictive, or prescriptive), urban mobility data specifications and integration of cloud/fog/edge infrastructure.	**Chapter 2**	Chapter 2 provides the rationale and motivation behind the EMERALDS Reference Architecture by analysing requirements identified within the EMERALDS Project. The RA is introduced and compared to the SotA. Additionally, there are sections that are analysing various aspects of the EMERALDS toolset, such DevOps best practices that are followed by the project partners, KPIs aggregated table and a report on how 'emerald' services are coping with Extreme scale mobility data.
	Following this, a definition and preliminary analysis of functional and non- functional requirements will be provided.	**Chapters 2, 3, 4, 5**	Chapters 2, 3, 4, 5 present the revised reference architecture, along with functional and non-functional requirements of the toolset and breakdown per component.
	Compliance with reference architectures and guidelines from BDVA and International Data Standards Organizations (NIST, CEN, CENELEC, ETSI) as well as links to State-of-the-art BDA, High Performance Data Analytics and Edge-to-Fog-to-Cloud reference architectures will be assessed in the view of	**Chapters 6, 7**	Chapter 6 examines the compliance of the EMERALDS RA with major reference architectures such as BDVA, IDS and Gaia-X. Chapter 7 highlights potential standardizations efforts within the Mobility Data Analytics Domain.

	contributing to the efforts towards a Common European Data Space.		
	EMERALDS will undergo a bottom-up approach to clearly identify how each of the EMERALDS toolset components and their mutual interactions, map to and address the requirements of the use case partners. Interoperability with the AI4EU AI on Demand platform will be pursued, while the delivered AI/ML methods, processing throughout the computing continuum tools, Data Management and Distribution tools, and visual analytics dashboards opt to achieve high configurability, user-friendliness, modularity, high scalability, and adaptiveness to infrastructure. The architecture also aims to enable efficient and flexible access to data, by eliminating long data-to-query times, supporting cross-format queries and dynamic data workloads.	**Chapters 3, 4, 5**	Chapter 3 provides an in-depth analysis of all 'emerald' services and software components, their individual requirements, architecture, and KPIs. Chapter 4 presents the Visual analytics and Dashboard services, mainly related to the CARTO Solution. Chapter 5 is focused on the analysis of the Mobility AI-as-a-Service platform, that will be developed as part of the project.

Table 2 - Matrix of Alignment

Executive Summary

The objective of this deliverable is to present the Reference Architecture for the EMERALDS project, outlining the various components and tools, and illustrating how they are integrated into a unified EMERALDS toolset that can be merged into commercial and open-source platforms, thereby supporting an as a Service functionality across different tiers of the computing continuum. The key aspects covered in this deliverable include reference architecture design, functional and non-functional requirements, infrastructure/resource provisions, components' high-level descriptions and technical KPIs.

EMERALDS's vision is to design, develop and create an **urban data-oriented Mobility Analytics as a Service (MAaaS) toolset**, consisting of the proclaimed **EMERALDS services**, compiled in a proof-of-concept prototype, capable of exploiting the untapped potential of extreme urban mobility data. The toolset will enable the stakeholders of the urban mobility ecosystem to collect and manage ubiquitous spatio-temporal data of high-volume, high-velocity and of high-variety, analyse them both in online and offline settings, import them to real-time responsive AI/ML algorithms and visualise results in interactive dashboards, whilst implementing privacy preservation techniques at all data modalities and at all levels of a data workflow architecture. The toolset will offer advanced capabilities in data mining (searching and processing) of large amounts and varieties of urban mobility data.

In the process of developing the EMERALDS reference architecture, a thorough analysis was conducted by reviewing the D5.1 – Use Cases Scoping Document. The intertwined study of both project research streams, the technical specification and Toolset architecture D2.1 and the business requirements and testing, validation environments in D5.1, constitute the key means of verification for the achievement of Milestone 1, M9: **Reference Architecture & Performance KPIs**. The assessment aimed to extract valuable insights into the diverse range of functional and non-functional end-user requirements that have been collected. This analysis served as a foundation for the shaping and the designing of each 'emerald' service and the overall reference architecture.

The need to exploit extreme urban mobility data and offer analytics on spatio-temporal data of high-volume, high-velocity and of high-variety required special design decisions on the architecture of the EMERALDS Toolset. To tackle such challenges, the project enshrines the deployment of EMERALDS services across the computer continuum, the scalability of the offered services and the use of a federated learning (FL) approach that would maximize the utilization of edge and fog devices while reducing the bandwidth requirements for data transfers.

The deliverable presents a detailed list of 'emeralds', organized into different groups based on a taxonomy serving the project's research goals in the fields of extreme scale data mining, filtering, aggregation and analytics. The following EMERALDS typologies have been identified: 1) Privacy-aware in situ Data Harvesting, 2) Data Fusion and Management, 3) Extreme-Scale Cloud and Fog Data Processing, 4) Extreme Scale Mobility Data Analytics at the Computing Continuum, 5) Active & Federated Learning over Mobility Data and 6) Security and Data Governance. Additionally, the deliverable presents the layout of a Mobility AI-as-a-Service (MAIaaS) platform that serves as a development hub for the EMERALDS services as well as a model inference orchestrator.

The RA adheres to the guidelines from BDVA, GAIA-X and the International Data Spaces Association (IDSA) to realize the concept of a Common European Data Space. It incorporates industry requirements, security aspects, interoperability, and data governance. Importantly, the RA serves as a blueprint for the project, outlining essential components and technical offerings for achieving EMERALDS objectives.

A mapping of the EMERALDS reference architecture to the relevant Big Data, AI and Computer Continuum architectures is carried out, establishing a connection to efficiently account for further developments in the relevant project areas and future EU strategic agendas.

1 Introduction

1.1 Purpose and scope of the document

This document aims to provide a comprehensive overview of the EMERALDS conceptual architecture's refined specification. The reference architecture is based on an in-depth comprehension of the underlying technologies sourcing from ongoing research activities, an up-to-date review of the State-of-the-Art (SotA) in the relevant project areas, adherence to relevant reference architectures and models, and end-user requirements. The deliverable outlines the role, functionality, technology stack, and interconnections of each 'emeralds' component, consolidating the technical offering in a unified EMERALDS Toolset.

The Toolset utilizes dedicated components from partners and open software, and distributed systems including edge/fog nodes and cloud nodes. Moreover, D2.1 elaborates on the functional requirements, technical development aspects and the definitions of technical KPIs.

Workflows permitting the establishment of robust extreme mobility data analytics pipelines across the computing continuum are supported based on varying configurations of individual 'emeralds'/ services and possible interconnections amongst them. Consequently, urban mobility data analytics problems are benefited through the demonstration of the enhanced capabilities offered by the toolset services and targeted experimentation in Use Cases environments can be facilitated through specified integration points and technical considerations. Last, D2.1 represents the high-level technical research stream of EMERALDS which is combined with the mobility-domain expertise research stream encapsulated in D5.1 with their intertwining evidently paving the ground for the following stages of the implementation plan as well as concluding the work plan with the achievement of MS1.

1.2 Relation to Work Packages, Deliverables and Activities

The EMERALDS Reference Architecture serves as the backbone of the envisioned technological developments within the frame of the project and constitutes the lighthouse research output of WP2. In this direction, D2.1 comprises a robust foundation for the development of a versatile palette of services, known as 'emeralds.' Architectural design is pursued with the aim to facilitate their seamless integration into a holistic urban mobility data analytics ecosystem. In the scope of aligning the design specification of the Toolset with the real-world challenges addressed by the use cases, this document is directly tied to the analysis of the deliverable D5.1 - Use Cases Scoping Document in WP5.

The present deliverable comprises an analysis of all activities that have been carried out within WP2 towards the design of the EMERALDS Toolset reference architecture, the definition of technical characteristics, functional and non-functional requirements and KPIs to assess and quantify the performance metrics of each software component constituting the toolset. It also marks the "means of verification" of **MS1: Reference Architecture & Performance KPIs**, which is related to WP2, and WP5. Therefore, this deliverable has a significant contribution to all the technical WPs, deliverables, and activities of the project. It sets the stage for most of the technical deliverables and the related activities to be implemented.

Moreover, the architecture takes into account technical KPIs, documented in dedicated deliverables (D3.1, D4.1 for core software modules and D5.1-D5.7 for use case validation), as guiding metrics throughout the development process. The analysis of the defined use case KPIs, key resources and datasets was used to identify the functional requirements of the EMERALDS services to be developed.

The output of this document shall be used as a guideline for the tasks and deliverables of the following work packages:

- **WP3 - Mobility Data Processing at the Computing Continuum**, which aims at creating a set of data processing and management tools that can be used to enable and facilitate the creation of useful data analytics methods.
- **WP4 - Extreme Scale Mobility Data Analytics & Machine Learning,** which is focused on the development of mobility data analytics and AI tools for the mobility as a service toolset that build on the data processing services developed in WP3 to enable the use cases in WP5.

Additionally, the D2.2 and D2.3 Containerized EMERALDS Toolset v1 and v2 are directly connected to the Reference architecture of the EMERALDS toolset as it will be defined in this document.

Finally, this document is directly related to **D2.6 - Security and Data Governance Layer**, a crucial domain regarding the security of the data on the move and at rest.

The project planning undergoes two main iterations: the first implementation cycle that spans until M24, when the first version of the EMERALDS services will be available and tested by the Use Cases, while the second implementation cycle focusing on refinements and addressing any new or modified business requirements. A detailed plan of the implementation activities is depicted in Figure 1-1.

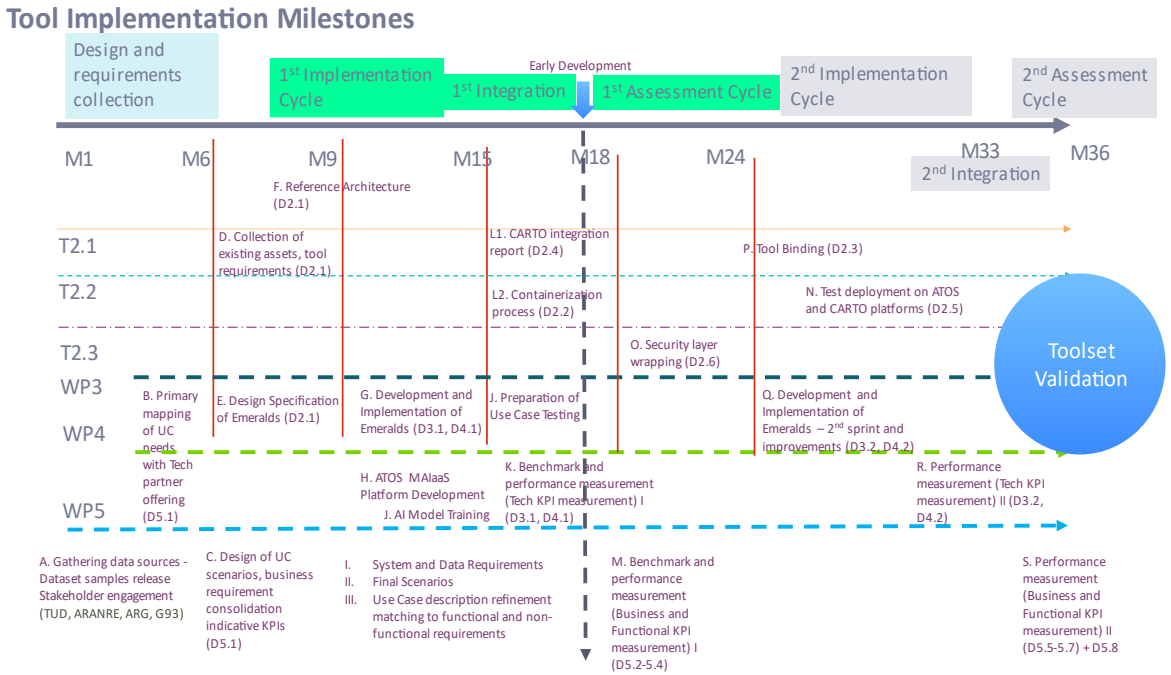

Figure 1-1 - EMERALDS Toolset Implementation Plan

1.3 Contribution to WP2 and Project Objectives

This document is the key output of **T2.1 – 'Reference Architecture and Containerization of Services'** of WP2 of the Description of Action (DoA). As the primary task undertaken in WP2, T2.1 contributes to the fulfilment of Project **Objective 1 (O1)**, **Design a service-oriented reference architecture of a palette of services ('emeralds') for extreme scale urban mobility data analytics,** underpinned by a distributed computing environment utilizing edge/fog nodes and cloud nodes, also ensuring that both edge and cloud processing contribute towards establishing a robust processing pipeline. Furthermore, **O1** emphasizes the need for the architecture to be tailored to the requirements of extreme data processing along the computing continuum. This is achieved through tapping into potential of unexplored computational resources and capabilities of distributed computing environments, as well as the introduction of novel algorithmic methods on data acquisition, filtering, management, and analytics.

The reference architecture provides the foundation for the technical implementation of the project, permitting multi-platform awareness and addressing essential aspects like interoperability, scalability, security, and data governance. Hence, **D2.1 supports developments foreseen and addressed within all project WPs and Objectives (O1-O5)**. In addition, interoperability is investigated in the scope of enabling the combined execution of EMERALDS services, and to a broader extent of the EMERALDS toolset with two established mobility data analytics-as-a-service platforms (one developed and operated by ATOS and a commercial cloud platform operated by CARTO).

The architecture's design considers the technical KPIs, which are instrumental in assessing the architecture's effectiveness and suitability for extreme data analytics. Activities within WP2 directly contribute to shaping the architecture in line with these functional, non-functional requirements and technical KPIs.

1.4 Structure of the document

The structure of the document is as follows:

- Chapter 2 provides the rationale and motivation behind the EMERALDS Reference Architecture by analysing requirements identified within the EMERALDS Project. It presents the revised reference architecture, along with functional and non-functional requirements of the toolset. Additionally, there are sections that are analysing various aspects of the EMERALDS toolset, such DevOps best practices that are followed by the project partners, KPIs aggregated table and a report on how 'emerald' services are coping with Extreme scale mobility data. This chapter reflects on challenges addressed on WP2.
- Chapter 3 provides an in-depth analysis of all 'emerald' services and software components, their individual requirements, architecture, and KPIs, according to the plan activities of WP3, WP4 and part of WP2 (T2.3)
- Chapter 4 presents the Visual analytics and Dashboard services, mainly related to the CARTO Solution, which are part of T2.1, T2.2 of WP2.
- Chapter 5 is focused on the analysis of the Mobility AI-as-a-Service platform, that will be developed as part of the project. Chapter 5 is linked to WP4, where the MAIaaS platform is outlined.
- Chapter 6 presents the relevance and compliance of the EMERALDS project with major reference architectures such as BDVA, IDS and Gaia-X.
- Chapter 7 draws the key takeouts from the Reference Architecture Design and concludes the document.

2 Overview of EMERALDS Reference Architecture

In the rapidly evolving landscape of modern interconnected mobility systems, the generation of vast amounts of data has become an unparalleled reality. From the bustling streets of urban centers to the intricate networks of transportation, every move, every transaction, and every interaction contributes to an avalanche of data that holds unprecedented potential. However, this wealth of data, while holding the promise of transforming how we envision and manage urban mobility, also poses an intricate challenge. The need to decipher this data deluge, to extract meaningful insights, and to translate them into actionable strategies has emerged as a critical imperative for a sustainable and viable urban future.

EMERALDS project envisions the creation of a versatile suite of specialized software modules, encompassing containerized versions of the tools and software stacks developed and showcased within the project (WP5, including 3 Use Cases and 2 Early Adoption Demonstrators). This collection of tools will be consolidated into a proof-of-concept prototype known as the urban data-oriented Mobility Analytics as a Service (MAaaS) toolset (**T2.1**). The primary objective of this toolset is to harness the untapped potential embedded in extreme urban mobility data. This endeavour is driven by the aspiration to provide solutions that underpin data-driven real-time applications and informed decision-making processes, such as intelligent traffic management, crowd management and public transport planning.

After the seamless integration of these tools into the EMERALDS MAaaS toolset, a two-fold deployment strategy will be executed on distinct cloud-based platforms, each catering to deliver solution for specific user segments. The first platform (ATOS Mobility AI as a Service) will cater to urban data scientists and technically proficient users (**T4.3**), constituting an open research environment aligned with the EU AI on Demand ecosystem. This ecosystem will not only facilitate the reproducibility of methodological advancements but also foster the open accessibility of cutting-edge technologies. The second platform (CARTO) will extend its scope to encompass a broader user base (**T2.2**). It will be optimized to accommodate extreme scale cloud-native data management, robust caching mechanisms, immersive visualizations, and intuitive dashboards. This deployment environment will effectively serve the diverse requirements of users with varying degrees of technical proficiency, leveraging the advanced capabilities of the CARTO platform.

To enable the seamless execution of analytics methodologies, these methods will be encapsulated as services, colloquially referred to as 'emeralds.' Each service will be neatly categorized within the corresponding tool category and configured to operate seamlessly with other services, following the release of software developments carried out in WP2, WP3 and WP4. This encapsulation of services will be facilitated by the containerization of services (**T2.1**), ensuring efficiency, scalability, and ease of deployment. A primary definition of an 'emerald' software component can be summarized as follows:

An **'emerald'** in the context of the EMERALDS project refers to a specialized software module or service that is developed and integrated within the Mobility Analytics as a Service (MAaaS) toolset. **'emeralds'** are designed to encapsulate advanced analytics methods, algorithms, and tools that focus on extracting meaningful insights and intelligence from extreme-scale urban mobility data. When '**emeralds'** combine their added functionalities to craft innovative tools, they constitute EMERALDS **Toolset**. Each '**emerald's'** unique offering complements others, creating a harmonious synergy that unlocks new capabilities of urban mobility data analytics in the form of EMERALDS **services**.

2.1 Project Specification

The conceptual architecture of the EMERALDS project is presented in Figure 2-1. In this diagram, the interconnection between research activities in WP level is established and the foundations for the toolset components are clearly laid out, along with their respective data exchanges and key cross-cutting concerns, such as privacy-awareness, data protection, cyber security. The goal of this design is to permit seamless and secure edge to cloud extreme scale mobility data analytics. In Section 2.3 of this document, the process of specifying the conceptual architecture to a reference architecture that can be instantiated across the use cases and early adoption demonstrators, meantime serving their respective objectives and envisioned functionalities is presented.

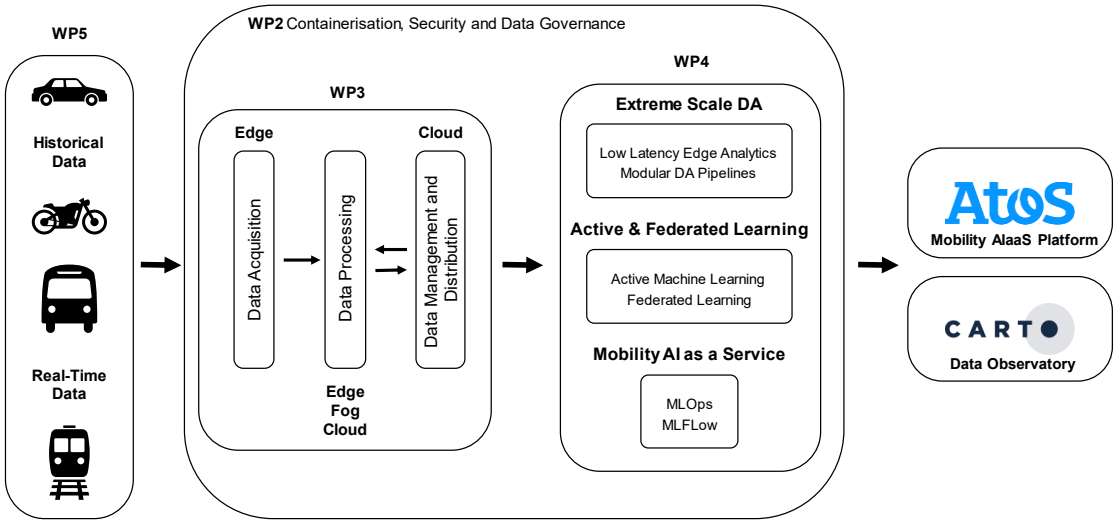

Figure 2-1 - EMERALDS Conceptual Architecture

In the frame of T2.1, the EMERALDS reference architecture is based on project specifications, as derived in the process of designing the architecture and collecting feedback horizontally from technical and use case partners. On the one hand, technical partners were engaged throughout this process, providing feedback and design specifications for the 'emeralds' components, developed within their respective tasks (WP3, WP4). On the other hand, use case partners were paired with technical partners based on the use case requirements (as recorded in D5.1 and further elicited in D5.2, D5.4, and D5.6) and mapping of 'emeralds' to the use cases' objectives.

The EMERALDS reference architecture serves as a critical backbone for the use cases that ensures every component fits together seamlessly and delivers their intended functionalities. It provides a standardized framework that allows different software modules and tools developed for the use cases to communicate and integrate effectively. Each use case can follow consistent design principles and practices, utilizing a valuable asset for future collaborations and innovations beyond the project's scope.

In this regard, the reference architecture follows the project specifications as defined in the grant agreement document:

- Design a service-oriented reference architecture of a palette of services ('emeralds') for extreme scale urban mobility data analytics, underpinned by a distributed computing environment that includes edge, fog and cloud nodes, that contribute towards establishing a robust processing pipeline.
- Develop a multi-platform aware architecture, including interoperability, scalability, security, and data governance specifications, and considering the technical KPIs documented in the respective deliverable.
- Ensure that the architecture is capable of handling extreme data scale analytics,
- Consider key cross-cutting concerns such as privacy-awareness, data protection, cyber security, and data standards. Enable secure and seamless cloud to edge extreme scale mobility analytics.
- Develop security wrappers for data manipulation across the Computing Continuum (CC). Utilize Trusted Execution Environments (TEE) for secure execution of routines.
- 'emeralds' of type (processing, cleansing, compression, analytics, AI /Explainable AI and FL)

EMERALDS use cases are discerned in the following categories: a) real-time analytics tasks, b) short term forecasting based on a predefined time horizon, and c) individual/ crowd mobility analytics and d) traffic-related analytics. All use cases addressing human mobility, either as individuals or crowds, avoid the use of personal data throughout the development, testing and validation processes.

In the context of EMERALDS, the following **Key Design Considerations** are addressed pertinent to intrinsic aspects of Extreme Data:

Extreme Data Volume and Velocity: Dealing with the sheer volume of data generated by urban mobility sources, such as GPS trajectories, poses challenges in terms of storage, processing, and real-time analysis. This is assigned to the tasks relevant to data management and processing.

Extreme Data Variety and Heterogeneity: Urban mobility data comes in various formats and from diverse sources, such as GPS sensors, social media, and traffic cameras. Integrating and processing these heterogeneous data types requires advanced data fusion and integration techniques.

Real-time Processing Capabilities: Processing and analyzing data streams in real-time to enable timely decision-making presents challenges in terms of low-latency processing and handling dynamic data streams.

Privacy and Security Preservation: Urban mobility data often contains sensitive information, necessitating the implementation of robust privacy-preserving mechanisms while still enabling effective analysis.

Improved Data Quality: Guaranteeing consistent quality and accuracy of data is challenging, particularly when dealing with data from various sources that present inconsistencies, errors or format issues. Main data quality characteristics are harmonized with the ISO/IEC 25012 Data Quality model.

Enhanced Resource Efficiency: Efficiently utilizing computing resources across edge, fog, and cloud nodes, requires intelligent resource allocation and management.

Extreme Data Analytics and Machine Learning: Developing accurate and efficient machine learning models that can handle the complexity and scale of urban mobility data, particularly when predicting events, patterns, and risks.

Visual Analytics: Designing effective visualizations that convey insights from complex mobility data to different types of users, ranging from technical experts to decision-makers, is a challenge.

A mapping of the Key Design Considerations to the EMERALDS Technical Activities foreseen within the work plan is set out in Table 3

Table 3 - Extreme Data Key Design Considerations mapped to EMERALDS technical activities.

Design Considerations	Elaborated in
Extreme Data Volume and Velocity	T3.1, T3.2, T3.3
Extreme Data Variety and Heterogeneity	WP3, T4.1, T4.2
Real-time Processing Capabilities	T3.1, T4.2, T4.3
Privacy and Security Preservation	T2.3, T3.2, T4.1
Improved Data Quality	T4.2, T3.3
Enhanced Resource Efficiency	T2.1, T4.1, T4.3
Extreme Data Analytics and Machine Learning	T4.2, T4.3
Visual Analytics	T4.1, T2.2

The aforementioned design considerations span across the areas of data acquisition, management, processing, filtering privacy, analytics and visualization. In Chapter 3 technical specifications of the 'emeralds' components are delivered, showcasing their efficacy in providing extreme data solutions in the areas of better technologies, tools and solutions for data mining (searching and processing) of large, constantly growing amounts and varieties of data, and/or extremely sparse/dispersed/heterogeneous/multilingual data (stored centrally or in distributed/decentralized systems), in particular IoT and urban mobility data.

In this regard, the project aims to deliver specified tools based on the following categorization:

A-1. **Privacy-aware in situ Data Harvesting** consist of the "Privacy aware data ingestion" for privacy-ensuring techniques for data collection and ingestion, and the "Extreme scale Stream processing" for data stream broker and remote tasks orchestrator.
This group of services is the result of **T3.1** and is mainly deployed on edge nodes, even though not restricted from the other areas of the CC.
A-2. Extreme-scale Cloud/Fog Data Processing consist of "Extreme-scale map-matching" which offers map-matching of GPS coordinates to an underlying road network, "Weather enrichment" for mobility data with external weather data sources, "Spatio-temporal querying" for efficiently querying over large collections of spatio-temporal datasets and "Hot-spot analysis" for discovering of hot-spots in an urban context and is the outcome of **T3.2**.
This group of services is able to handle massive amounts of mobility data by taking advantage of the two design decisions, the capable to horizontally scale as it is based on Apache Spark framework and by revisiting the problem of processing of urban mobility data under constrained movement, as dictated by the underlying road network.
A-3. **Mobility Data Fusion and Management** consist of "Mobility/trajectory data compression" for optimized storage and data management for urban trajectory data, "Sensor data fusion" for integrating data that originate from different sources and "Traffic State Estimation" offering means of interpolating the missing traffic information towards a better coverage, produced by **T3.3.**
B-1. **Extreme Scale Mobility Data Analytics at the Compute Continuum** consists of "Trajectory/Route Forecasting and Origin/Destination Estimation" for providing a real-time route evaluation framework for event/incident prediction and route forecasting, "Probabilistic approach for trip chaining" for estimating which route probabilities are more likely to produce current data, "Trajectory data / travel time analysis" which offers trajectory analytics based on the open source Python library

MovingPandas and “Extreme Scale Map Matching” which offers map-matching for FCD to retrieve trajectories in order to estimate speeds on a given network. The software components are the outcomes of **T4.1**.

B-2. **Active and Federated Learning over Mobility Data** consist of “Traffic state / flow forecasting” for traffic state prediction using Neural Networks, “Crowd density forecasting” using Graph Neural Network, “Parking garage occupancy forecasting”, “Active Learning & XAI for crowd/flow forecasting”, “Active Learning model for risk classification” and “Federated Learning models for mobility data” for federated learning from data that cannot be shared between organizations. Implementation is undertaken in **T4.2**.

B-3. **Mobility AI-as-a-Service** consists of a family of services related to data ingestion, ML experimentation, training, testing and monitoring modules, an ML model registry/repository and finally the federated learning module. These services shall be used mainly as a development platform for other “intelligent” emeralds and secondary as an inference hub for the trained models, thus acting as MLOps platform (**T4.3**).

In addition, as an horizontal feature, the. **Security and Data Governance Layer** consists of “Trust-Execution Environment” for providing a secure and isolated environment within a computing device, “Secure Communication channels” for securing data pipelines across the CC and the “Intrusion Detection in specialized Hardware” for identifying intrusion attempts on the execution nodes and provide adequate countermeasures to eliminate the thread. This group of emeralds is focused on applying the state-of-the-art security best practices primarily on edge nodes, whose hardware resources are limited. The development of lightweight tools and library on the security domain is prioritized and is the main result of **T2.3**.

2.2 Functional and Non-Functional Requirements

Requirements analysis plays a pivotal role in the EMERALDS project, as it is a critical process that enables the assessment of the project's success. In this section, the *architectural requirements* of the EMERALDS toolset are outlined along with the positioning of individual components across layers of the reference architecture. The way combinations of distinct ‘emeralds’ address the requirements is explained in Chapter 3.

Requirements are generally split into two types: *Functional* and *Non-functional*:

- Functional requirements pertain to the specific functionalities and features that the ‘emeralds’ must present to fulfill their intended purposes. They outline the actions and behavior that the services should exhibit, defining **what** the ‘emeralds’ can achieve concerning urban mobility data analytics.
- Non-functional requirements encompass a diverse set of quality attributes, addressing various aspects of the toolset such as performance, scalability, reusability and security to name a few.

Matching functionalities of the ‘emeralds’ to the Use Case Business requirements is performed through a series of dedicated workshops between the use case owners and the Project’s Technical Partners organized under WP5, within the activities of T5.1 Use Cases Orchestration & Validation and for detailed planning within the respective use case tasks (T5.2, T5.3, T5.4).

2.2.1 Non-Functional

One of the paramount requirements of the EMERALDS project and key aspect for the success of the project is the **Performance**. Emphasis is placed on optimizing the system's speed and resource utilization during operation. This becomes particularly critical for use cases where real-time data should be processed in a timely manner, data-intensive tasks necessitate execution within clustered

nodes across Computing Continuum, and data processing on edge nodes where the hardware resources are limited. The end-to-end measurement of service improvements upon individual or combined stages of data workflows will be evaluated in terms of performance and speed, through testing, measurement and validation procedures planned as part of WP5 (approach introduced in Chapter 2 of D5.1).

The performance requirement is intrinsically linked to the **scalability** of the EMERALDS toolset. Scalability refers to the system's ability to accommodate growth and expansion without compromising its performance, efficiency, or correctness of the outcome. The architecture should be capable of handling extreme data scale, not exhaustively pertaining volume but also extreme sparsity, heterogeneity of sources as well as formats, variety, and distributed data streams. Each 'emerald' service should be able to scale, preferable horizontally, to adapt to higher data volumes or complex tasks. Of course, the efficient resource utilization from a system using the 'emerald' services requires the allocation of the required resources to maintain the desired level of performance, a pre-requisite that only can be met on Cloud environments. Other technique may also apply to meet the scalability requirements, such as load distribution either on swarms of edge devices or on HPC clusters. It is obvious that the sufficient usage of resources across the compute continuum is imperative for meeting the performance expectations.

Extending the concept of scalability, **elasticity** may also be considered an important requirement for the 'emerald' services. Elasticity specifically focuses on the system's ability to adapt its resources automatically and dynamically in response to changes in demand. This allows for the optimal utilization of the platform hardware resources.

In the fields of extreme scale urban mobility data analytics, the **accuracy** is equally important with the performance of a tool. A highly accurate service provides reliable and precise outcomes, generating insights that closely align with the expected values. Ensuring accuracy often involves using more sophisticated algorithms, refining data quality, and reducing potential sources of errors. But a highly accurate 'emerald' may lead to increased computational complexity and longer processing times, which impacts the overall performance. Finding the right balance between performance and accuracy is a critical aspect of the EMERALDS toolset design.

One of the main non-functional requirements foreseen within the EMERLADS toolset is the ability to integrate to multiple and diverse platforms, as it will also be demonstrated in project's Use Cases. Therefore, **Re-usability** and **Interoperability** build up another set of requirements that needs to be considered as part of the development process. Reusable 'emerald' services can be leveraged across multiple use cases and scenarios without the need for significant modifications. Additionally using open standards (for instance the Open Geospatial Consortium standards) and common protocols, the toolset may facilitate seamless integration with any system thereby enabling data flow between heterogeneous platforms. These are core requirements for the development of Urban Mobility Data Analytics as a Service platforms.

As EMERALDS is expected to help in the design and operation of data flows and data analytics across the compute continuum, **security** is a cornerstone requirement for the acceptance of the services by any partner or service provider. Security addresses various aspects related to safeguarding the system and its data. Some key security considerations include:

- **Encryption:** Utilize encryption techniques to protect sensitive data at rest as well as on the move. For the second part, the use of secure communication protocols (e.g., SSL, HTTPS) during data transfer between different nodes across the compute continuum is a strong prerequisite.
- **Access Control**: Implementing policies to restrict access to different data resources based on user roles and permissions. As part of the access control mechanism, the Authentication and

Authorization of the MAaaS platforms should be considered. Additionally, implementing an auditing and logging mechanism allows the system administrators to track system activities, detect intrusion attempts and facilitate incident investigations.

- **Compliance**: Ensuring that the system adheres to relevant security regulations, industry standards, and best practices. A fundamental aspect of compliance, particularly concerning data protection, lies in adhering to the stringent GDPR regulations. To bolster data protection, crucial actions include in-situ data anonymization and cleansing, running analytics services on data owners' infrastructure, and facilitating federated learning to minimize data exposure.

2.2.2 Functional

The EMERALDS Project specifications mandate the creation of a toolset capable of operating seamlessly across the compute continuum, integrated into multiple Mobility Analytics as a Service (MAaaS) platforms, and efficiently managing with data of great volume and variety. As such the project aims to encompass numerous use cases with mobility as their common theme. A comprehensive analysis of such cases is documented in the D5.1 'Use Case Scoping Document', which delves into the specific requirements of the project's Use Case partners. Each 'emerald' service will either address select user stories mentioned earlier or delve into cutting-edge research areas within mobility analytics, with the goal to surpass existing state-of-the-art tools or set the threshold for new ones. Consequently, these services will offer distinct functionalities and features. Nevertheless, all services would adhere to the global requirements as a unified roadmap for their own development process.

The primary objective is to architect a **service-oriented reference framework** consisting of a diverse palette of services, referred to as 'emeralds,' dedicated to facilitating extreme scale urban mobility data analytics. This architecture will be supported by a distributed computing environment encompassing edge, fog, and cloud nodes, ensuring that processing capabilities from all the compute continuums contribute to establish a robust processing pipeline.

The architecture of the EMERALDS toolset will demonstrate **awareness and adaptability across multiple platforms**. This entails addressing critical aspects such as interoperability, scalability, security, and data governance through well-defined specifications. Technical Key Performance Indicators (KPIs), documented in the respective deliverable, will be considered during the development process.

The EMERALDS project goal is to formulate a solution for **handling extreme scale mobility data Operations** across the Compute Continuum (CC), by leveraging distributed data resources. This involves addressing challenges in data acquisition, processing, fusion, and management. The focus is on developing advanced tools with the capability to handle massive data volumes, with extreme sparsity, heterogeneity in data sources and formats, and diverse distributed data streams.

Table 4 presents a comprehensive overview of the functional requirements that constitute the foundation of the EMERALDS architecture. These requirements encompass a wide range of critical aspects, ensuring the successful realization of the service-oriented reference framework for extreme scale urban mobility data analytics. Each requirement is strategically aligned with the project objectives, such as scalability, security, and advanced data processing capabilities, while also acknowledging the importance of adhering to technical KPIs for performance evaluation.

Table 4 - EMERALDS Functional Requirements

FR #	Functional Requirement	Description	EMERALDS Category
FR1	**Service-Oriented Framework**	Provide a modular and flexible service-oriented framework where a range of services (emeralds) can be integrated to facilitate extreme scale urban mobility data analytics.	Privacy-aware in situ Data Harvesting, Extreme Scale MDA at the CC, Mobility AI-as-a-Service
FR2	**Distributed Computing Environment**	Seamlessly incorporate edge, fog, and cloud nodes into the computing environment, enabling efficient processing and data flow across different compute continuums.	Privacy-aware in situ Data Harvesting, Extreme Scale MDA at the CC, Active & Federated Learning over Mobility Data
FR3	**Robust Processing Pipeline**	Ensure that processing capabilities from all compute continuums collaboratively contribute to the establishment of a robust processing pipeline for analytics tasks.	Extreme-scale Cloud/Fog Data Processing, Mobility Data Fusion and Management
FR4	**Extreme Scale Data Handling**	Support efficient data acquisition, processing, fusion, and management, addressing challenges posed by massive data volumes, sparsity, and diverse data streams.	Extreme-scale Cloud/Fog Data Processing Mobility Data Fusion and Management Extreme Scale Mobility Data Analytics (MDA) at the CC
FR5	**Heterogeneous Data Support**	Accommodate various data sources, formats, and types, enabling seamless integration and processing of diverse mobility data.	Mobility Data Fusion and Management, Mobility AI-as-a-Service
FR6	**Advanced Analytics Tools**	Advanced analytics tools capable of handling complex mobility data analysis tasks, such as trajectory analysis, pattern discovery, and event detection.	Extreme Scale Mobility Data Analytics (MDA) at the CC, Active & Federated Learning over Mobility Data, Mobility AI-as-a-Service
FR7	**Real-time Processing**	support real-time processing capabilities, enabling timely analysis and insights from streaming mobility data.	Extreme-scale Cloud/Fog Data Processing
FR8	**Secure Data Handling**	Incorporate robust security mechanisms to protect sensitive mobility data throughout its lifecycle, including data transmission, storage, and processing	Privacy-aware in situ Data Harvesting, Security and Data Governance Layer

FR #	Functional Requirement	Description	EMERALDS Category
FR9	**User-Friendly Interfaces**	User-friendly interfaces should be provided for configuring, deploying, and managing the EMERALDS services, making the tools accessible to users with varying technical backgrounds	Visual Analytics and Dashboards

2.3 EMERALDS Reference Architecture

The contents of this section are currently undergoing development and are expected to be refined in the next months. As the project progresses, the specific details of the software design will be documented in the corresponding deliverables (D3.1, D3.2 D4.1, D4.2, D2.2-D2.6, due in M15, M18, M24 and M33). In this manner, the agile approach advocated by EMERALDS is materialized, permitting thorough reviewing of software design and alignment with the project objectives and requirements.

EMERALDS reference architecture will be based on a service-oriented approach for the development of a toolset that can be operated across the compute continuum and enables extreme scale urban mobility data analytics.

The toolset consists of a set of services, known as 'emeralds', of a wide range of functionalities and roles. To maintain coherence in the toolset presentation, the components have been categorized into six subcategories, as they have been defined in the initial proposal:

- Privacy-aware in situ Data Harvesting
- Extreme-scale Cloud/Fog Data Processing
- Mobility Data Fusion and Management
- Extreme Scale Mobility Data Analytics (MDA) at the CC
- Active & Federated Learning over Mobility Data
- Mobility AI-as-a-Service

Table 5 summarizes the list of emeralds services. It also provides information about the owner and a short description of each service. A more detailed analysis of each 'emerald' including functionality and technical specification is provided in Chapter 3.

Table 5 - EMERALDS Design Specification Overview

Generic Information		Description	Category	Target TRL
Owner	'emerald'			
UPRC, AIT	**Privacy aware data ingestion**	This 'emerald' offers privacy-ensuring techniques for data collection and ingestion, like continuous privacy monitoring, while ensuring optimal data	Privacy-aware in situ Data Harvesting	4

Generic Information Owner	'emerald'	Description	Category	Target TRL
		ingestion using methods like real-time compression etc.		
UPRC	**Extreme scale Stream processing**	This 'emerald' offers tasks such as monitoring, orchestrating and optimizing the processing of Data Streams that takes place in real time by implementing methods that are optimized for low power/ embedded systems. Such methods can include cleaning augmentation, segmentation, enrichment etc.	Privacy-aware in situ Data Harvesting	4-5
UPRC	**Extreme-scale map-matching**	This 'emerald' offers map-matching of GPS coordinates to an underlying road network. Typically, UCx offers a dataset of GPS coordinates of moving vehicles, and then we obtain the road network (typically OSM) and return back the dataset, but each GPS is w associated with a specific road segment and a position on this road segment.	Extreme-scale Cloud/Fog Data Processing	4
UPRC	**Weather enrichment**	This 'emerald' offers enrichment of mobility data from UCx with external weather data sources, e.g., ECMWF. The enrichment can be performed in various settings: batch (for historical data), online (for streaming data), in a scalable way (for huge data sets).	Extreme-scale Cloud/Fog Data Processing	3
UPRC	**Spatio-temporal querying**	This 'emerald' offers efficient and scalable querying over large collections of spatio-temporal data. Typical queries that can be supported include range queries, k-nearest neighbour queries, and joins. The current focus is on spatial joins: given 2 large datasets S and T, find the pairs (s,t) with distance lower than a user-specified distance threshold. Implementation in Apache Spark.	Extreme-scale Cloud/Fog Data Processing	3

Generic Information Owner	'emerald'	Description	Category	Target TRL
UPRC	**Hot-spot analysis**	This 'emerald' offers discovery of hot-spots in an urban context, where hot-spots are road segments with unusually high traffic. A dataset of GPS coordinates is required as input. The discovered hot-spots are based on the information present in this GPS dataset. At a technical level: (1) we use the Getis-Ord statistic to discover hot-spots, (2) the 'emerald' will be implemented using big data technologies (e.g., Apache Spark) to cope with large datasets.	Extreme-scale Cloud/Fog Data Processing	4
ULB	**Mobility/trajectory data compression**	This 'emerald' offers optimized storage and data management for urban trajectory data, integrated into MobilityDB. This should include trajectories of different transport modes: public transport, trains, micro-mobility, etc	Mobility Data Fusion and Management	3-4
ULB	**Sensor data fusion**	This 'emerald' offers methods for integrating data that originate from different sources, that typically lacks keys for integration	Mobility Data Fusion and Management	3-4
ULB	**Traffic State Estimation**	Means of interpolating the missing traffic information towards a better coverage, taking into account that traffic data is typically sparse, both in the spatial coverage of the segments of the network, as well as the temporal coverage.	Mobility Data Fusion and Management	3-4
UPRC	**Trajectory/Route Forecasting and Origin/Destination Estimation**	This 'emerald' uses the pre-processed/augmented data streams that are provided by T3.1 in order to provide a real-time route evaluation framework for event/incident prediction and route forecasting.	Extreme Scale MDA at the CC	4
INLE	**Probabilistic approach for trip chaining**	This 'emerald' enhances existing approach by estimating which route probabilities are more likely	Extreme Scale MDA at the CC	Not Available

Generic Information Owner	'emerald'	Description	Category	Target TRL
		to produce current data and use them for forecasting.		
AIT	**Trajectory data / travel time analysis**	This 'emerald' offers trajectory analytics using the open-source Python library MovingPandas which covers trajectory data processing (such as trip/stop extraction; speed and travel time computation; trajectory smoothing; outlier detection) as well as visualization (in Jupyter notebook environments and Panel data apps)	Extreme Scale MDA at the CC	4-5
Sistema	**Extreme Scale Map Matching**	This 'emerald' offers map-matching for FCD to retrieve trajectories in order to estimate speeds on a given network.	Extreme Scale MDA at the CC	4
UPRC, AIT	**Traffic state / flow forecasting**	This 'emerald' offer traffic state prediction using NN; prediction horizon: 15-30 min in the future (to enable anticipating traffic measures)	Active & Federated Learning over Mobility Data	4
AIT, INLE, UPRC, TUD	**Crowd density forecasting**	This 'emerald' offers crowd-density forecasting models (short & mid-term) using GNN; later (in Y2/3), we envision to extend/generalize the model for multi-functional areas (shopping, commute, tourist)	Active & Federated Learning over Mobility Data	2-4
AIT, UPRC	**Parking garage occupancy forecasting**	This 'emerald' offers parking forecasting models (short-term); e.g., using GNN	Active & Federated Learning over Mobility Data	2-4
AIT	**Active Learning & XAI for crowd/flow forecasting**	This 'emerald' offers explainability features to existing GNN-based crowd/flow forecasting models provided by the use case partners.	Active & Federated Learning over Mobility Data	2-4
AIT	**Active Learning (AL) model for risk classification**	This 'emerald' offers a new risk detection model that can take crowd information and other data sources, and output different categories of risks or risk category probabilities	Active & Federated Learning over Mobility Data	2-4

Generic Information Owner	'emerald'	Description	Category	Target TRL
AIT	**Federated Learning (FL) models for mobility data**	This 'emerald' offers FL methods that can be run on in-vehicle edge devices, e.g., for data privacy protection or for communication bandwidth saving	Active & Federated Learning over Mobility Data	2-4
ATOS	**Data ingestion interfaces**	This 'emerald' offers data ingestion interfaces for ML components developed in T4.2 from already harvested datasets (sources) to the MLOps framework.	Mobility AI-as-a-Service	4
ATOS	**ML experimentation module**	This 'emerald' offers ML experimentation modules for ML components developed in T4.2	Mobility AI-as-a-Service	4
ATOS	**ML training and testing module**	This 'emerald' offers ML training and testing modules for ML components developed in T4.2	Mobility AI-as-a-Service	4
ATOS	**ML Models (and tools) repo**	This 'emerald' offers an ML models and tools repo for ML components developed in T4.2	Mobility AI-as-a-Service	4
ATOS	**Federated Learning module**	This 'emerald' offers FL modules for ML components developed in T4.2	Mobility AI-as-a-Service	4
ATOS	**ML Monitoring tools**	This 'emerald' offers ML monitoring tools for ML components developed in T4.2	Mobility AI-as-a-Service	4

As the 'emerald' components are designed to operate across the compute continuum and support different operation needs from data ingestion to analytics both on cloud and edge nodes, it was imperative to determine the placement of the 'emerald' components on the compute continuum. The outcome of this exercise is reflected in Table 6.

Table 6 - EMERALDS Services Distribution across the Edge/Fog/Cloud Continuum

ID	Category	'emerald' Service	Edge	Fog	Cloud
A-1	Privacy-aware in situ Data Harvesting	Extreme scale Stream processing	**X**	**X**	
		Privacy aware data ingestion	**X**	**X**	
A-2	Extreme-scale Cloud/Fog Data Processing	Extreme-scale map-matching			**X**
		Weather enrichment			**X**
		Spatio-temporal querying			**X**

ID	Category	'emerald' Service	Edge	Fog	Cloud
		Hot-spot analysis			X
A-3	Mobility Data Fusion and Management	Mobility/trajectory data compression			X
		Sensor data fusion			X
		Traffic State Estimation			X
B-1	Extreme Scale MDA at the CC	Trajectory/Route Forecasting and Origin/Destination Estimation	X	X	X
		Probabilistic approach for trip chaining		X	X
		Trajectory data / travel time analysis			X
		Extreme Scale Map Matching			X
B-2	Active & Federated Learning over Mobility Data	Traffic state / flow forecasting			X
		Crowd density forecasting			X
		Parking garage occupancy forecasting			X
		Active Learning & XAI for crowd/flow forecasting			X
		Active Learning (AL) model for risk classification			X
		Federated Learning (FL) models for mobility data	X	X	X
B-3	Mobility AI-as-a-Service	Data ingestion interfaces	X	X	X
		ML experimentation module			X
		ML training and testing module			X
		ML Models (and tools) repo			X
		Federated Learning module	X	X	X
		ML Monitoring tools			X

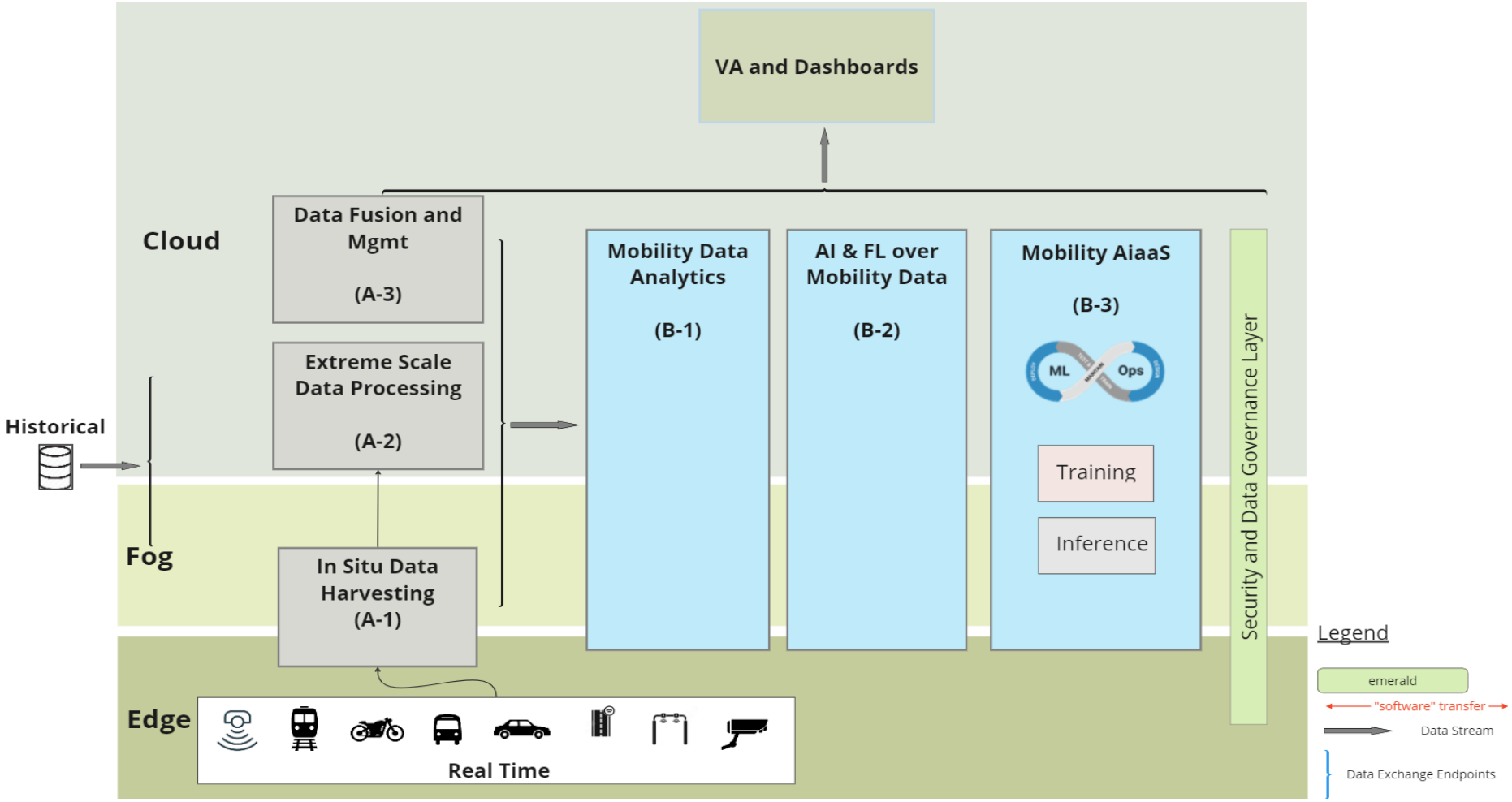

Figure 2-2 -EMERALDS Reference Architecture

Figure 2-2 illustrates an updated version of the reference architecture as initially presented in the Figure 2-1. It also presents all the 'emeralds' placed across the CC and grouped according to their main functionalities into the 6 main groups, previously mentioned, plus the Security layer:

Regarding the **Visual Analytics and Dashboard** services, **EMERALDS advances** beyond the state-of-the-art by using scalable and performant cloud data-warehouses for processing mobility data easily. Capabilities to extend Cloud Data Warehouses opting to support PostGIS and expose functions suitable for more complex Spatial Analysis compared to an exclusive PostGIS implementation are pursued (**T2.2,** D2.4 v1 – D2.5 v2). All these functions will be fully integrated into the CARTO platform, thus giving users the ability to explore, visualize, and analyse their spatial data without switching contexts or the need to ETL data.

The platform is capable to ingest massive amounts of mobility data, effectively apply analytic methods and present them in a feature rich graphical environment. Additionally, the ATOS platform offers a Jupyter Notebook[1] Instance as a Service, that can be used for simple visualizations of the generated results of the ML/AI 'emerald' services. Users of the EMERALDS Toolset may design their own visualizations services based on the interoperability of the data and standards of the EMERALDS components.

Since an important aspect of the toolset is the creation of effective Mobility Data Pipelines, it is also important to place the EMERALDS components across such a hypothetic scenario. Figure 2-4 acts as a roadmap for implementing such pipelines.

Additionally, the EMERALDS Toolset aims to become available through a publicly accessible repository in various forms of software deliverables, such as code libraries, AI/ML models, execution packages and docker containers[2].

Regarding the infrastructure, the Toolset takes advantage of all forms of compute instances available, starting from public Cloud Providers such as GCP, AWS and Azure, to private cloud data centres and single board computers such as Raspberry PI[3], Nvidia Jetson[4].

Five major phases or steps are commonly found in big data analytics pipelines, representing a structured approach to handling and extracting value from large and complex datasets (Figure 2-3).

Data Acquisition and Recording, where the raw data for the analysis are collected. The raw data is then processed and cleaned to remove any errors or inconsistencies, involving tasks such as data ***preprocessing,*** data transformation or removal of outliers with the aim to guarantee that data is accurate and reliable for further analysis. In the next step, ***data integration, aggregation and representation*** diverse formats and types of data are combined into a unified format, resolving any schema conflicts, facilitating the analysis stage. In the ***query processing, data modelling and analysis*** phase, various techniques are applied in order to extract insights from the processed data. Once the analysis is complete, the results need to be ***interpreted*** and translated into meaningful information.

Figure 2-3 - State-of-the-play Big Data Analytics Pipeline

[1] https://jupyter.org/

[2] https://www.docker.com/resources/what-container/

[3] https://www.raspberrypi.org/

[4] https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/

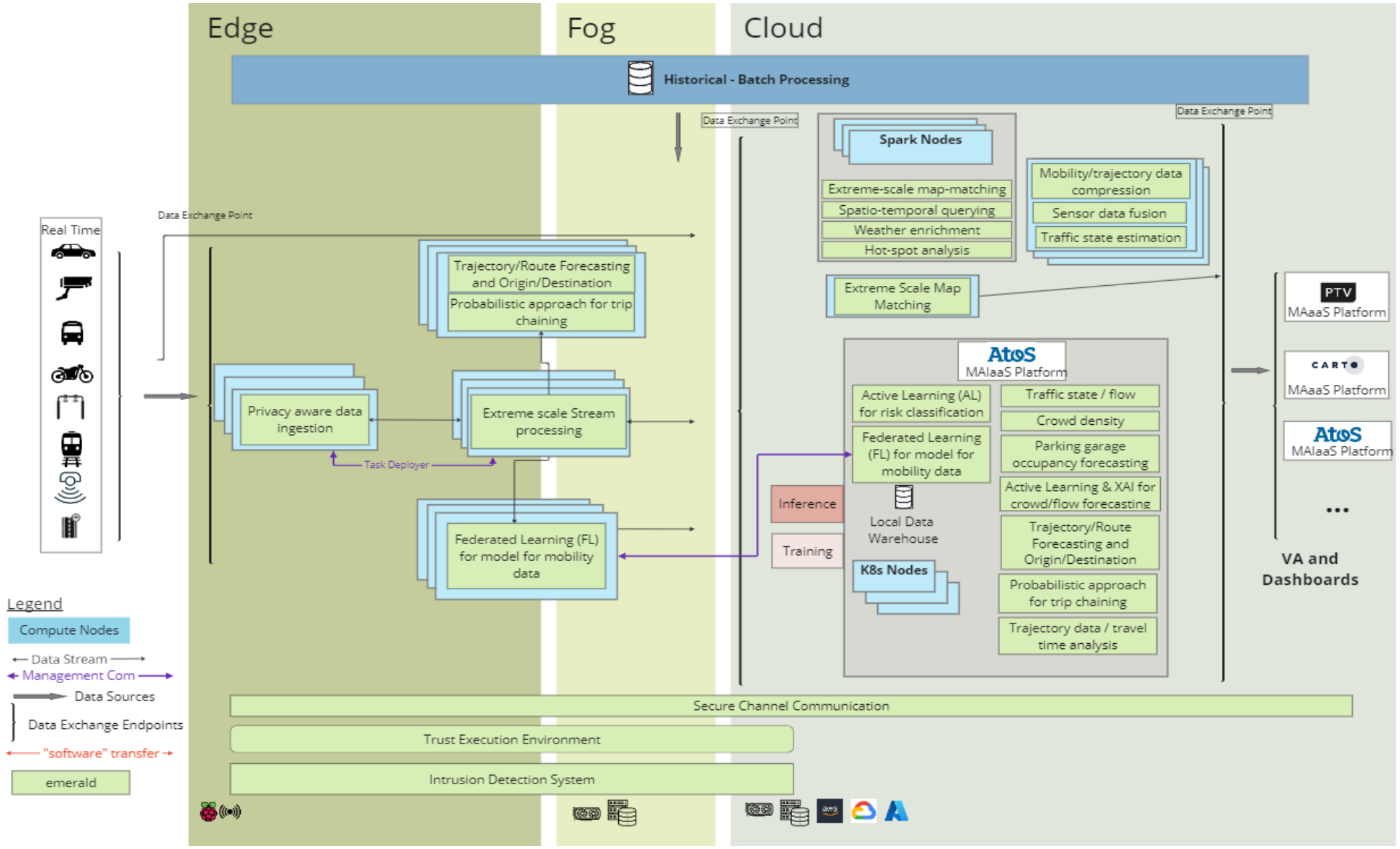

Figure 2-4– EMERALDS services positioning across Data Pipelines

Figure 2-4 presents a hypothetical data pipeline. Such a pipeline can be established by combining the EMERALDS toolset with external software components, either commercial or open-source.

The entry point for the pipeline is the data input sources, which are split into two main categories,

- **Real-time** data that are mainly being provided by sensors located on the Edge and formatted as data streams. Their processing is triggered by events and are often small in size.
- **Batch data** that are provided mainly in tabular format and can be stored and consequently accessed across the compute continuum. They come in large volumes.

The EMERALDS Reference Architecture offers a number of different paths for the pipeline. These paths are related with services running on Edge and Fog nodes and are the following:

- **Data Ingestion and processing.** Provides the entry point of data flows into the data pipeline. For real-time data processing the event-driven processing is the preferred architectural approach. In such cases a Message Broker (e.g., Apache Kafka, RabbitMQ[5]) or the "Extreme Scale Stream Processing" 'emerald' is dedicated for supporting data transfer across the various nodes. For historical/batch data this step requires the extraction of data from a data storage.
- **Analytics in-situ.** By using pretrained ML models running on edge nodes, it allows for faster analysis, without the need for data transfer on the next layers of the CC and especially outside the boundaries of data owner infrastructure.
- **Federated Learning.** As part of the ML training process, data located on the edge can be used for training machine learning models using the federated learning methodology. In this case, edge nodes exchange models' parameters and not datasets with the core node, in our case the MAIaaS platform.

As a next logical step of the Data Pipeline, EMERALDS is offering two groups of services running on Cloud Infrastructure.

- **Extreme Scale Data Processing.** The 'emeralds' of this group can handle massive volume of data, mainly in batch form and perform tasks related to data cleansing, filtering, normalization and fusion, spatio-temporal querying, and descriptive analytics.
- **Extreme Scale Data Analytics.** This group is related with the MAIaaS platform and the MLOps process to develop and train new or enhanced machine learning models. To that end, MAIaaS platform holds to dual role, as a development hub for training the ML models and as an inference point for the trained ones. The platform is designed to operate its own data management layer – data ingestion, storage, and versioning and provide Visual Analytics tools.

Regarding the Data Manager, EMERALDS toolset is not bind to any specific Database for storage, but promotes the use of MobilityDB[6], an open-source geospatial trajectory data management and analysis platform, implemented as an extension to PostgreSQL and PostGIS.

The last step of the pipeline is the **Visualization** of the results. To that end, CARTO offers its own production grade (TRL 9) platforms, that specializes in location intelligence, spatial analysis, and data visualization. CARTO platform is using Cloud Data Warehouses, such as Google BiqQuery, Redshift, and Snowflake to store vast amount of geospatial data and run CARTO analytics toolbox[7].

[5] https://www.rabbitmq.com/

[6] https://mobilitydb.com/

[7] https://docs.carto.com/faqs/analytics-toolbox

The architecture has followed a bottom-up approach with a focus on the areas of expertise of each technical partner, with ultimate goal to define a new state of the art for these areas. As it will be described in more details on Chapter 3, each component requires different prerequisites and more specifically different execution environments, from single board computers such as raspberry pi's, NVidia Jetson, beaglebone to name a few, to Apache Spark clusters, Kubernetes cluster and public cloud providers infrastructure. Moreover, the machine learning/artificial intelligence (ML/AI) based 'emeralds' can be benefited from the execution on GPU equipped computers, such as Google Coral dev board [8]when referring to edge nodes or properly configured servers on fog and cloud nodes. Last, there are 'emeralds' focusing on tackling security concerns of the edge devices using TEE technics and establishing secure ecosystems during data on the go and data at rest. A further categorization can be defined according to the existing information.

- **Intelligent**. Services that have been based on ML, FL and explainable AI models.
- **Infrastructure**. Services required by the MAIaaS Platform or related to security of data.
- **Application**. Services that are related with data ingestion, processing, fusion, and analytics.

The objective is to provide a versatile toolset which can be integrated into other platforms in the mobility analytics domain. These platform integrations can be regarded as instances of this reference architecture. Furthermore, it is also anticipated that these platforms will selectively incorporate a subset of the 'emerald' services in their respective data pipelines. This approach ensures flexibility, adaptability and the efficient utilization of the EMERALDS toolset across a range of mobility analytics platforms.

A series of virtual meetings fleshed out the bottom-up approach that was applied in order to clearly identify how each of the EMERALDS toolset components and their mutual interactions, map to and address the requirements of the use case partners. Based on the output of Deliverable 5.1 Use Cases Scoping Document and the definition of the 'emeralds' service the following Table 7 presents the mapping of the "Use Cases and Early Adopters to 'emeralds' components" as it has been agreed during the time of writing this report. Changes are expected based on the data available, feasibility due to time constraints or the emergence of new business requirements.

Table 7 - Use Cases and Early Adopters to Emeralds components

'emerald'	UC1: Hague	UC2: Rotterdam	UC3: Riga	EA: York	EA: CARTO
Privacy aware data ingestion			Yes		
Extreme scale Stream processing	Yes	Yes	Yes		
Extreme-scale map-matching	Yes	Yes	Yes (conditional)		
Weather enrichment	Yes	Yes	Yes		
Spatio-temporal querying	Yes	Yes	Yes (conditional)		
Hot-spot analysis	Yes	Yes	Yes	Yes (conditional)	
Mobility/trajectory data compression	Yes	Yes	Yes		

[8] https://coral.ai/products/dev-board/

Sensor data fusion	Yes	Yes	Yes		Yes
Traffic State Estimation		Yes			
Trajectory/Route Forecasting and Origin/Destination Estimation	Yes	Yes	Yes		
Probabilistic approach for trip chaining			Yes		
Trajectory data / travel time analysis		Yes	Yes	Yes	Yes
RT Extreme Scale Map Matching				Yes	
Traffic state / flow forecasting		Yes			
Crowd density forecasting	Yes		Yes		
Parking garage occupancy forecasting	Yes				
Active Learning & XAI for crowd/flow forecasting	Yes	Yes			
Active Learning (AL) model for risk classification	Yes (conditional)				
Federated Learning (FL) models for mobility data	Yes (conditional)		Yes (conditional)		
Data ingestion interfaces	Yes (conditional)	Yes (conditional)	Yes		
ML experimentation module	Yes	Yes	Yes		
ML training and testing module	Yes	Yes	Yes		
ML Models (and tools) repo	Yes	Yes	Yes		
Federated Learning module	Yes		Yes		
ML Monitoring tools	Yes	Yes	Yes		

2.4 Infrastructure

An analysis of the infrastructure requirements essential for the successful operation of the EMERALDS toolset is the aim of this section. Infrastructure requirements are categorized into three pillars, related to each distinct phase of the EMERALDS components lifecycle. These are the ***development*** process, the ***orchestration*** infrastructure, and the ***execution*** environment.

2.4.1 Development Process

The first pillar focuses on the **development** process of the emerald's components. The use of best practices during this phase is encouraged, such as versioning control with the use of a common repository for all open-source 'emeralds' components with code review processes, documentation, and continuous integration/continuous delivery (CI/CD) as part of "Deliverable 2.2 - Containerized

EMERALDS Toolset v1" and "Deliverable D2.3 – Containerized EMERALDS Toolset v2". For the latter, processes will be established to ensure the quality of the delivered software, like automated testing with profiling and code coverage, and static analysis tools for identifying code smells, security vulnerabilities, enforcement of code guidelines, and measurement of technical debt. Additionally, the development process follows a continuous improvement methodology which is applied in project management level by splitting it into two major phases, with the first package delivered on M15 followed by a demonstration period (M24) before moving to the second phase on M33, where the technical efforts uptake performance optimizations, bug fixes, and further development of new or existing features.

Regarding the development process for the "Intelligent" EMERALDS services, additional activities are required beyond the selection of the appropriate model and the implementation of the corresponding service. MLOps practices are to be followed such as data engineering, ML model engineering, model testing, validation, and deployment. Towards this end, ATOS will design and implement a tailored platform (T4.3 "Mobility AI-as-a-Service"), which serves among others as a development environment for the "Intelligent" EMERALDS services. The platform will offer an Integrated Development Environment (IDE) as a service), based on Jupyter[9] Notebooks, along with MLOps pipelines that will allow modelers to create, train and test their AI/ML models. Furthermore, the platform will also support Federated Learning training and distribution processes by offering and installing agents on remote nodes that will manage the iterative learning process. More details regarding this platform are described in Chapter 5.

2.4.2 Deliverable Methods of EMERALDS Toolset

The 'emeralds' components will be offered jointly as a toolset, but also as individual solutions in the following forms. First, as a code library through a common code repository. External platforms may use any 'emerald' service according to the instructions of that service. Extending and managing integration is considered to be taken in the deployment platform end. This approach is inherently restricted, as it requires a common technology stack used by the platform and the services. Starting with the challenges of the execution environments, several concerns that need to be addressed, such as dependency management, execution engine, isolation, deployment and configuration complexity, compilation process, logging, and monitoring frameworks are recognized. On the other hand, this delivery method is the easiest solution to be implemented for platforms or products that are built based on monolithic architecture, legacy systems and for services running on edge devices where resource utilization is crucial and the use of a proper framework with low hardware footprint is preferred. Deliverables D3.1, D3.2, D4.1 and D4.2 investigate technologies and frameworks to achieve the desired performance on edge devices.

A second approach lies on the distribution of the "emerald" components as containers. This approach enhances portability, isolation, scalability, monitoring, and the effortless deployment process. In effect, this method suits better for cloud native applications and thereby for 'emeralds' of T3.2, T4.1, T4.2 and T4.3. Typically, a performance penalty is expected due to the extra layer of virtualization, but the benefits outweigh this drawback, especially if the platform that integrates the EMERALDS Toolset can be scaled horizontally. For the purpose of offering both delivery methods, a code repository and a container registry is foreseen as part of D2.2 and D2.3.

2.4.3 DevOps Practices

DevOps is a set of practices and tools that integrates and automates the work of software development (Dev) and IT operations (Ops) as a means for improving the software development life

[9] https://jupyter.org/

cycle. Since the creation of a one-fits-all Mobility Analytics Platform is not a goal of the EMERALDS project, the operations part of the DevOps could be explored only as part of a Lab setup, for testing and demonstration purposes.

Key functionalities usually covered by a continuous development (CD) pipeline are the following:

- **Automated Deployment**: Using the container registry, external platforms can easily deploy the 'emeralds' components of their choice. Automation is a key concept to ensure reproducibility, avoid errors and increasing deployment speed.
- **System Provisioning**: Automating the configuration and provisioning of the execution environments to achieve consistency between development, testing and production environments. It is also a key factor for system reconfiguration after catastrophic events.

Given that EMERALDS encompasses the entire compute continuum, it is of great importance to comply with DevOps best practices, also specified for the management of edge nodes. This is especially evident when considering that in a production ready application the number of edge devices might be in the order of thousands, such as a sensor network within a city. Well defined processes should be configured to enable periodic updates of the software components running on edge or fog nodes, for remote attestation, and for SecOps practices such as security configuration, certificates renewal, trusted environments setup – processes that are partially explored in Deliverable 2.6.

Furthermore, the federated learning related tasks for predictive analytics on edge nodes presents another use case for DevOps and MLOps processes to be applied. In this direction, cases where model retraining is required given new datasets available on distributed data lakes located on Data Owners facilities are investigated.

Typically, an **orchestration** platform follows a pyramid architecture, with a centralized service managing the entire fleet of nodes and devices. Depending on the size of the device's network, additional management nodes may be deployed to distribute the orchestration load. Figure 2-5 presents this approach, where the light blue boxes represent the management nodes, while the dark green the EMERALDS Execution Nodes. An alternative option is offered for nodes acting as both management and execution modes.

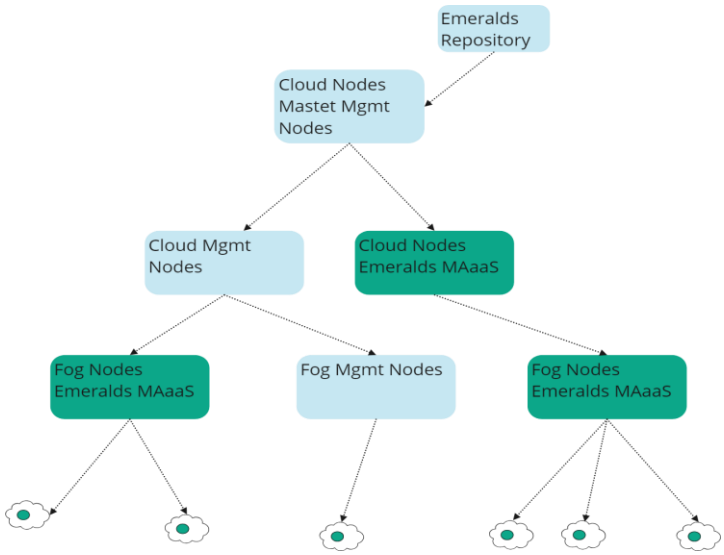

Figure 2-5 - Orchestration Pyramid Architecture

Agents deployed on all network devices is a common approach to allow bi-directional communication between the master node and the slaves, increased security, easier software compliance across edge

devices and independence in case of network failures. The bi-directional communication may allow the use of an event driven approach for their management, thus reducing network bandwidth requirements and increase resource utilization on the management nodes. In this way, it is possible to achieve zero-ops (no human operations), except from the initial registration phase –where patterns that allow automated service discovery and registration should be explored, always considering security previsions by blocking compromised nodes to gain access to the network. A tailored intertwined DataOps, MLOps and DevOps chain approach, encompassing the seamless delivery and execution of EMERALDS services also enabling extreme data handling across the computing continuum by means of efficient utilization of distributed computational resources is depicted in Figure 2-6.

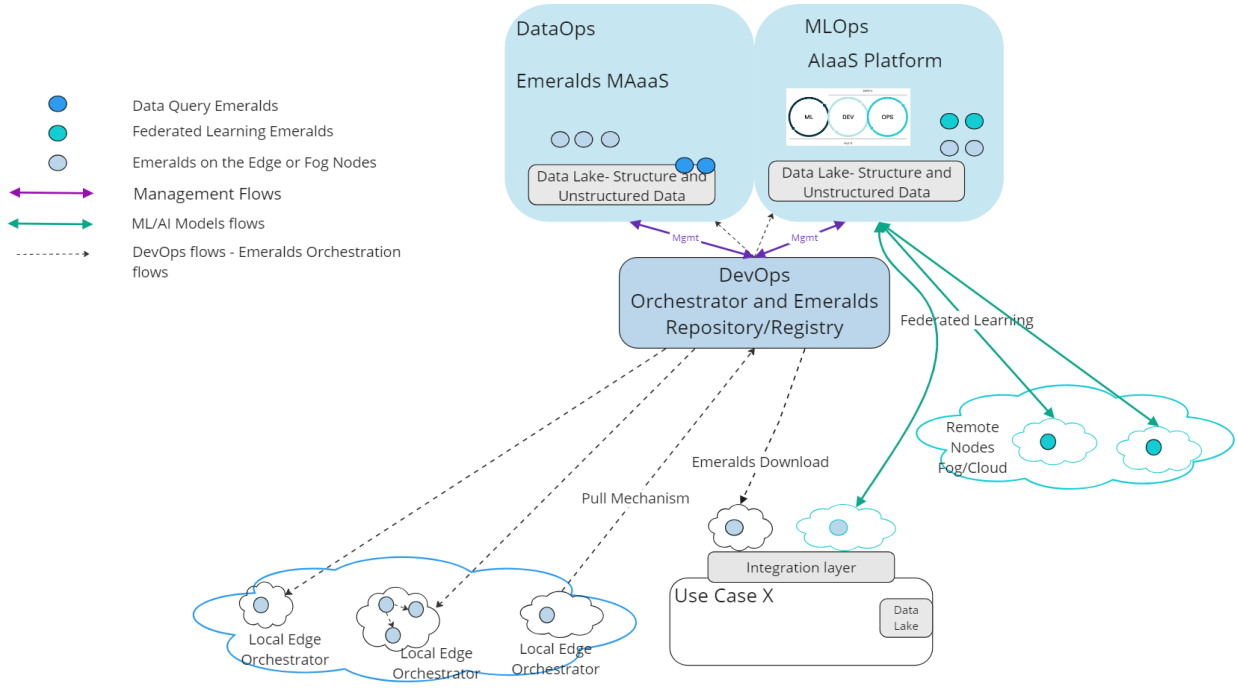

Figure 2-6 – Infrastructure Architecture

Another important aspect of the platform would be the ability to provide additional functionalities for the **execution** environment such as monitoring and logging capabilities to allow for better observability of the solution, self-healing mechanisms for responding to failures and automatically taking corrective actions – node or service restarts and redistribution of loads, and integration points for streamline data pipelines, such as message brokers (e.g. MQTT[10] Broker, Apache Kafka), and HTTP/Rest APIs.

Existing or new platforms that would like to incorporate the EMERALDS toolset on their solution, would **need to either offer their own orchestrator** or take advantage of open-source solutions already available such as KubeEdge [11]and IBM Edge Computing[12].

[10] https://en.wikipedia.org/wiki/MQTT

[11] https://kubeedge.io/

[12] https://www.ibm.com/edge-computing

As mentioned, the ATOS MAIaaS platform will act as a development platform for 'emeralds' "intelligent" components. For demo only purposes the MAIaaS platform will also act as execution environment for the "intelligent" components (e.g., inference), meantime allowing the execution of other 'emeralds', which can be delivered in a containerized format. DevOps practices are applied mainly for the deployment and monitoring of the federated learning 'emerald' components on edge nodes, to improve the overall development experience and to provide quantitative measures of tool execution. In addition, the 'emerald' "Extreme scale Stream processing" is designed to operate as a highly effective Data Stream Broker for edge and fog nodes, while implementing basic DevOps practices. Further information can be found on section 3.1.2.

2.5 Key Performance Indicators (KPIs)

In this chapter, we provide the definition of the technical KPIs for each 'emerald' service and the MAIaaS platform. Technical KPIs play a pivotal role in the EMERALDS project as they serve as quantifiable benchmarks for assessing the performance and effectiveness of various technical aspects within the project's scope. These KPIs, meticulously documented in project deliverables, provide a structured framework for evaluating critical factors such as interoperability, scalability, security, and data governance. Setting specific, measurable targets, technical KPIs guide the development process, in order to warrant that the resulting tools and services meet predefined criteria for functionality and efficiency. They also enable objective performance assessments during assessment and validation cycles, assisting the project partners to fine-tune and optimize their solutions.

The baseline KPI values, the measurements methodology and the actual measurements are undertaken in the corresponding deliverables D3.1, D3.2, D4.1, D4.2 and D2.6. The development process pursued in previously mentioned WPs, involves the measurement of services' KPIs within technical partners' development environments.

These KPIs are also measured in the WP5 Use Cases after being integrated to the corresponding Mobility Analytics solutions. This environment can be considered as pre-production for the verification of the 'emerald' Service and the outcome of it can be compared with the KPIs measured during the development process. As a general note, the impact of the EMERALDS Toolset in UC of WP5 is analogous to the number of toolset services that are being utilized, and it will indicate the incurred improvement on the urban mobility data analytics workflows.

Since the overarching EMERALDS project goal is the creation of a toolset rather than a Mobility Analytics platform, system wide KPIs can only be defined within the scope of specific Use Cases, as those defined in WP5.

Additionally, the impact KPIs defined will be provided as part of the WP2, WP3 and WP4 related deliverables, based on the technological advancements of the 'emerald' services. Figure 2-7 presents the different categories of KPIs along with some indicative examples defined within the EMERALDS project.

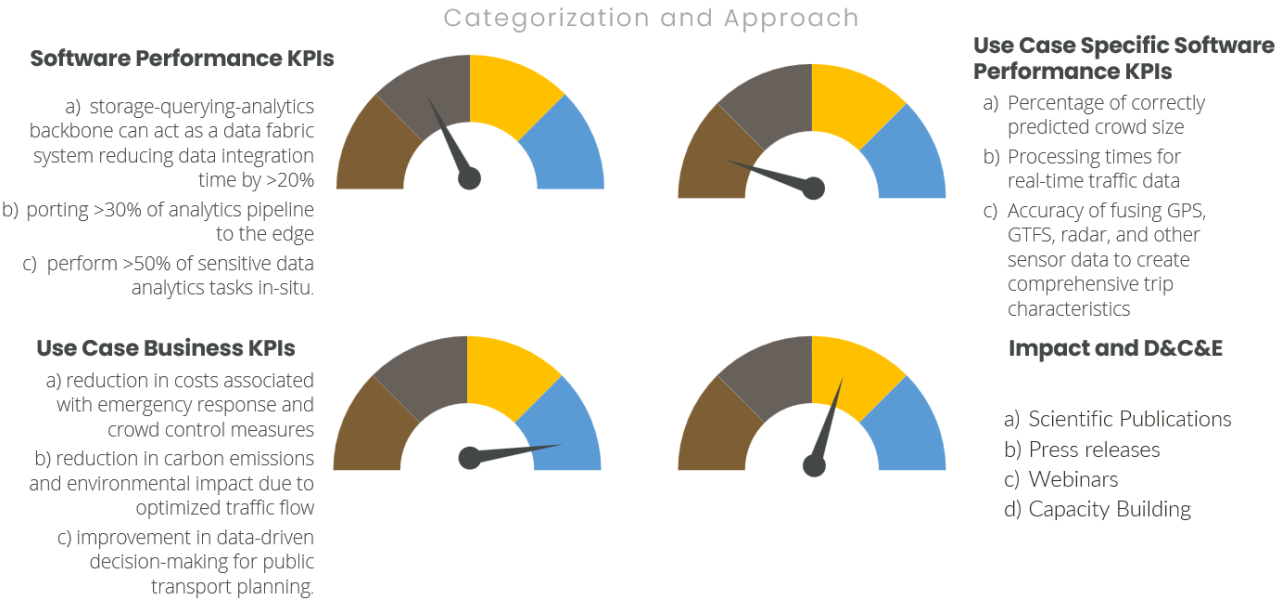

Figure 2-7 - KPIs Categorization during EMERALDS Project

Three global KPIs that should be attained by the EMERALDS toolset and individual components whenever this is applicable are targeted. These are:

- Reduce data integration time by >20%
- Port >30% of analytics pipelines (including AI inference) at the edge.
- Perform >50% of sensitive data analytics task in-situ/without data leaving the source.

Their measurement requires data pipelines to be assessed and compared with and without the use of the EMERLADS services that would be required for the specific task. Such scenarios can only be executed during the project timeline, by the EMERALDS Use Cases and at least for the last two KPIs, the use of edge nodes is a prerequisite.

It also worth mentioning that the defined thresholds of the global KPIs should be considered as indicative rather than unchangeable.

In Table 8 the technical KPIs are being defined by each 'emerald' component for the evaluation of the delivered software. They can be categorized into the following generic areas:

- Performance indicators, such as throughput, latency, concurrency, and response time
- Resource Utilization, such as memory, CPU resources consumption, and elasticity/scalability
- Quality indicators, such as error and accuracy rates
- Training and Inference time indicators,
- Model Explainability
- Security indicators, such as network security and infrastructure security coverage.

It is important to emphasize that for "intelligent" EMERALDS services, the KPIs measurement is a combination of the software and model quality attributes along with the data sources utilized during training within specific use case scenarios. Therefore, varying trained "instances" might result on distinct measured KPIs values.

Table 8 - Technical KPIs list

'emerald'	Technical KPI	Description
Privacy aware data ingestion		
	Privacy preservation	# of privacy risks evaluated per sec
	Real time privacy evaluation	Evaluation for multiple objects per sec
Extreme scale Stream processing		
	Ingestion throughput	rows/sec, size of input data
	Low latency data processing	Minimizing the total latency for each datapoint that is transmitted to and from the Orchestrator.
Extreme-scale map-matching		
	Performance: Efficiency/Scalability	Time required for map-matching (for larger datasets, larger clusters) and resources that can be exploited
	Quality/Accuracy	Errors (%) of map-matched data. If ground truth is available.
Weather enrichment		
	Performance: Efficiency/Scalability	Time required for enrichment (for larger datasets, larger clusters)
	Quality/Accuracy	Ratio of interpolated to non-interpolated values
Spatio-temporal querying		
	Performance: Query Efficiency	Query execution time (for larger datasets, larger clusters)
	Performance: Data shuffling	Measurement of the size of data that was transferred on the network due to shuffling for the computation of join.
Hot-spot analysis		
	Performance: Efficiency	Time required for discovering hot-spots (for larger datasets, larger clusters)
	Accuracy over performance	provide approximate hot-spots that are very close to the exact ones.
	Quality/Accuracy	Visual inspection of discovered hot-spots.
Mobility/trajectory data compression		

'emerald'	Technical KPI	Description
	Data size	Compression factor= data size before compression / data size after compression
	Data quality	In case of lossy compression, some metric of assessing the loss of data is needed.
Sensor data fusion		
	Linkage rate	When fusing source, A with source B, then # data object in A correctly enriched / |A|
	Fusion quality	Some metrics of assessing the accuracy/confidence of fusion.
Traffic State Estimation		
	Prediction accuracy	Measure how accurately the service predicts missing traffic data (e.g., MAE, RMSE)
	Imputation Success Rate	Calculate the percentage of missing values successfully imputed by the service.
Trajectory/Route Forecasting and Origin/Destination Estimation		
	Forecast accuracy	Additional information to be provided on D4.1
	Real time forecasting	Number of object/s that forecasting has been applied to
Probabilistic approach for trip chaining		
	Trip Chaining accuracy	Increased #matched trips of UC3 approach by ~12%
	Ingestion throughput	rows/sec, size of input data
Trajectory data / travel time analysis		
	Performance: Execution Time	Performance measurement of the trajectory cleaning in terms of execution time.
	Performance: Execution latency	Improved time to first result for network-constrained movement analyses
RT Extreme Scale Map Matching		

'emerald'	Technical KPI	Description
	Data Processing Latency	Reduction of data processing latency throughput
	Memory Usage	Utilization of the Random Access Memory
	Scalability: Ingestion rate	Increase data ingestion rate to scale up to 1M datapoints/s
Traffic state / flow forecasting		
	Prediction accuracy	Metrics based on RMSE and MAE
Crowd density forecasting		
	Prediction accuracy	Metrics based on MERRMSE, MAE, MAPE
Parking garage occupancy forecasting		
	Prediction accuracy	Metrics based on MERRMSE, MAE, MAPE
Active Learning & XAI for crowd/flow forecasting		
	Explanation quality	Fidelity Score (XAI)
	Prediction accuracy	Metrics based on R2, RRMSE, MAE, MAPE
Active Learning (AL) model for risk classification		
	Explanation quality	Fidelity Score (XAI)
	Prediction accuracy	Metrics based on F1-score, AUC
	Classification accuracy	More information to be available on D4.1
Federated Learning (FL) models for mobility data		
	Prediction accuracy	Metrics based on R2, RRMSE, MAE, MAPE
Data ingestion interfaces		
	Data Ingestion Latency	It measures the time it takes for data to be ingested and made available for processing or analysis after it is received by the data ingestion interface. It reflects the efficiency of the data ingestion process and indicates how quickly data can be utilized by downstream systems
ML experimentation module		

'emerald'	Technical KPI	Description
	Experiment Iteration Time	It measures the time it takes to complete a full iteration of an ML experiment within the experimentation module. It includes activities such as data preparation, model training, hyperparameter tuning, and evaluation.
	User Engagement or Adoption Rate	It measures the level of engagement or adoption of the ML experimentation module by data scientists or users within the organization. It can be tracked by monitoring the number of active users, frequency of usage, or the percentage of users who regularly use the module.
ML training and testing module		
	Training time	This KPI measures the time it takes to train ML models within the training module. It reflects the efficiency and speed of the training process. Tracking this metric helps identify bottlenecks or areas for optimization, allowing for faster model development and experimentation.
	Available datasets for training/testing	Provides the number of available datasets uploaded in the MLOps framework for training/testing ML models
ML Models (and tools) repo		
	Model Repository Utilization	It measures the utilization rate or adoption of the ML Models (and Tools) repository within the organization. It can be tracked by monitoring the number of models or tools stored in the repository, the frequency of updates or additions, or the number of unique users accessing the repository.
Federated Learning module		
	Maximum Number of Devices	It measures the maximum number of devices achieved that can participate in the Federated Learning process concurrently. It reflects the scalability and capacity of the system to handle a large number of participating devices.

'emerald'	Technical KPI	Description
	Error Tolerance	It measures the tolerance for errors or failures during the Federated Learning process. It quantifies the acceptable level of errors, such as communication failures, device dropouts, or unreliable data sources, before the system's performance or model accuracy is affected.
ML Monitoring tools		
	Anomaly Detection and Alert Resolution Time	It measures the time it takes to detect anomalies in model behaviour or performance and resolve associated alerts using the ML Monitoring tools. It reflects the efficiency and effectiveness of the monitoring process in identifying and addressing issues.
Secure Communication Channels and Trust-Execution -- Security and Data Governance		
	The end-to-end data flow from the edge to the cloud, will be 100% encrypted.	No unencrypted data will ever be transmitted by/to any processing layer.
	Realise a secure computing framework at all the processing layers.	
Intrusion Detection System -- Security and Data Governance		
	Packet inspection efficiency on the edge. Measurement: Throughput ≥ 1 Gbps.	The intrusion detection module will parse packets executing pattern matching on the packet payload as efficient as possible.

2.6 Extreme Scale Data Analytics in EMERALDS

In general, there are four characteristics that must be part of a dataset to qualify it as **big data.** These are volume, velocity, variety, and veracity. In the domain of Urban Mobility (UM) these characteristics are accompanied by challenges that we will briefly analyze.

Volume: Nowadays the abundance of location tracking technologies creates huge amounts of data and opens possibilities for applications in almost every aspect of our modern lives. In the UM domain, vast amounts of data are generated across city limits from various sources, such as GPS devices on cars and/or passengers, traffic sensors and signals, public transportation systems, road network, weather reports, environment sensors, and incidents reports to name a few. The data can be further categorized as static, such as historical data, city maps or public transportation routes and dynamic, based on their frequency of updates. All these data must be ingested, processed, transferred, and stored.

Velocity: The number of sources that stream data is becoming overwhelming. Billions of people are commuting daily, and the generated data needs to be ingested and stored. This requires a large number of compute nodes that would be able to perform the required tasks in a performant way and will be able to withstand sudden bursts of data input, during peak hours. Additionally, the storage layer should be designed to support increased write rate and throughput.

Variety: In urban environments, not all locations or transportation modes are equally represented in the data. Mobility data is generated by a wide array of sources, including smartphones, vehicle sensors, public transportation systems, and even social media. Each of these sources use different data formats, communication protocols, and data collection frequencies, leading to significant heterogeneity in the data. The incoming data will be available in different forms, and formats. The mobility analytics tools should be able to work with different kinds of data, such as video feeds, spatio-temporal, weather reports, and social media feeds. Even for the same data source form, different representations may be available such as GTFS[13] and SIRI[14].

Veracity: The quality of the incoming data plays significant role in the raw-to-knowledge-data Pipelines. Within the EMERALDS project the training of the machine learning models is strongly affected by the dataset quality. Raw data might be inaccurate (GPS signals), erroneous or noisy and special services are required to detect these anomalies and correct them whenever this is possible.

EMERALDS takes great advantage of the fact that most data is characterized as spatio-temporal and builds on top of that to tackle the Extreme Scale data analytics and processing challenges.

Especially regarding the Variety, EMERALDS project addresses the lack of common data systems, like those existing for relational and spatial data, integrating algorithms, specialized analyses, and prototypes, since they base on different data models, specialized indexes, and different architecture requirements. Forming a combined approach to data integration, transformation and analytics across the E2F2C continuum is investigated within WP3 and WP4.

Table 9 presents each ‘emerald’ and their main benefits against the 4V’s.

[13] https://gtfs.org/

[14] https://en.wikipedia.org/wiki/Service_Interface_for_Real_Time_Information

Table 9 - EMERALDS services in achieving Extreme Scale Data Capabilities

EMERALDS Services	Volume	Velocity	Variety	Veracity
Privacy aware data ingestion	Large throughput in the form of multiple large data streams	Real-time processing: applicable to streaming data	GPS, Weather, Static data etc	Large throughput in the form of multiple large data streams
Extreme scale Stream processing	Large throughput in the form of multiple large data streams	Real-time processing: applicable to streaming data	GPS, Weather, Static data etc	Large throughput in the form of multiple large data streams
Extreme-scale map-matching	Can work on huge data sets (implementation in Apache Spark)	Real-time processing: applicable to streaming data	GPS data, Graph data (road network)	The usual problems related to GPS data (cleaning, etc.)
Weather enrichment	Batch processing of historical data: Requires large volumes weather data			The usual problems related to GPS data (cleaning, etc.)
Spatio-temporal querying	Can work on huge data sets (implementation in Apache Spark)			The usual problems related to GPS data (cleaning, etc.)
Hot-spot analysis	Can work on huge data sets (implementation in Apache Spark)			The usual problems related to GPS data (cleaning, etc.)
Mobility/trajectory data compression	Mobility data, as it comes from source, include high redundancy.		Compression should apply to different types of temporal/spatiotemporal data	

EMERALDS Services	Volume	Velocity	Variety	Veracity
Sensor data fusion	Mobility data, as it comes from source, include high redundancy.		Different types of temporal/spatiotemporal data	
Traffic State Estimation	Handles large data volumes			
Trajectory/Route Forecasting and Origin/Destination Estimation	Large throughput in the form of multiple large DataStream	Real-time analytics: applicable to streaming data	Analytics on multiple sources such as GPS, Weather, Static data etc	This 'emerald' needs to deal with potential data quality issues
Probabilistic approach for trip chaining	This 'emerald' handles large data volumes		Spatial, timeseries from >=2 sources	If required data cleaning, data preprocessing
Trajectory data / travel time analysis			This 'emerald' aims to support a variety of different movement data formats	This 'emerald' aims to support data quality assessment processes
Extreme Scale Map Matching	Processing and storing large volume GPS datasets.	Consuming GPS DataStreams with minimum latency		This 'emerald' is correcting trajectory routes, given the fuzziness of GPS DataSets
Traffic state / flow forecasting	This 'emerald' handles large volumes of training data	This 'emerald' aims to process live data streams		This 'emerald' needs to deal with potential data quality issues.
Crowd density forecasting		This 'emerald' aims to process live data streams	Analytics on multiple spatiotemporal sources such as crowd time series, Weather, Static data etc	This 'emerald' needs to deal with potential data quality issues
Parking garage occupancy forecasting		This 'emerald' aims to process live data streams	Analytics on multiple spatiotemporal sources such as occupancy time series, Weather, Static data etc	This 'emerald' needs to deal with potential data quality issues

EMERALDS Services	Volume	Velocity	Variety	Veracity
Active Learning & XAI for crowd/flow forecasting		This 'emerald' aims to process live data streams	Analytics on multiple spatiotemporal sources such as crowd time series, Weather, Static data etc	This 'emerald' needs to deal with potential data quality issues
Active Learning (AL) model for risk classification		This 'emerald' aims to process live data streams	Analytics on multiple spatiotemporal sources such as crowd time series, Weather, Static data etc	This 'emerald' needs to deal with potential data quality issues
Federated Learning (FL) models for mobility data	This 'emerald' enables scalable ML by distributing model training over multiple clients	This 'emerald' aims to process live data streams		This 'emerald' needs to deal with potential data quality issues
Data ingestion interfaces	This 'emerald' handles large data volumes	This 'emerald' aims to process live data streams		
ML experimentation module	This 'emerald' handles large data volumes			
ML training and testing module	This 'emerald' handles large data volumes			
ML Models (and tools) repo				
Federated Learning module	This 'emerald' handles large data volumes	This 'emerald' aims to process live data streams		
ML Monitoring tools				

2.7 State of the Art on Urban Mobility Data Analytics Platforms

This section presents an analysis on the state-of-the-art BDA, High Performance Data Analytics and Edge-to-Fog-to-Cloud reference architectures, as best practices to be drawn throughout the technical implementation of the project. Figure 2-8 presents an indicative set of open-source solutions.

The BDA architecture described in a former study[i], proposes a flexible approach based on a distributed computing platform for real-time traffic control taking into account a systematic analysis of the requirements of existing traffic control systems. Apache Kafka is employed as the key tool for building data pipelines and stream processing, whilst HDFS manages data storge. Kafka's built-in mechanisms state data analytics more scalable and reliant. The analytics system is capable of efficiently handling a substantial number of data sources concurrently, even when these sources are transmitting data at high rates. Ultimately, the end-to-end system can accommodate large volumes of data flows without cutting down on performance.

The model put forth in previous research[ii], is designed to harness the resources available within autonomous vehicles. This innovative architecture is composed of two main components: a distributed data storage mechanism optimized for real-time analysis and an in-vehicle cloud server tool tailored for batch processing of offline data. Furthermore, a comprehensive workflow model was meticulously designed to facilitate the seamless processing of big data. This workflow model is specifically engineered to enable the real-time examination of streaming data, aligning with the demands of dynamic environments.

System architectures supporting BDA in Intelligent Transportation Systems usually focus on satisfying specific predefined objectives (GPS data mining, traffic flow predictions, predicting Traffic Accidents and more) and have traditionally relied on ad hoc solutions. While they might achieve these immediate goals, they tend to lack the flexibility required to adapt to diverse applications and various data sources. This inherent inflexibility gives rise to rigid systems that struggle to accommodate new requirements and novel datasets.

datACRON[15] was an H2020 RIA project specializing in real-time threat and abnormal activity detection, prediction of trajectories and important events related to moving entities, together with advanced visual analytics methods, over multiple heterogeneous, voluminous, fluctuating, and noisy data streams from very large fleets of moving entities spread across large geographical areas. D1.2 Architecture Specification presented a design that catered for combining real-time and historical data. The proposed architecture resembles a Lambda Architecture[iii] comprising distinct layers, it begins with the foundational Data Store where data is stored in a structured manner, followed by the Data Processing layer that facilitates efficient data access. The stack then features specialized modules for Trajectory Detection and Prediction, as well as Event Detection and Forecasting, which leverage the capabilities of the Data Processing layer to handle complex analytical tasks. Finally, the Visual Analytics layer operates at the top of the stack, serving a dual role in facilitating data exploration by directly accessing underlying data and collaborating with other analytics modules to empower users in exploring, configuring, and validating patterns, models, events, and trajectories.

The H2020 MARVEL[16], Multimodal Extreme Scale Data Analytics for Smart Cities Environments, developed an architecture design for Edge-to-Fog-to-Cloud ubiquitous computing that supports multimodal perception and intelligence for audio-visual scene recognition, event detection and situational awareness in a Smart City Environment. Overall, seven subsystems are interconnected, consisting of 29 technological components with a range of functionalities. The MARVEL architecture

[15] https://cordis.europa.eu/project/id/687591

[16] https://cordis.europa.eu/project/id/957337

enables the collection, analysis, and mining of AudioVisual (AV) data, develops joint representations and models for improved analytics and classification. An optimised E2F2C processing and deployment subsystem based on Knot (previously Karvdash)[17] distributes tasks and services across all levels and possible deployment points/execution sites in the available E2F2C infrastructure.

Commercial mobility analytics platforms have witnessed significant growth in recent years due to the increasing demand for data-driven insights in urban planning, transportation management, and related fields. Some prominent products in this domain include: UrbanSDK, ESRI ArcGIS Velocity, Tetralytics, CARTO, NEC "Data Platform for Hadoop", Hortonworks Data Platform, and Swisscom Heatmaps. Kepler.gl is a powerful open-source geospatial analysis tool for large-scale data sets and similarly *deck.gl* is a WebGL-powered framework for visual exploratory data analysis of large datasets.

In EMERALDS, the reference architecture foresees partial integration with two Mobility Analytics as a Service platforms: on one hand the ATOS (Chapter 5) Mobility AI as a service platform which supports a) interoperability between computed or ingested data from the combined execution of tools in all levels of the architecture, b) seamless data transfer from processing to analytics to visualizations, c) model configuration (training, inference, offloading) for stakeholders, mobility operators and end, tools according to their workflow requirements and explore analytics services on all time-horizons (hindsight, insight, foresight) while being supported in time-critical operations and decision making, on the other hand, the CARTO platform is established in spatio-temporal data manipulation and geospatial data visualization, relying on PostgreSQL advanced query tools combined with a scalable cloud technology stack, presents a user-friendly experience with the developed 'emerald's services and tools targeting non-expert users. As an end point of the architecture, the CARTO MAaaS platform houses data visualizations, dashboards, and VA tools, controlled with appropriate user-defined queries.

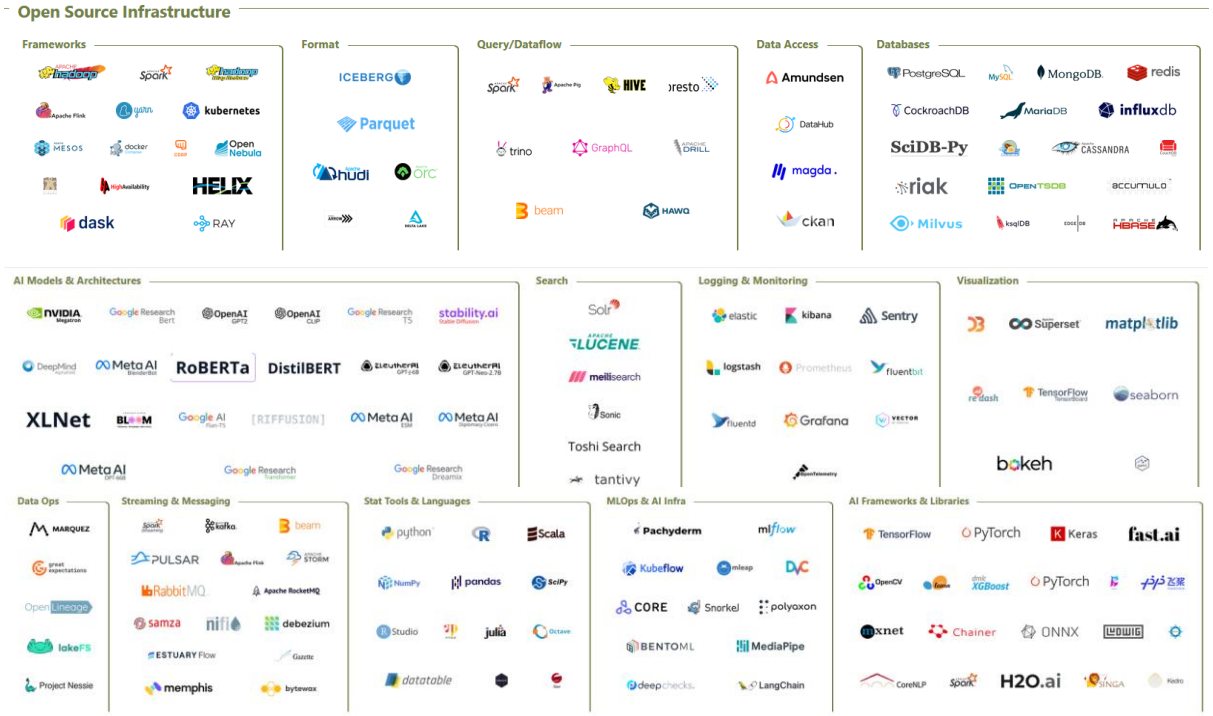

Figure 2-8 Open-Source Infrastructure Technology Stack for the treatment of Big Data Analytics and AI processes[18]

[17] https://github.com/CARV-ICS-FORTH/knot

[18] https://mad.firstmark.com/

3 EMERALDS Services Design Specification

In this Chapter, detailed descriptions of the design specifications for the services that constitute the EMERALDS toolset are provided. Each 'emerald' represents a building block of the reference architecture, catering to a unique facet of the project's overarching mission. Requirements, KPIs and baseline measurements, and component's respective structure are introduced, therefor marking the ground for the next phases of development and testing.

3.1 Privacy-aware in situ Data Harvesting

3.1.1 Privacy Aware Data Ingestion

Human mobility is a very sensitive field of study since it encompasses the continuous patterns of people during their daily lives. As a result, analysts need to be aware that in such scenarios, the user's privacy can be very easily compromised not only due to bad practises during the data ingestion and management phases, but also because adversaries can extrapolate info that is essentially hidden inside the patterns that are formed by moving objects. Thus, the data providers that need to always be aware of this possibility must find ways of combating all the possible data breaches in the most effective way possible. On the one hand, we propose the **use of IoT/Edge processing** as a viable alternative to a centralized only approach that can ensure privacy by keeping prying eyes away from sensitive data while also improving user experience through reduced latency and processing times. Additionally, we are implementing a **continuous mobility-based privacy evaluation service** that constantly detects and possibly mitigates privacy breaches that might occur by a malicious attack performed by an adversary.

Requirements

This 'emerald' offers privacy-ensuring techniques for data collection and ingestion, like continuous privacy monitoring, while ensuring optimal data ingestion using methods like real- time compression and processing. By creating and deploying predefined workflows to the Edge/Fog nodes that are situated near to where data is ingested, this 'emerald' will ensure that user privacy is guaranteed in a twofold manner: i) not transferring possibly sensitive data to a centralized server and ii) by continuously evaluating the data received with respect to privacy breaches that might occur by an adversary (like deanonymisation attacks).

More specifically, regarding the privacy evaluator, our goal is to develop a service that can evaluate the privacy risk for a streaming data source in real time while being seamlessly integrated with the Streaming Orchestrator, as described in section 3.1.2. Following the Streaming Orchestrator workflow, the Real-Time Privacy Evaluator will be a containerized module that can be deployed to any Compute Continuum level (with the Edge being the point of emphasis), following a continuous execution plan that will enable it to provide the user with a constant stream of information regarding the level of privacy risk of each data source. This service will include multiple privacy attacks like the Home-Work attack, the Location attack as well as their variants like the Location Sequence Attack etc (iv). The main objective of this service will be to adapt these paradigms to a real-time scenario and to propose and develop new ones that will provide the user with a more holistic view of the privacy risks of any data source at hand.

Architecture

Current implementations of the attacks are mainly based on interpreted languages like Python. This is a major limiting factor, since Python based tech-stacks are rarely optimised enough to be deployed to a limited-resource environment like the Edge. Our service will be based on Rust a compiled language based on LLVM [19]that is more performant but is also easily integrated to a low-resource or even bare-metal scenario. Our target TRL is level 4.

KPIs

The KPIs that we will put our focus on are:

- The number of records that can be evaluated in real-time depending on the node that is used (Edge vs Fog etc.)
- The total percentage of processing/ingestion jobs that will take place in-situ instead of at the Cloud/Datacenter.

3.1.2 Extreme Scale Stream Processing

One of the main goals of the EMERALDS project is to utilize the compute capabilities of the ubiquitous IoT/Edge computing nodes to offload a lot of the data processing tasks from the (by design) centralized Cloud to the hyper-distributed Fog/Edge infrastructure that is available. To achieve such an ambitious goal, an intelligent Orchestrator that acts as an intermediary between Compute Continuum (CC) levels and devices is needed. Focus is given on the design and implementation of an Orchestrator capable of handling multiple concurrent devices and multiple high-throughput data sources, essentially acting as both a job and a data broker, released as a re-useable software component ('emerald').

The Compute continuum is a network that includes large amounts of devices that need to communicate efficiently with each other to share data as well as to coordinate about the dynamic scheduling of each of the multiple jobs that can be assigned to the network at any one time. Even though this challenge has been studied extensively as part of the distributed computing field (especially for tools like Spark[v] and Hadoop[vi]), in that case the compute nodes that are available adhere to a set of strict rules on their respective architecture, resources and locality. The CC, however, cannot and should not enforce such standards, something that greatly influences the way each node communicates and interacts with each other. To this end, the Orchestrator that will be purpose built for the CC, will be able to handle multiple architectures, be language and platform agnostic and manage issues like moving, redistributing, or assigning nodes and uncertain node availability.

Requirements

This 'emerald' will be built as an API interface over HTTP where one or more nodes could be assigned the role of the host at any given time. By sending specific HTTP requests to the host node, each node can indicate its availability and share metadata regarding the location, computational capabilities, architecture etc. Based on that metadata, the host can assign specific tasks to the node, tasks that are accompanied by access to specific data-streams that the node will have access to. Essentially, the Orchestrator will include two main components, the ***Job Scheduler/Deployer*** and the ***Streaming Data Broker***. The former will be responsible for delegating specific tasks to the appropriate nodes of the CC and the latter will oversee sending and receiving data through multiple high throughput data streams, acting as an intermediary between the many and possibly inconsistent data streams that may be

[19] https://llvm.org/

available. This way, the Scheduler/Deployer combo will be able to scale based on the data at hand, allowing for a single or multiple master nodes based on data locality and traffic (Figure 3-1).

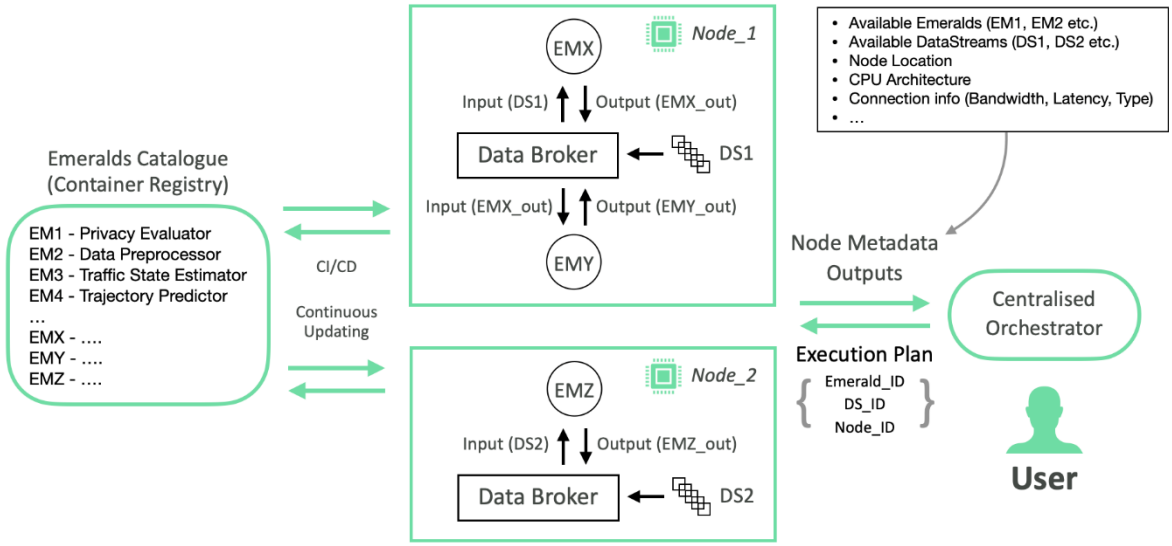

Figure 3-1 - Extreme Scale Stream Processing Architecture

Architecture

Our Orchestrator will be prototyped and implemented primarily using Python with the target TRL being level 4-5. In order to increase performance, other (compiled) languages might eventually be used for specific parts of the implementation. Some processing and analytics modules that will accompany and demonstrate the Orchestrator will also be implemented, regarding areas like data cleaning, compression etc. Of course, as part of the overall integration process, other 'emeralds' that will be developed during the project will take advantage of the streaming and allocation services that the Orchestrator has to offer, including but not limited to the Privacy Aware Data Ingestion, the Trajectory/Route forecasting and the Traffic State Predictor 'emeralds'.

KPIs

The main KPIs of this 'emerald' are:

- Maximizing the total amount of processing and analytics jobs that will be able executed in-situ/at the Edge (global KPI) using the Orchestrator module.
- Minimizing the total latency for each datapoint that is transmitted to and from the Orchestrator.

3.2 Data Fusion and Management

3.2.1 Mobility/Trajectory Data Compression

Due to the large volumes of transportation and traffic data, the application of data summarization techniques becomes necessary for compressing the data while preserving its key characteristics. This enables more efficient storage, faster processing, and effective analysis.

Requirements

This 'emerald' develops both theoretical frameworks and practical prototypes for data summarisation techniques, with the goal of effectively making mobility data suitable for more streamlined analysis. Exploiting the inherent constraints of trajectories within the road network enables the attainment of even more significant compression ratios.

"**Trajectory Simplification**" as one of the suggested summarisation methods for this 'EMERALD', is an approach that removes detailed movements and preserves the essential and key points of trajectories. On the other hand, "**Trajectory Smoothing**" technique targets noise reduction and mitigations irregularities, such as sudden changes in direction. This provides more coherent data, leading to more insightful data analysing. Furthermore, this 'EMERALD' aims to integrate the "**Lossless Data Compression**" method, a strategy that selects data points and retains trajectories based on sampling rate and temporal properties (time-related characteristics). Lastly, a technique known as "**Task-aware Simplification**" is also considered here which dynamically adjusts the level of data summarisation.

The compression results can be assessed using two key performance indicator types, ***data size*** and ***quality***. The former evaluates the compressed trajectories compared to the originals, while the latter examines the quality of outcomes in terms of retaining data integrity. Introducing such metrics is part of the future research within T3.3 and will be presented in D3.1.

Development Process

Furthermore, to provide support for the practical application of the aforementioned techniques, MobilityDB, an open-source geospatial trajectory database system developed by ULB will serve as one of the platforms and testbeds. MobilityDB is an SQL database for moving object trajectories, utilising PostgreSQL's extensibility features to provide the necessary support for storing and querying geospatial trajectory data.

KPIs

- The Compression ratio, as the ratio of the original data size to the compressed data size.
- Data Quality. In case of lossy compression, some metric of assessing the loss of data is needed. As part of the research, what quantitative metrics can be defined for this KPI, such as PSNR (Peak Signal-to-Noise Ratio) or Mean Square Error (MSE) to name a few, is explored.

3.2.2 Sensor Data Fusion

The Sensor Data Fusion component involves integrating road-level data with satellite and terrestrial imagery through the development of data streaming integrators. The purpose of data fusion and management is to achieve a comprehensive view of the transportation and traffic ecosystem by integrating data from diverse sources such as sensors, cameras, GPS devices, and other data collection systems.

Requirements

As a high-level perspective, the procedure of collecting micro mobility data from various data sources to generate Mobility GUIs and applications can be categorised into three main components: **Collectors, Harvesters, and Linkers**. Figure 3-2 below provides an overview of this concept.

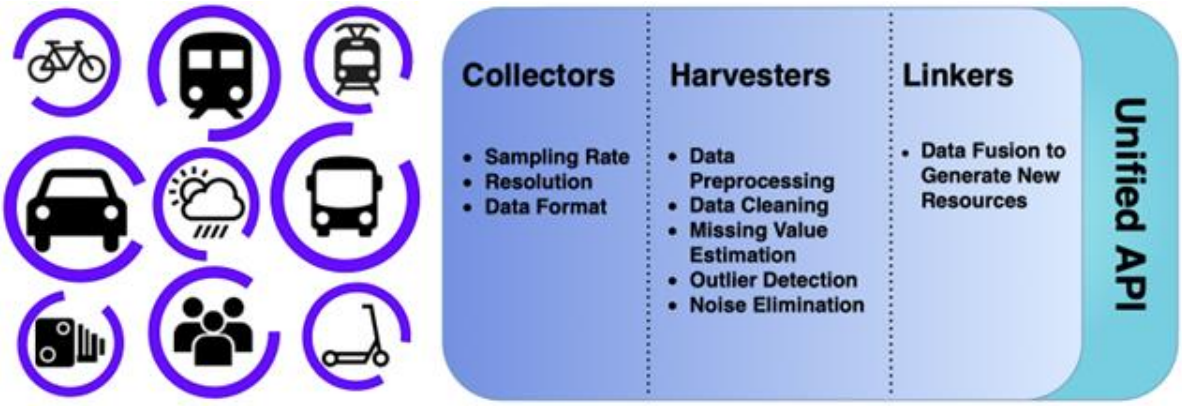

Figure 3-2 - Sensor Data Fusion

To start, the **Collectors** gather data from diverse data sources that introduce heterogeneity in terms of file formats, sampling rates, and data quality. The large scale of urban mobility data, with millions of citizens using location-aware applications, adds further complexity.

Moving forward, **Harvesters** as the following core component, receive the data collected by Collectors and perform a simple transformation of individual sources through data preprocessing. This stage includes tasks such as data cleaning, noise elimination, outlier detection and missing value estimation.

Lastly, the **Linkers** component involves applying data fusion to the pre-processed data in order to produce new resources. The new resources will enrich the existing data, thereby improving subsequent processing task such as mobility analytics. The focus of this 'emerald' is on the last layer of linkers, i.e., data fusion. This layer however is greatly affected by the preceding layers of collectors and harvesters. The methods applied in the three layers need to be consistent and complementary to each other.

As an example, a linker uses the collected number of vehicles in road segments at the time of snapshot to provide the average speed of road segments. This is a difficult process because the recorded data normally does not include vehicle IDs, which makes it challenging to track the vehicles. As a solution, the linker needs to use several collectors and aggregates them to return the desired result.

KPIs

Regarding evaluation of the fused data, the **linkage rate** KPI is employed to quantify the enhancement achieved through the fusion compared to the original distinct data sources. Defining **quality KPIs**, as another type of indicators for this 'EMERALD', constitutes a part of the future research endeavours.

3.2.3 Traffic State Estimation (Multi-Modal)

This 'EMERALD' encompasses an investigation on the macro-scale of estimating and predicting traffic at the road network level to address the issue of missing values within the transportation network data. To elaborate, traffic data collectors encounter random disruptions, such as hardware or software malfunction, that occasionally prevent data uploads. This leads to the "missing value problem" which has negative impacts on data analysis. To overcome the problem, the imputation of missing data has become a necessity. Due to traffic flow dependencies on its spatial and temporal neighbours, using spatiotemporal data is useful **for extracting motion patterns in order to effectively estimate the missing data**. To this end, we leverage the structure of the transportation network as well as the temporal data recorded by collectors at road segments as the base spatiotemporal knowledge.

Requirements

Regarding implementing this 'EMERALD', the project incorporates machine learning and deep learning techniques to analyse mobility data. These libraries provide a range of algorithms and models for spatiotemporal feature extraction, enabling the system to learn from the data and generate accurate results.

Architecture

To this end, state-of-the-art approaches, including the utilization of **Graph Convolutional Neural Networks (GCNNs)** suitable for graph-based structures, have been explored. These cutting-edge techniques contribute to improving the handling of missing values and enhancing the overall analysis of traffic data within the tool.

KPIs

- Prediction Accuracy, by measuring how accurately the service predicts missing traffic data. KPIs can include Mean Absolute Error (MAE), Root Mean Square Error (RMSE), or other regression metrics.
- Imputation Success Rate, which is the calculated percentage of missing values successfully imputed by the service.

3.3 Extreme-Scale Cloud and Fog Data Processing

3.3.1 Extreme-Scale Map-Matching

Map-matching refers to the process of mapping raw positions of moving objects to positions on the road network. Typically, the positions of moving objects are provided from GPS, which is known to be noisy and contain small errors. It is therefore practically infeasible to analyse the movement of a vehicle in an urban area, without first performing the map-matching step, which converts the GPS coordinates to positions on the road network. The problem is far from trivial due to many factors, including the complexity of the road network (e.g., consider two roads in very close proximity, the main road and a parallel side road), the sparsity (low sampling rate) of GPS input positions, the inaccuracy of the input data, etc.

Requirements.

Given a GPS data set of vehicles in an urban area as well as the graph representing the road network, the aim is to map each GPS position to a position on one of the road segments of the road network.

Architecture.

To produce a highly scalable map-matching service, we will design and implement a prototype based on big data processing frameworks, notably **Apache Spark**. The focus will be on techniques for parallelizing processing to exploit a set of available workers, aiming at minimizing the processing time required to perform the map-matching task. As such, the 'emerald' will be implemented as a Spark job that can be executed on a set of worker nodes in parallel. Its input is the GPS data set, which can be provided as a sequence of spatio-temporal points, and an OpenStreet map (OSM) that contains the road network of the urban area of interest. Its output is another GPS dataset that also contains for each input GPS position the mapped position on the road network.

Development Process.

First, we will evaluate existing solutions for map-matching and perform a comparative evaluation. After identifying potential advantages/disadvantages, we will select a method that performs well based on the evaluation criteria (see below). The selected method will be evaluated using both data from the project as well as publicly available data, to guarantee reproducibility. As soon as we have a stable working prototype with acceptable quality results, the next step will be to focus on performance by parallelizing the solution in order to be applicable to extreme-scale data sets.

KPIs

- Quality: measure the accuracy, i.e., the percentage of mapped positions which were correct. Various additional metrics can be used, for example: route mismatch fraction, length index, precision, and recall. This evaluation requires the availability of ground truth, to make the comparison.
- Performance: how fast is the map-matching performed, measured in terms of throughput (processed input records per second).
- Scalability: how more resources can be exploited, i.e., how throughput can be increased when we use more workers.

3.3.2 Weather Enrichment

This 'emerald' pertains to the ***enrichment of spatio-temporal data with the prevailing weather information***. This is offered by an integrator mechanism that accepts as input data longitude, latitude, and data information, and delivers as output the data with weather attributes. The mechanism can be employed either in centralized or distributed environments for large-scale data enrichment.

Requirements.

In order to operate, the 'emerald' needs to have access to weather data. These data are provided by some organizations like NOAA (National Oceanic and Atmospheric Administration) and ECMWF (European Centre for Medium-Range Weather Forecasts) in gridded binary format (GRIB). This is a standardized format by the World Meteorological Organization's Commission for Basic Systems for storing and exchanging historical and forecast weather data. In both centralized or distributed modes, its performance depends on the frequency in which the GRIB files are loaded in memory to access the weather information. Once a GRIB file is loaded in-memory, it can be used for the enrichment of several records. However, subsequent records may require to be enriched from different GRIB files, as a GRIB file covers the globe's meteorological data for a specific temporal period.

Architecture.

The main components of the Weather integrator are a) the spatio-temporal record parser and b) the caching mechanism. The ***spatio-temporal record parser*** is used for reading one-by-one the records that will be enriched with weather information. It is also used for writing the output. The caching mechanism is used for retaining GRIB files in-memory. The components are written in Java programming language. The 'emerald' runs in Java environment as regards its execution in centralized mode. For the distributed mode it can run as a consumer/producer in the Apache Kafka data streaming platform. Specifically, it can consume a topic and produce at the same time the enriched records in another topic. Also, it can read GRIB files in two modes: either locally or from HDFS (Hadoop distributed fie system). It expects records in CSV or JSON format as input. It outputs the enriched records in the same format. It can match as a part of data pipeline with the extreme scale map-matching 'emerald' which enriches spatio-temporal data with road-network information. The current

TRL level is 3 (experimental proof of concept), so the aim is to reach 5 (technology validated in relevant environment).

Development Process.

The development of this 'emerald' includes code refactoring and continuous testing to ensure the correctness of the output results. The development concerns the improvement of the accuracy of the weather attribute values from a GRIB file. So far, the values of the weather attributes were provided by querying a specific spatial cell, the one in which a spatio-temporal point is enclosed in. To get more accurate values, we will develop procedures that will do spatial interpolation on the values of neighbouring cells that have a spatial proximity to a given point.

KPIs

- Performance measurement of the enrichment in terms of execution time.
- Measurement of the difference of the interpolated values with the non-interpolated ones. This will indicate the fixed error.

3.3.3 Spatio-Temporal Querying

This 'emerald' offers spatio-temporal querying, that is scalable querying of big data that have a spatial dimension and a temporal dimension. Based on the needs of the project's use cases, it is possible to **extend the querying** algorithms also towards **spatio-textual data objects**, such as those derived from social media applications (e.g., tweets). More specifically, at the current time, the aim for this 'emerald' is to support a generic operator that takes as input two data sets, namely the spatial distance join operator. The spatial distance join operator takes as an input two data sets (R, S) and identifies the object pairs $\{r_i, s_i\}$ whose distance is at most equal to a user-specified threshold ε (i.e., $dist(r_i, s_i) \leq \varepsilon$). Application examples of this operator can be found in urban planning, cartography, neuroscience and astrophysics.

Requirements

Given two data sets and a distance threshold ε, the aim is to identify the object pairs {ri, si} that fulfil the condition $dist(r_i, s_i) \leq \varepsilon$. A naive solution would be to calculate the distance of all objects and then keep those that are equal or less than a threshold O(|R|*|S|). For small data sets this would be adequate, but for large sets the cost is prohibitive. For this reason, spatial partitioning techniques are employed, from which the computational cost breaks in partitions. The partitions are assigned to a number of nodes and are processed independently.

Architecture

The Spatio-temporal querying 'emerald' is based on **parallel processing** in order to support scalability. With this in mind, the design takes into consideration the aspects of a parallel environment. The implementation will rely on the Apache Spark big data processing framework. As an input, two sets of spatial points will be provided, stored in a distributed file storage system, such as HDFS. The current TRL level is 3 (experimental proof of concept), so the aim is to reach 5 (technology validated in relevant environment).

Development Process

The spatial distance join operator will be developed as a job that will accept as an argument the full paths of the two data sets and the distance threshold. The implementation will take place on the Apache Spark Framework. Also, a brute-force approach will be applied in order to validate the correctness of the results.

KPIs

- Performance measurement of the spatial distance join in terms of execution time.
- Measurement of the size of data that was transferred on the network due to shuffling for the computation of join.

3.3.4 Hot-Spot Analysis

Hot-spot analysis is the process of identifying spatial or spatio-temporal areas with high concentration of objects, where the high value is statistically significant. By objects we refer to entities that have a spatial position (x, y) at a specific timestamp (t). The areas are typically application-dependent 2D boxes (in the case of static spatial data), or 3D cubes (in the case of moving objects). Examples include hot spots in urban areas based on tweets, hot spots at sea based on the movement of vessels, etc. An interesting extension of hot-spot analysis for urban environments concerns the ***identification of congested road segments***, which is the primary focus of this 'emerald'.

Requirements

Given a GPS data set of vehicles in an urban area as well as the graph representing the road network, the aim is to identify the top-k hot-spots road segments, that is road segments with statistically significant high concentration of vehicles. It is important to note that hot-spot analysis goes well beyond plain counting techniques (e.g., based on location) and that the temporal nature is important (i.e., a congested road may cause other neighbouring roads to become congested in the near future).

Architecture

The 'emerald' for hot-spot analysis is based on parallel processing to ensure scalability. Its design is in accordance with data-parallel frameworks for big data processing and analysis that rely on **MapReduce**, such as Apache Spark. As such, the 'emerald' will be implemented as a Spark job that can be executed on a set of worker nodes in parallel. Its input is the GPS data set, which can be provided as a sequence of spatio-temporal points, and an OpenStreet map (OSM) that contains the road network of the urban area of interest. Its output is a set of k road segments, each associated with a hot-spot value.

Development Process

First, pre-processing is required to map the GPS records to road segments and assign to each position a value that indicates congestion. This value will be related to the speed of the vehicle, and its comparison with the free flow speed of this road segment will be an indicator of congestion. Then, the graph that represents the road network will be ready for processing, so as to **compute for each road segment the hot-spot value and return the top-k hot-spots**.

As this is a problem with open research questions, the TRL level is relatively low, so the aim is to reach TRL4.

KPIs

- Performance measurement hot-spot analysis in terms of execution time.
- Trade accuracy for performance, i.e., provide approximate hot-spots that are very close to the exact ones.
- Check if hot spots are rationale in comparison with a simple counting approach for result quality. Also, visual inspection of the discovered hot-spots on a map by a human.

3.4 Extreme Scale Mobility Data Analytics at Computer Continuum

3.4.1 Trajectory/Route Forecasting and Origin/Destination Estimation

The purpose of this 'emerald' is to produce reliable models of urban movement by utilizing the Compute Continuum and its capabilities to forecast future locations, trajectories, and routes, focusing on efficiency and effectiveness. By leveraging advanced data analytics and machine learning/AI techniques – including Deep Learning (CNNs, RNNs etc) and Machine Learning based ones (Ensembling, boosting, etc.) - this ''emerald'' aims to provide transformative insights into the complex urban mobility landscape of modern EU cities. Through this approach, the ''emerald'' seeks to address the challenges posed by the dynamic and complex nature of today's large scale urban environments. By understanding and predicting movement patterns, it will enable urban planners, policymakers, and businesses to optimize transportation systems, reduce congestion, and enhance overall urban liveability.

Requirements

The models that will be part of this ''emerald'' will be designed with accuracy and efficiency in mind, targeting computational platforms that can provide timely inference near to where data is harvested and processed. More specifically, these models will include:

- **Forecasting Future Locations, Trajectories and Routes**: By analysing vast volumes of historical movement data, this set of models will be able to predict where individuals and objects are likely to be in the future, with prediction horizons ranging from short to long term forecasting. By utlizing the future location prediction method, the ''emerald'' will predict the paths that moving objects, such as vehicles and pedestrians, may take in a given urban environment, providing insights into their complete future trajectories and routes.
- **Origin/Destination Estimation**: Models will be developed using historical data from urban public transport trips, with the goal being the efficient and accurate estimation of entry and exit bus stops.

Architecture

The tech stack that will be used is primarily **Python based**, as are most ML/AI tools that are developed today. If and where needed, other options that might be more optimized for low- powered scenarios will be explored. All the models will be tested on devices from all CC levels (Edge, Fog, and Cloud), especially with inference in mind. Regarding integration, the Stream Orchestrator (as described in section 3.1.2) will be used by each model to be easily deployed and given access to data (in the form of data-Streams). Target TRL for this ''emerald'' is 4.

KPIs

The KPIs that will be used to track and report progress include both quality (I.e., accuracy/loss) and performance based (I.e., inference times) measurements. Additionally, the global KPI about analytics that need to be performed in-situ will also be tracked.

3.4.2 Probabilistic Approach For Trip Chaining

This 'emerald' is a **probabilistic model**. It enhances existing approach by estimating which route probabilities are more likely given existing data. The model will be able to output route estimation for new input data as well. This model will be developed both locally and in the cloud. It will run in the cloud.

Business Requirements

[D5.1 UC3] It will improve the determination of entry and exit stops thus making the ridership estimates more robust. This estimation will be used for optimizing Riga's transport network and its efficiency.

Functional requirements

The model should determine entry and exit stops based on input data. Currently available data are ticket validations, vehicle messages (GPS data) and route network (schedule) data – GTFS.

Architecture

This 'emerald' is developed in Python and mainly interacts with the MAIaaS (ML Development & MLOps) platform (T4.3) shown in Figure 5-1.

The following inputs and components are required to develop the model:

- Historic training data in files or databases available through the MAIaaS platform data storage / repository
- Model development, experimentation, evaluation, and monitoring components (incl. Jupyter notebooks and appropriate python libraries for probabilistic modelling)
- Distribution component to deploy the trained model for inferencing and store it in a model repository.

The following inputs and components are required to run the model inference on live data:

- Live data available via APIs (may be simulated, if necessary)
- Trained models are served either by the MAIaaS platform or can be downloaded by the EMERALDS public registry.
- A front end to display the prediction results is required.

Development process

Development will start with a basic model using only the latest historical data. The data that will be used to develop the final model will be decided after data preprocessing where some data will be excluded (for example: erroneous values or non-representative behavior during the COVID-19 pandemic). During the validation process, the aim will be to improve the existing model developed by Grupa Ltd, (G93). Current TRL and its advancement will be defined after current model handover.

KPIs

- Percent point improvement over current algorithm's entry & exit stops estimation percentage rates.
- Performance metrics, Ingestion throughput and input data size.

3.4.3 Trajectory Data / Travel Time Analysis

This 'emerald' is a **data science library**. It offers **trajectory analytics** by leveraging and further improving the open-source Python library **MovingPandas** which covers trajectory data processing (such as trip/stop extraction; speed and travel time computation; trajectory smoothing; outlier detection) as well as visualization (in Jupyter notebook environments and Panel data apps). Specific envisioned improvements include, for example, more better trajectory cleaning options as well as support for network-constrained movement data.

It addressed the following **Business requirements:**

- [UC3] Excess route segment travel times analysis enables the quantification of the impact of delays on public transport passengers.

- [EAD2] Trajectory analytics and visualizations broaden the analytical toolbox of the CARTO spatial data science offering.

Requirements

This 'emerald' serves an essential role in the data science workflow cycle, (as illustrated in Figure 3-3 since it enables rapid prototyping of data preprocessing, cleaning, and exploratory data analysis for movement data.

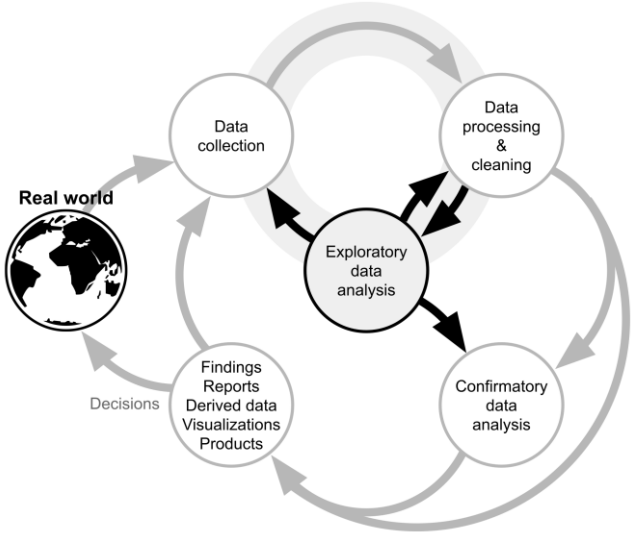

Figure 3-3 - Data science workflow cycle. Image source[20]

Architecture

This 'emerald' is developed in Python. MovingPandas builds on GeoPandas[21] and uses HoloViz[22]/GeoViews[23] tools for interactive visualizations. The full list of dependencies is available in:

https://github.com/movingpandas/movingpandas/blob/main/environment.yml

If no visualizations are required, the following shorter list of requirements may be used:

https://github.com/movingpandas/movingpandas/blob/main/environment-minimum.yml

MovingPandas can read from and write to all file formats and databases supported by the underlying Pandas and GeoPandas libraries. The minimum information required are x, y, t, id (where x/y are the coordinates, t is the timestamp, and id is the ID of the moving object or trajectory), for example a CSV file that looks as follows:

```
id; x; y; t
1; 0; 0; 2022-01-01 12:00:00
1; 1; 1; 2022-01-01 12:05:05
2; 3; 2; 2022-01-01 12:01:00
```

[20] Graser & Dragaschnig (2020) Open Geospatial Tools for Movement Data Exploration

[21] https://geopandas.org

[22] https://holoviz.org

[23] https://geoviews.org

Alternatively, GIS file formats (such as GeoPackage and Shapefile) or GPX files may be used as input. Additionally, MovingPandas support OGC MovingFeatures JSON.

The library may be used, for example, on the ATOS MAIaaS platform to facilitate model development or in CARTO data science notebooks.

Development process

This 'emerald' is developed as an open-source library on Github.

The TRL goal for new functionality added to this 'emerald' is TRL4-5.

KPIs

- Performance measurement of the trajectory cleaning in terms of execution time.
- Improved time to first result for network-constrained movement analyses

3.4.4 Real-Time Extreme Scale Map Matching

The PTV platform allows customers to get access to the PTV product portfolio. The PTV analytics tool application that will be used in the project, is deployed into the PTV platform, which is hosted in Microsoft Azure. The application has three major contact points with the PTV platform. If users open the PTV analytics tool in their browser, they first retrieve the basic single page application without any data from a hosting service. The page will then authenticate the user using the authentication service of the PTV platform. Afterwards, the page can use its now authenticated connection to communicate with the PTV platform API management that facilitates the **access to platform services within Kubernetes clusters**. This allows access to the PTV analytics tool API services and to their functionalities.

Requirements

For the EMERALDS project, a tool called "Extreme Scale Map Matching" will be designed to **map-match GPS data on a static map to retrieve trajectories and estimate the speeds on the graph network links**. The minimum input data will be DeviceId, Event Time in UTC, Coordinates, which will be ingested as a real time data stream. The static map (street graph) will be ingested as a file. Ten parallel clients are estimated to request with a rate of one request per minute giving output binary rate of about 50 MB/seconds. The tool will interface with external applications over REST API.

Architecture

The "Extreme Scale Map Matching" will be developed as part of the project toolset. The tool will be deployed into a Microsoft Azure environment, and it will be used as a proxy to the PTV analytics tool REST API services which allow the integration with external applications using the provided API keys. Figure 3-4 describes the aforementioned process.

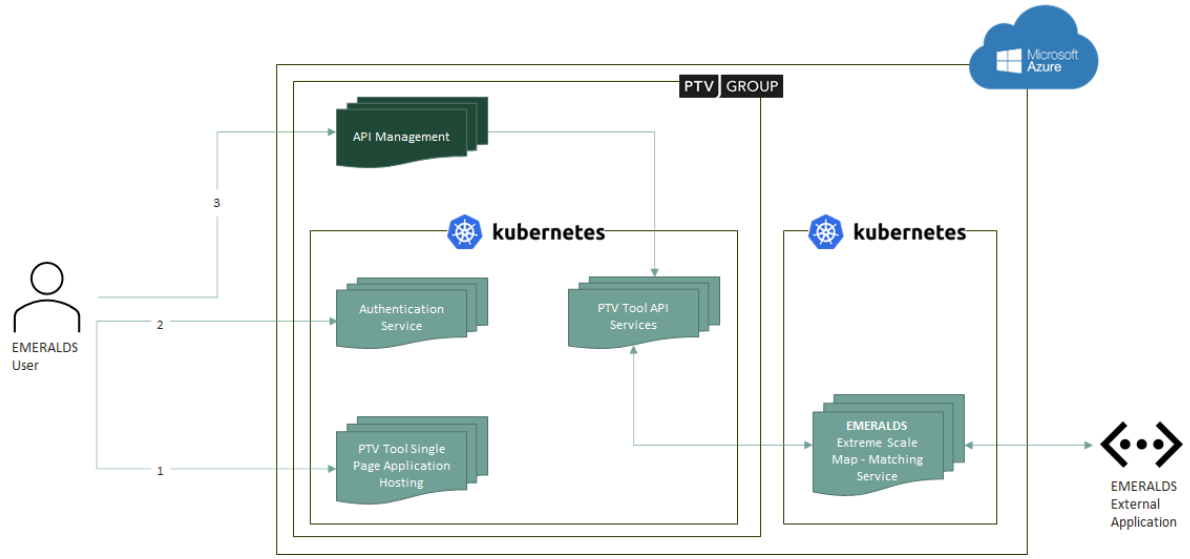

Figure 3-4 - PTV and Extreme Scale Map Matching Architecture

KPIs

- Data Processing Latency: the latency between having a new data point available at the input and having the relevant speeds updated on the output is supposed to be smaller than the state of the art.
- Memory usage
- The expected data source input rate for the project is 2k data points/s and the tool will be designed to scale up to 1M data points/s

3.5 Active & Federated Learning Over Mobility Data

This section will mainly cover the federated and active learning as defined in the T4.2. consisting of:

1. **Forecasting Models**
    - Traffic state / flow
    - Crowd density
    - Parking garage occupancy
2. **Active Learning & XAI**
    - for crowd/flow forecasting
    - for risk classification
3. **Federated Learning (FL)**
    - In the cloud and using edge devices

3.5.1 Traffic State / Flow Forecasting

This **forecasting model 'emerald'** offers traffic state prediction using NN; prediction horizon: 15-30 min in the future (to enable anticipating traffic measures)

Requirements

The purpose of this 'emerald' is the development of a set of models that target a) traffic state and b) traffic flow forecasting. With the large number of sensors that moving objects are equipped with today, vast amounts of movement data are generated daily. Combining that with the more ample and easily available access to compute that municipalities and organisations/companies have today, creating ML/AI models that can accurately describe, analyse and forecast movements is easier and more valuable than ever.

In a highly urbanised environment, where people move constantly from place to place using a predefined/fixed network of roads, accurately modelling and forecasting traffic can provide value in more ways than one. The importance of traffic state/flow forecasting can be seen through its impact on:

- **Public safety.** Congestion increases the likelihood of accidents. Combining that with the fact that emergency response efforts can face delays due to the same traffic congestions, it is clear that effective forecasting models can not only reduce overall congestion but can also emergency services to pick the fastest route while proposing measures that can alleviate traffic pressure, allowing help to arrive sooner.
- **The economy.** Traffic congestions increase fuel consumption and decrease productivity, driving up transportation costs for businesses and citizens. Additionally, time wasted in congestion greatly impacts traveller satisfaction, something that can lead to long term loss for businesses that rely on tourism and hospitality. By providing accurate information regarding traffic state, drivers can choose alternative routes, reducing overall congestion. Travellers can also know what to expect in terms of traffic related delays, allowing them to plan their trips better, leading to a better overall experience.
- **The environment.** Congestion can lead to increased emissions of pollutants that harm the environment (around 40% of global emissions come from cars). Using traffic forecasting models can enable urban planners to build more optimised road networks, decreasing congestion and, consequently, reducing emissions.

Architecture

The design and implementation process of the aforementioned forecasting models, both traffic state and traffic flow forecasting methods will be considered. The difference between the two lies on the forecasting target, with the former predicting overall congestion at specific road segments and the latter, predicting the rate of traffic flow between specific road segments or intersections. Machine Learning methods like ensembling (ex. LightGBM[vii], XGBoost[viii] etc.) and Deep Learning methods like **Recurrent, Convolutional and Graph Neural Network** will be developed and tested. The large-scale and heterogenous data sources that will be provided by the project partners will be used, creating comprehensive training and validation sets that include large sets of useful features.

Regarding the tech stack, Python will be used because of its specialization in ML/DL and AI. Additionally, the models will be turned into standalone modules that will be deployed to the CC through the Orchestrator module that is described in section 3.1.2.

KPIs

The KPIs for this 'emerald' are based on **prediction model accuracy** metrics like RMSE and MAE.

3.5.2 Crowd Density Forecasting Model

This 'emerald' is a **machine learning model**. It offers short & mid-term crowd-density forecasting. This model will be developed and run in the cloud. It addressed the following **Business requirements:**

- [UC1] Crowd predictions help to improve the short and mid-term personnel planning for crowd management (i.e., how many crowd managers should be on duty on a given day and where should they be deployed)
- [UC3] Crowd predictions help to improve the planning of public transport offerings and may be used to inform passengers.

Requirements

Given the current weather forecasts, event calendar, and knowledge of historic crowd density patterns, the models should forecast the crowd density for the next hours / days. (More input data sources may be added as they become available extending the feature space of the ML algorithms.)

Architecture

This 'emerald' is developed in Python. Its main interactions are with the MAIaaS (ML Development & MLOps) platform (T4.3) shown in Figure 3-5, which is a subset of the MAIaaS architecture as presented in Figure 5-1 - EMERALDS' MAIaaS architectureFigure 5-1

The following inputs and components are required to develop the model:

- Historic training data in files or databases available through the MAIaaS platform data storage / repository
- ML development, experimentation, evaluation, and monitoring components (incl. Jupyter notebooks and appropriate Python environments using PyTorch or TensorFlow stacks)
- Distribution component to deploy the trained model for inferencing and store it in a model repository

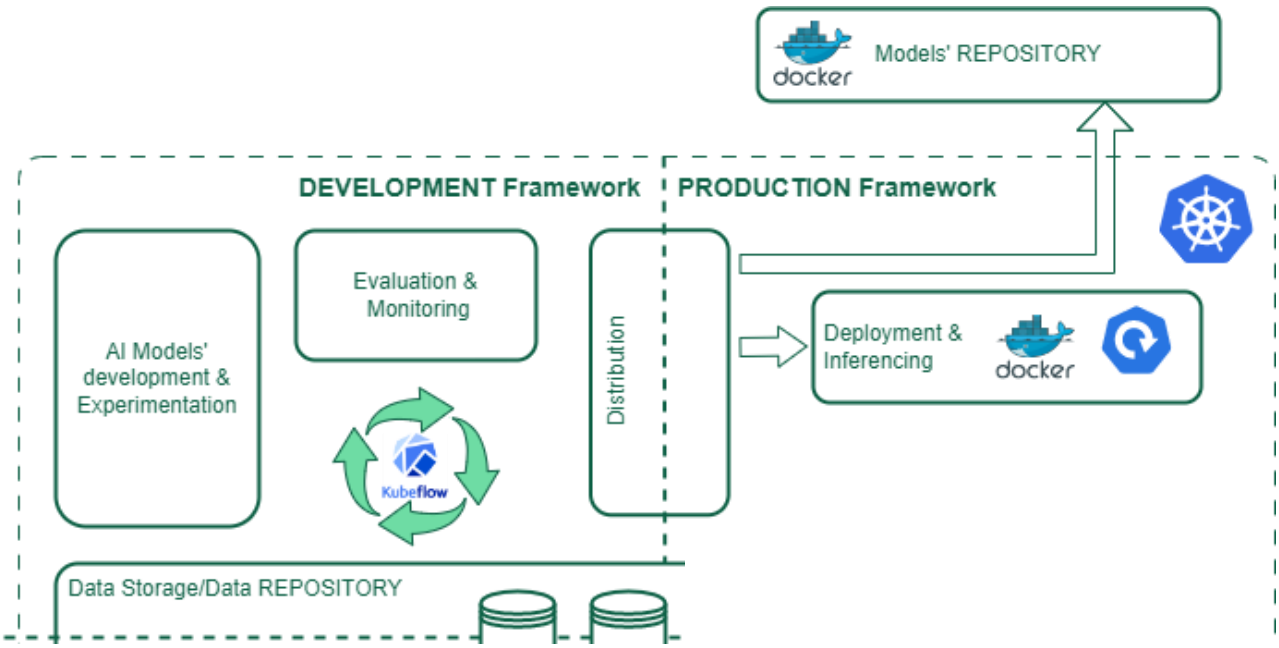

Figure 3-5: Key MAIaaS components for ML model development – Refer to

The following inputs and components are required to run the model inference on live data:

- Live data available via APIs (may be simulated, if necessary)
- Trained model served by the MAIaaS platform
- A frontend to display the prediction results is required.

Development process

Starting from a basic mid-term model with few data source, the model development aims to add increasingly more data source, refine the model architecture, and branch into short-term prediction. Consequently, there will be multiple versions of the prediction models. Which datasets will be integrated depends on the data feasibility. With the term data feasibility, we referring to the potential exists in the data and whether they can be used to make better decisions.

The TRL goal of this 'emerald' development is to advance from TRL 2 4.

KPIs

- **Metrics:** Prediction accuracy (e.g., RMSE, MAE, MAPE)
- **KPI:** improvement over state of the art
- **Baselines:** Baselines will be established using well-known time series forecasting approaches, such as ARIMA. Where possible, existing crowd forecasting models will be used as additional baselines.
- **Limitations:** Since model performance is a function of model design, training data quality, and the complexity of the (crowd) behavior phenomenon in the region of interest, the possibilities for comparisons of models trained for different regions with different data are limited.

3.5.3 Parking Garage Occupancy Forecasting Model

This 'emerald' is a **machine learning model**. It offers mid-term **parking forecasting** and addresses the following **business requirement:**

- **[D5.1 UC1] [Objective 3]** Parking garage occupancy predictions for the four garages at Scheveningen Beach for the next 10 days help manage the traffic.

Requirements

Given the current weather forecasts, event calendar, and knowledge of historic parking occupancy, the models should forecast the parking occupancy for a predetermined prediction interval, e.g., the next x days. (More input data sources may be added as they become available.)

Architecture

Like other models in this section, this model is developed in Python. Its main interactions are with the MAIaaS (ML Development & MLOps) platform (T4.3) shown in Figure 3-5.

The following inputs and components are required to develop the model:

- Historic training data in files or databases available through the MAIaaS platform data storage / repository
- ML development, experimentation, evaluation, and monitoring components (incl. Jupyter notebooks and appropriate Python environments using PyTorch or TensorFlow stacks).
- Distribution component to deploy the trained model for inferencing and store it in a model repository.

The following inputs and components are required to run the model inference on live data:

- Live data available via APIs (may be simulated, if necessary)
- Trained model served by the MAIaaS platform.
- A frontend to display the prediction results is required.

Development process

Starting from a basic mid-term model with few data sources, the model development aims to add increasingly more data sources, refine the model architecture, and branch into short-term prediction. Consequently, there will be multiple versions of the prediction models. Which datasets will be integrated depends on the data feasibility.

The TRL goal of this 'emerald' development is to advance from TRL 2 to 4.

KPIs

- Metrics: Prediction accuracy (e.g., R2, RRMSE, MAE, MAPE)
- KPI: improvement over state of the art
- Baselines: Baselines will be established using well-known time series forecasting approaches, such as ARIMA or GradientBoostingTrees.
- Limitations: Since model performance is a function of model design, training data quality, and the complexity of the (parking) behavior phenomenon in the region of interest, the possibilities for comparisons of models trained for different regions with different data are limited.

3.5.4 Active Learning & XAI For Crowd/Flow Forecasting

This 'emerald' encompasses **AL&XAI tools for crowd/flow and parking forecasting models** presented in previous sub-sections. Therefore, this 'emerald' addresses the following **business requirements**:

- **[D5.1 UC1 – UC3]** The forecasting models should be trustworthy and explainable and accept expert input.

Requirements

Given the current weather forecasts, event calendar, and knowledge of crowd and density flows, the model's performance will be iteratively improved through AL&XAI (expert input) to forecast the short/mid-term flow and density of the crowd more accurate.

Architecture

This 'emerald' is developed in Python. The AL&XAI user interface (using e.g., AL library GRAFANA and XAI library SHAP) will be developed in Jupyter notebooks or data apps (using e.g., Panel apps that can be run inside the MAIaaS Jupyter Lab). The main methodology is illustrated in Figure 3-6.

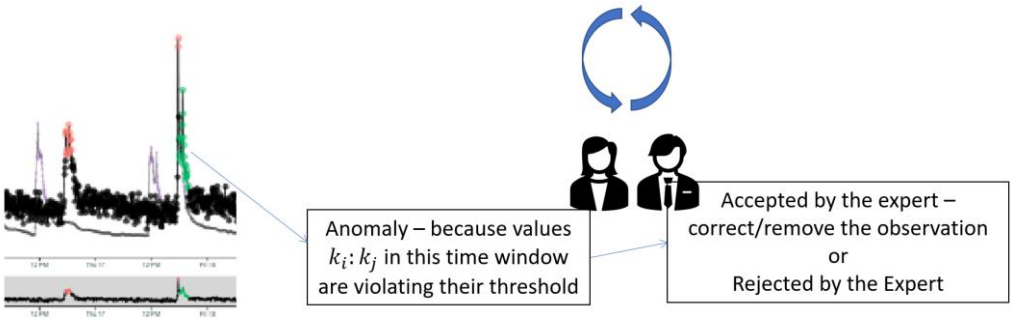

Figure 3-6 - Illustration of the eXplainable Active Learning Framework for a regression task

The following inputs and components are required to develop the AL&XAI methods:

- Platform for Python-based AL development (Suitable Python environment & Jupyter Lab instance in MAIaaS)
- Distribution component to deploy the trained model for inferencing and store it in a model repository.

The following inputs and components are required to run the model inference on live data:

- Live data available via APIs (may be simulated, if necessary)
- Trained model served by the MAIaaS platform.
- Finally, to engage the UC partners in the active learning workflow, a suitable frontend is required. (A set of samples will be automatically selected and presented to the data expert. The criteria for sample selections are, samples that the model classify/forecast with low confidence and/or samples that are falsely predicted/forecasted.)

Development process

AL&XAI methods for mobility use cases will be developed based on the current state of the art in time series and GeoXAI research. The TRL goal of this 'emerald' development is to advance from TRL 2 to 4.

KPIs

- Metrics: Explanation quality (e.g., Fidelity score) and prediction accuracy (e.g., R2, RRMSE, MAE, MAPE)
- KPI: improvement over state of the art
- Baselines: The effectiveness of the proposed framework will be validated through a comparative analysis against the following state-of-the-art baseline methods such as QBC (Query by Committee), D-QBC (Density-Weighted Query by Committee), EMCM (Expectation-Maximization Clustering Method)).

3.5.5 Active Learning (AL) Model For Risk Classification

This 'emerald' is a **machine learning model**. It offers risk classification based on training data provided by UC partners. It addresses the following business requirement:

- [UC1] Risk classification helps with personnel planning for crowd management. Classification rules provided by UC domain experts may be refined through AL.

Requirements

Given the current weather forecasts, event calendar, and knowledge of crowd and density flows, the models should detect and classify the risks as defined by the police/experts. The AL component will query expert feedback in situations where the machine learning model cannot make a clear decision.

Architecture

Similar to the previous sub-section, this 'emerald' is developed in Python. The AL&XAI user interface (using e.g., AL library AliPy) will be developed in Jupyter notebooks or data apps (using e.g. Panel apps that can be run inside the MAIaaS Jupyter Lab). The main methodology is illustrated in Figure 3-7.

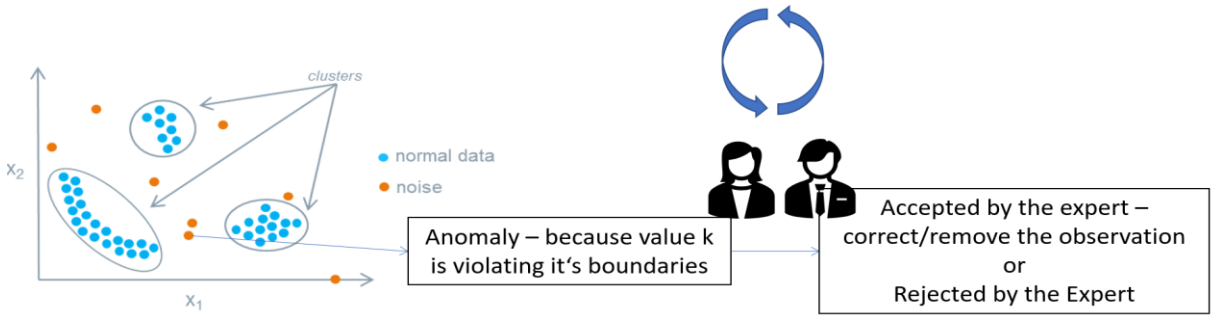

Figure 3-7 - Illustration of the eXplainable Active Learning Framework for a Classification task

The following inputs and components are required to develop the model:

- Historic training data in files or databases available through the MAIaaS platform data storage / repository
- Platform for Python-based AL development (Suitable Python environment & Jupyter Lab instance in MAIaaS)
- Distribution component to deploy the trained model for inferencing and store it in a model repository.

The following inputs and components are required to run the model inference on live data:

- Live data available via APIs (may be simulated, if necessary)
- Trained model served by the MAIaaS platform.
- To present the errored sample predictions to the use case partners, a frontend to display the data and their explanations is required.

Development process

The development of this model depends on the preparation of suitable training data by the UC partner. During the model development process, the uncertain samples are automatically selected, presented, and explained to the data expert. Consequently, the process will continue to refine the model architecture, and/or enhance the data quality. AL&XAI methods for mobility use cases will be developed based on the current state of the art in time series and GeoXAI research. The TRL goal of this 'emerald' development is to advance from TRL 2 to 4.

KPIs

- Metrics: Explanation quality (e.g., fidelity score) and Prediction accuracy (e.g., F1-score, AUC and classification accuracy)
- KPI: improvement over state of the art
- Baselines: The effectiveness of the proposed framework will be validated through a comparative analysis against the following state-of-the-art baseline methods such as QBC (Query by Committee), D-QBC (Density-Weighted Query by Committee), EMCM (Expectation-Maximization Clustering Method)).

3.6 Security and Data Governance Layer

The Security and Data Governance Layer in the EMERALDS project, which is directly tied to "Task 2.3 Security & Data Governance" (M1-M36) and D2.6 "Security and Data Governance Layer" (due M24), is the foundation of the framework's cyber security modules and enhancements, aiming to ***secure the project's devices and transmitted data, and alert and deflect any potential cyber threats***. As such, the Security and Data Governance Layer is different from the rest of the EMERALDS services that the

project will provide, as it consists of a layer running “horizontally” across all devices and services. Our goals are to establish (i) a secure environment by ensuring trust between edge nodes, (ii) secure communication channels for the protection (encryption/decryption) of the data flow across edge nodes and services, and (iii) a monitoring system that inspects packet flows, identifies threats and generates alerts.

TUC’s input in WP2 “Reference Architecture and Toolset Integration” is mostly technical and aims to deliver at least two (2) components for the provision of cybersecurity protections and measurements on the edge nodes. More specifically, TUC is exploring the deployment and use of (a) secure and isolated environments known as Trust-Execution Environments, that can provide multiple functionalities with regards to cybersecurity e.g., trusted code execution, remote attestation, and secure communication between trusted entities and (b) the portability of an intrusion detection system to hardware devices. It is also important to note that regarding secure communication channels TUC will also explore possible applications of Virtual Private Networks (VPNs) and lightweight cryptographic libraries, ensuring that all nodes’ traffic, irrespectively of each node’s limitations, will be transmitted securely in an encrypted form.

In the following subsections we provide information and technical details regarding the components that will be incorporated into the EMERALDS Security and Data Governance Layer.

3.6.1 Trust-Execution Environment

A Trust-Execution Environment (TEE), or Trusted Execution Environment, is a secure and isolated environment that lies in the hardware chip within a computing device. It enhances security by providing a higher level of isolation for trusted code execution and attestation, and is typically implemented in hardware, firmware or both. The purpose of a TEE is to establish a trusted area within a computing system and is commonly used for the trusted execution of critical applications and the protection of sensitive information.

Requirements

A TEE offers several key features that enhance security:

- **Isolation**: The TEE creates a separate execution environment that is isolated from the rest of the system, including the operating system and other applications. This isolation prevents unauthorized access or interference with the trusted code and data.
- **Trusted Code Execution**: The TEE ensures that the code executed within the environment is protected against tampering, unauthorized modifications, or reverse engineering. It verifies the integrity of the code before execution and ensures that it has not been tampered with.
- **Secure Data Storage**: The TEE provides a secure storage area, often referred to as a secure enclave or secure container, where sensitive data can be stored and accessed only by authorized processes running within the TEE. The data is encrypted and protected from unauthorized access, even if the device is compromised.
- **Secure Communication**: The TEE enables secure communication channels between the trusted environment and external entities, such as other devices or remote servers. This ensures that sensitive data and cryptographic operations can be performed securely, protecting the confidentiality and integrity of the communication.
- **Attestation**: TEEs often provide mechanisms for attesting the integrity and security of the trusted environment to external entities. This allows other parties to verify that the code and data within the TEE have not been compromised or tampered with.

Architecture

TEE technology can be used in various devices and applications, including mobile devices, Internet of Things (IoT) devices, cloud computing, digital rights management, secure enclaves for executing sensitive algorithms, and secure authentication mechanisms. Examples of TEE implementations include ARM TrustZone and Intel Software Guard Extensions (SGX).

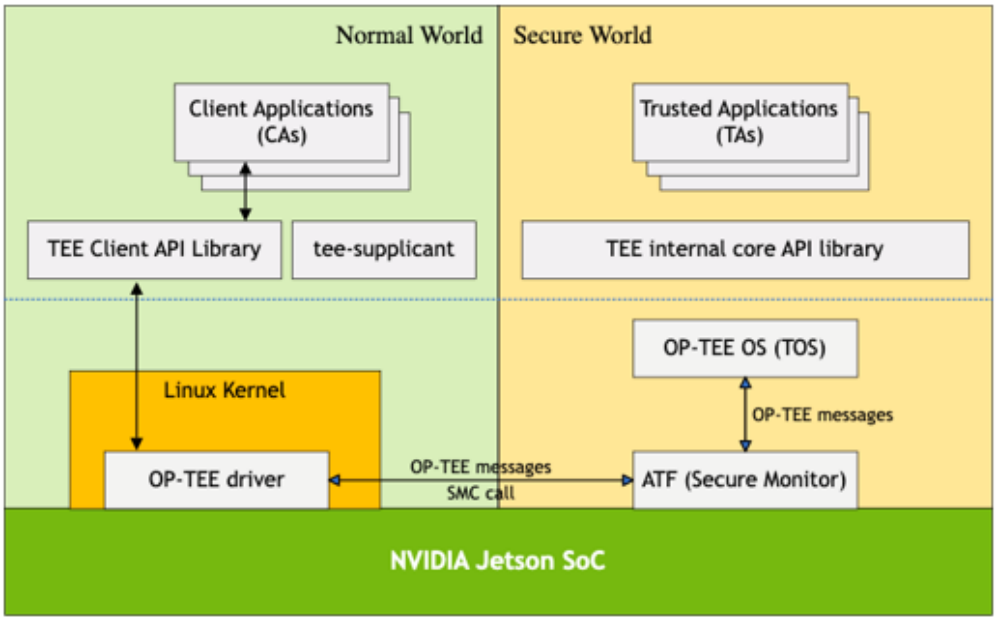

Figure 3-8 - Communication between Normal World and Secure World [24]

For the TrustZone API for example, the main focus is on Trusted Applications, which are programs that are executed securely. These applications cannot be accessed or altered, and as it can be seen in Figure 3-8 they can only be invoked by the Normal World or the Client. When the Secure World receives a request for a specific trusted application execution, it will respond accordingly to the Normal World as the execution is finished. It should be noted that each device (that is compatible) can establish a TEE, but it is unfeasible to have a single TEE for multiple edge nodes, as each trusted OS lies in the hardware chip of each device.

In EMERALDS, as part of the "Security and Data Governance Layer", TUC is exploring remote attestation techniques, for both ARM TrustZone and Intel SGX, in order to verify edge nodes and ensure their trustworthiness utilizing a Trusted Execution Environment. The protocol design that we follow for remote attestation will enable a server to send a request to an edge node, and in turn the corresponding trusted application running in the nodes' TEE will execute the protocol in the secure OS and respond to the server. In the case of an attack or compromise, the edge node will respond incorrectly to the server's request, or not respond at all, meaning that it cannot be trusted any more.

[24] https://docs.nvidia.com/jetson/archives/r35.1/DeveloperGuide/text/SD/Security/OpTee.html

3.6.2 Secure Communication Channels

Requirements

The second pillar of the "Security and Data Governance Layer" aims to ensure that all communications and data flows between the project's devices are secure. In the following we describe the mechanisms that TUC explores and develops in order to ensure communications security between edge nodes and the project's centralized entities. Since we are still early in the development of this component, we are currently exploring various technologies that could enable secure communications and assessing their performance, advantages and drawbacks. The most promising ones are presented in the following.

Architecture

Secure Communication channels via TEE

Initially, secure communication channels can be established by leveraging the previously presented component of the TEE, as it offers secure storage for storing key-pairs and encryption/decryption functionalities. In general, we wish to explore whether those properties can be applied for communication between multiple edge nodes. More specifically, it can offer:

- Encryption: To ensure secure communication, encryption is essential. The use of strong encryption algorithms such as AES (Advanced Encryption Standard) or RSA (Rivest-Shamir-Adleman) to encrypt the data transmitted between the communication endpoints within the TEE. Encryption ensures that even if the communication is intercepted, the data remains unreadable.
- Key Management: Proper key management is crucial for maintaining the security of the communication channel. Safely generate, store, and handle encryption keys within the TEE environment. Keys should be securely provisioned, rotated regularly, and protected against unauthorized access.
- Secure Channels for Data Transfer: If the TEE communicates with entities outside the trusted environment, ensure that the channels used for data transfer are also secure. Use encrypted channels such as VPN (Virtual Private Network) connections, IPsec (Internet Protocol Security), or secure messaging protocols.

Secure Communication channels via VPN

A Virtual Private Network (VPN) is a technology that establishes a secure and encrypted connection between a device and a remote server. This connection allows devices to access the internet or other resources while ensuring their online activities are private, secure, and protected from potential threats such as hackers, data snoopers, and government surveillance. VPNs are commonly used to enhance privacy, bypass geo-restrictions, and maintain data security, especially when using public Wi-Fi networks. TUC will explore the VPN solution for secure communication channels by developing upon open source tools i.e. OpenVPN, WireGuard.

Additionally, since we are dealing with edge devices and some of them might have limited computing capabilities and not supporting a TEE environment, we are also exploring the option of providing a striped-down, lightweight version of a TLS library as part of the "Security and Data Governance Layer", ensuring secure communications even for devices with limited capabilities.

KPIs

- Encryption rate of the data flows. Ensuring that all data flows between the project's devices and cloud services are end-to-end encrypted, and that no unencrypted data will ever be transmitted by/to any processing layer.

3.6.3 Intrusion Detection In Specialized Hardware (FPGA)

An Intrusion Detection System (IDS) in specialized hardware is a program that runs in a device such as a GPU or FPGA and scans all incoming network traffic (to a host or a whole network) in order to identify intrusion attempts and alert the user. Generally, an IDS acts as an alerting system, but there also exist some tools such as Snort[25], that provide intrusion prevention capabilities by using some logic to detect and drop suspicious packets. Such systems are also called Intrusion Prevention Systems (IPS).

Architecture

Snort inspects every packet of the traffic flow, and based on predefined rules, it sends alerts with the rule's corresponding message. Snort is an open-source tool that has been under constant development for many years, along with a large community surrounding it. In the figure below (Figure 3-9), the overall architecture is presented, from the network backbone and packet capturing by the sniffer, towards the preprocessing and scanning (Detection Engine) steps of the packets, resulting in the corresponding logs.

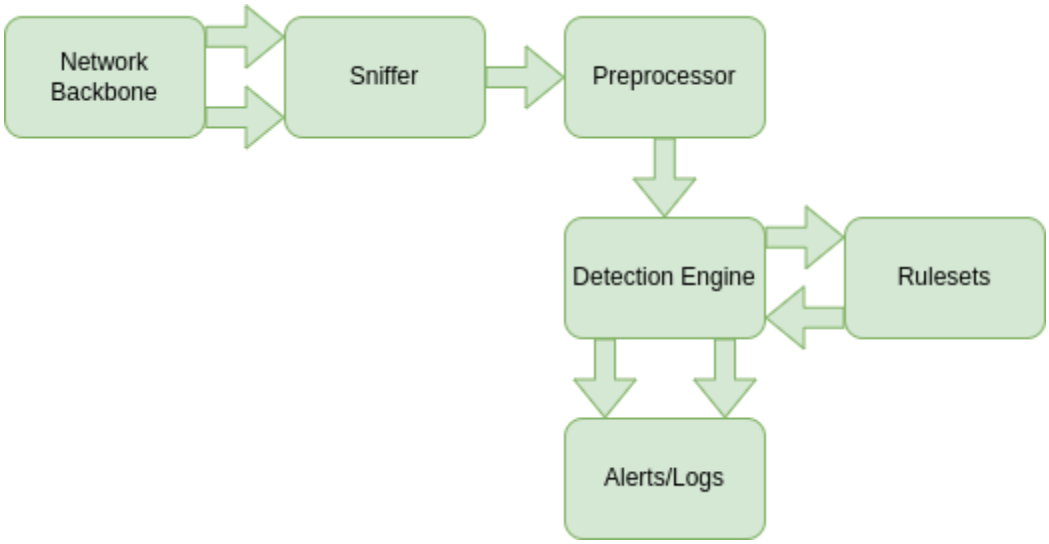

Figure 3-9 - Snort architecture

Snort uses predefined rules during the "sniffing" process. These rules generally determine the type of the action, e.g., alert/log or drop, what type of message should be displayed if a match was found, as well as IP/port specifications and patterns. Snort also executes efficient string searching in the packet payload by utilizing pattern matching algorithms. Since the rules are predefined, the tool builds an automaton in the initialization step, which is essentially a tree containing all the patterns. Afterwards each packet is scanned through the tree and if it reaches a leaf node, it means a predefined pattern has been found in the current packet payload, resulting in an alert. The most common pattern matching algorithm used is known as Aho-Corasick. Since the tool is open source, the dedicated community which supports it offers a reliable set of rules generated from past attacks. A device that runs Snort will have to download these rules and host them in the local memory.

TUC explores the feasibility of porting this specific tool to specialized hardware, e.g., Field-Programmable Gate Array (FPGA), which is an integrated circuit designed to be configured after manufacturing and deploying it into EMERALDS "Security and Data Governance Layer". This can be

[25] https://www.snort.org/

done by utilizing the Vivado High-Level Synthesis (HLS)[26] development environment by Xilinx, a staple framework for FPGA developers and engineers. Integrating a hardware-based intrusion detection and prevention system into EMERALDS will enhance our capabilities in detecting, preventing and mitigating any potential threats on the edge at an early stage, utilizing the processing capabilities and speed that a hardware-based solution can provide (Figure 3-10).

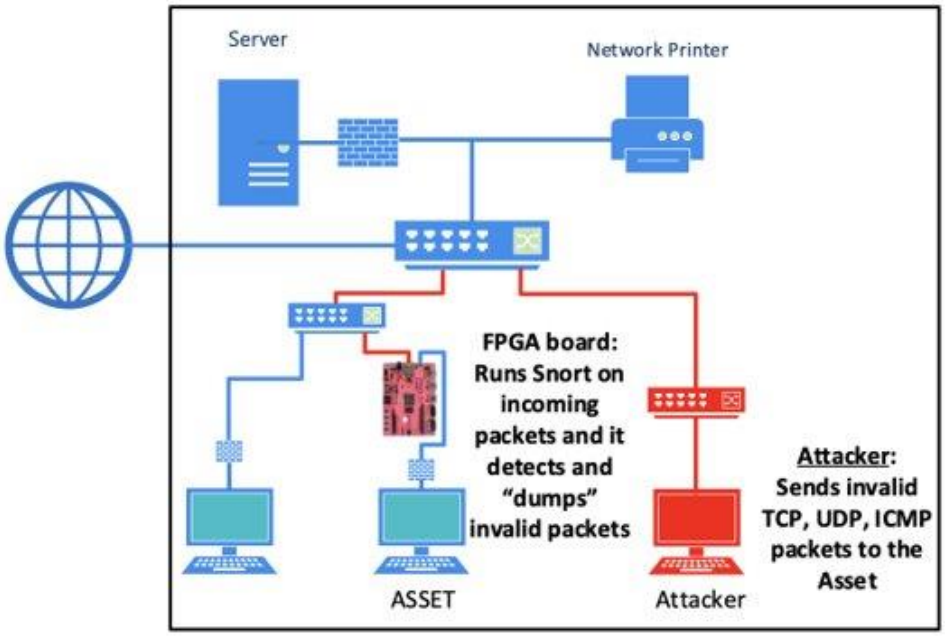

Figure 3-10 - Example FPGA-IDS deployment[ix]

KPIs

- The intrusion detection module will parse packets and perform pattern matching on the packets' payload as efficiently as possible, aiming to achieve a throughput of at least 1 Gbps, as defined in the task description and the Chapter Summary.

3.6.4 Federated Learning (FL) Models For Mobility Data

This 'emerald' is an FL model for federated learning from mobility data that can't be shared between organizations or is collected by in-vehicle edge devices, e.g., for data privacy protection or for communication bandwidth saving. This 'emerald' may address the following **business requirements**:

- [UC1] Crowd predictions may be improved by learning from different sensor systems that collect crowd data but cannot share training data with each other (including e.g., camera-based and smart phone-based systems)
- [UC3] Passenger prediction may be trained on board of the buses, thus reducing the need to transfer all training data to a central location.

The specific UC to demo this 'emerald' will be fixed at a later stage of the project. So far, no UC partner has committed themselves to a FL-specific task or to providing edge devices.

[26]https://www.xilinx.com/support/documentation-navigation/design-hubs/dh0012-vivado-high-level-synthesis-hub.html

Requirements

Given the locally available mobility data (e.g., crowd or passenger data), locals models are trained which are then transmitted to the central FL server and aggregated into a joint global model.

Architecture

Similar to the previous sub-section, this 'emerald' is developed in Python. It builds on the Federated Learning module of the MAIaas platform described below. The main methodology is illustrated in Figure 3-11.

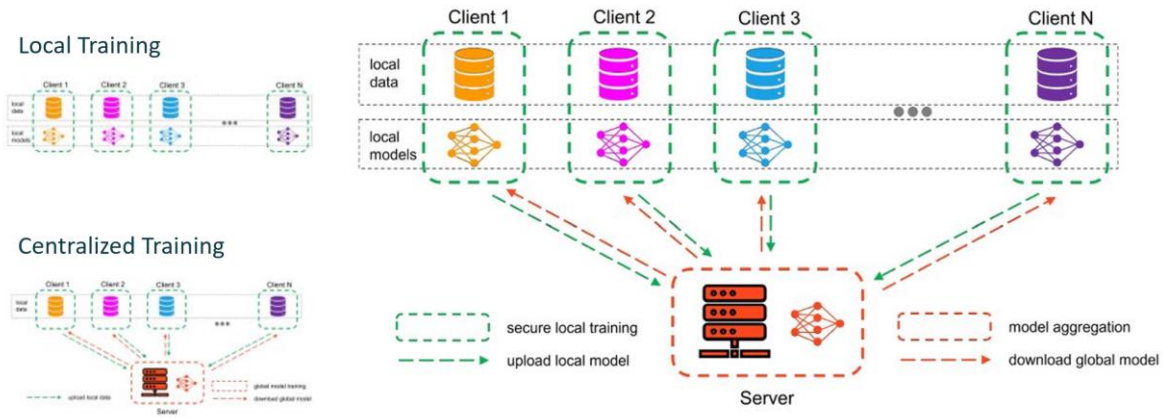

Figure 3-11 - Illustration of the FL methodologies for crowd density and forecasting one edge devices/cloud.

The following inputs and components are required to develop the model:

- Federated learning framework (for example, ***Flower FL framework***) on edge/cloud written in python language.
- Distribution component to deploy the trained model for inferencing and store it in a model repository, such as the ATOS MAIaaS Model Repository

The following inputs and components are required to run the FL workflow on live data:

- Live data available via APIs (may be simulated, if necessary)
- Trained model served by the MAIaaS platform (e.g., cloud models) or UPRC edge devices.
- Communication between FL clients and server to enable exchange of model updates.

Development process

Starting from centralized UC-specific models, federated versions will be developed. The TRL goal of this 'emerald' development is to advance from TRL for ML 2 to ML 4.

KPIs

- Metrics: Prediction accuracy (e.g., R2, RRMSE, MAE, MAPE)
- KPI: improvement over state of the art
- Baselines: The effectiveness of the proposed framework will be validated through a comparative analysis against the following state-of-the-art baseline methods such as FedAvg (Federated Averaging), FedProx (Federated Proximal), MOCHA (Memory-efficient, Online, Constrained, High-dimensional, Agnostic), etc.
- Limitations: As model effectiveness relies on factors like model design, quality of training data, and the intricacies of predefined risks, the scope for comparing models trained across various regions, risks, and datasets is restricted. Additionally, challenges such as computational constraints and communication bottlenecks may arise during the federated training process.

4 Visual Analytics & Dashboard Services

This Chapter delves into the process entailing the development of dedicated visual analytics tools for geospatial and location intelligence, contributing to the visual representation of extreme-scale mobility data. Visual analytics plays a pivotal role in transforming complex urban mobility data into actionable insights.

The high-level framework for visual analytics and dashboards in EMERALDS is presented in this chapter. It describes the essential elements and how they work together, placing special emphasis on the application of cutting-edge data visualization techniques. It also describes the foundational technology stack used to build dynamic and interactive dashboards. The various data visualization techniques used within EMERALDS are explained and how these components empower users to unlock the value of extreme-scale urban mobility data is illustrated.

The primary Visual Analytics (VA) and Dashboard Services are provided through the CARTO Mobility Analytics Platform. Furthermore, ATOS MAIaaS offers a basic dashboard service as part of its Jupyter Notebook as a Service offering, while exploring the potential integration of widely recognized Data Visualization tools such as Metabase[27], Apache Superset[28], or Redash[29] into the platform. Last, the EMERALDS toolset is designed to facilitate the creation of tailor-made Visual Analytics Services for specific use cases, allowing for a high degree of customization and flexibility.

4.1 CARTO Visual Analytics Services

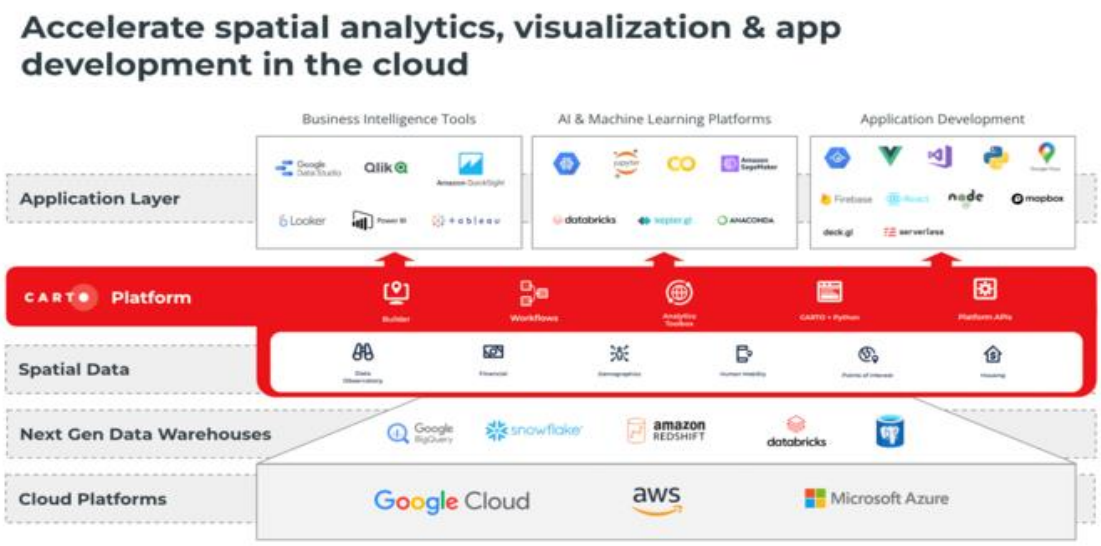

Figure 4-1 CARTO overview

The CARTO Platform is the world's leading Location Intelligence platform that empowers organizations with spatial data and analysis capabilities. It serves as a powerful tool for Data Scientists, Developers, and Analysts to optimize business processes, predict future outcomes, and make data-driven decisions using Spatial Data Science. CARTO enables users to leverage spatial data for various applications, such

[27] https://www.metabase.com/

[28] https://superset.apache.org/

[29] https://redash.io/

as efficient delivery routes, strategic store placements, behavioural marketing, and much more. The platform's technology stack includes a robust set of tools for data visualization, geospatial analysis, and spatial data management. It offers a user-friendly interface that allows users to create interactive and insightful dashboards, maps, and applications without the need for extensive coding knowledge.

CARTO's integration with the EMERALDS project takes advantage of its spatial data analysis capabilities, data ingestion and visualization features to enhance the MAaaS toolset (Figure 4-1). The CARTO Platform serves as a core component of the EMERALDS ecosystem, providing advanced location intelligence capabilities to support decision-making by public authorities and individuals in the mobility sector.

4.1.1 Relevant Components of the CARTO Platform:

- Spatial Data Visualization: CARTO offers powerful mapping and visualization tools to create stunning and interactive maps, heatmaps, and data-driven visualizations. Users can easily overlay different datasets, analyse patterns, and gain valuable insights from spatial data.
- Location Analysis: The platform provides a wide range of location-based analytics, such as spatial aggregations, proximity analysis, and route optimization. These tools enable users to discover spatial patterns, optimize delivery routes, and make informed decisions based on location insights.
- Spatial Data Science: CARTO's Analytics Toolbox allows Data Scientists and Analysts to perform advanced geospatial analysis, predictive modelling, and machine learning using spatial datasets. It is a set of UDFs and Store Procedures to unlock Spatial Analytics directly on top of your cloud data warehouse platform. It is organized in a set of modules based on the functionality they offer. This integration facilitates data-driven decision-making and the ability to predict future outcomes with a spatial context.
- Geospatial Data Management: CARTO offers a data management system tailored to handle large-scale spatial datasets efficiently. Users can upload, store, and manage spatial data seamlessly, ensuring the availability of reliable and up-to-date information. It allows the users to connect to multiple cloud data warehouses, explore their geospatial data, geocode their tables, enrich their data with a wealth of vetted datasets to enhance their geospatial analysis, and access the different CARTO tools.

Open Geospatial Consortium (OGC) standards provide a framework for interoperability and compatibility among various geospatial data sources and services. Therefore, adhering to OGC standards is fostered within EMERALDS through the requirements gathered from CARTO and other industry stakeholders, and was prioritized throughout the design of the RA. The tools and services can seamlessly integrate with existing geospatial data infrastructure, exchange data with other systems, and be readily adopted by organizations that follow these standards. File formats designed for geospatial data storage (i.e., GeoParquet, FlatGeoBuf, Geopackage, and Shapefile) are considered for data exchanges with the aim to enhance storage and retrieval of geospatial mobility data, improving the overall performance and usability of the analytics tools and services.

The following three figures illustrate the CARTO spatial extension to data warehouse (Figure 4 3), the structure of Analytics Toolbox (Figure 4-4) and how the Analytics toolbox acts as a “middleware” between the data warehouses and the visualization services (Figure 4-5).

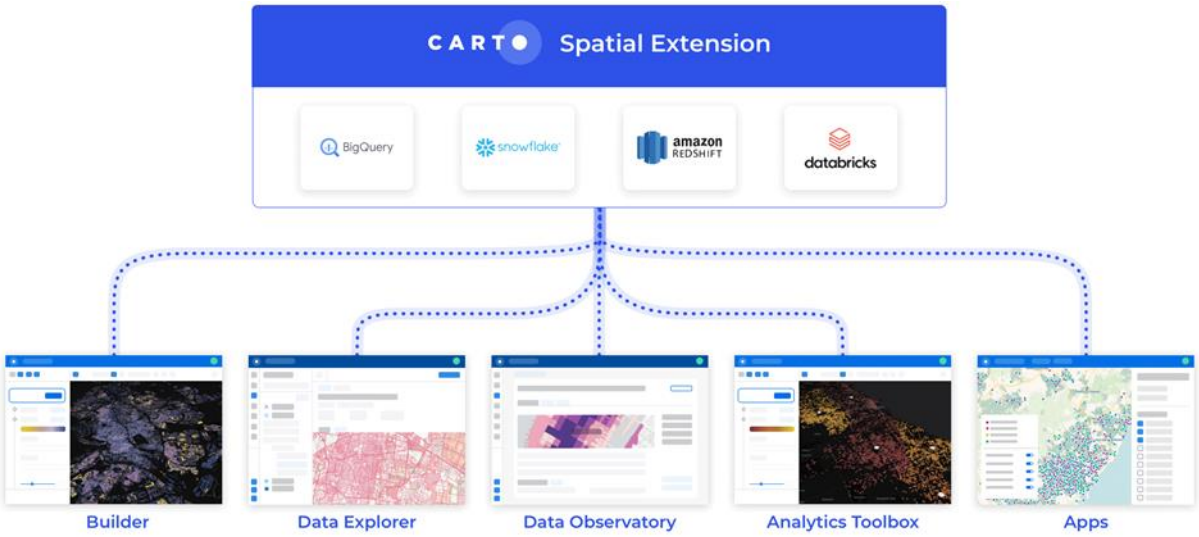

Figure 4-2 - CARTO spatial extension

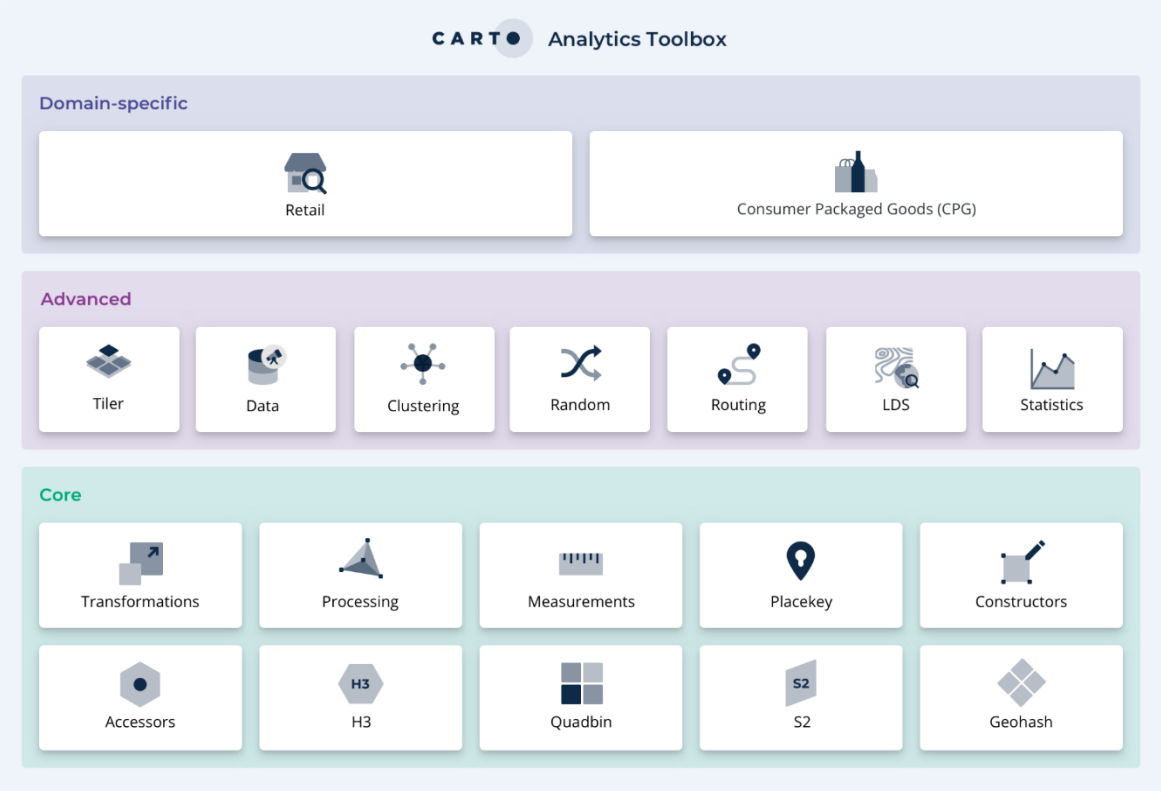

Figure 4-3 - CARTO's Analytics Toolbox Structure

Figure 4-4 - How CARTO's Analytics toolbox is connected with Data warehouse and Visualization Services

4.1.2 Consuming External Datasets

CARTO's platform allows users to consume external datasets, enhancing the platform's capabilities beyond the data available natively. External datasets can be integrated into dashboards and maps to provide additional context and insights. CARTO allows the users to connect to their data warehouse or using remote files (such as CSVs, GeoJSONs or shapefiles). The data can be added in seconds with a simple drag-and-drop interface. With integrations for leading cloud data platforms and analytics tools, including Google BigQuery, Snowflake, Amazon Redshift and Databricks, the users can connect to their existing analytics stack and enrich with external data from CARTO Data Observatory.

4.1.3 Integration of MobilityDB with CARTO

MobilityDB is a specialized extension for PostgreSQL databases that handles temporal and spatiotemporal data, such as moving object trajectories. The integration of MobilityDB with the CARTO Platform brings several benefits:

- **Temporal Analysis**: MobilityDB enables CARTO to analyze time-based data, such as movement patterns, travel times, and historical trends, providing a comprehensive understanding of mobility dynamics.
- **Real-Time Tracking:** By leveraging MobilityDB's capabilities, CARTO can support real-time tracking and monitoring of moving objects, allowing users to track vehicles, assets, or individuals in space and time.
- **Predictive Modeling:** The integration of MobilityDB enhances CARTO's Spatial Data Science capabilities by enabling the incorporation of temporal aspects into predictive models, resulting in more accurate predictions and better decision-making.
- **Advanced Mobility Solutions**: With MobilityDB integration, CARTO can offer advanced mobility solutions, including route prediction, traffic forecasting, and dynamic mobility simulations, which are essential for the success of the EMERALDS project.

In conclusion, the CARTO Platform's integration with the EMERALDS project harnesses the power of location intelligence and spatial data science to revolutionize mobility analytics. By combining CARTO's visualization features, spatial data analysis tools, and external data consumption capabilities with

MobilityDB's temporal analysis capabilities, the integration sets the stage for more efficient mobility solutions, saving time, money, and reducing environmental impact. The product provides a comprehensive, user-friendly platform that empowers organizations and public authorities to make data-driven decisions, optimize mobility processes, and shape the future of transportation.

4.1.4 Visual Analytics and Dashboards

CARTO Platform offers robust visualization and dashboard services tailored to mobility services, empowering organizations to effectively analyze and communicate spatial data related to transportation and movement. Leveraging its advanced Geographic Information System (GIS) capabilities, CARTO enables users to create dynamic and insightful visualizations that depict various mobility aspects such as traffic patterns, transit routes, and urban planning. The platform's intuitive interface allows users to easily import, manipulate, and visualize location-based data, transforming it into interactive maps and graphics. Furthermore, CARTO's dashboard functionality enables the assembly of interactive and customizable dashboards that consolidate multiple visualizations into a single interface, providing stakeholders with a comprehensive view of mobility trends, challenges, and opportunities. This integrated approach facilitates data-driven decision-making for improving transportation systems, optimizing routes, and enhancing overall urban mobility planning.

4.2 Open-Source VA Services

In the frame of T2.1 the visualization needs of Use Cases when deemed necessary will be supervised and supported by means of adapting established open-source visualization tools i.e., Bokeh, Grafana. Outcomes of AI/ML components can be accessed via customizable widgets, relying on well-structured data schemas depending on the use case scenario.

For handling the visualization of extensive datasets generated by the EMERALDS Toolset workflows, end-users may leverage the power of Open Source WebGL technologies, such as Kepler.gl and deck.gl, to enhance the visualization capabilities. These WebGL-based tools will enable interactive and high-performance visualizations, allowing users to explore and gain insights over the data produced by the EMERALDS.

The integration of these tools with the generated datasets requires the creation of connectors and adapters that will allow the seamless ingestion of the data. This process will require careful consideration of data formats, structures, and schemas to ensure compatibility. To this end, the promotion of standardized formats for the generated data, is examined as part of D3.1, D3.2, D4.1, and D4.2.

5 Mobility AI-as-a-Service (MAIaaS)

The rise of Machine Learning (ML) has shifted AI development from code-centric to data-centric approaches, defining Machine Learning Operations (MLOps) as an extension of DevOps (Development Operations) methodologies, focused on ML models' creation. The EMERALDS' Mobility AI-as-a-Service (MAIaaS) platform will leverage existing MLOps advancements while addressing project's mobility-specific use-cases. It covers the entire ML model lifecycle, integrating data sets, hyperparameter search, model verification, and continuous learning, while addressing challenges like security, privacy, and latency constraints through AI deployment across the computing continuum from use cases' edge to project's cloud. The platform's key objectives include facilitating data scientists' work, industrializing (develop, deploy and exploit) ML models, improving model adaptability, and offering a catalogue of pre-trained ML models accessible via APIs. This platform will thus support the models developed within WP3 and WP4. This way, EMERALDS seeks to break barriers and drive widespread adoption of data-driven models, elevating mobility with innovative AI- powered solutions.

5.1 MAIaaS functional requirements

The functional requirements of the EMERALDS' MAIaasS platform encompass a set of essential capabilities that ensure seamless and efficient management of EMERALDS' ML models throughout their lifecycle. These requirements cover: data integration, model experimentation/training, model deployment & monitoring, federating learning features and pre-trained ML models' distribution. From a functional point of view, these can be classified as follows:

Data Management & Integration

- The platform must offer data management and version control capabilities to handle vast datasets used in model training and deployment. Ensuring data consistency and traceability allows data scientists to confidently track various experiments based on specific datasets.
- Additionally, the platform should facilitate storing curated datasets coming from EMERALDS' scenarios, granting data scientists access and utilization of these data within the platform, streamlining their workflow and enhancing overall productivity.

ML/DL models' development

- The platform must empower project's data scientists with streamlined and efficient processes, collaborative capabilities, and standardized tools. It should offer model management, tracking, and version control, enabling teams to monitor changes and promote collaboration seamlessly.
- Automation plays a significant role in this context, with a need for automating model training and testing processes. By automating these tasks, the platform accelerates model delivery and enhances overall productivity.
- To optimize model performance, data scientists require the ability to use tools for visualizing training metrics and hyperparameters. Additionally, automatic hyperparameter tuning is essential for maximizing model performance without the need for manual selection.

- Standardization and homogenization of basic development tools are necessary. By having common development tools, team members can execute and reproduce code easily, facilitating knowledge exchange and efficient code corrections.
- The platform should also provide a catalogue of common Docker images that satisfy development libraries and package needs. Hence, data scientists can streamline this management, alleviating concerns about building and running containers.

Models' deployment

- It is a must to ensure seamless and efficient deployment, monitoring and maintenance of model in a productive environment. Data scientists seek to monitor the performance and accuracy of deployed model continuously, allowing for regular updates and maintenance to uphold optimal performance.
- Furthermore, they emphasize the importance of model explainability. By applying explainability techniques, they aim to understand how models derive results, enabling then to comprehend the decision-making process and assess the impact of model predictions on human decision-making.
- The EMERALDS' MAIaaS platform will also support the cloud deployment of validated ML models for their exploitation (when needed) by the project's use cases.

Models' distribution

- Centralized model repositories are vital for data scientists to share knowledge, promote collaboration, and facilitate access to a wide range of models with the platform.
- Data scientists emphasize real-time model serving to swiftly cater to application and systems' needs, enabling users to access predictions promptly and efficiently.

Federated learning

- The functional requirement related to federated learning emphasizes the data scientists' desire to leverage this approach to preserve data privacy and data ownership. This enables data scientists to collaborate on model development without compromising sensitive data, protecting user privacy, and maintaining data ownership rights. MAIaaS platform will also support project's federated learning PoCs.

5.2 EMERALDS' MAIaaS architecture: MLOps approach

The architecture of MAIaaS platform is designed to achieve EMERALDS ML models' development and deployment, leveraging Kubernetes[30] as its underlying infrastructure whilst providing a robust and scalable foundation of the entire ML lifecycle.

First target of this platform is to support AI models required to implement EMERALDS' mobility services within all project's use cases. This introduces a first division of the architecture (Figure 5-1), between the cloud components, supporting ML development and inferencing, and the use cases' edge layer, providing data and running some of these models. In turn, the cloud core is divided into two related blocks: the development framework, with all components devoted to ML models experimentation, evaluation, and monitoring; and the production framework, for deploying and exploiting the functional versions of the models in the cloud.

[30] https://kubernetes.io/

Data management is a transversal layer that provides datasets for both: development (training and testing) and exploitation (inferencing) of ML Models. The platform supports data storage and persistence for structure and unstructured data using tools like Minio[31] or PostgreSQL[32]based ones. This capability enables data scientists to securely store and manage their curated datasets, ensuring data consistency and traceability.

For model development, Kubeflow[33] has been implemented in the platform, a powerful toolset specifically designed for ML workflows. Kubeflow covers model development, experimentation, evaluation, and monitoring during training, streamlining the entire model development process. Data scientists can utilize Jupyter Notebooks[34] and pipelines already included in Kubeflow tool to experiment with different models and hyperparameters effectively. Hence, data scientists can train/experiment with their models by using curated data, which was stored previously in the platform by them, having an entire development environment in one place.

Once model has been developed, to facilitate model versioning, reproducibility and distribution, the platform provides access to a model and tools repository. Tools like MLFlow[35] registry allow data scientists to upload models and artifacts, promoting collaboration and knowledge sharing. Additionally, Docker is employed to manage and share containerized images of models or services associated with those models, easing deployment, and eliminating concerns about environment consistency.

Model deployment is Kubernetes-based as well, with each model containerized using KServe[36], making them readily available for real-time inference. This approach ensures scalability and reliability for serving models in a production environment. Monitoring tools, such as Evidently, are also deployed as components in the Kubernetes cluster, enabling continuous monitoring of model performance and drift detection. If any data drift is detected the platform supports dockerized pipeline creation by data scientists for automatic model training and deployment. Hence, the model will keep up to date in an automated way.

The platform embraces federated learning as a key feature. By deploying the centralized module in our Kubernetes cluster, we enable federated learning capabilities, preserving data privacy and ownership while allowing collaboration across multiple distributed data sources. This module will also manage the sub-models' deployment in edge nodes, as part of the FL infrastructure.

[31] https://min.io/

[32] https://www.postgresql.org/

[33] https://www.kubeflow.org/

[34] https://jupyter.org/

[35] https://mlflow.org/

[36] https://kserve.github.io/website/0.10/

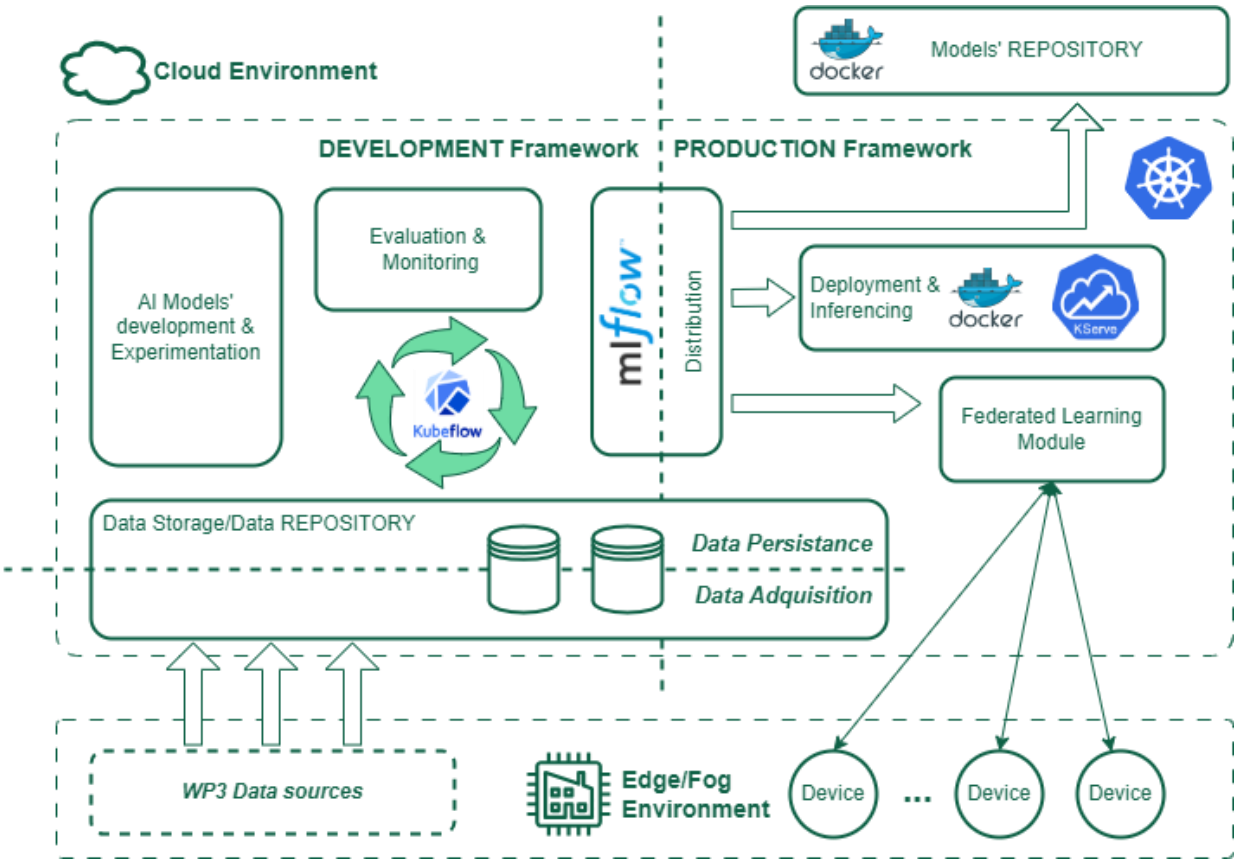

Figure 5-1 - EMERALDS' MAIaaS architecture

5.3 Interaction with the related EMERALDS services

The MAIaaS platform is connected to the various EMERALDS services, such as analysis and inference, all focusing on mobility-related tasks. The platform acts as a development hub where partners can efficiently create and deploy ML models tailored for EMERALDS. Once models are trained and optimized within the platform, they can seamlessly integrate with EMERALDS' analysis and inference services. Hence, the MAIaaS platform hosts and facilitates ML model inferencing, ensuring optimized, production-ready deployments aligned with the project's mobility-focused solutions. The integration ensures that the ML models developed on the platform effectively contribute to the addressing mobility challenges and provide valuable insights for EMERALDS' broader goals. By foresting collaboration and alignment with EMERALDS' specialized services, the platform becomes a key enabler in driving impactful innovations for the mobility domain.

5.4 Platform KPIs

The EMERALDS' MAIaaS platform defines a set of figures or KPIs that covers various aspects of the platform's functionality and are crucial in evaluating its impact on model development, deployment, and monitoring processes. These KPIs are initially classified from two points of view:

Platform's performance, which evaluates relevant processes' execution. This includes:

- Data Ingestion Latency is a critical KPI that measures the time taken for data to be ingested and made available for processing or analysis after its arrival at the data ingestion interface.

A low latency indicates and efficient data ingestion process, ensuring that downstream systems can quickly access and utilize the data for ML model development.

- Anomaly Detection and Alert Resolution Time is a critical KPI for model monitoring. It measures the time taken to identify anomalies in model behaviour or performance and resolve associated alerts using the ML Monitoring tools. A swift resolution time indicates an effective monitoring process, ensuring the platform detects and addresses issues promptly.
- The Maximum Number reached of Models in Training Phase is an important KPI that evaluates the platform's resource capacity and robustness to handle multiple models being trained concurrently. A higher value signifies the platform's ability to efficiently manage model training demands and indicates its scalability.
- Also, the Maximum Number reached of Models in Production is an essential KPI that measures the maximum number of ML models deployed in production. This reflects the platform's robustness in handling model deployment demands, ensuring a smooth transition from development to real-world deployment.
- For federated learning, the Maximum Number of Devices reached is a crucial KPI. It measures the maximum number of devices participating in the federated learning process concurrently, reflecting the platform's scalability and capacity to handle many distributed devices, ensuring efficient and privacy-preserving model training.

Platform's usability, that reflects how well the platform is adopted by project's data scientists:

- The number of Available Datasets for Training/Testing is another essential KPI that reflects the richness of data available within the MLOps framework. It indicates the diversity and volume of datasets uploaded for ML model training and testing, contributing to the platform's capacity for addressing a wide range of use-cases.
- User Engagement or Adoption Rate is a pivotal KPI that measures the level of engagement and adoption of the ML experimentation module by data scientists and users with the project. Monitoring the number of active users, frequency of usage, and the percentage of regular users provides insights into the platform's popularity and value among its users.
- Model repository Utilization is a significant KPI that tracks the adoption and utilization of the ML Models (and Tools) repository within the project. Monitoring the number of models or tools stored, frequency of updates, and the number of unique users accessing the repository gauges its effectiveness in promoting knowledge sharing and collaboration among data scientists.

6 Compliance with Reference Architectures

EMERALDS as a project derives valuable insights and lessons from the guidelines and recommendations provided by the Big Data Value Association (BDVA), a leading organization focused on promoting data-driven innovation across Europe, on the design of data sharing architectures. The key learnings identified are:

a) Building systems that can handle vast and voluminous data while remaining flexible enough to upscale or downscale based on traffic, transactions and demand. These ***scalability*** features are widely acknowledged throughout the process of designing and developing 'emeralds' and are incorporated in the reference architecture. ***Horizontal scalability*** is addressed via the use of auto-scaling infrastructure such as Kubernetes[37] for the **MAIaaS Platform** and Apache Spark Clusters for the **Extreme Scale Data Processing** 'emeralds'. Additionally, the development of 'emeralds' software components is based on microservice design principals – such as stateless and event driven and deployed as containers. The ***vertical scalability*** is not pursued actively due to its physical scalability ceiling on processing extreme scale datasets. Nevertheless, the use of it can be helpful during the training phase of machine learning 'emeralds' components, in order to reduce the time of each MLOps cycle.

b) BDVA advocates for ***interoperability*** and adherence to industry standards to facilitate seamless data exchange and integration[x]. To this end, EMERALDS reference architecture includes standardized interfaces and data formats, enabling a variety of codes, systems and components to interact.

c) Integration of ***privacy aware data processing and secure data transactions across the compute continuum*** in alignment with the BDVA's emphasis on data privacy, security and compliance with regulatory requirements such as GDPR, Data Act and the newly introduced Trustworthy AI act[xi].

d) ***Reusability*** is a central property promoted on 'emeralds' design. In EMERALDS, individual components are developed independently and reused across different use cases, on track with modular and reusable design that simplifies development and code reuse.

e) The architecture incorporates ***AI-driven analytics*** modules that enable predictive modelling, pattern recognition, and real-time decision-making based on mobility data. Significant potential has been recognized in advanced analytics and AI to derive insights from large datasets, according to the BDVA.

f) ***Optimizing data processing and analytics*** by leveraging both edge and cloud computing resources. The architecture presented embraces a hybrid approach, edge nodes for real-time processing and cloud resources for more complex analytics tasks and therefor paving the way for dealing with extreme scale data challenges.

g) BDVA promotes ***collaboration*** within a larger innovation ecosystem, fostering partnerships between academia, industry, and research institutions[xii], such as the collaboration framework underpinning the EMERALDS architecture. RTOs, established industry partners, SMEs and local authorities are linked through an environment that is designed to support collaboration and knowledge sharing, facilitating the exchange of data, insights, ideas and innovative approaches to urban mobility research questions.

The International Data Spaces Association and Gaia-X initiatives share similar goals and principles as their main objective is to establish secure, trusted, and interoperable data ecosystems that prioritize data sovereignty, security, and data sharing among organizations and industries. The two

[37] https://kubernetes.io/

initiatives have identified the synergies between their goals and have been working together to align their efforts.

The design of the EMERALDS project reference architecture has been influenced by these key principles in the following ways:

a) IDS and Gaia-X promote the concept of **Data Sovereignty**, meaning that the data should remain under the control and ownership of the data provider. EMERALDS adheres to this principal by making available the emerald services to data owner rather than transferring the data outside its facilities. This achieved with the development of Federated Learning 'emeralds' as described on section 3.6.4 and the deployment most of the 'emeralds' as containers, a form that can be easily integrated to the IDS Application Store concept. Under this concept the EMERALDS services can be downloaded and executed in the IDS Connector component and apply their services on the Datasets. Figure 6-1 illustrates these two cases along with the corresponding interactions.

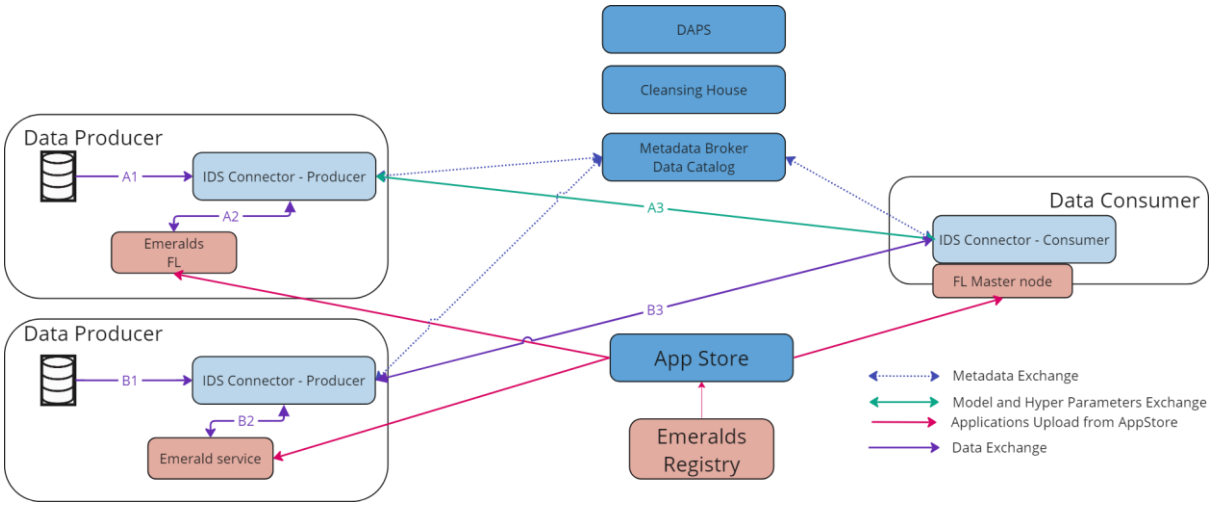

Figure 6-1 – IDSA representation of EMERALDS reference architecture, assuming 'emeralds' to be consumed as services of a Data Space App Store

b) Similar to BDVA, IDSA and Gaia-X also promote **interoperability** and **standardization** as key design principals. EMERALDS project is abiding these requirements through the implementation of the emerald services on top of common protocols and formats.

c) **Trustworthiness** and **Security** are another two key principles of the two initiatives, that EMERALDS project has taken into consideration in the Reference Architecture. To that end, The Security related 'emeralds', described in section 3.6, are offering trusted execution environment, encryption in communications and intrusion detection. Additionally, the "**Privacy aware data ingestion**" offers privacy aware insights to the available datasets, thus increasing End User's trust to the analytics platform that integrates them.

7 Conclusions and next steps

This architecture transcends traditional boundaries by encompassing a distributed computing environment, including edge, fog, and cloud nodes, fostering collaboration across all compute continuums to establish a robust processing pipeline. It serves as the foundational blueprint for designing and implementing a diverse array of specialized software modules, collectively referred to as 'emeralds,' geared towards extreme-scale urban mobility data analytics.

Leveraging distributed computing environments for extreme urban mobility data analytics offers a multitude of benefits:

a) scalability to handle the vast volumes of data generated from an increasing number of sensing, monitoring and streaming devices

b) Capability to design robust and fault tolerant systems. Distributed environments are designed to withstand hardware failures, maintaining data integrity and reliability.

c) the interplay between these computing continuums allows for real-time processing, enabling rapid insights and timely decision-making, critical for urban mobility management

d) foster interoperability, allowing diverse data sources and formats to be integrated efficiently

Specifically, the EMERALDS project is pushing the boundaries of the Mobility Data Analytics field by introducing the following key features:

- Tailored extreme-data processing, management and analytics features on urban mobility data triggering the development of novel solutions in the mobility domain.
- A Platform for Developing Custom Mobility Data Analytics and Prediction Models as a Service, utilizing best practices from the MLOps domain.
- Streamlining processes for federated learning considering extreme variety and veracity of data sources
- Allow the creation of new Mobility Data Analytics platforms by using subset of the EMERALDS toolset.
- Use of the production grade Dashboard and Visual Analytics services offered by CARTO in a newly introduced chaining of DataOps with MLOps, linking the output of sophisticated Machine Learning ‘emeralds’ with the spatio-temporal querying.
- efficient and flexible access to data, by eliminating long data-to-query times, supporting cross-format queries and dynamic data workloads.

In the following months, the EMERALDS implementation plan foresees the setup of a Continuous Integration/Continuous Deployment (CI/CD) stack to support the entire software lifecycle processes, from testing of workflows up to the release of the fully tested and deployed solutions in the use cases and proof-of-concept environments M18 D2.2.

A solid continuous integration plan will be developed, that will analyse all software resources (e.g., mechanisms, modules, components, services) available, and identify, specify, and document the integration points amongst these resources. Both inter-module and inter-component integration will be included. An early containerization of tools from D3.1, D4.1 will be performed on M18 (D2.2) and distributed for the first validation cycle throughout the use cases, while the final proof-of-concept prototype Toolset released on M33 (D2.3). The containerised Toolset is developed with the aim to achieve interoperability with two analytics-as-a service platforms (one operated by ATOS and the other established commercial cloud platform run by CARTO), whilst users will be able to call-out selected methods matching the needs of their extreme data workflow task in the form of software micro-services (‘emeralds’).

8 References

[i] Amini, S.; Gerostathopoulos, I.; Prehofer, C. (2017) Big data analytics architecture for real-time traffic control. In Proceedings of the 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Naples, Italy, 26–28 June 2017; pp. 710–715.

[ii] Daniel, A.; Subburathinam, K.; Paul, A.; Rajkumar, N.; Rho, S. (2017). Big autonomous vehicular data classifications: Towards procuring intelligence in ITS. Veh. Commun., 9, 306–312.

[iii] Patan, R., & Babu, M. R. (2018). A novel performance aware real-time data handling for big data platforms on Lambda architecture. International Journal of Computer Aided Engineering and Technology, 10(4), 418-430.

[iv] Pellungrini, R., Pappalardo, L., Pratesi, F., & Monreale, A. (2017). A Data Mining Approach to Assess Privacy Risk in Human Mobility Data. ACM Transactions on Intelligent Systems and Technology (TIST), 9, 1 - 27.

[v] Zaharia, M.A., Chowdhury, M., Franklin, M.J., Shenker, S., & Stoica, I. (2010). Spark: Cluster Computing with Working Sets. *USENIX Workshop on Hot Topics in Cloud Computing*.

[vi] Yao, Y., Gao, H., Wang, J., Sheng, B., & Mi, N. (2021). New Scheduling Algorithms for Improving Performance and Resource Utilization in Hadoop YARN Clusters. IEEE Transactions on Cloud Computing, 9, 1158-1171.

[vii] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. NIPS.

[viii] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[ix] E. Papadogiannaki, G. Chrysos, K. Georgopoulos, S. Ioannidis. (2023). A Reconfigurable IDS Framework for Encrypted and Non-Encrypted Network Data in Supply Chains. In 2023 International Conference on Engineering and Emerging Technologies (ICEET) (accepted, not published yet)

[x] Bertels, Natalie, Freek Bomhof, Håkan Burden, Susanne Stenberg, Katerina Yordanova, Matia Trino, Götz Brasche, Valerio Frascolla, and Ana Garcia Robles. (2023) "The Digital Decade Policy Programme of the European Commission." BDVA

[xi] Timan, T., & Mann, Z. Á. (2019). Data protection in the era of artificial intelligence. *Trends, existing solutions and recommendations for privacy-preserving technologies*. October 2019. BDVA

[xii] Zillner, S., Bisset, D., Milano, M., Curry, E., Södergård, C., & Tuikka, T. (2020). Strategic research, innovation and deployment agenda: Ai, data and robotics partnership.

Ethics Checklist and Questionnaire

THIS FORM NEEDS TO BE FILLED-IN BY THE DELIVERABLE LEADER BEFORE, DURING AND AFTER THE WORK LEADING TO THE RELEVANT DELIVERABLE. DELIVERABLE LEADERS ARE ENCOURAGE TO DISCUSS EACH ACTIVE QUESTIONAIRE WITH THE ETHICS COMMITTEE. EACH OPEN QUESTIONNAIRE SHOULD EITHER BE STORED IN THE PROJECT MS TEAMS DIRECTORY OR A LINK MADE AVAILABLE TO THE RELEVANT SHARED DOC. COMPLETED FORMS NEED TO BE SUBMITTED AS PART OF THE DELIVERABLE Q/A PROCESS. IN THE EVENT OF COMMENTS AND/OR QUESTIONS BY THE ETHICS COMMITTEE, THE DELIVERABLE LEADER HAS TO PROVIDE RELEVANT RESPONSES AND/OR CLARIFICATIONS IN A TIMELY MANNER

A. PERSONAL DATA

1. Has ***personal data*** going to be processed for the completion of this deliverable?

-No

- If “yes”, do they refer only to individuals connected to project partners or to third parties as well?

2. Are “***special categories of personal data***” going to be processed for this deliverable? (Whereby these include personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, and trade union membership, as well as, genetic data, biometric data, data concerning health or data concerning a natural person's sex life or sexual orientation)

-No

3. Has the ***consent*** of the individuals concerned been acquired prior to the processing of their personal data?
   - If “yes”, is it based on the Project’s Informed Consent Form, either on the provided Template or on other attached herein Template?
   - If “no” is it based on a different legal basis?

-No personal data or individual consent required for D2.1

4. In the event of processing of personal data, is the processing:
   - obviously “***Fair and lawful***”, meaning executed in a fair manner and following consent of the individuals concerned or based on another - acknowledged as adequate and proportionate as per above - legal basis?
   - Performed for a ***specific*** (***project-related***) ***cause*** only?
   - Executed on the basis of the principle of ***proportionality and data minimisation*** (meaning that only data that are ***necessary*** for the processing purposes are being

processed and such deductive reasoning is documented)?

- Based on ***high-quality, updated and precise*** *personal data*?

5. Are there any provisions for a storage limitation period of the personal data-in case of storage- after which they must be erased?
6. Are all ***other lawful requirements*** for the processing of the data (for example, ***notification of the competent Data Protection Authority(s)*** or undergoing a ***DPIA procedure*** and consulting with the competent DPA, if and where applicable) adhered to and on what legislative basis are such notifications justified as necessary or dismissed as unnecessary?
7. Have individuals been ***made aware of their rights*** on the processing of the personal data as per the GDPR and the relevant and executive national legislation (particularly the rights to access, rectify and delete the personal data and their right to lodge a complaint with the relevant Competent Authority) and if yes, by what demonstrable means (e.g. the informed consent form as per above or as per other Templates, attached herein?)
8. Even if anonymized or pseudonymized or aggregated data are referred to, does the dataset contain ***location data*** that could potentially (even via the combined use of other datasets) be ***traced back*** to ***individuals***? If yes, what specific measures are taken to ensure this data (i) is anonymized or pseudonymized and (ii) cannot be used to track individuals without their consent? If no, what is the scientific methodology used to collect and gather said data?
9. In the context of risk assessment, prediction and forecasting, as foreseen in the scope of the EMERLADS project, during traffic, population movement monitoring or weather events, ***is there any risk*** that personal data could be inadvertently revealed in the event of an ***emergency*** or ***unusual event***, because of the dataset usage, either on its own or combined with other openly available datasets, triggering identification or unwanted disclosure of PII? What measures are in place to protect - still identifiable if the dataset allows such extraction - **personal data** in these circumstances?
10. For the use case of Trip Characteristics Inference as per the EMERALDS project scope, are there specific measures to ensure that **inferences made about trip characteristics** cannot be linked back to ***specific individuals*** or reveal ***sensitive information*** about their ***habits*** or ***routines*** ie. by identifying specific individuals' absence or presence routines whether in the home or in a professional environment or in other premises?
11. Are there any potentially personal identifiable information (PII) in the datasets, **disclosable by combination with other datasets**, either open data or proprietary (e.g., E-tickets validation data)? If yes, how is PII adequately anonymized or pseudonymized or how other datasets that by combination may result in unwanted or illegal disclosures or identification before any processing takes place?

B. DATA SECURITY

1. Have proportionate security measures been undertaken for protection of the data, taking into account project requirements and the nature of the data?
   - If yes, brief description of such measures (including physical-world measures, if any)
   - If yes, is there a data breach notification policy in place within your organization (including an Incident Response Plan to such a breach)?

- No personal data required for D2.1

2. Given the **large-scale nature** of some datasets, are there specific measures in place to protect **included personal data** at scale at the data source or in the possession of data processors?

3. Regardless of personal data*,* in the case of Multi-modal integrated traffic management as defined under the EMERALDS scope, are there specific measures in place to ensure **the availability and integrity of data spanning multiple modes of transport** from being disclosed in other manners than the ones intended and covered under an open data scheme?

4. Are there specific measures in place to secure ***sensitive infrastructure data***, if present?

C. DATA TRANSFERS

1. Are personal data transfers beyond project partners going to take place for this deliverable?

- No

- If "yes", do these include transfers to third (non-EU) countries and if what policies apply?

-No data transfers beyond project partners required for this deliverable

2. Are personal data transfers to public authorities going to take place for this deliverable?
3. Do any state authorities have direct or indirect access to personal data processed for this deliverable?
3. Taking into account that the Project Coordinator is the "controller" of the processing and that all other project partners involved in this deliverable are "processors" within the same contexts, are there any other personal data processing roles further attributed to any third parties for this deliverable? And if any, are they conformed to the GDPR provisions?
4. Given the geographical diversity of the datasets, are there measures in place to ensure compliance with specific personal data protection regulations **in different jurisdictions** ie. at the place of the data source establishment as well as at the place of the establishment of a Data processor?
5. Are there additional protocols for data transfers involving **sensitive infrastructure data**, if present?

D. ETHICS AND RELATED ISSUES

1. Are personal data of children going to be processed for this deliverable (ie. "underage" signified e-tickets)?

-No

2. Is ***profiling*** of identifiable individuals in any way enabled or facilitated for this deliverable?

-No

3. Are ***automated decisions*** for identifiable individuals made or enabled on the basis this deliverable?

-No

4. Have partners for this deliverable taken into consideration system architectures of ***privacy by design*** and/or ***privacy by default***, as appropriate?

- This deliverable introduces the design specification of the project's security layer.

5. Have partners for this deliverable taken into consideration gender equality policies or is there an explicit reasoning that dismisses such risk as unsubstantiated or such need as irrelevant as per the methodology of work and production of the deliverable?

-No

6. Have partners for this deliverable taken into consideration means of protecting the confidentiality of the dataset if it is not signified as open data?

-No

7. Are there additional considerations around the collection and processing of ***location data*** and data ***that could potentially be used to infer patterns about individuals' movements***?

8. Have partners identified any ***additional ethical issues*** related to the processing of sensitive infrastructure data?

-No

9. Are shared economy (ie. "Uber" transfer services or "Lime" Scooters or other solution) or other shared mobility infrastructures used by the data sources? If yes, are there measures in place to ensure that the processing of ***shared mobility data*** respects privacy rights?

-No

10. In the context of Traffic Flow Data Analytics, are there specific considerations to ensure that the **analysis of traffic flow data** does not infringe on privacy rights or reveal sensitive information about individuals' movements or routines?

- Not applicable in this deliverable

11. Is the Project taking into account the need for an all people-inclusive policy in the future within its overall goals and not only the ''tech-savvy'' (i.e. elderly people not familiar with some tech devices, poor people) and does it entail possible proposals for that?

-Yes