UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Industrial Engineering

Application of the Minimum Interoperability Mechanisms (MIMS) Prospective Standard to the EU Data Space for Smart Communities for certification of the context information manager (MIM1) for the company Libelium under the EU compliance testing infrastructure.

End of Degree Project

Bachelor's Degree in Industrial Organization Engineering

AUTHOR: Ferri Piqueras, Mateo

Tutor: Orozco Messana, Javier

External cotutor: BRYNSKOV, MARTIN

ACADEMIC YEAR: 2023/2024

Mateo Ferri

APPLICATION OF THE MINIMUM INTEROPERABILITY MECHANISMS (MIMS) PROSPECTIVE STANDARD TO THE EU DATA SPACE FOR SMART COMMUNITIES FOR CERTIFICATION OF THE CONTEXT INFORMATION MANAGER (MIM1) FOR THE COMPANY LIBELIUM UNDER THE EU COMPLIANCE TESTING INFRASTRUCTURE

PORTADA DEL TFG, ELEMENTO NORMALIZADO QUE TE LO DAN CUANDO SOLICITAS LA DEFENSA

AKNOWLEDGMENT

My co-supervisors for the Final Degree Project: Martin Brynskov (DTU) and Javier Orozco Messana (UPV).

Antonio Jara (Libelium).

Ramón Ferri Tormo (Ayuntamiento de Valencia).

Jesper Nyland and Shahrzar Pour (DTU).

Carlos Andrés Romano (UPV)

RESUMEN

La estrategia europea para los datos tiene como objetivo crear un mercado único de datos que garantice la competitividad global y la soberanía de los datos de Europa. Esto conducirá a la creación de espacios comunes de datos europeos. Garantizarán que haya más datos disponibles para su uso en la economía y la sociedad, manteniendo al mismo tiempo el control de las empresas y los individuos que generan los datos.

Un elemento clave para cumplir este objetivo clave del mercado de innovación de la UE es garantizar la interoperabilidad de datos y servicios. La Ley de Europa Interoperable (publicada el 30/11/2022) introduce un marco de cooperación para las administraciones públicas de toda la UE que ayuda a construir un intercambio transfronterizo seguro de datos y acordar soluciones digitales compartidas, como software de código abierto, directrices y listas de verificación. , marcos y herramientas de TI. Este paquete legislativo de la UE está siendo adoptado actualmente por todos los Estados miembros de la UE para fomentar prácticas comunes de cooperación en materia de interoperabilidad basadas en el actual Marco Europeo de Interoperabilidad (EIF) que se implementan en arquitecturas determinadas bajo un plan común.

En este contexto, la UE está desarrollando SIMPL (Secure Interoperability Middleware Platform Layer) como la futura plataforma de middleware inteligente y segura que respalda el acceso a datos y la interoperabilidad entre espacios de datos europeos. Esta solución de plataforma de software permitirá la federación de aplicaciones, herramientas y bases de datos basada en la nube en toda la UE, proporcionando una experiencia de usuario perfecta. Sin embargo, un elemento clave de esta plataforma es un estándar ampliamente aceptado y respaldado a nivel mundial. UIT-T (comité de normalización de las Naciones Unidas para el sector de las telecomunicaciones, https://www.itu.int/en/ITU-T/about/Pages/default.aspx) SG20-TD288-R1/GEN (2023-09) Y. El borrador de recomendación de MIM define el concepto, propósito y estructura de los Mecanismos Mínimos de Interoperabilidad (MIM) que proporcionan los requisitos para implementar las capacidades mínimas pero suficientes necesarias para lograr la interoperabilidad basada en un terreno común mínimo.

La piedra angular de la interoperabilidad para todas las arquitecturas de datos comunes es un enfoque estandarizado para la gestión de la información contextual. La implementación del software de este controlador de intercambio de datos en un entorno de nube debe ser totalmente interoperable según la propuesta de estandarización Y.MIM para garantizar una adopción adecuada de los principios de los espacios de datos de la UE.

La tesis propuesta verificará todas las capacidades y los requisitos correspondientes identificados en la implementación ITU Y.MIM de MIM1 desarrollando un procedimiento de certificación de cumplimiento para una arquitectura siguiendo la propuesta de estándar actual. El procedimiento incluye la preparación de conjuntos de datos de muestra siguiendo todos los modelos y servicios de datos relevantes en MIM1, junto con el informe de certificación correspondiente. La implementación física se llevará a cabo en una instalación piloto del Espacio de Datos de la UE para Comunidades Inteligentes para el Aarhus City Living Lab propuesto por Libelium.

Para la implementación práctica de este proyecto, la empresa Libelium ofreció su arquitectura actual para certificación. El trabajo del TFG implementará el procedimiento de certificación en la instalación piloto de Libelium del Espacio de Datos de la UE que actualmente respalda el nodo SIMPL para la instalación de pruebas y experimentación de la UE para comunidades inteligentes que requiere certificación en MIM1. Libelium proporciona este marco de prueba y experimentación al Proyecto Europa Digital CitCom.AI (https://citcom.ai/). Los conjuntos de datos desarrollados en el TFG se han preparado a partir de casos de uso existentes tomados de pruebas reales en el Centro de Experimentación y Pruebas de Ciudades Inteligentes para Comunidades Inteligentes en Dinamarca.

Bajo este entorno, la tesis propondrá y validará un procedimiento para la certificación de cumplimiento (incluidos conjuntos de datos de prueba) de la nueva arquitectura Context-broker bajo MIM 1 para su certificación.

La evaluación de cumplimiento para la certificación de la arquitectura Libelium falla en varios de los requisitos. El TFG propone una arquitectura alternativa basada en un broker de mensajes de código abierto (RabbitMQ) y vuelve a implementar las pruebas de cumplimiento que arrojaron una certificación positiva. Más allá de la certificación, la tesis incluye un análisis de costos para transformar la arquitectura anterior a la nueva solución compatible, incluidos todos los cambios propuestos.

ABSTRACT

The European strategy for data aims to create a single market for data that will ensure Europe's global competitiveness and data sovereignty. This will lead to the creation of Common European data spaces. They will ensure that more data becomes available for use in the economy and society, while keeping the companies and individuals who generate the data in control.

A key element for fulfilling this key EU innovation market objective is ensuring interoperability of data and services. The Interoperable Europe Act (published 30/11/2022) introduces a cooperation framework for public administrations across the EU that helps build a secure cross-border exchange of data and agree on shared digital solutions, such as open-source software, guidelines, checklists, frameworks, and IT tools. This EU legislative package is currently under adoption by all EU Member States fostering common interoperability cooperation practices based on the current European Interoperability Framework (EIF) which are implemented on given architectures under a common blueprint.

Within this context the EU is developing SIMPL (Secure Interoperability Middleware Platform Layer) as the future smart, secure middleware platform Supporting data access and interoperability among European data spaces. This software platform solution will enable the EU wide cloud-based federation of applications, tools, and databases providing a seamless user experience. However, a key element of this platform is a widely accepted standard supported at world level. ITU-T (UN standardization committee for the Telecommunication sector, https://www.itu.int/en/ITU-T/about/Pages/default.aspx) SG20-TD288-R1/GEN (2023-09) Y.MIM draft recommendation defines the concept, purpose, and structure of Minimal Interoperability Mechanisms (MIMs) that provide the requirements for implementing the minimal but sufficient capabilities needed to achieve interoperability based on a minimal common ground.

The interoperability keystone for all common data architectures is a standardized approach for context information management. The software implementation of this data exchange controller on a cloud environment must be fully interoperable under the proposed Y.MIM standardization proposal to guarantee an adequate uptake of the EU Data Spaces principles.

The proposed thesis will verify all capabilities, and the corresponding requirements, identified on the ITU Y.MIM implementation of MIM1 developing a compliance certification procedure for an architecture following the current standard proposal. The procedure includes the preparation of sample datasets following on all relevant data models and services on MIM1, together with the corresponding certification report. Physical implementation will take place on a pilot installation of the EU Data Space for Smart Communities for the Aarhus City Living Lab proposed by Libelium.

For the practical implementation of this project, the company Libelium offered his current architecture for certification. The TFG work will implement the certification procedure on the Libelium pilot installation of the EU Data Space which is currently supporting the SIMPL node for the EU Testing and experimentation facility for Smart Communities which requires certification on MIM1. This testing and experimentation framework is provided by Libelium to the Digital Europe Project CitCom.AI (https://citcom.ai/). The datasets developed on the TFG have been prepared from existing use cases

taken from real trials on the Smart Cities Testing and Experimentation Facility for Smart Communities in Denmark.

Under this environment the thesis will propose and validate a procedure for the compliance certification (including test datasets) of the new Context-broker architecture under MIM 1 for its certification.

The compliance assessment for certification of the Libelium architecture fails on several of the requirements. The TFG proposes an alternative architecture based on based on an open-source message broker (RabbitMQ) and implements again the compliance tests which rendered a positive certification. Beyond the certification the thesis includes a cost analysis for transforming the old architecture to the new compliant solution, including all proposed changes.

ABSTRACT

L'estratègia europea per a les dades té com a objectiu crear un mercat únic de dades que garantisca la competitivitat global i la sobirania de les dades d'Europa. Això conduirà a la creació d'espais comuns de dades europees. Garantiran que hi haja més dades disponibles per al seu ús en l'economia i la societat, mantenint al mateix temps el control de les empreses i els individus que generen les dades.

Un element clau per a complir este objectiu clau del mercat d'innovació de la UE és garantir la interoperabilitat de dades i servicis. La Llei d'Europa Interoperable (publicada el 30/11/2022) introduïx un marc de cooperació per a les administracions públiques de tota la UE que ajuda a construir un intercanvi transfronterer segur de dades i acordar solucions digitals compartides, com a programari de codi obert, directrius i llistes de verificació. , marcs i ferramentes de TU. Este paquet legislatiu de la UE està sent adoptat actualment per tots els Estats membres de la UE per a fomentar pràctiques comunes de cooperació en matèria d'interoperabilitat basades en l'actual Marc Europeu d'Interoperabilitat (*EIF) que s'implementen en arquitectures determinades sota un pla comú.

En este context, la UE està desenvolupant *SIMPL (*Secure *Interoperability *Middleware *Platform *Layer) com la futura plataforma de *middleware intel·ligent i segura que recolza l'accés a dades i la interoperabilitat entre espais de dades europees. Esta solució de plataforma de programari permetrà la federació d'aplicacions, ferramentes i bases de dades basada en el núvol en tota la UE, proporcionant una experiència d'usuari perfecta. No obstant això, un element clau d'esta plataforma és un estàndard àmpliament acceptat i recolzat a nivell mundial. *UIT-T (comité de normalització de les Nacions Unides per al sector de les telecomunicacions, https://www.itu.int/en/itu-t/about/pages/default.aspx) SG20-TD288-R1/GEN (2023-09) I. L'esborrany de recomanació de *MIM definix el concepte, propòsit i estructura dels Mecanismes Mínims d'Interoperabilitat (*MIM) que proporcionen els requisits per a implementar les capacitats mínimes però suficients necessàries per a aconseguir la interoperabilitat basada en un terreny comú mínim.

La pedra angular de la interoperabilitat per a totes les arquitectures de dades comunes és un enfocament estandarditzat per a la gestió de la informació contextual. La implementació del programari d'este controlador d'intercanvi de dades en un entorn de núvol ha de ser totalment interoperable segons la proposta d'estandardització I.*MIM per a garantir una adopció adequada dels principis dels espais de dades de la UE.

La tesi proposada verificarà totes les capacitats i els requisits corresponents identificats en la implementació *ITU I.*MIM de MIM1 desenvolupant un procediment de certificació de compliment per a una arquitectura seguint la proposta d'estàndard actual. El procediment inclou la preparació de conjunts de dades de mostra seguint tots els models i servicis de dades rellevants en MIM1, juntament amb l'informe de certificació corresponent. La implementació física es durà a terme en una instal·lació pilot de l'Espai de Dades de la UE per a Comunitats Intel·ligents per al *Aarhus *City *Living *Lab proposat per *Libelium.

Per a la implementació pràctica d'este projecte, l'empresa *Libelium va oferir la seua arquitectura actual per a certificació. El treball del *TFG implementarà el procediment de certificació en la instal·lació pilot de *Libelium de l'Espai de Dades de la UE que actualment recolza el node *SIMPL per a la instal·lació de proves i experimentació de la UE per a comunitats intel·ligents que requerix certificació en MIM1. *Libelium proporciona este marc de prova i experimentació al Projecte Europa Digital Citcom.ai (https://citcom.ai/). Els conjunts de dades desenvolupades en el *TFG s'han preparat a partir de casos d'ús existents presos de proves reals en el Centre d'Experimentació i Proves de Ciutats Intel·ligents per a Comunitats Intel·ligents a Dinamarca.

Sota este entorn, la tesi proposarà i validarà un procediment per a la certificació de compliment (inclosos conjunts de dades de prova) de la nova arquitectura *Context-*broker baix *MIM 1 per a la seua certificació.

L'avaluació de compliment per a la certificació de l'arquitectura *Libelium falla en diversos dels requisits. El *TFG proposa una arquitectura alternativa basada en un *broker de missatges de codi obert (*RabbitMQ) i torna a implementar les proves de compliment que van llançar una certificació positiva. Més enllà de la certificació, la tesi inclou una anàlisi de costos per a transformar l'arquitectura anterior a la nova solució compatible, inclosos tots els canvis proposats.

TECHNICAL VOCABULARY

IaaS = Infrastructure as a Service

EU = Europe

AI = Artificial Intelligence

DTU = Technical University of Denmark

DS = Data Space

MIMs = Minimal Interoperability Mechanisms

ITU-T = International Telecommunication Union - Telecommunication Standardization Sector

SIMPL = Secure Interoperability Middleware Platform Layer

TEF = Testing and Experimentation Facilities

DS4SSCC = Data Space for Sustainable Smart Cities and Communities

OASC = Open and Agile Smart Cities

NGSI-LD = Next Generation Service Interfaces-Linked Data

API = Application Programming Interface

URI = Uniform Resource Identifier

HATEOAS = Hypermedia as the Engine of Application State

FIWARE = Future Internet Ware

JSON = JavaScript Object Notation

XML = Extensible Markup Language

RESTful API = Representational State Transfer Application Programming Interface

INDICE

1 Introduction

There are different Infrastructure as a Service (IaaS) providers that offer a wide range of services and capabilities to companies and developers over the internet, enabling access to virtual computing resources and other services. Some of the most notable examples include Azure, Amazon Web Services, Google Cloud, IBM Cloud, and Alibaba Cloud, which are among the leaders in the world. Interestingly, there is no European provider at this level. "Data Space" is a European concept born out of Europe's need for its own cloud infrastructure.

Over the years, the value of data has increased dramatically due to the wealth of useful information it can contain. The digital world has advanced at an incredibly fast pace, yet Europe has lagged in providing infrastructure for storing and working with data in the cloud. This has led to the widespread use of American or Chinese options worldwide, including in Europe, such as Azure, Amazon Web Services, Google Cloud, IBM Cloud, and Alibaba Cloud.

Therefore, the European Strategy for data [50] aims to create a single market for data that ensures global competitiveness and data sovereignty for Europe. This will lead to the establishment of European common data spaces, ensuring more data availability for use in the economy and society while maintaining data sovereignty.

The European Commission has launched numerous projects, demonstrating its commitment to creating a single market for data and its willingness to promote a society based on the efficient use of that data. The "Digital Europe Programme" [28] initiative aims to provide funding for these projects. It supports and is fully aligned with the European Union's goals of a green and digital transition, as well as the Sustainable Development Goals [52].

A key element in achieving this objective is ensuring the interoperability of data and services, as it is an indispensable and critical feature for information sharing. To this, the European Union introduces a cooperation framework through the European Interoperability Act [24]. And to promote common cooperation practices in interoperability, the European Interoperability Framework (EIF) [13] is implemented. In this context, the EU is developing SIMPL [25] (Secure Interoperability Middleware Platform Layer), funded under the "Digital Europe Programme" [28], which aims to support European common data spaces and all major data initiatives financed by the European Commission. SIMPL is an intelligent middleware that promotes interoperability in data spaces and contributes to the EU's data strategy.

Another project funded under the Digital Europe Programme is CitCom.AI Testing and Experimentation Facilities [29], focusing on the testing and validation of data-driven services and products within the EU. CitCom.AI TEF belongs to the Smart Cities domain and, along with Health, Agriculture, and Manufacturing, forms the four Test and Implementation Facilities [53].

The CitCom.AI TEF [29] project is directly related to the Data Space for Sustainable Smart Cities and Communities (DS4SSCC) [48] project. The testing and experimentation of the CitCom.AI TEF project are conducted within the data space of smart and sustainable cities and communities under the DS4SSCC project, using a layer that enables secure and reliable interoperability in the data space (SIMPL) [25].

A key element of these projects is a widely accepted and globally endorsed standard. ITU-T [1] (United Nations standardization committee for the telecommunications sector) SG20-TD288-R1/GEN (2023-09) Y.MIM. This draft defines the concept, purpose, and structure of Minimum Interoperability Mechanisms (MIM) [36], which provide the requirements for implementing minimal yet sufficient capabilities necessary to achieve interoperability based on a common minimum ground.

Under this standard, this work will develop a certification procedure to assess compliance with Context Information Management (MIM 1) requirements, aiming to measure the level of interoperability of CitCom.AI data space relevant architectures. This procedure will be applied to the architecture of Aarhus Living Lab, which is relevant to the company Libelium and to the CitCom.AI and DS4SSCC projects, directly aligned with SIMPL.

The sustainable development goals most aligned with this context are:

Figure 1: Source: own elaboration. https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/

1.1 Scope & Description

Throughout my university degree in Industrial Management Engineering (Ingenieria en Organización Industrial) at the Political University of Valencia (Universidad Politecnica de Valencia (UPV)). During my period at the university, I learned to understand and know the characteristics of different systems analysing and studying them seeking to improve processes, reduce wastes and making the systems more effective and efficient. Subjects like *Estudio del Trabajo, Métodos Cuantitativos de Organización Industral, Diseño de Sistemas Productivos y Logísticos, Control Estadístico de Calidad, Sistemas Integrados de información, Organización de la Producción* and more, helped me to acquire the knowledge needed to study and analyse different systems and processes. However, I could not achieve the approach proposed on this thesis without all the knowledge and aptitudes learned and obtained during my university period from other subjects and Erasmus periods.

The approach of this work is the Interoperability in the context of Data Spaces. This thesis is embraced by a European project named CitCom.AI Testing and Experimentations Facilities (TEF) [2], which will be explained later in this document. Thanks to the Erasmus internship period I have been able to work in the CitCom.AI TEF in Copenhagen, in the Technical University of Denmark (DTU) [3], with some of the experts in this domain, and with the person who is the head of the DTU Smart Communities Centre in Denmark, the coordinator of the CitCom.AI TEF. This thesis also has been possible due to Libelium, it is a Spanish technological company that is one of the participants of the CitCom.Ai TEF and they hired me in the middle of the Erasmus internship, and they provided me a lot of documents, information and training and helped me with this thesis. Another entity that helped me in this work is the Council of Valencia (Ayuntamiento de Valencia), concretely the Service of Smart Cities (Servicio de Ciudades Inteligentes) which is also one participant in the TEFs, and they helped me providing information about some Use Cases of the TEF in Valencia to my personal better understand of the domain.

The scope of this work combines the knowledge acquired throughout the degree, along with those acquired during the Erasmus period, internships at DTU and the time worked at Libelium also with the data provided by the company, the city council of Valencia and the Technical University of Denmark, all of these as participants in CitCom.AI TEF.

1.2 Object & Objectives

The area of application of this project is Data Spaces. Data Spaces are the key elements within the digital EU Strategy [50] in the Digital Europe Programme [28] for a common infrastructure on data and services (including AI applications). Based on a review of the current literature, this project explains what the concept of data space is, trying to give the reader a global vision of key technical and non-technical elements that Data Spaces encompass, like on one hand: definition and characteristics, and on the other hand global context, benefits, the reasons why this concept was created and developed, etc. The keystone of the common data spaces is the interoperability, which term is going to be explained and contextualize in the European context.

Data spaces and interoperability are going to be a key component in Industry 4.0, which will become more efficient as it becomes more interoperable with all actors in its supply chain.

The objective of the practical part of this work is to develop a certification procedure for compliance with the requirements identified in the ITU Y.MIM [1] implementation of MIM 1: Context Management Information. The physical implementation of this certification procedure is intended to be carried out on a pilot testing installation of the European Union Data Space for Smart Cities and Communities, Aarhus Living Lab. Moreover, on the same Aarhus Living Lab pilot, but with another architecture focused on meeting these requirements of MIM 1, the certification procedure for compliance will be implemented to analyse the compliance of this new improved architecture since it should meet the requirements of MIM 1.

This compliance certification procedure for the requirements is an important part of task 3 of the Smart Cities project CitCom.AI TEF, as well as the mapping and implementation of this in the Aarhus Living Lab use case, a test pilot of the CitCom.AI TEF project, from the northern node.

1.3 Motivation & Justification

As a student, I have had the opportunity to come to Copenhagen, Denmark. To learn and study what is going to be the future of digital communications and the ways to manage data and to enhance my English as a personal goal. This internship has place at the Technical University of Denmark.

Additionally, as a student I am studying a master in Artificial Intelligence, which is a very cutting-edge topic at this moment and very related with CitCom.AI TEF [29]. With this thesis I am positioning myself in an innovative context that is aiming to change the communications in Europe and in the world in the next years, looking for the efficiency and the continuous improvement. For all this reasons, I am very motivated with this work, and I am doing my best in the process.

1.4 Thesis structure

The structure of this thesis is:

Introduction: This section provides a broad framework of the study, including the scope and description of the work, objectives, motivation, and justification of the study, followed by a summary of the thesis structure.

Background: This part covers non-technical elements, such as an overview of Data Spaces and their value, the relevance of interoperability and Minimal Interoperability Mechanisms. It also discusses technical elements, including definitions and features of EU Data Spaces, principles of interoperability, design patterns, and the use of a Message Broker.

Methodology: The analysis of the requirements of the MIM 1 and the Message Broker tool used in the research are detailed, establishing the methodological framework for the study's implementation and evaluation.

Implementation: Describes the practical application of the research in the Aarhus Living Lab, including the current test architecture and the proposed architecture, both assessed with the certification procedure explained at the Methodology.

Evaluation: Assesses the outcomes of the implementation and the economic impact of the study, providing a critical appraisal of the findings and their significance in the project context.

Conclusion: Offers a summary of the findings, cost evaluation, and contributions of the study, as well as suggestions for future research.

References: Lists all sources cited in the document, providing the necessary academic and technical backing for the research.

Annexes: Includes additional material supporting the study.

2 Background

2.1 Non-Technical elements

2.1.1 Overview of Data Spaces

2.1.1.1 Data Spaces

A **Data Space** is defined as "A distributed system defined by a governance framework that enables secure and trustworthy data transactions between participants while supporting trust and data sovereignty. A data space is implemented by one or more infrastructures and enables one or more use cases."

This concept will be widely and technically explained in the next part of the work: "EU Data Spaces" in "Technical Elements", aiming to align the concept with the focus of this work.

2.1.1.2 Example of Data Space

Based on the book "The Evolution of Data Spaces" [4] is going to be explained one example of a Data Space, concretely a Data Space for mobility.

The Figure 2 shows the architecture model of a Mobility Data Space. The architecture of this Data Space consists of three layers:

1. At the zenith of this intricate structure lies the realm of connectivity, a vibrant tapestry weaving together the diverse threads of the mobility data ecosystem. Here, a myriad of players - from commuters navigating the urban sprawl to public transport entities and avant-garde car-sharing ventures - engage in a dynamic dance of data exchange. Their goal is dual-pronged: firstly, to elevate the comfort of those traversing the city's arteries, and secondly, to deftly orchestrate the pulsating rhythms of traffic and the ceaseless flow of passengers.

2. Nestled in the heart of this system is the middle layer, a digital crucible where the tangible world is mirrored in the ethereal realm of data. This layer is the cornerstone of the mobility data ecosystem, meticulously crafting digital doppelgängers of the myriad entities that populate the mobility landscape. This act of creation is not mere mimicry; it's a vital cog in the machine of data sharing, necessitating the fusion of disparate data streams from a kaleidoscope of sources. This melding births innovative mobility solutions. Yet, to weave these varied data strands into a coherent tapestry, a lingua franca of data must be established. Take, for example, the electric mobility sector, where charging stations are not merely points on a map, but complex entities defined by a harmonized lexicon of attributes - type, locale, charging modes, power levels, and more.

3. Anchoring this architectural triptych is the foundational layer, a robust software infrastructure that underpins the creation, stewardship, and dissemination of these digital twins' data. This layer is a testament to the philosophy that, in the vast, multifaceted domain of application areas, a decentralized structure trumps a centralized one in adaptability and efficacy. Nevertheless, this decentralized approach does not forsake the need for uniform software components and gateways, pivotal in knitting together data providers and users into the fabric of the data ecosystem. Moreover, it

demands a suite of communal services, ensuring that data not only flows but converses fluently across the distributed software landscape.

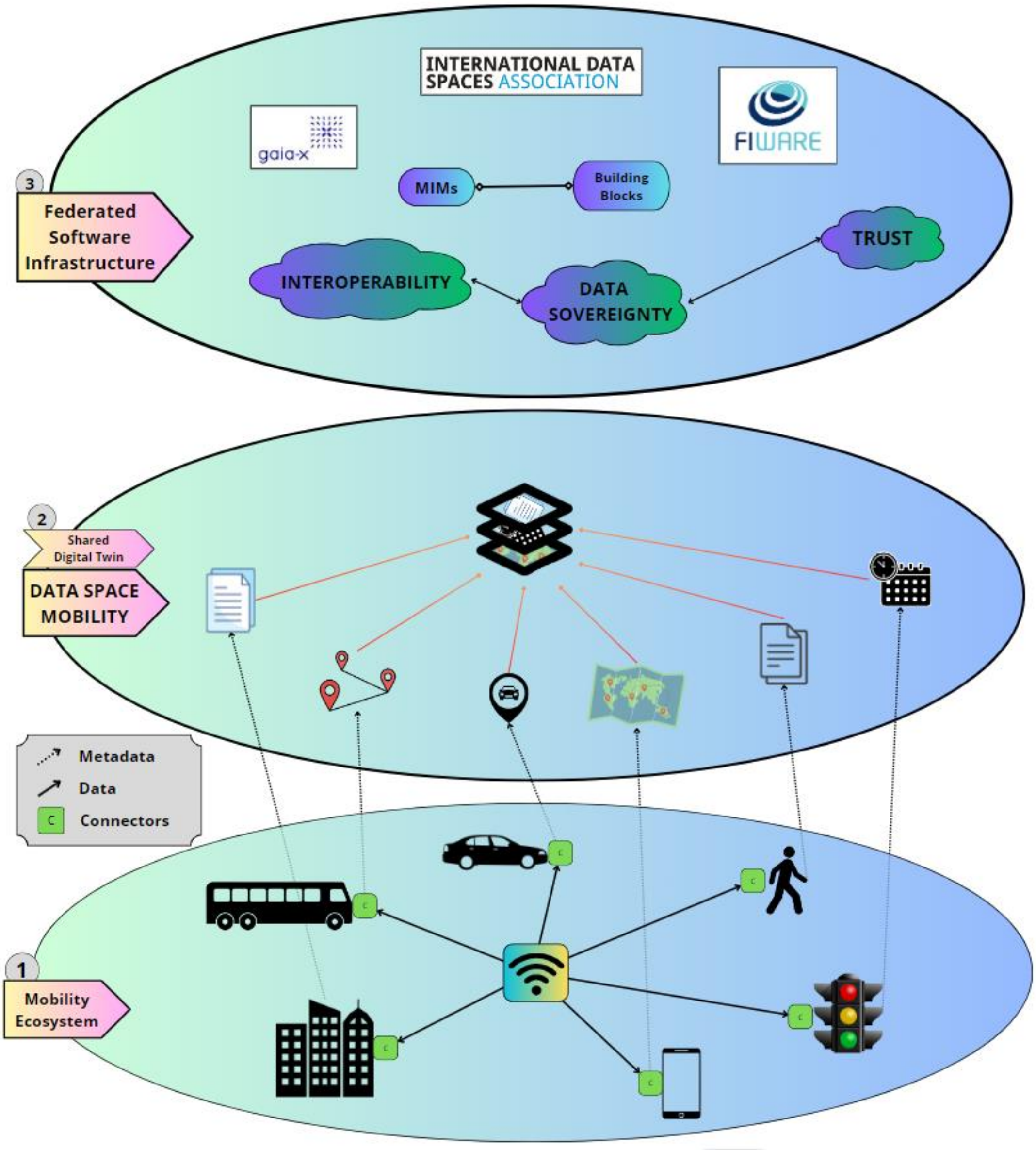


Figure 2: Source: own elaboration.

2.1.2 The value of Data

At this point, one important aspect to consider is the Data Market, to understand how is valued the data and what is understood by the Data Market.

The European Data Market Study [54] defines data markets as marketplaces where digital data, and insights from data, are exchanged as products or services that result from the elaboration of raw data. Examples of these are credit scores, client profiles, or insights on the traffic in a particular city. On the demand side, different companies and governments acquire data or these data insights to use in their decision-making processes and make their business more efficient. Governments, as demanding entities, use all available data to make the best decisions and achieve data-driven applications that will benefit citizens and businesses: for example, improve health care, create safer and cleaner transport systems, generate new products and services, reduce the cost of public services, improve sustainability and energy efficiency…

To understand the actual value of data for business, both public and private, is necessary to know the roles that data takes at this moment because data has been continuously changing over the las years.

Based in the book “Designing Data Spaces” [4], nowadays, four types of roles of data can be identified.

1. Data enables the possibility to achieve operational improvements in a company. An effective and efficient data management is necessary to integrate and automatize processes in a business.
2. Data has become a product. The information of data from one source can be used for other purposes and gain value with it, this gives data a variable value in the market depending on the usefulness of that data.
3. Data is useful for the business innovation. Data-driven services like end-to-end maintenance services needs to be provided by data from different sources.
4. Data is considered a resource for the long-term sustainability of the economy. The European Commission launched the EU Data Market Study in 2013 to measure value and study the EU Data Economy. According to the EU Data Market Monitoring Tool (2020) [5], the EU Data Market Study estimates that data will reach a value of 550 billion Euro in the EU 27 in the most realistic scenario.

Despite the context of this work being related to Smart Cities and Communities, it is fully aligned with the estimation of data value in business and enterprises, as in both cases, data is used to create products and services.

2.1.3 Relevance of Interoperability

To fully leverage the value of the data presented in the previous section, it is necessary that the use of the data be efficient, reusable, and scalable. The value of data is significant because data can be reused without losing value. It is not just about the value a piece of data provides to one company, but rather the potential value that any company can extract using that data. For instance, one company may use customer data to determine the location of a new store, while another may use the same data to offer a home delivery food service, among many other possible solutions for that same data. The same applies to sensor data; for example, a meteorological agency uses sensor data to provide weather forecasts, but an agricultural company can use that data to conduct their own studies and improve crop yields. This exploitation of data involves two distinct elements:

- Data holds high value as long as it is understandable and relevant within a certain context.
- A dataset that can only be utilized by one company holds lesser value compared to a dataset that can be leveraged by multiple companies to derive benefits.

For data sharing between systems and to harness their value, the involved systems must be interoperable with each other. This entails that there exists a certain form of communication among the systems sharing information, enabling them to connect and exchange data.

The concept of interoperability will be technically explained later in this document, but for the context of this work, the accepted definition is as follows: "Interoperability is about the degree to which two or more systems can usefully exchange meaningful information via interfaces in a particular context."

As stated in the definition, interoperability is not a binary condition of either "yes" or "no"; there are different levels of interoperability depending on the exchange of information between systems. For instance, Tim Berners-Lee's 5-Star Model [55] is a framework for rating the quality of data on the web based on its level of openness and availability, in other words, how interoperable the data is. Utilizing the information from this model, the author of this work has created the following representation (Figure 3):

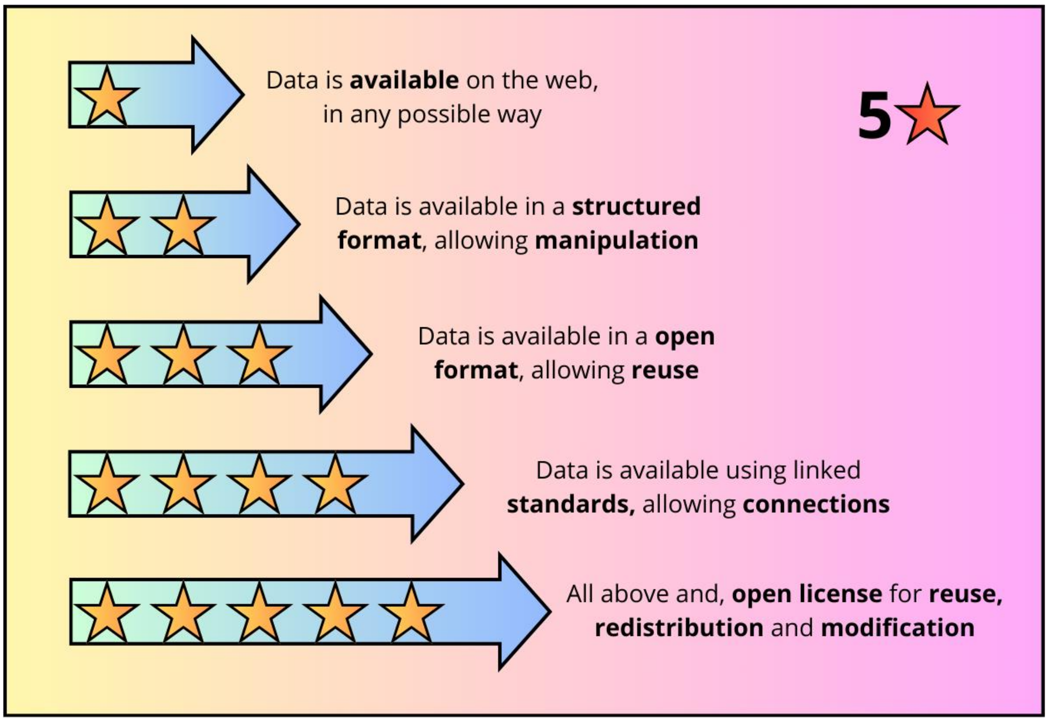


Figure 3: Source: own elaboration.

For the context of this work, we will outline the main barriers to interoperability in cities and communities, which contribute to the complexity of the issue and the existence of numerous European projects promoting interoperability.

CHALLENGES

There are many problems and challenges in the cities and communities in the mission to enable data and services to work together and to support a scalable and open market of solutions. In this point are presented some of the main problems presented in webinars, conferences and meetings inside the European projects involved in the Interoperability context.

1 Cities and communities consist of many autonomous or semi-autonomous agencies, both public and private, that provide services, and it is difficult, if not impossible to ensure that all agencies in a city follow identical processes and standards.

2 Cities and communities also are managed independently of each other and make decisions based on their own needs and background.

3 Many cities and communities have contracts that may require the use of proprietary solutions.

4 There are many Standards Development Organizations, each building families of standards from different viewpoints, and these standards are not always compatible with each other.

5 The very detail and complexity of the standards landscape may not allow to the smaller cities and communities or those with limited resources from to attempt standards implementation.

2.1.4 Minimal Interoperability

Minimal interoperability in this context refers to the minimal level of interoperability that all parties in a system must accomplish to interoperate with other parts within the system. If one part does not accomplish that minimal level, it will not have a seamless communication with others within the system.

For example, in a meeting about Data Spaces, there could be participants of all kinds, such as a person who is an expert in interoperability, another who is an expert in European regulations, another person who is an expert in use cases, and many more participants who are experts in different subjects. The objective of the meeting is, for example, to organize among all the parties of the meeting to issue a statement to the cities about the importance of Data Spaces in Europe. In order to release that statement, all the experts must participate by contributing their knowledge so that the statement has the correct information in each area. The maximum level of interoperability would be for all participants to be experts in everything, in interoperability, in regulations, in use cases, in organization… which is practically impossible. The minimum level of interoperability in that meeting would be that everyone, at least, knew the concept of data spaces, therefore, in order to follow the thread of the meeting or contribute something, all participants must know the concept of data spaces. However, that is not the only requirement to meet the minimum interoperability in that meeting, and that is, if each participant speaks a different language, understanding among the participants would be impossible, therefore, a common language is necessary. If all participants share a common language and are knowledgeable about the concept of Data Spaces, the meeting can be carried out optimally and there will be fluid communication among the participants.

Therefore, in any ecosystem, for seamless communication and the desired minimum interoperability to occur, participants of said ecosystem must meet some minimum requirements. Each ecosystem will have its own requirements and standards. Of course, participants may have a greater capacity to interoperate among themselves and this will improve communication, but it is optional.

The concept of minimum interoperability provides a scenario where there are minimum requirements that allow a participant with low resources not to have a higher expense than strictly necessary to be interoperable within the ecosystem. For participants who are already part of another ecosystem and have their own requirements developed and implemented, but who want to be part of another ecosystem, it is not necessary to change their entire way of communicating, but by meeting the minimum requirements of the new ecosystem they are joining, they allow a minimum level of interoperability between the participants of the new ecosystem.

Projects like SIMPL [25] or TEFs [29] are the most important projects of Data Spaces in Europe are directly aligned with the concept of interoperability. These projects will be explained in this document, and the common key element of those platforms that allows interoperability is a widely accepted standard supported at world level. ITU-T [1] (UN standardization committee for the Telecommunication sector) SG20-TD288-R1/GEN (2023-09) Y.MIM draft recommendation. Concretely, defining the format and concept of the Minimal Interoperability Mechanisms (MIMs).

2.2 Technical elements

2.2.1 EU Data Spaces Definition and other concepts

There is still a lack of a generally accepted definition of the term data space. Even in central texts (Regulation (EU) 2021/694), a definition is often avoided. The definition of Data Space could be different depending on the context it is applied.

Here there are some definitions gathered from different sources.

- From the European Commission in the official webpage [6]: "Data spaces are decentralised infrastructures, where diverse actors can share and use data in a secure, reliable and trustworthy manner, following common governance, organisational, regulatory and technical mechanisms."
- Open DEI [7]: "A data space is defined as a decentralised infrastructure for trustworthy data sharing and exchange in data ecosystems based on commonly agreed principles.
- Gaia-X [8]: "Data Spaces are defined as: A federated, open infrastructure for sovereign data sharing, based on common policies, rules and standards."
- The Data Space Business Alliance [9] define a Data Space as a data ecosystem built around commonly agreed building blocks, enabling an effective and trusted sharing of data among participants to create value.
- Internacional Data Spaces [10]: Data Spaces are protected environments in which participants can freely exchange data by adhering to a clear set of rules that protect data sovereignty and guarantee transparency and fairness.

All these definitions are similar and are provided by different actors very related with the concept. However, for the scope of this work, and for the relationship of this work with the projects CitCom.AI TEF [29] and Data Spaces for Smart and Sustainable Cities and Communities (DS4SSCC) [48], the definition taken by this work as the most accurate and tight to the reality is the one given by Data Space Support Centre [45]. The main reason for this election is because:

- As it is explained in the Anex 1 in this work, Data Spaces Support Centre is coordinating all data spaces in different domains in Europe to ensure all of them are interoperable, so it is the union and knows all parties, this results on the fact that definition has been co-created with different data space initiatives and endorsed by a relevant team of experts in the field.

Data Space is defined by the DSSC [45] in its glossary and its concepts are recent, and are being revised and uploaded, for this reason there are two versions of the glossary at the day this work done (05/01/2024). The definitions of data spaces in the glossary are:

Glossary v1.0 (March. 2023) [11]: **Data Space** is "An infrastructure that enables data transactions between different data ecosystem parties based on the governance framework of that data space. Data space should be generic enough to support the implementation of multiple use cases."

Glossary v2.0 (Sept. 2023) [12]: **Data Space** is "A distributed system defined by a governance framework that enables secure and trustworthy data transactions between participants while supporting trust and data sovereignty. A data space is implemented by one or more infrastructures and enables one or more use cases."

Which are the parts of a Data Space? What is a Data Space made of? The simple answer for these questions is **Building Blocks**, basically explained is like this: imagine a Data Space as a table, to build a table you need four legs and a board, so each leg and the board are Building Blocks.

A Building Block is "A basic unit or component that can be implemented or a capability that can be deployed and combined with other building blocks to achieve the functionality of a data space. Data space building blocks are divided into organisational and business building blocks and technical building blocks."

This definition is given by Data Space Support Centre, from the glossary v2.0. [12]

The more traditional way of handing local data ecosystems was Data Lake where data is brought from different sources into a central repository where all the data can be effectively in one place and then, all allowed entities can reuse as needed. The difference between Data Lake and Data Spaces is that in Data Lake, everything is gathered into a single place and Data Space is about setting up a data ecosystem where members agree the standards and the governance rules.

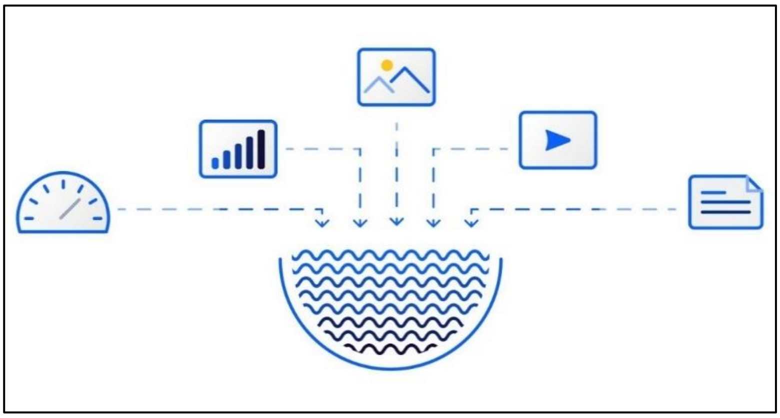


Figure 4: Source: Adservio (2021)

The Figure 4 shows very accurately the concept of Data Lake, all kind of data is stored in one place. This way of handling data is affective, but is not the only way, nor the best.

2.2.2 EU Data Spaces Characteristics & Principles

Regarding what Open DEI published in the book "Design Principles for Data Spaces" [4]. Open DEI proposes in the book four principles on which to base the construction of data spaces, with the objective of make data spaces a solid and sustainable foundation for the next growth cycle within the digital economy of the EU. There are 4 design principles of Data Spaces defined by Open DEI:

1. Data Sovereignty: is the capability of a natural person or a corporate entity for exclusive govern and control of the data generated within its borders.
2. Data level playing field; implies that new entrants face no insurmountable barriers to entry because of monopolistic situations. When data level playing field exists, players compete on quality of service, and not on the amount of data they control. A data level playing field is a pivotal condition to create a fair data sharing economy.
3. Decentralised soft infrastructure: The data sharing infrastructure is not a monolithic centralised IT infrastructure. Instead, it is the de facto collection of interoperable implementations of data spaces which comply to a unified set of agreements in all disciplines: functional, technical operational, legal, and economic. A 'soft infrastructure' as it is merely invisible and made up of agreements. Out of the principle of data sovereignty follows functional and non-functional requirements of interoperability, portability, findability, security, privacy, and trustworthiness.
4. Public-Private governance; For the design creation and maintenance of the data level playing field a sound governance is essential. All stakeholders need to feel represented and engaged. These include users (persons, business) or provider of data services as well as their technology partners and professionals.

Open DEI also provides in the same book the taxonomy of Building Blocks (BB)comprised in a data space design. This taxonomy and building blocks are accepted and assumed as correct by Data Space Support Centre and DSSC and so on by CitCom.AI, all projects on which this work is related on and based. The Building Blocks are divided into Technical BB and Governance BB:

Technical Building Blocks

- Data interoperability
- Data sovereignty
- Data value creation

Governance Building Blocks

- Business agreements
- Operational agreements
- Organizational agreements

2.2.3 Interoperability

2.2.3.1 Definition

Here are some technical definitions of interoperability from some sources.

The definition of interoperability given in the **European Interoperability Framework** (EIF) [13]: "the ability of organisations to interact towards mutually beneficial goals, involving the sharing of information and knowledge between these organisations, through the business processes they support, by means of the exchange of data between their ICT systems.".

The definition of Interoperability in the **EIF4SCC** [14] it is very similar to the definition given in the EIF, but with the difference that is more focused on the local systems within the cities and communities:

"The ability of organisations and individuals to interact towards the delivery of services in cities and communities, through the exchange of data, information, and knowledge, enabled by aligned processes and digital technologies, taking into account security and privacy issues."

It is also given in the **NIST** [15] book in 2010: "Interoperability is the capability of two or more networks, systems, devices, applications, or components to exchange and readily use information securely, effectively, and with little or no inconvenience to the user."

Interoperability is also defined in the book **Software Architecture in Practice** (3rd edition) [16]:

"Interoperability is about the degree to which two or more systems can usefully exchange meaningful information via interfaces in a particular context."

For this work, this definition is the most general, it tackles most of the possible scenarios where interoperability can be identified, using common vocabulary, and making the concept understandable. However, there are some aspects for this definition (explained below) that will help to understand why this concept is the most appropriate.

- The definition includes the ability to exchange information, what is syntactic interoperability, but also with the word "meaningful" refers to the ability to correctly interpret the information being exchanged.
- The definition uses the word information, and it tackles all, it can be data, services, applications…

2.2.3.2 Types of Interoperability

The kinds of interoperability are considered mainly by the European Commission through the "European Interoperability Framework" and the European Research Cluster on the Internet of Things through the "IoT Semantic Interoperability".

European Interoperability Framework. [13]

- Legal Interoperability: legal interoperability refers to the ability of organizations to align their operations and services with the applicable legal framework. This type of interoperability ensures that interactions between entities are conducted in accordance with current laws, regulations, policies, and standards.
- Organizational Interoperability: this level of interoperability focuses on optimizing and coordinating work processes, decision-making, and information exchange.
- Semantic Interoperability: it deals with ensuring that the meaning of information is understandable and preserved throughout its exchange between systems and organizations. This aspect is crucial for allowing computer applications and users to unambiguously interpret shared data.
- Technical Interoperability: refers to the ability of systems to connect and operate together. This level of interoperability encompasses aspects such as application programming interfaces (APIs), communication protocols, data formats...

Internet of Things Semantic Interoperability: Research Challenges, best practices. [17]

- Technical Interoperability: It primarily revolves around communication protocols and is crucial for enabling devices to connect and communicate effectively, laying the foundation for seamless interaction between various technological systems.
- Syntactical interoperability concerns the structure and format of data. It ensures that the messages exchanged adhere to a well-defined syntax and encoding.
- Semantic interoperability focuses on the meaning and interpretation of content. It aims for a mutual comprehension of the content exchanged, ensuring that information is not only transmitted but also accurately interpreted by different stakeholders. This level of interoperability is crucial for achieving meaningful communication between participants.
- Organizational Interoperability: this level of interoperability is fundamental for achieving meaningful communication across diverse entities. It relies on the successful integration of technical, syntactical, and semantic interoperability.

2.2.3.3 Interoperability in EU Framework

Cities and communities are confronted with complex challenges, such as climate change, housing quality, health and social issues, energy efficiency and urban mobility. Interoperability is crucial to overcome these challenges in cities and communities. Lack of interoperability leads to fragmented service delivery at local level as well as a lack of communication among different platforms, technologies, and stakeholders, resulting in suboptimal services to the public sector, but also for the private sector.

This point aims to cover the concept of Interoperability in Europe. Starting from the CADDIA Programme [18] and INSIS Programme, both from 1985 until 1992, concerning the coordination of the activities related to the implementation of a long-term programme for the use of telematics for Community information systems.

The next programme is the Interchange of Data between Administrations (IDA) [19].

The first phase of the Interchange of Data between Administrations (IDA I) contributed to the establishment of large telematic networks in the areas of employment, health, agriculture, statistics, and competition (started in 1995).

The IDA II programme (1999-2004) supports the implementation of projects of common interest relating with the areas in the IDA I.

The Interoperable Delivery of Pan-European eGovernment Services to Public Administrations, Business and Citizens (IDABC) [20] is an eGovernment programme for the period 2005-2009. It succeeds the IDA project and covers the objectives of the IDA programme but also aims to create pan-European eGovernment services for businesses and citizens as its tittle suggests.

The IDABC programme ended on 31 December 2009. [21] The Commission proposed a follow-on programme on Interoperability Solutions for European public Administrations (ISA programme).

The ISA^1 [22] programme facilitates efficient and effective electronic cross-border and cross-sectoral interaction between public administrations across Europe supported by interoperable electronic public services. (2010 – 2015)

The ISA^2 [23] is the follow-up programme to ISA^1, which ran from 2010-2015. This programme aims to ensure that interoperability activities are well coordinated at EU level, also to develop and operate solutions for the public administrations based on businesses and citizens' needs and to put in place the necessary instruments to boost interoperability at EU and national level. These instruments are, for example:

- A revised European Interoperability Framework (EIF).
- A revised European Interoperability Strategy (EIS).

The following picture is from the official website of the European Union:

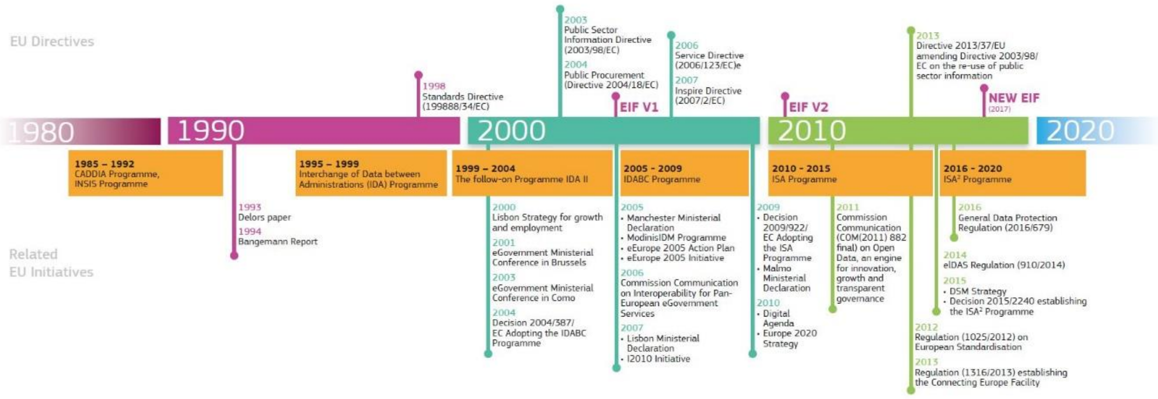


Figure 5: European Interoperability Timeline

The author of this work has made another picture of the history, more focused on the visualization and not in the details aiming to complete the information at this point with an image.

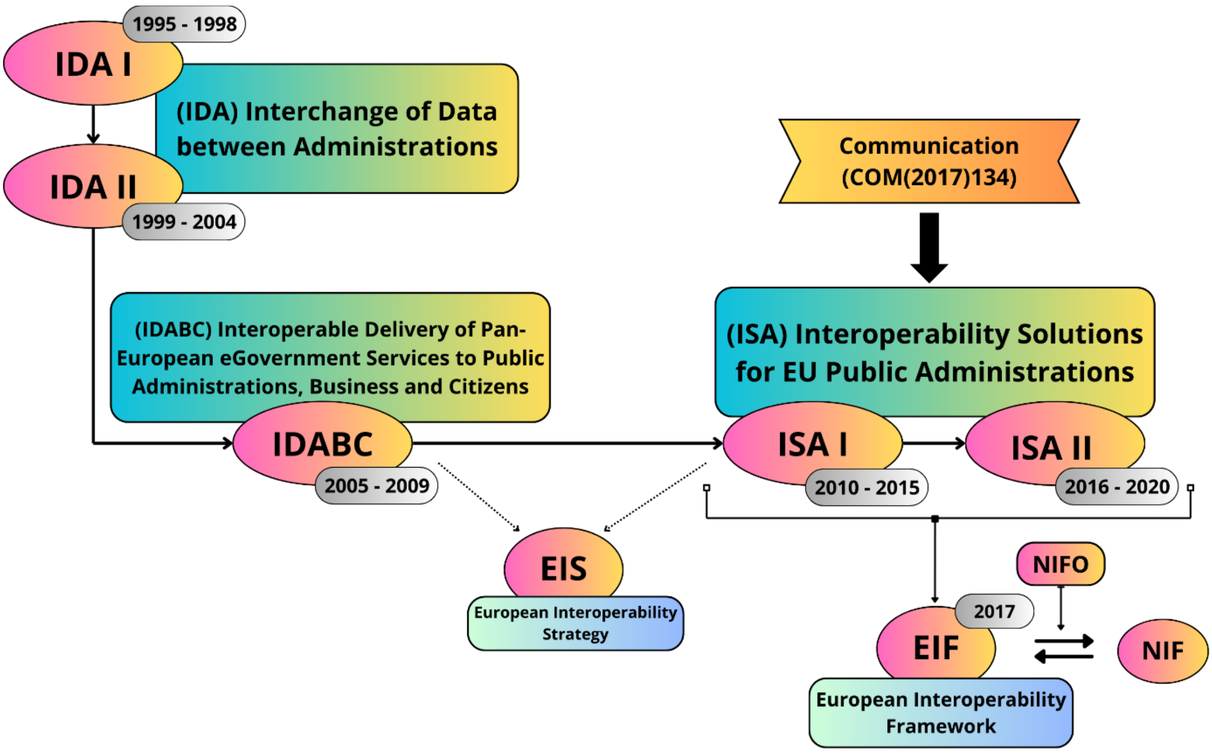


Figure 6: Source: own elaboration.

The **European interoperability framework** (EIF) [13] is a commonly agreed approach to the delivery of European public services in an interoperable manner. Through a set of recommendations, it defines basic interoperability guidelines in the form of common principles, models, and recommendations.

European Interoperability Framework EIF for eGovernment had been in place since 2004, but local context was not included. The EIF4SCC, for the first time introduces the concepts and ideas adapted to the local context.

The aim of the **European Interoperability Framework for Smart Cities and Communities** (EIF4SCC) [14] is to provide leaders of EU local administrations with definitions, principles, recommendations, practical use cases drawn from cities and communities from around Europe and beyond, and a common model to facilitate delivery of services to the public across domains, cities, regions, and borders.

Another key element is the **European Interoperability Act [24]**, proposed by the European Commission in November 2022, is a significant legislative initiative aimed at enhancing cross-border cooperation and interoperability across the public sectors within the European Union. It's designed to facilitate the seamless exchange of data and the adoption of shared digital solutions.

The Act outlines a structured EU cooperation framework involving public administrations, supported by both public and private actors, to develop, share, and reuse interoperable solutions.

This Act represents a significant step towards achieving a high level of public sector interoperability across the EU, ensuring a more integrated, efficient, and innovative digital public sector.

The Cloud Federation / Smart Middleware for a European cloud federation and for the European data spaces, **SIMPL** [25] (Secure Interoperability Middleware Platform Layer) is being developed by the European Commission financed under the DIGITAL Europe programme [26]. SIMPL is a secure middleware platform developed to facilitate data access and ensure interoperability within the Common European Data Spaces.

Among other things, SIMPL Will:

- Give data providers full control over who accesses their data in such data spaces (data sovereignty).
- Ensure that data sets and their infrastructure can be seamlessly interconnected and made interoperable.
- Allow the replacement or addition of components without affecting the rest of the system.
- Ensure trust, confidence and compliance with regulations are built into the system. This implies an effortless sharing of resources between participants, regardless of their data processing environment. It creates an abstraction layer that enables data to flow across multiple providers and Member States.

These characteristics are completely aligned with the objectives of the Minimal Interoperability Mechanisms (MIMs) [31].

MIMs, conceptually, refer to the idea of establishing a minimum set of capabilities and requirements that enable minimal yet sufficient interoperability between systems, platforms, or applications. Within the context of interoperability are included; trust, transparency, security, data models, among others, are included.

The relationship between both is direct, as while MIMs establish the general principles and requirements for interoperability, SIMPL provides technical solutions and the necessary infrastructure to put these principles into practice. Specifically, one of the practical implementations that SIMPL aims to develop is:

“Simpl-Labs: an environment for data spaces to assess the open-source software and assess their level of interoperability with Simpl. .... Furthermore, more mature data spaces will be able to use Simpl-Labs to assess their level of interoperability with Simpl-Open.”

The MIMs define a series of requirements and capabilities that allow measuring the level of interoperability of a data space. To achieve this, a certification procedure is necessary to measure the compliance with the requirements of an architecture, which is what this work proposes, specifically for the Context Information Management MIM 1.

PARTICIPANTS

Open and Agile Smart Cities (OASC) [30]

Data Space Business Alliance [46]

Living-in.EU [47]

Data Space Support Centre [45]

These and others are described in the Anex 1.

2.2.3.1 Testing and Experimentation Facilities

The Data Space for Smart, Sustainable Cities and Communities (DS4SSCC) [48] project aims to create a data space for smart cities and communities focused on facilitating the implementation of the European Green Deal and the Sustainable Development Goals. This data space utilizes SIMPL [25] for secure and efficient access to and exchange of data.

The testing and experimentation facilities (TEFs) [27] will offer a combination of physical and virtual facilities, in which technology providers can get primarily technical support to test their latest AI-based software and hardware technologies in real-world environments. This will include support for full integration, testing and experimentation of latest AI-based technologies to solve issues/improve solutions in each application sector, including validation and demonstration.

The easiest way to understand the TEFs is to look at them as a sort of safety filter between emerging digital technology, AI, robotics etc., and the citizens. This filter – the four TEFs – acts to test these technologies before they reach infrastructure, society, companies, and consumers.

TEFs will serve technology providers who want to develop their AI/Robotic solution, SMEs will be able to use TEFs without paying for the services and larger companies will have to pay based on the established price list.

The TEFs are divided into four domains, each covering distinct areas: Agriculture and food, manufacturing, healthcare and finally cities and communities.

The **agrifood TEF [56]** deals with the agricultural sector and food production. This could be everything from testing a robotic tractor to artificial intelligence in crop rotation software.

The **TEF-Health [57]** deals with the healthcare sector, from next generation artificial hearts to the use of machine learning in medical imaging and diagnostics for example.

The **AI-Matters TEF [58]** will test technology within manufacturing, from robots in plastic moulding to drones in industrial warehouses.

The Testing and experimentation Facility for smart cities and communities is under the programme Digital Europe Programme (DIGITAL). The Digital Europe Programme (DIGITAL) [26] is a new EU funding programme focused on bringing digital technology to businesses, citizens, and public administrations. And in the work programme part: Digital Europe Work Programme 2021-2022 [28].

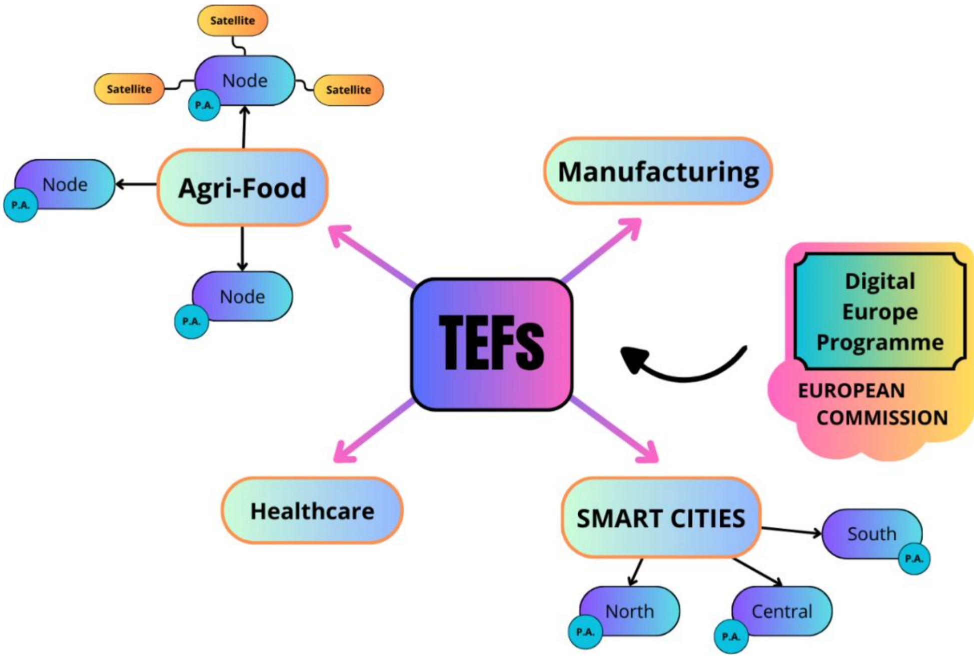


Figure 7: Source: own elaboration.

The **CitCom.ai TEF** is a bit different. With an initial focus on power, mobility and connectivity, its job is to test AI and robotics before they get into places where humans live and move around. This could be self-driving cars, but it could also be telecommunications data retrieval software. Or the robotic tractor being used in a municipal park. CitCom.ai works as a sort of cross-domain filter between technologies, infrastructures, and citizens where they live.

The unique project founded under this topic (Smart Cities and Communities) is the “European Artificial Intelligence and Robotics Testing and Experimentation Facility for Smart and Sustainable Cities and Communities, or **CitCom.AI** [29].

The EU network of permanent testing and experimentation facility (TEF) for smart cities and communities is established to act as a platform where companies can assess and experiment with different AI products. This initiative aims to support the acceleration of trustworthy AI development in Europe.

The following figure shows how CitCom.AI is divided into nodes and its focuses.

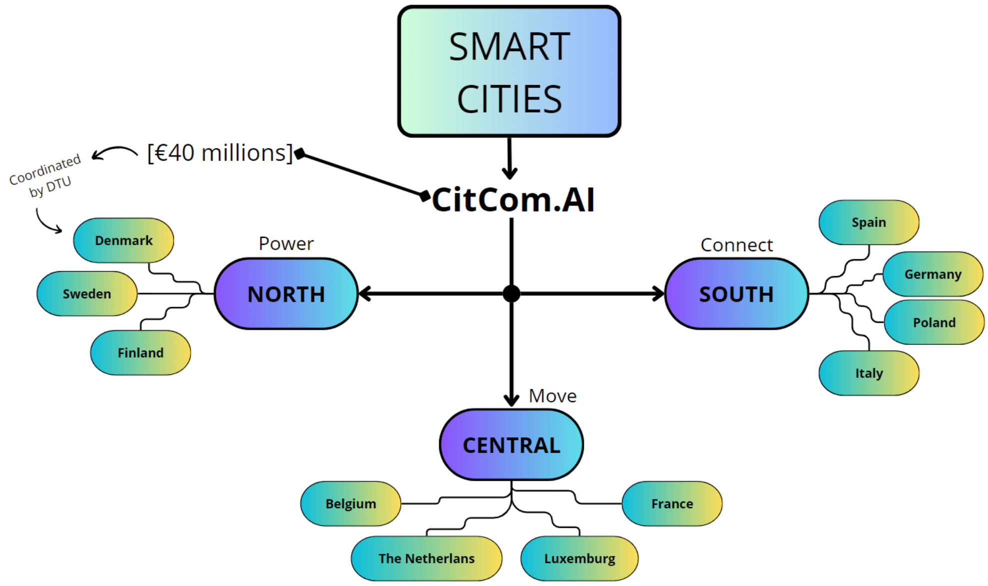


Figure 8: Source: own elaboration.

2.2.3.2 Minimal Interoperability Mechanisms (MIMs)

The **Open and Agile Smart Cities** (OASC) [30] created the MIMs, aligned with its mission to unite cities and communities with the objective to build a global market for solutions, services, and data, based on the needs of those cities and communities.

The **ITU-T Y.MIM** [1] standard is focused on the Minimal Interoperability Mechanisms for Smart and Sustainable Cities and Communities. It is an initiative of the International Telecommunication Union (ITU). Drawing on previous experience in the development and implementation of MIMs, it proposes a draft Recommendation that defines the concept, purpose, and structure of the MIMs to establish a standard that originates from a common ground.

The definition given in the official document [31] provided by OASC is: "**Minimal Interoperability Mechanisms** refer to standardized protocols and frameworks designed to enable seamless interoperability between various systems and applications. "

The primary objective of the MIMs is to establish a foundation for interoperability, overcoming the challenges posed by system heterogeneity, different data formats, and varying communication protocols.

[31] A **MIM** technically, is a description of a common set of requirements that will provide a **Minimal** but sufficient set of capabilities that a city needs to achieve a certain objective, along with a description of the **Mechanisms** by which one or more different technical solutions can address those capabilities. It also provides guidance to help gain a useful level of Interoperability between the different mechanisms that may be used to achieve that set of capabilities.

There are ten MIMs at 30th of November of 2023, and all of them are accepted in the format of 2022. However, the OASC has described MIMs 1, 2, 4 and 7 in a new format at various level of completeness.

Each one of the MIMs is focused on one domain to achieve full interoperability, security, integration, comprehensive data… The current MIMs are [32]:

MIM 1 – Context Management Information

MIM 2 – Data Models

MIM 3 – Contracts

MIM 4 – Trust

MIM 5 – Transparency

MIM 6 – Security

MIM 7 – Places (geo-temporal data)

MIM 8 – Indicators

MIM 9 – Analytics

MIM 10 - Resources

2.2.4 Design Pattern

In the term of Smart Cities, there are many systems distributed across multiple sectors, departments, domains, etc. Creating a module where these systems are implemented and connected is complex because of the following questions; how will these systems in the module interact with each other? how will they connect? How will they exchange information?

Before answering these questions, it is important to know what "module" means. Relating concepts, this "module" is mostly the same as a "Data Space" in its most basic way (connection and interaction between systems). This module must be built using a specific software architecture.

For this point, the author's foundation is grounded in and has drawn upon the insights presented within the referenced book: *Software Architecture in Practice (third edition)* [16], referenced below. Specifically, the chapters: 1.3 Architectural patterns, 4. Understanding quality attributes, 13. Architectural tactics and patterns.

The objective of this point is to discern an architectural pattern that is employed within the realm of Data Spaces, or at least, one that closely resembles it. Types of patterns can be divided according to the type of architectural elements they use. For example:

- Layered pattern, is a common module type pattern. It is a system formed by layers.
- Multi-tier pattern, Competence centre and platform are allocation patterns.
- Component and connector type patterns are:
    - Shared data (or repository) pattern
    - Client-server pattern

The most similar are the component and connector type patterns, but any of those represent the interaction part of the systems, the client-server pattern is about control access and quality of the services shared between participants, basically it is the next step after the interaction part. Delving more into the component and connector patterns, there is a Broker pattern, which is very similar to the Data Spaces basic architecture.

The following picture is intended to represent the most prototypical architecture in a single data space where there are providers and consumers.

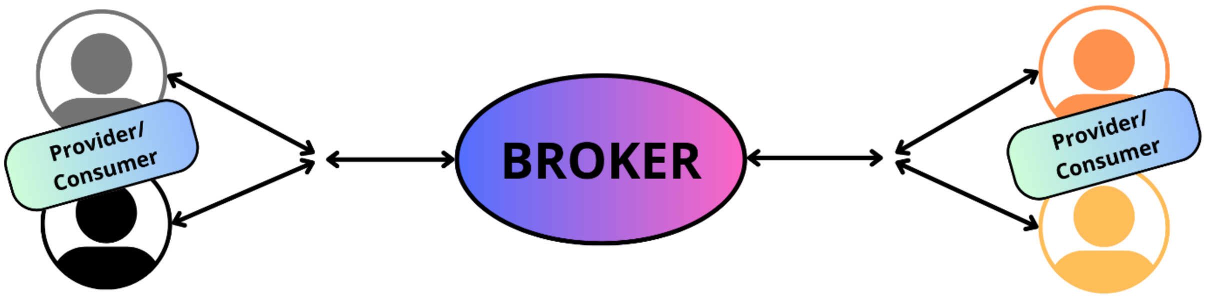


Figure 9: Source: own elaboration.

This architecture characterizes the Broker pattern, which type is component and connector pattern.

This architecture presents different users that can be providers, giving some data but at the same time they want some other data, so here there are different users that can be providers and consumers. The objective here is to enable the connection and interchange of information between users in the system but making a separation between them. In the reality users do not necessarily know the location, the nature, and other things about the other users in the system, that is why the broker pattern is the key.

The broker pattern enables decoupling the relationship between consumer and providers by inserting a broker as an intermediary. When a consumer needs some information, it queries the broker via consumer interface. Then, the broker sends a request to the provider. The information result from the request is communicated from the provider back to the broker, which then returns the result back to the client.

This way of communication the consumer can be completely ignorant of the characteristics and location of the provider. This type of communication makes easy for example when a provider leaves the system, or another one enters because the only component in the system that needs to know about the changes is the broker.

The question here is, why this pattern allows or enables the decoupling of the different participants in the system?

It must be something technical inside this architecture that allows the separation between participants, which gives the critical value and the critical peculiarities to these types of systems allowing data sharing between participants and enhancing the full value of the data.

2.2.5 Message Broker

Based on several studies about comparing message queueing systems, in this work are going to be briefly analysed some Message Broker Systems

The knowledge for the development of this point has been extracted from "A fair comparison of message queuing systems" [33], the comparation between RabbitMQ and ActiveMQ [34], and the comparison between RabbitMQ and Kafka [35].

With the objective to choose some Message Broker Systems to analyse and compare, there are some features that must have been considered:

1. Community activity: The more popularity of the products, the more contributors discovering bugs and fixing bugs are, and as a result, the system will be more stable and faster improved.
2. Communication protocol: The protocols used are usually different, but if a message broker system uses a open communication protocol, it is more accessible and understandable.
3. Consumption mode: there are two types; push and pull. Message broker systems that can have one or both.
4. Message queueing model: there are two models: point-to-point and publish-subscribe. Message broker systems that can have one or both.
5. Usability: This aspect concerns about the easiness of installation, the completeness of documentation, the functionalities of management and monitoring, etc.
6. Scalability: the capability of the message queueing system of continue scaling as demand increases.
7. Latency: measures how long it takes for the message to be transmitted between different parts.

Based on these features, for example, ones with relative popularity, with active communities, with characteristics that made the system representative in different aspects like language, functionality support, easily installation and monitoring, etc. the message queueing systems selected are the following three:

1. **Kafka:**

Kafka makes use of zero-copy to avoid repeated copy operations. Kafka makes use of zero-copy to avoid repeated copy operations.

Kafka uses batching of data to improve the throughput and reduce the overhead.

Kafka employs data compression techniques to improve the efficiency of data transmission.

2. **RabbitMQ:**

RabbitMQ supports various standardized protocols and support multiple functionalities, such as delay queueing.

RabbitMQ provides a Web UI for the system, where users can manage the queues, the exchangers, and manage them easly.

RabbitMQ provides a web tutorial in which is easily explained with a tutorial and some examples, the installation and the implementation of the tool.

RabbitMQ uses a protocol developed with Erlang, which means that support distributed computation.

3. RocketMQ

RocketMQ stores all messages in the same physical file. The centralized store design improves the efficiency when increasing number of topics. This means that when the number of topics increases will cause a decline of Kaska's performance, while RocketMQ keeps a stable throughput.

The community of RocketMQ is not very active compared to RabbitMQ and Kafka.

The monitoring of the system is limited and significantly less intuitive than the RabbitMQ one.

3 Methodology

For the practical part of this work, all the identified capabilities and their corresponding requirements under the draft recommendation of the Minimal Interoperability Mechanisms (MIM 1), which is a globally accepted and endorsed standard ITU-T Y.MIM, will be verified. The procedure includes preparing sample datasets following all relevant models and data services in MIM1, developing a certification procedure for compliance with the requirements and capabilities of MIM 1 for any architecture.

The physical implementation will take place in a pilot installation of the EU Data Space for Smart Communities for the Aarhus City Living Lab proposed by Libelium. This involves developing a data space as a test environment for information sharing. The testing architecture will be analysed for its level of compliance with the requirements of MIM 1 using the previously created procedure.

With this information, the benefits or strengths of this architecture and its limitations in the context of technical interoperability, data sharing, and context information management will be inferred and presented. Once the architecture has been analysed, efforts will be made to enhance and modify the structure and standards used to increase the level of interoperability of the architecture and ensure compliance with the requirements of MIM 1.

This new architecture will be studied to review its compliance with the requirements of MIM 1 using the developed certification procedure.

During the evaluation, a brief comparison of the implementation will be conducted, and the usability of the compliance certification procedure for MIM 1 requirements will be assessed. This assessment will include analysing whether it was possible to evaluate the architectures based on the certification results. Additionally, an estimation of the economic impact resulting from the implementation will be presented.

3.1 MIM 1 Analysis

At this stage, the objective is to develop a certification procedure for compliance with the requirements and capabilities of MIM 1. This procedure will enable the analysis of different information sharing architectures, determining whether they comply with the requirements and capabilities of Context Information Management MIM 1 based on certain characteristics of those architectures.

From the final version 6.0 of the MIMs: (MIM 1) [36]:

Capabilities	Requirements
C1: Applications are able to access data from different sources (cities, communities, vertical solutions).	**R1**: A uniform interface shall be used; the context management API
	R2: Information from all sources shall use the same concepts, so called data information models
C2: Applications are able to use both current and historical data, use geospatial querying and be automatically updated when the source data changes.	**R3**: The uniform interface shall support retrieval of current data
	R4: The uniform interface shall support retrieval of historical data
	R5: The uniform interface shall support geospatial querying
	R6: The uniform interface shall support subscription to changes
C3: Applications can discover and retrieve data relevant to their context from a variety of sources, including from within larger data sets and with default limits and page sizes.	**R7**: Relevant data sources to any required context (at least location and time period) shall be discoverable and retrievable according to their context
	R8: Specific subsets of data relevant to the context shall be retrievable from within larger data sets and with default limits and page sizes

C1: Applications are able to access data from different sources (cities, communities, vertical solutions).

R1: **A uniform interface shall be used; the context management API.**

First, what is an API?

From Wikipedia [37]: "An application programming interface (API) is a way for two or more computer programs to communicate with each other."

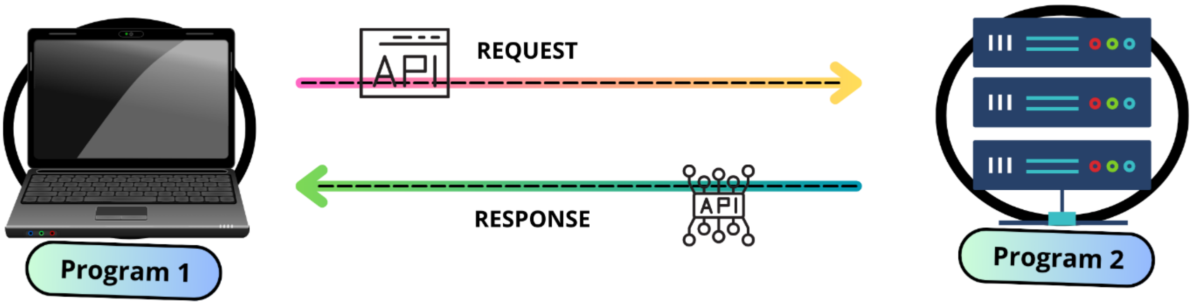


Figure 10: Source: own elaboration.

This image shows two programmes using the same way of communication, so one understands the other.

The meaning of this requirement is that the interactions between different software components shall use a uniform interface. Using a uniform interface, applications will be able to interact with other software components.

For this task, it is necessary to identify whether a system, or a set of applications, employs a uniform interface to evaluate compliance with requirement 1. To be able to identify this, it is necessary to understand the concept of a uniform interface in the context of software components.

The term "uniform interface" is a fundamental principle in the design of software architectures. The use of a uniform interface implies using a standardized set of operations for interaction among the different components of a system.

There are some limitations that a software architecture must be compliant to use a uniform interface.

- Resource identification: resources are identified by URIs[38] (Uniform Resource Identifier). According to the definition given by Roy T. Fielding an URI is an identifier consisting of a sequence of characters enabling uniform identification of resources via separately defined set of naming schemes.
- Resource manipulation [39]: client or consumer have the ability to manipulate the representation of the resources. When client request for some resource, server will respond with the representation of that resource. The format is chosen by the client, it could be JSON, XML, HTML, Plan Text, images, etc. For example, if the client wants the message in one format, that should be specified in the "Content Negotiation" as:
  "application/json" or "application/xml".

- Self-descriptive messages [40]: each message must contain enough information to be understood by the receiver without more information.
- Hypermedia as the Engine of Application State (HATEOAS). it just means that navigating through the application using links. For example, representing resources with links.

To be compliant with this requirement the architecture must use a uniform interface characterized with these limitations.

C1: Applications are able to access data from different sources (cities, communities, vertical solutions).

The following image shows an example of an architecture that contains different sources within a city, connected with one application, which intends to provide different uses like, better rated hotels in a specific area, empty parking spots in the city centre, transportation maps, location of buses and metro… All these are different sources and each one can use its own language to keep and share the data.

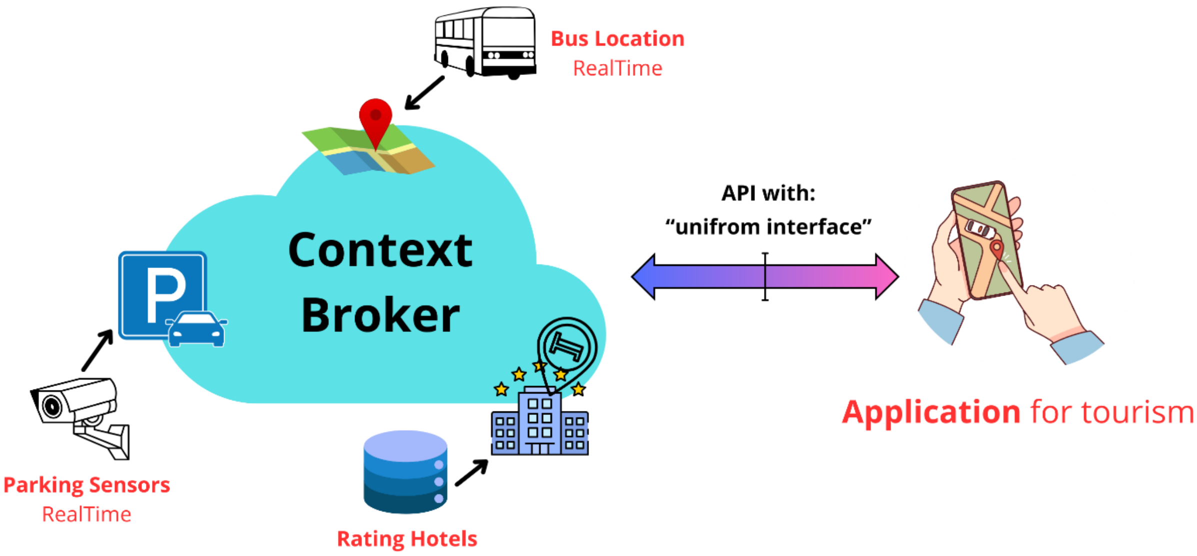


Figure 11: Source: own elaboration.

Applications cannot read and understand all the formats provided by the different sources, so this Context Management API ensures that the application always receives the data and information in the same format.

To sum up, any application that uses an API with uniform interface, could access to the data and information and is able to read the context information correctly.

This enables other applications to be able to access different context information from different sources in the system just using a uniform interface. The importance here is that data could be used not only by one application, but by all the applications that need it, harnessing the full value of the data.

There is a table of mechanisms which explains the technical specifications that address the requirements. For MIM 1, there is not only one mechanism to cover the requirements, but there are also two sets of technical specifications (mechanisms) that can address the requirements: FIWARE NGSI-LD [40] and the Open Geospatial Consortium Standards [41] approach.

It is important to point out that there are no standards adopted by every use case. Each city has its own decision process, its preferences, some quirks, thus, there are likely to be other technical solutions used.

The objectives of MIM 1 are: bring together information from different sources (using a uniform interface, Requirement 1) and the other objective is to enable comprehensive and integrated use, sharing and management of data. In other words, it is not only to be able to read the data, but to **understand** it correctly.

R2: **Information from all sources shall use the same concepts, so called data information models.**

First, what is a data information model?

Data modelling is defined by IBM [42] in its webpage as "Data modelling is the process of creating a visual representation of either a whole information system or parts of it to communicate connections between data points and structures."

The FIWARE NGSI API defines its own data model for context information, it is contained in the Context Information Management API (NGSI-LD), and it is based on an **information model**. The **information model** NGSI-LD specifies the data representation mechanisms that NGSI API shall use.

This model NGSI-LD has high complexity, so here is going to be presented the NGSI-v2 model, this way allows to focus more on the importance of the existence of the same data information models in the system to achieve the minimal level of interoperability.

The main elements in the NGSI-v2 information model are:

- Context Entities: are simple entities that represent a "thing". Following the standard definition in the document ITU-T Y.2060 [43], "With regard to the Internet of Things, a thing is an object of the physical world or the information world that is capable of being integrated into communication networks. For example, a sensor, bins, a house, an issue in a ticketing system…
- Properties or Attributes: are characteristics of context entities. For example, for an entity -*car*-, the speed of the car could be an attribute -*current_speed*-.
- Metadata: it is contained as an optional part in the attribute values, and it contains the metadata.

Note that in the NGSI-LD model metadata is implemented in the Properties or Attributes element and appears other main element named -Relationships- which relate entities.

The following image shows the information model of NGSI, in the left the NGSI-v2 information model and in the right NGSI-LD, which allows to compare the structure.

The figure 12 and figure 13 and the insights are based on the documents of the specifications of NGSI-v2 [44] and NGSI-LD [40].

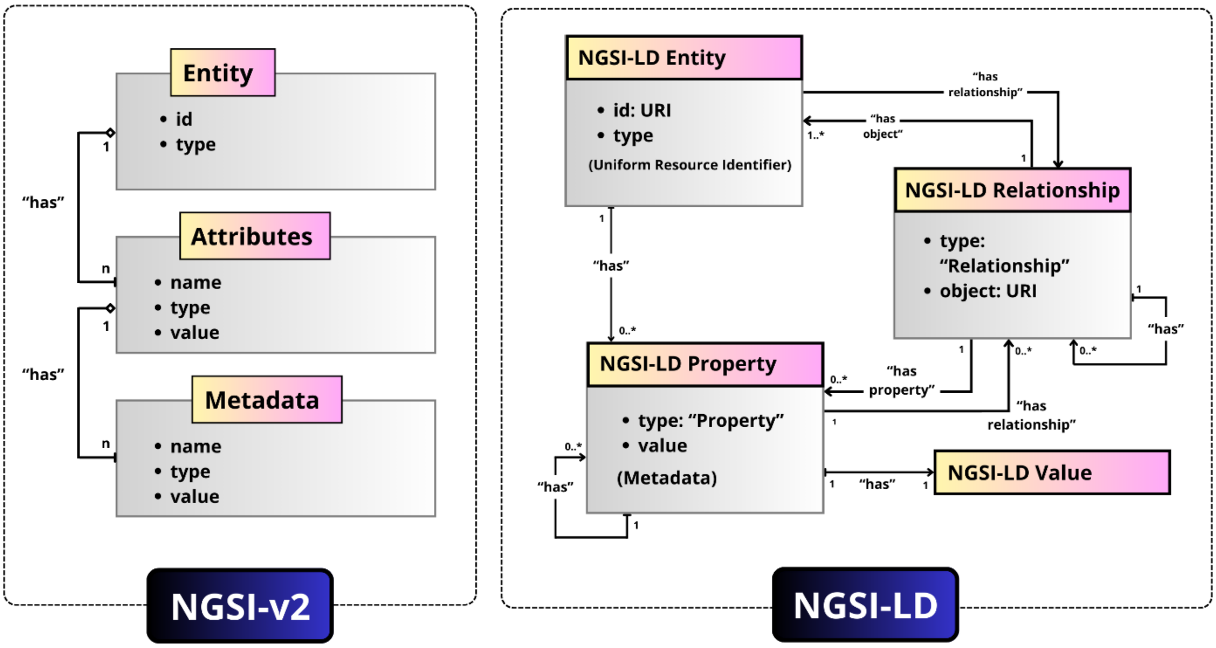


Figure 12: Source: own elaboration.

Now, there are two examples, one of NGSI-v2 and other of NGSI-LD in application/json type. This is to visualize how the data information models are seen in practice. Pragmatically, these are a response bodies containing a list of entities from a specific query made by the consumer.

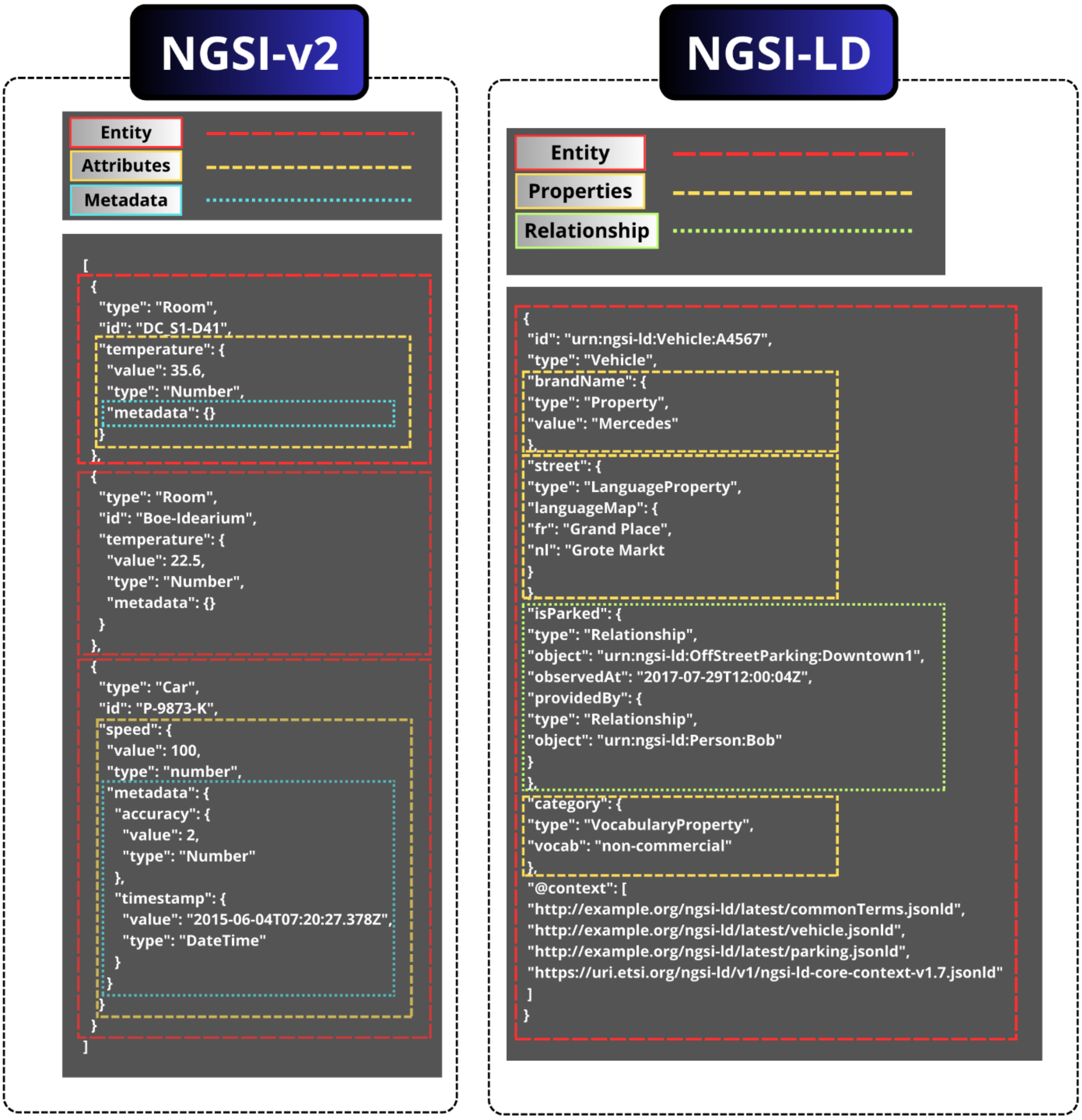


Figure 13: Source: own elaboration.

Summarizing, if one participant is defining one entity (using the standard) with “type” and “id”, other participants should know that there is a “type” which contains the name of the entity and there is an “id” which contains the set of characters that identify that exact entity.

R3: The uniform interface shall support retrieval of current data.

The consumer must be able to retrieve real-time data from different data sources through a single operation. Providers send real-time data, current data, and the consumer can make a query or operation and receive that current data through the uniform interface. This requirement, through a generic RESTful API that complies with the specifications of a uniform interface by definition, is exemplified as follows:

The provider has sent the following information: a thermometer showing a temperature value of 20.

To retrieve the current temperature data from the thermometer, the request should be made as follows:

Figure 14: Source: own elaboration.

R4: The uniform interface shall support retrieval of historical data.

The consumer must be capable of retrieving historical data from different data sources through a single operation. Providers send historical data, and the consumer can make a query or operation to receive that historical data through the uniform interface. This requirement, through a generic RESTful API that complies with the specifications of a uniform interface by definition, is exemplified as follows:

The provider has been sending data, and the architecture has been storing them: a thermometer recorded a temperature of 20 on day 1, 22 on day 2, and 26 on day 3.

To retrieve the historical temperature data from the thermometer, the request should be made as follows:

Figure 15: Source: own elaboration.

The response in JSON format should be: (XML format is also shown to further generalize the definition, but both are standard formats and can be easily transformed from one to the other. JSON is chosen for its better readability for the human eye).

Figure 16: Source: own elaboration.

R5: The uniform interface shall support geospatial querying.

The consumer must be able to retrieve data from a specific geographic area through a single operation. For this scenario to be possible, the data must include geographic information; otherwise, access to such information will not be available. Assuming that the data contains geographic information, the consumer can make a query or operation and receive certain current data through the uniform interface. This requirement, through a generic RESTful API that complies with the specifications of a uniform interface by definition, is exemplified as follows:

Figure 17: Source: own elaboration.

In this example, the consumer retrieves temperature data from the location with latitude 40 and longitude 70 as the central point and with a radius of 100 meters. All data sent by thermometers located within that perimeter will be retrieved.

It is important to note that geospatial location can have various ways of representing locations, and the query must use the same location language.

R6: The uniform interface shall support subscription to changes.

To subscribe to data changes using a RESTful API, a mechanism is used that involves sending an HTTP POST request to create a subscription along with a URL to which the API will send notifications about the changes the consumer has subscribed to. Illustrated:

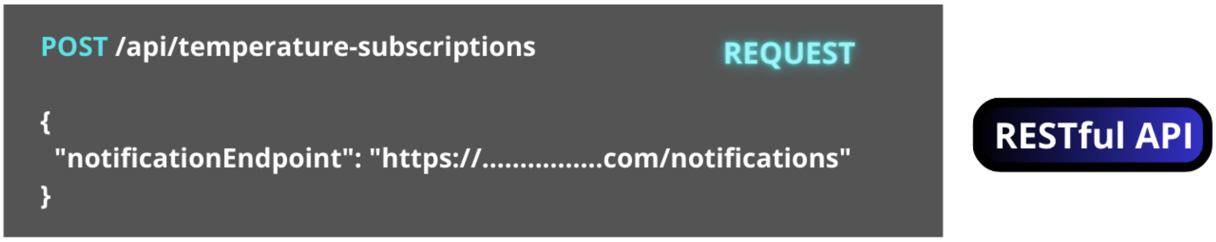


Figure 18: Source: own elaboration.

R7: Relevant data sources to any required context (at least location and time period) shall be discoverable and retrievable according to their context.

For this purpose, using a RESTful API, the requirement limits at least location and time period. To achieve this, the query parameters 'lat' for latitude, 'lon' for longitude are used for the specific location, and 'startDate' and 'endDate' for the time period. It is worth noting that this is using a generic RESTful API, and not all APIs handle queries in the same manner.

If the architecture returns data within these parameters upon making the request with the specified parameters, it fulfills the requirement.

R8: Specific subsets of data relevant to the context shall be retrievable from within larger data sets and with default limits and page sizes.

Prior to fulfilling this requirement, the architecture must support historical data retrieval (R4). For this requirement, it should be possible to retrieve smaller sets of data from the entire dataset (historical data) and specifically related to a particular context. Furthermore, specific limits must be established for this subset of data.

To confirm whether this requirement is met in the architecture, simply make a request, and append the following at the end:

```
”page=X & limit=XX”
```

And check if it returns the data.

3.2 Tool Message Broker

The object of this point is to choose one Message Broker system to develop the test environment architecture for the Aarhus Living Lab.

To achieve this, in the context of this thesis, is appropriate to lay out which are the necessities of the message broker system.

This work aims to analyse the level of compliance of the requirements of MIM 1 in the architecture simulated environment, that shows the process of data sharing aiming to achieve different levels of interoperability. This must be done and simulated in a very simple way due to the difficulty of the process and the need to adapt the work to a final degree project challenge.

For these reasons, the message broker system selected must be easily understandable, installable, and implementable. Should have and interface to see how is working and the parts of the message broker. It also should allow to see what the functionalities of the broker and the queues are and easily understandable, in other words, it must be easily and highly usable.

Seeing the technical characteristics of the three options available at the “2.2.4 Message Broker” in this work. The one that is more suited is the RabbitMQ message broker system:

RabbitMQ provides a tutorial, is easy to install and to implement Use Cases, and the only application needed is Docker. Docker has an environment, and the user does not need to install all the pack from RabbitMQ because Docker has installed already. It also provides a Web UI for the system, where users can manage the queues, the exchangers, and manage them easily, which is very important from the point of view of this work to see how it works in the interior, the current state of the queues and the exchanges. Another point is that, as student I have knowledge of programming using Python, and it is possible to implement RabbitMQ using Visual Studio Code for example, which is going to be used in this work.

The version of RabbitMQ that is going to be used is RabbitMQ 3.12.10 release (21 November 2023).

There is a hole tutorial developed by the author of this work to implement RabbitMQ in Visual Studio Code, from zero and running it, using Docker (Anex 2).

4 Implementation

4.1 Aarhus Living Lab

4.1.1 Current Architecture (testing pilot)

This is the current architecture of the Aarhus Living Lab.

This architecture has been developed as a testing environment for data sharing at the Technical University of Denmark. It enables the study of which requirements of MIM 1 (Context Information Management) it fulfills and the possibilities and opportunities derived from meeting certain requirements, as well as the limitations the architecture faces due to non-compliance with other specific requirements.

All of this is clearly focused on the opportunities and limitations of the architecture in the context of technical, semantic, and syntactic interoperability.

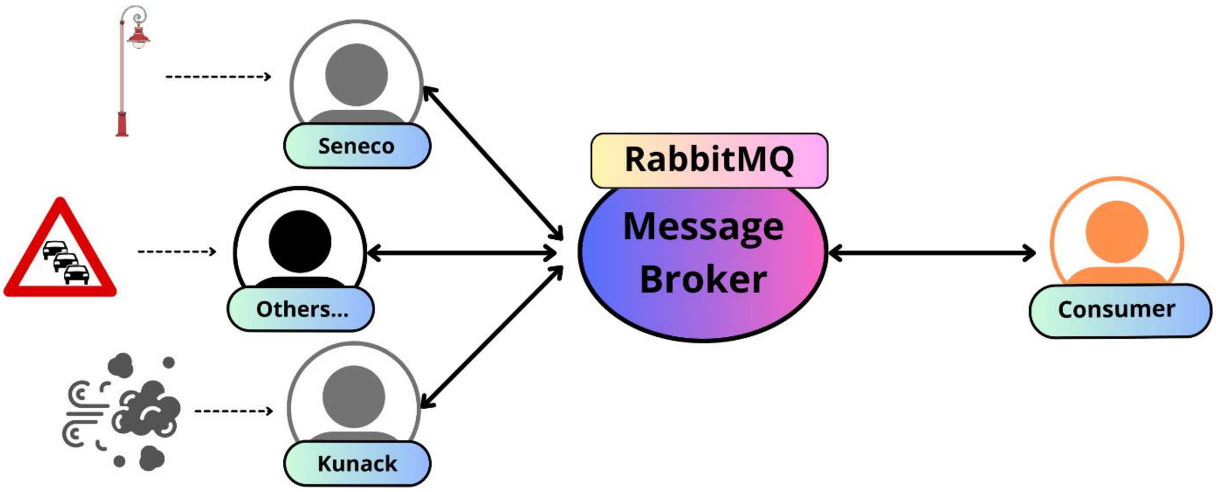


Figure 19: Source: own elaboration.

How it works?

This architecture has been created using the RabbitMQ tool and employs the Provider and Consumer design pattern. In this case, there are multiple providers, which are various data sources such as Seneco, provider of streetlight data, and Kunack, provider of air quality data, and one or multiple consumers.

Characteristics of this architecture:

- All data is sent as messages, in the form of a text string.
- All providers send their data (messages) to an Exchange, and each message is associated with a routing_key and a queue. Therefore, each provider that sends a certain data must associate it with a routing_key and a queue. This is visually better understood in Figure XX.
- This Exchange (in this architecture, there is only one) is responsible for sending the data to the queues to which they are associated.

- Any consumer that wants to access specific data must connect to the routing_key of the data it wants to receive.

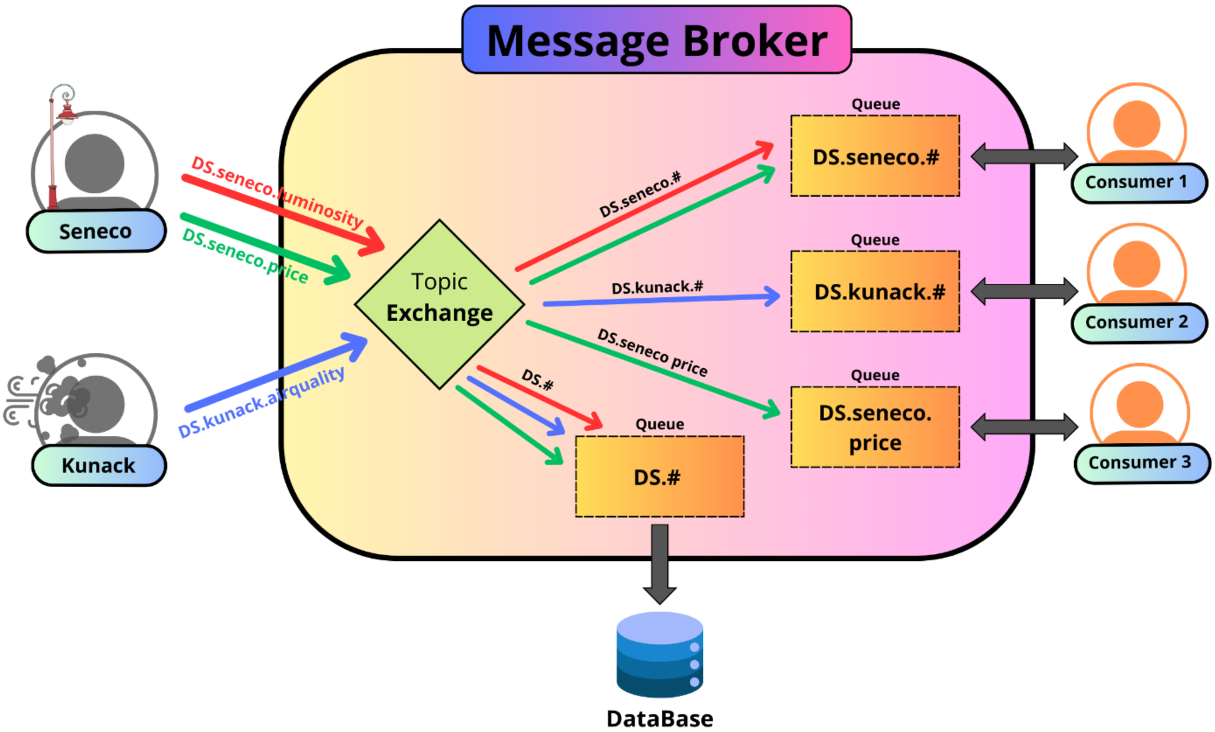


Figure 20: Source: own elaboration.

Seneco sends two types of data: luminosity data with the routing_key "DS.seneco.luminosity" and price data with the routing_key "DS.seneco.price". Kunack only sends air quality data with the routing_key "DS.kunack.airquality".

In this architecture, there are four consumers:

- Consumer 1: retrieves all data sent by Seneco via the routing_key "DS.seneco.#", meaning all routing_keys that start with DS.seneco, regardless of what follows. This consumer collects data for both luminosity and price.
- Consumer 2: retrieves all data sent by Kunack via the routing_key "DS.kunack.#". Since Kunack only sends air quality data, this consumer collects only air quality data.
- Consumer 3: retrieves only price data from Seneco via the routing_key "DS.seneco.price". These data include daily prices and total prices per data transmission.
- Database: retrieves all data from all providers within the DataSpace via the routing_key "DS.#" and stores them in a single location.

Below is an example of how information sharing is performed in this system.

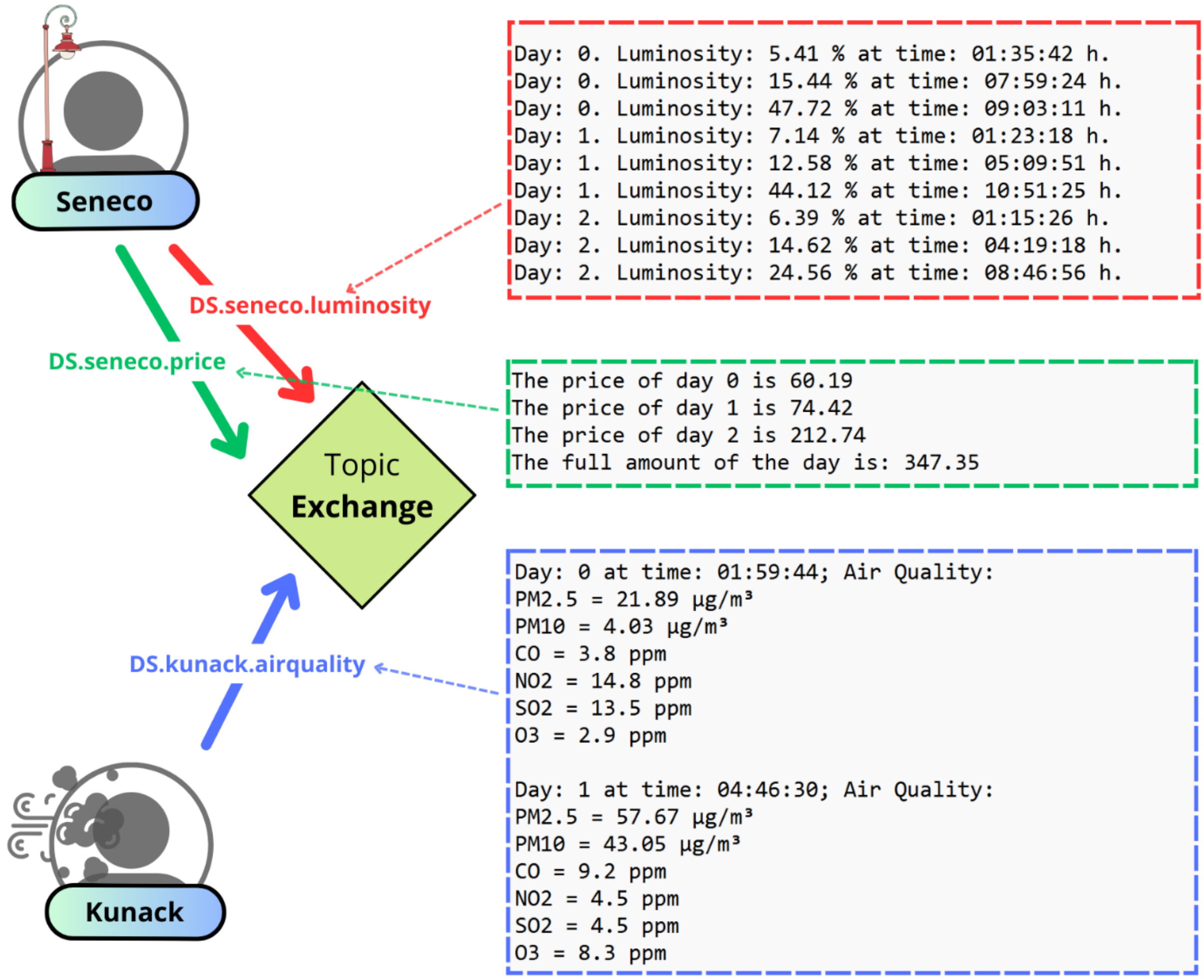


Figure 21: Source: own elaboration.

The providers send data in the form of text messages, linked to a routing_key as explained earlier. Consumers access this data through the routing_key, receive the data, and store it in text files. The data is visualized as follows.

Each consumer has a text file where the information is stored. Before any provider sends data, the text files of the consumers are as follows.

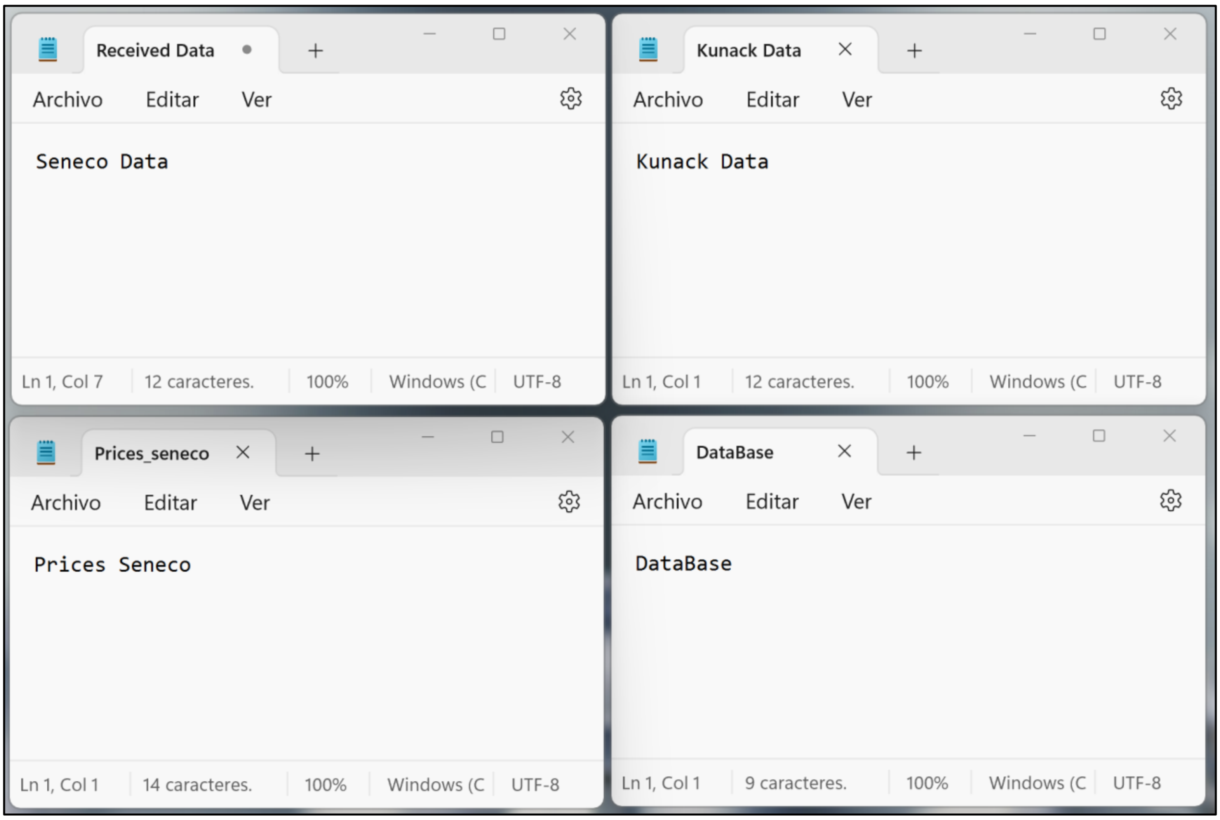


Figure 22: Source: own elaboration.

Once the providers send the data, each consumer accesses the data it wants.

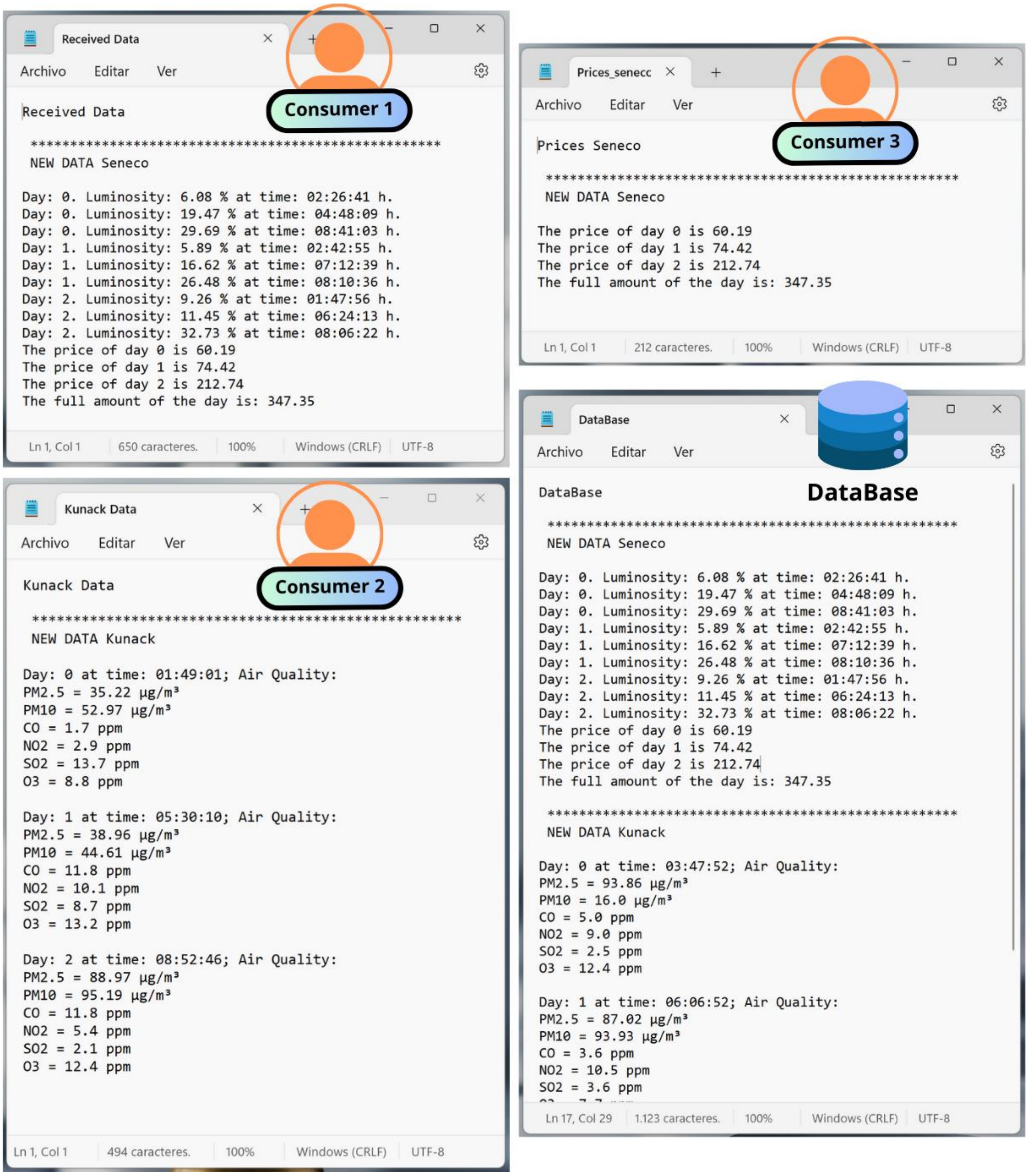


Figure 23: Source: own elaboration.

Essentially, as explained before, each consumer retrieves the data to which it has subscribed using the routing_key.

What is the level of compliance of this architecture with the MIM 1 requirements?

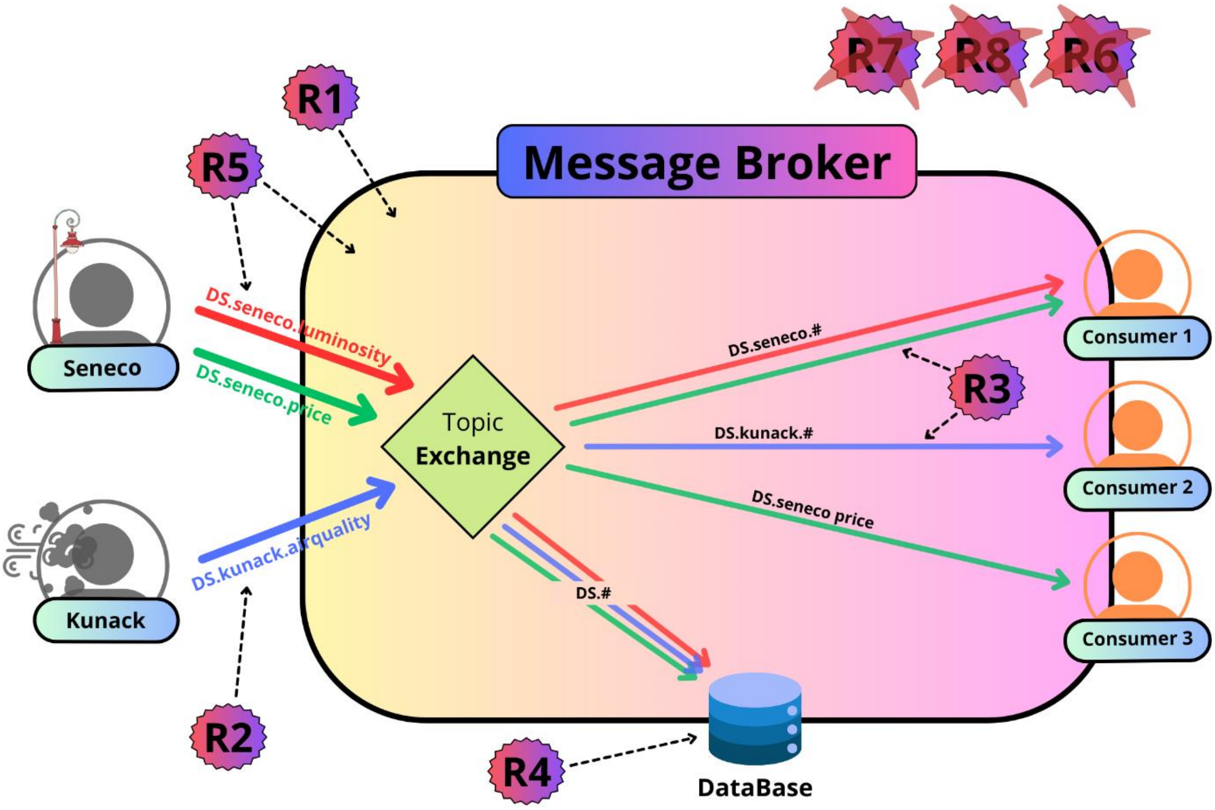


Figure 24: Source: own elaboration.

R1: A uniform interface shall be used; the context management API.

Applying the definition of a uniform interface explained earlier in section 3.1 MIM1 Analysis, "The use of a uniform interface implies using a standardized set of operations for interaction among the different components of a system."

In RabbitMQ, specifically in this architecture, any provider must:

1. Create a channel through which the information will pass.
2. Declare an Exchange on that channel, which will be responsible for sending the information to the appropriate destination.
3. Publish a message through that Exchange with a certain routing_key.

It is visualized as follows:

```
#Provider
channel = connection.channel()
channel.exchange_declare(exchange='AarhusLL', exchange_type=ET.topic)
channel.basic_publish(exchange='AarhusLL', routing_key='DS.seneco', body="Data")
```

Figure 25: Source: own elaboration.

Any consumer wishing to interact with the different participants must:

1. Declare a queue within the channel.

2. Bind the queue to the Exchange and to a routing_key. In this way, all information coming from the Exchange with the desired routing_key will be received and stored in that queue.

3. Consume the message from a queue.

It is visualized as follows:

```
#Consumer
channel.queue_declare(queue='SenecoTXT', exclusive=False)
channel.queue_bind(exchange='AarhusLL', queue='SenecoTXT', routing_key= 'DS.seneco.#')
channel.basic_consume(queue='SenecoTXT', auto_ack=True, on_message_callback="Data Reveived")
```

Figure 26: Source: own elaboration.

Therefore, a standardized set of operations is used for the interaction of the different participants in the system, fulfilling the definition of a uniform interface.

However, it does not meet the necessary characteristics explained also in section 3.1 MIM 1 Analyse. It does not use a uniform resource identification (URI); RabbitMQ, at least at this level, does not allow the identification of resources within the data. In fact, the data are text messages. This architecture does allow the modification of resources; the data are sent as text and received in the same way, so the consumer can store the text data in any file format, such as a text document, JSON, or XML. In fact, in this architecture, the data are saved in a text file. The messages themselves are descriptive; they contain the necessary information with the routing_key and the queue or the Exchange. The last characteristic, HATEOAS, is not fulfilled; within the data, it is not possible to navigate from one resource to another using links.

Therefore, requirement 1 is not fully met.

R2: Information from all sources shall use the same concepts, so called data information models.

The data structure is not the same for all; each provider sends data in the way they want, without a structure like the one explained in section 3.1 MIM 1 Analysis of entity, attribute, and metadata.

There is no standardized data information model; instead, each provider uses a different one. Additionally, each provider uses its own names to define characteristics within the data, such as price and luminosity, which another data source might call cost and illumination. Therefore, this requirement is not fulfilled in the architecture.

Below is a reminder of the capabilities and requirements table of MIM 1: Context Information Management:

Capabilities	**Requirements**
	R1: A uniform interface shall be used; the context management API

C1: Applications are able to access data from different sources (cities, communities, vertical solutions).	**R2**: Information from all sources shall use the same concepts, so called data information models

Requirement 1 is partially fulfilled, as it uses a uniform interface for the entire architecture, albeit without using URIs or links in the data. However, Requirement 2 is not met. Nonetheless, Capability 1: "Applications are able to access data from different sources (cities, communities, vertical solutions)" is indeed fulfilled in the architecture. Any consumer can access data from different providers or sources, thus decoupling the architecture and making providers and consumers independent.

R3: The uniform interface shall support retrieval of current data.

In this architecture, consumers may be connected or disconnected.

If a consumer is connected, it means it will receive all real-time data sent through the routing_keys to which it is subscribed. Therefore, it is indeed true that the interface of this architecture allows the retrieval of real-time data (current data).

R4: The uniform interface shall support retrieval of historical data.

In this architecture, historical data is accessible in two ways:

1. Manual retrieval: All data from the providers within the Data Space is received by the "Database" consumer and stored together in a single file. This file contains all historical data of the Data Space.

2. Through the uniform interface: The architecture itself operates in a way that if a consumer is not connected to the architecture, all data sent to that consumer during the period of disconnection is stored in the queue associated with that consumer. This way, historical data is stored in the queue while the consumer is disconnected, and when it reconnects, it receives all pending historical data.

Therefore, if a consumer wants to retrieve real-time data (it must be connected) and later wants to access this data, while being connected, this data has not been stored in the queue because it has already been received by the consumer. Hence, to access this data, the consumer must access the Database.

R5: The uniform interface shall support geospatial querying.

Whenever the data contains geospatial information, it can be accessed through querying via the uniform interface. For this to happen, the provider must send the geospatial data connected to a specific routing_key so that the consumer can query that particular routing_key and access the geospatial data.

R6: The uniform interface shall support subscription to changes.

This architecture does not allow subscription to changes. Simply put, data that has already been received cannot be modified. The data consists of text messages without a data information model, so there is no differentiation between entities, properties, or values; it is simply understood as text. Therefore, it is not possible to modify a value or a property.

R7: Relevant data sources to any required context (at least location and time period) shall be discoverable and retrievable according to their context.

This architecture does not store context information from the data sources. The closest approximation to fulfilling this requirement is the structure of the routing_key. For example, the routing_keys for Seneco are "DS.seneco.luminosity" and "DS.seneco.price", and the routing_key connected to consumer 1 is "DS.seneco.#", thus retrieving all data originating from DataSpace and Seneco. However, it does not gather context information from Seneco; instead, it retrieves the data it sends. Context information would include, for example:

```
Context Data: {
    Day {}
    Luminosity {"%"}
    Time {}
    Daily Price {}
    Price {}
}
```

It does not allow access to this data because it is not stored or managed anywhere in the architecture; the broker does not have that capability.

R8: Specific subsets of data relevant to the context shall be retrievable from within larger data sets and with default limits and page sizes.

As explained in requirement 6, the data consists of text messages without a data information model. Therefore, there is no differentiation between entities, properties, or values; it is simply understood as text. Because of this, the architecture interface does not allow access to parts of the data.

4.1.2 Proposed Architecture

To address the interoperability limitations of the previous architecture, a new architecture is proposed that uses FIWARE, specifically the NGSI-v2 or NGSI-LD standard.

Firstly, the Message Broker RabbitMQ will be replaced by the Scorpio Context Broker or the Orion Context Broker, both of which are FIWARE Context Brokers. The main difference between them is that the Scorpio Context Broker supports more complex queries than the Orion Context Broker. By "more complex," it refers to queries related to a specific context, small data packets within longer datasets, related to requirements 7 and 8.

In addition to the FIWARE Context Broker, QuantumLeap will be added. QuantumLeap is a component of FIWARE that is connected to the Context Broker. Its most important functions include facilitating queries on stored historical data and enabling subscription to context changes, even storing those changes in a time-series database.

PostgreSQL, a relational database management system (RDBMS), is connected to QuantumLeap to serve as its data storage backend and support for handling time-series data. Another important feature is scalability; this architecture is designed to be scalable so that more data sources can join the Data Space. PostgreSQL is designed to handle very heavy workloads with large amounts of data using recursive queries and window functions.

In this architecture, the consumer utilizes a RESTful API for communication with the Data Layer:

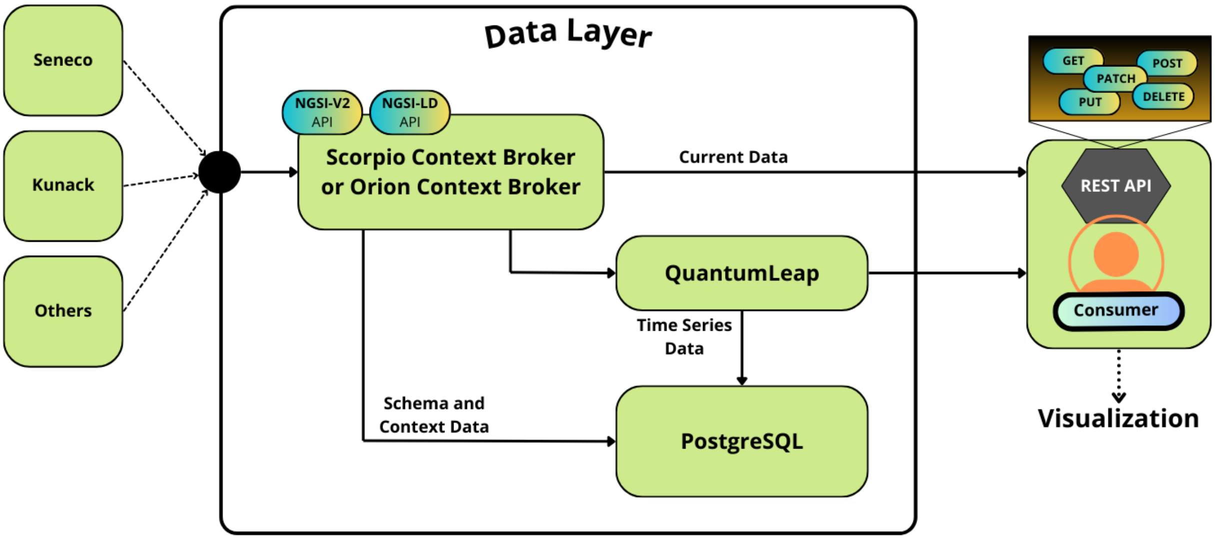


Figure 27: Source: own elaboration.

What is the level of compliance of this architecture with the MIM 1 requirements?

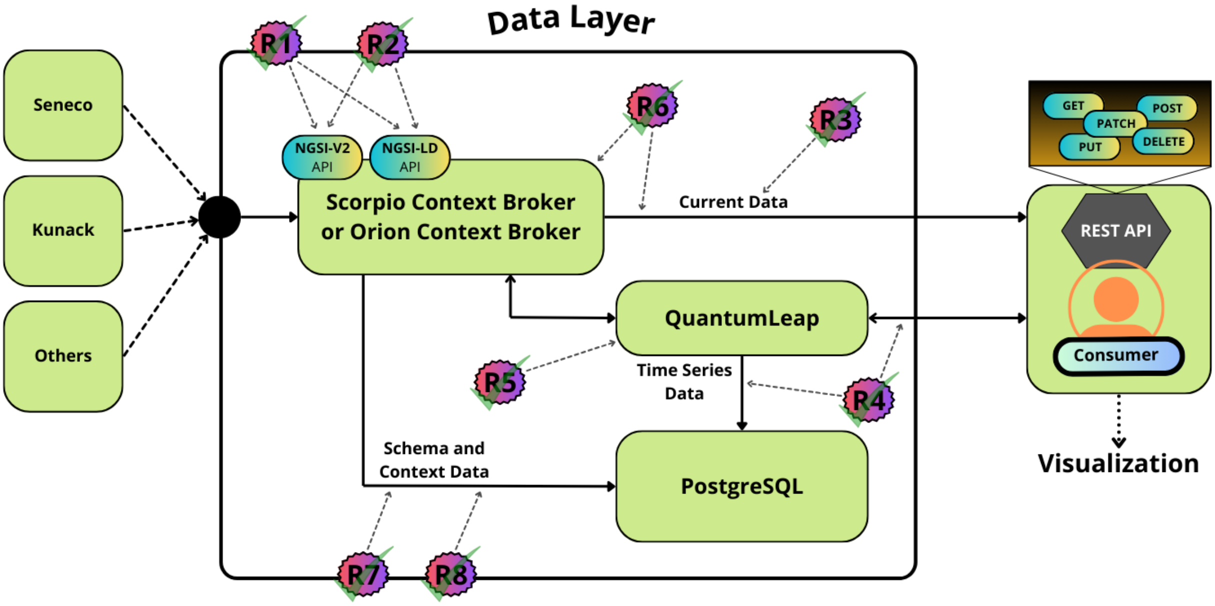


Figure 28: Source: own elaboration.

The mechanism that complies with the requirements of the MIM 1 is the NGSI-LD or NGSI-v2.

R1: A uniform interface shall be used; the context management API.

Scorpio Context Broker uses the NGSI-LD API. It uses a standardized set of operations for interaction among the different components of a system. Compliant with the requirements of a uniform interface explained at 3.1 MIM 1 Analyse, it uses methods HTTP, with self-descriptive messages, using URIs to identify entities and using links to give “@context” to the entities. The consumer uses RESTful API, which complies with the uniform interface constraint. The Orion Context Broker is another option, it is not as powerful or scalable for complex datasets, but it uses the NGSI-v2 API, the Context Management API, providing a uniform interface.

The main difference between both Context Brokers is that the Orion Context Broker, with the NGSI-v2 standard, does not utilize URIs or HETAOAS (links between data), resulting in limitations in organizing and accessing them. However, this makes the data simpler to handle within its constraints.

R2: Information from all sources shall use the same concepts, so called data information models.

Scorpio uses the NGSI-LD, and Orion Context Broker uses NGSI-v2, both use the information model explained at 3.1 MIM 1 Analyse to model entities with their properties and relationships. It allows finding information by discovering entities, following relationships and filtering according to properties, relationships, and related meta-information, through this approach, the integration of data and concurrent analysis is streamlined.

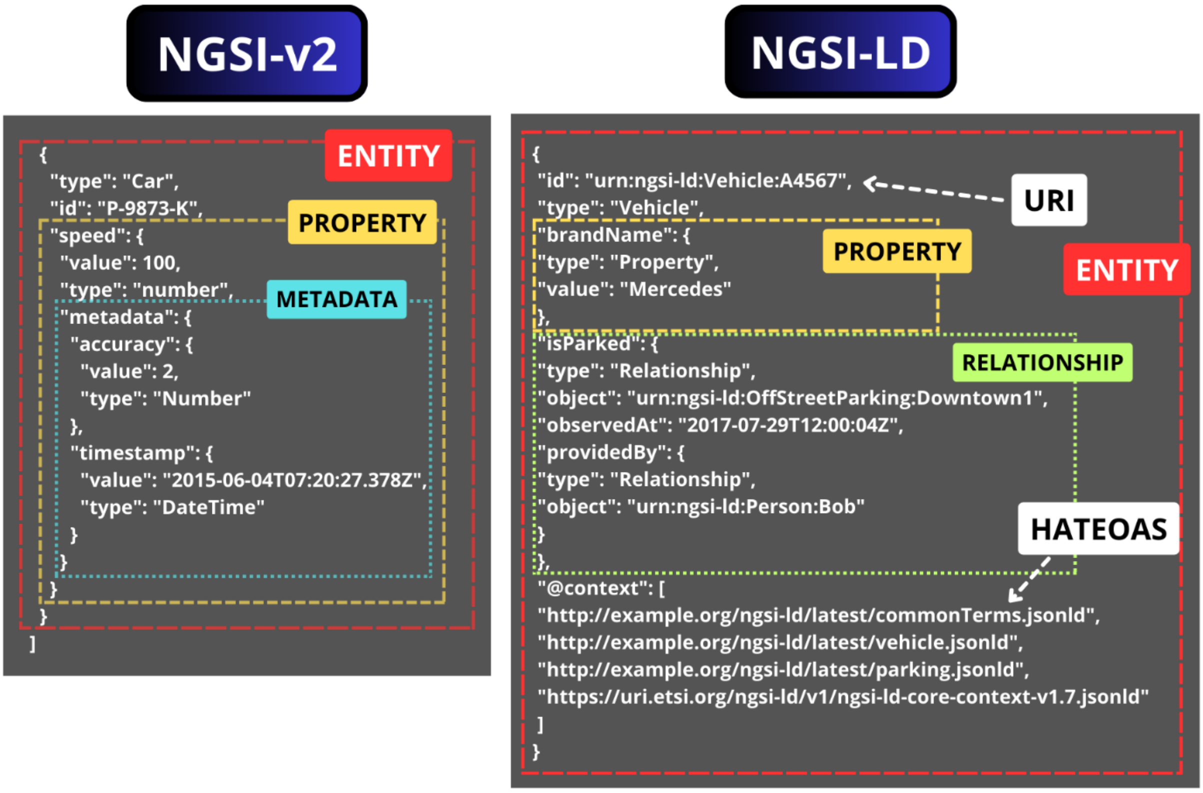


Figure 29: Source: own elaboration.

R3: The uniform interface shall support retrieval of current data.

The Context Brokers support real time data retrieval through the APIs (NGSI-v2 and NGSI-LD). The consumer can make a request using the HTTP method using the restful API to get certain data. For example:

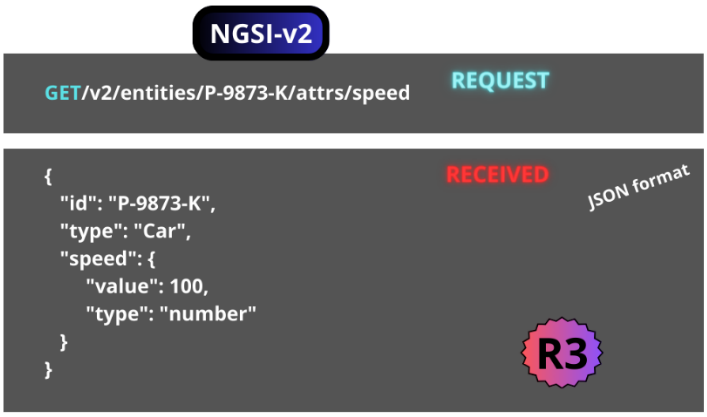


Figure 30: Source: own elaboration.

R4: The uniform interface shall support retrieval of historical data.

QuantumLeap subscribes to data changes managed by the Scorpio Context Broker. QuantumLeap stores time series data in PostgreSQL, allowing detailed analysis and reporting on historical data. This way, QuantumLeap, integrated with the Context Broker, allows the query and retrieval of historical data stored in PostgreSQL.

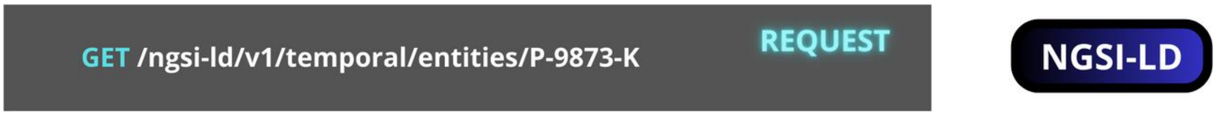


Figure 31: Source: own elaboration.

R5: The uniform interface shall support geospatial querying.

QuantumLeap is a REST service for storing, querying, and retrieving NGSI v2 and NGSI-LD spatial-temporal data. The consumer only has to use the REST API to query geospatial data to QuantumLeap. QuantumLeap, with the direct connection to the Context Broker is able to retrieve geospatial data from the stored data in PostgreSQL.

R6: The uniform interface shall support subscription to changes.

The uniform interface supports it. The API call to use is: “POST /subscriptions/”, through the REST API used by the consumer, directly to the Scorpio Context Broker, this is very similar to the RabbitMQ one, but using a Restful API using HTTP methods.

R7: Relevant data sources to any required context (at least location and time period) shall be discoverable and retrievable according to their context.

The API call that enables the discovery and retrieval of data sources relevant to any context is:

“GET /CSourceRegistrations”

On the other hand, context source registrations describe where and how additional context data can be obtained, which is a direct function of the Context Broker. Following the Figure, the Scorpio Context Broker stores that information in PostgreSQL. So, the flow of the request is first to the Scorpio Context Broker, and it gets the context information from the PostgreSQL.

R8: Specific subsets of data relevant to the context shall be retrievable from within larger data sets and with default limits and page sizes.

The Scorpio Context Broker can handle a query for a specific context, along with a specified limit (defined within the query itself) and returning a response with the information.

R7 and R8: the Scorpio Context Broker is necessary in this case to be compliant with these requirements due to the complexity of the requests according to a specific context and specific subsets of data within a larger dataset.

5 Evaluation

5.1 Evaluation of the results & Economic Impact

After analysing the first architecture in terms of its compliance with the MIM 1 requirements, the following information can be extracted:

Benefits:

1. Decoupling Providers and Consumers: This architecture allows for the decoupling of providers from consumers, which is beneficial for two main reasons:

If a provider makes any changes, such as leaving the architecture, it does not affect the structure. Connections are not established individually between providers and consumers but through a Message Broker. If a provider leaves the architecture, but another one providing the same type of data joins and connects to the Message Broker with the same routing_key, the consumers receiving that data do not know if it is from the original provider or a new one.

Scalability: The disconnection between providers and consumers allows for new providers to join the architecture without requiring any action from consumers. Otherwise, each new provider would need to establish connections with every consumer, leading to unsustainable individual provider-consumer connections as the architecture grows.

Message Sharing: Message sharing in this architecture is associated with routing keys, allowing consumers to subscribe to specific routes and receive only the desired data.

Data Storage: This architecture enables data storage in queues when they are not consumed. This allows consumers to access real-time data while connected and store historical data in a personal queue while disconnected.

Limitations:

Unique Interface: The uniform interface used is specific to the RabbitMQ message broker, making scalability difficult at the city, department, and other data source levels. Additionally, it does not use URIs, limiting scalability due to the absence of unique identifiers for each data. Each new participant in the data space must understand the meaning of the character string such as "Price" or "luminosity", as another data source referring to the same attributes but with different characters would not be associated with the same meaning. Moreover, it does not use HATEOAS (Hypermedia as the Engine of Application State), which makes establishing connections between complex-related data impossible.

Lack of Data Structure: Data in this architecture is not structured uniformly, and each provider can structure the data as desired, complicating the understanding and manageability of data by consumers.

No Subscription to Data Changes: This architecture does not allow subscription to data changes. If a certain value needs to be modified, it is necessary to resend the entire dataset, which burdens the structure computationally and spatially, making it unsustainable with many participants and thus less scalable.

For the second architecture:

With the implementation of the certification procedure, it is known which specific capabilities the architecture has based on the requirements it meets, following the table of MIM 1 capabilities:

- Compliance with R1 and R2: Applications are able to access data from different sources (cities, communities, vertical solutions).

- Compliance with R3, R4, R5, R6: Applications are capable of retrieving real-time and historical data, as well as performing geospatial querying and subscribing to changes.

- Compliance with R7 and R8: Applications are able to retrieve data relevant to a certain context, including subsets of data within large datasets and setting default limits and page sizes.

With this information, it is known that the architecture of the Aarhus Living Lab pilot can achieve the objectives of MIM 1, allowing the collection of information from different systems within the city, IoT devices, and other data sources that meet these requirements identified in the implementation of the ITU Y.MIM standard of MIM 1, being a widely accepted and globally backed standard. Not only that, but it also allows for the management, sharing, and integrated and comprehensive use of that data from other sources.

The architecture that uses the Message Broker with RabbitMQ is highly scalable and does not comply with the requirements or capabilities of MIM 1 according to the implementation of the certification procedure for compliance with the requirements developed in this work. This indicates that it does not achieve the objectives of MIM 1 and does not reach the minimum level of interoperability to retrieve data from different sources and make comprehensive and integrated use of that data, which is what is intended to achieve seamless communication and connection between participants in a data space, at least in the context information management in the interaction between systems.

On the other hand, the second architecture that uses the NGSI-LD mechanism (to comply with requirements 7 and 8) does comply with all the requirements of MIM 1 and therefore achieves the level of interoperability sufficient for data sharing with context information in the interaction between systems.

This work applies the certification procedure for compliance with the requirements of MIM 1, defined in format through ITU Y.MIM supported by the European Commission, knowing precisely when an architecture or a system reaches the minimum level of interoperability, which is a fundamental pillar of common data spaces. Thanks to this, it can be known that an architecture is interoperable regarding context information management and can interact with others within a common data space regarding that context.

Recalling the European data strategy, the objective is to create a single market that ensures global competitiveness and data sovereignty. To enter this global single market, participants must comply with the minimum level of interoperability, which in the case of MIM 1, allows access to information and data from different sources within the global market, understand that data, and manage, share, and use it comprehensively with different solutions.

The implementation of the compliance certification procedure allows us to determine the level of interoperability of each architecture and evaluate how each architecture would behave in a real-world information sharing scenario, as well as its limitations.

Thanks to this compliance certification procedure, two different architectures of the same pilot project, Aarhus Living Lab, have been studied and analysed. This pilot project is a very relevant use case in the CitCom.AI TEF project, aiming to understand its level of compliance with the requirements of MIM 1 and therefore the level of interoperability and capabilities of each architecture.

After evaluating this work, it is important to clarify that the procedure serves to analyse the compliance of an architecture with the requirements of MIM 1. If an architecture meets the requirements of MIM 1, it means it is interoperable in terms of Context Information Management, but it does not guarantee any other level of interoperability.

In fact, in a data space, when talking about interaction between systems, interoperability is not sufficient if the architecture only complies with the requirements of MIM 1, as shown in the OASC schema on MIMs.

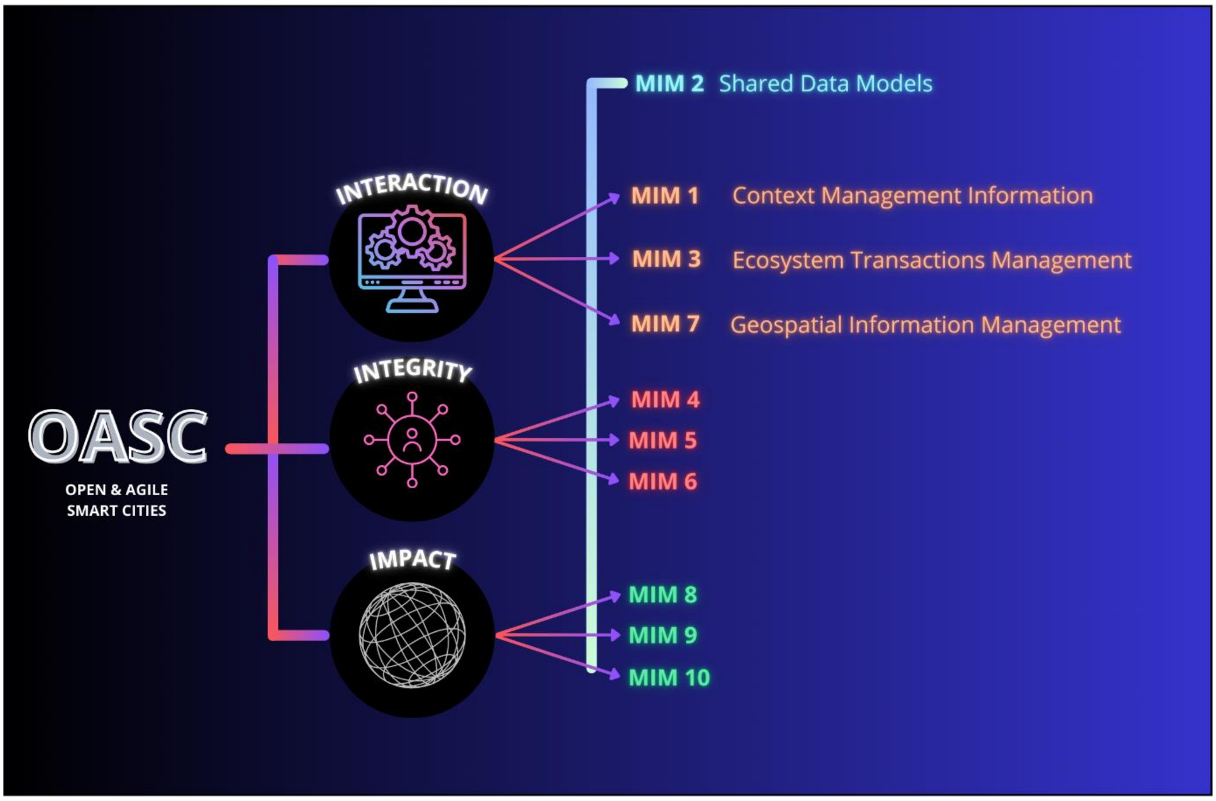


Figure 32: Source: own elaboration.

The interaction part is subdivided into three sections: Context (MIM 1), Contracts (MIM 3), and Geolocation (MIM 7). There is another aspect that is crucial and transversal, which is the Shared Data Models (MIM 2). Contracts and geolocation are part of the interaction but are not essential for data sharing. In other words, even if the requirements of contracts (MIM 3) and geolocation (MIM 7) are not met, data sharing can still occur.

However, what is crucial are the requirements of MIM 1 and MIM 2:

- MIM 1 establishes capabilities and requirements that allow access to information and data from different sources within the data space and the comprehensive and integrated use of that data.

- MIM 2, as a transversal actor, proposes the use of consistent and easily readable definitions for computers that are unique and understandable by all participants in the data space, whether they are cities, communities, businesses, or companies. This means the use of identifiers for each instance of each entity to ensure that all data space participants refer to the same thing, which, as explained earlier in this work, is semantic interoperability.

Continuing this work, following the certification procedure for MIM 1 requirements, and conducting another procedure to evaluate compliance with MIM 2 would be very interesting and valuable for related projects such as CitCom.AI, DS4SCC, SIMPL, and others.

Therefore, this work is of crucial importance in the context of Smart Cities, Industry 4.0, data sharing, data value exploitation, data spaces, interoperability, and many others in terms of analyzing the level of technical interoperability in Context Information Management. One limitation is that it is not possible to complete the analysis of interoperability in the interaction between systems, as the part of semantic interoperability covered by MIM 2 is missing.

ECONOMIC IMPACT

The specific economic impact generated by this work is practically impossible to estimate. However, it can be related to the projects it is involved in and their relevance and funding.

CitCom.AI TEF [29], funded with 40 million euros, directly uses the content of this work for the analysis of MIM 1 and its mapping and evaluation in various architectures relevant to the project, including Aarhus Living Lab, which has been analysed in this work.

SIMPL [25], funded with 65 million euros. DS4SSCC [49], funded with 19 million.

Not only in these projects but also in Industry 4.0, where the value of data is increasingly being leveraged to build solutions and improve sales forecasting, production planning, customer insights, and a myriad of potential enhancements that data usage enables in the industry.

Industry 4.0 is driven by technologies such as the Internet of Things, Artificial Intelligence, and data analytics. Therefore, this work focused on interoperability between systems and data spaces is fully aligned. Every business or company no longer only relies on its own data but, thanks to interoperability and data spaces, can use data, services, and information from other companies, either by purchasing them or exchanging them for their own data. Therefore, interoperability in industry and cities translates into two main positive impacts:

- Interoperability allows a company to use data from millions of other companies as long as it has the necessary permissions. It can now access not only its own data but also all data available in the market.

- A company's own data becomes valuable when it is interoperable, as other companies can now use that data either by purchasing it or exchanging it.

Delving into the economic analysis of this work, it will examine the savings arising from the use of an architecture known to meet the requirements of the MIM 1 compared to an architecture that does not meet the requirements of the MIM 1, which is known through the application of the procedure developed in this work.

For the "Current Architecture (testing pilot)" of the Aarhus Living Lab, the sharing of information and data is done through a RabbitMQ message broker and by using a specific routing_key.

The cost of RabbitMQ at this level of architecture is nil, it should be noted that this is at the local level.

– 0 €

When the data is sent in a text message format, without links between them or a defined and system-understandable data information model, it does not allow for subscribing to specific changes in data values. Instead, a hired operator must understand how the architecture works and what the structure of the data information model is, among other things, to obtain the relevant information from within the data sent by each provider. It is assumed that the operator must have training as a basic level computer engineer to be able to understand the architecture and the data structure for its subsequent use. Based on real examples of an operator transforming the raw data sent by providers into usable data for the consumer, it is estimated that an operator with basic level engineering training is capable of translating or transforming the data from 5 providers. A computer engineer earns an average of €24,000 a year. And one operator for each consumer since it is assumed that different companies use those data.

– 24.000€ * Y/5 * X € (X=number of consumers, Y= number of providers)

The providers only send the data in text message format through the routing_key, therefore, the cost would be for an operator who is able to connect through the "channel" with the RabbitMQ architecture and send the data with a routing_key. For this, a basic level computer engineer is required.

– 24.000 * Y € (number of providers).

For the consultation of historical and relevant data for a certain context, which is essential for the consumers, it would be necessary for each consumer, manually through an operator, to access the database where all the data is stored and manually take the relevant data. For this action of searching for relevant data within a database, it is estimated that a hired operator is needed for each consumer (since each one is independent) and it is not necessary for the operator to have training as a computer engineer. The average salary in Spain is around €20,000.

- 20.000 * X € (X= number of consumers)

The total expenditure, estimated broadly and taking into account the limitations of this type of architectures, the data spaces, and everything related due to being a leading-edge topic in full development with virtually no previous examples, is as follows:

Cost of Architecture 1: 0 + (24,000 * X * Y/5) + (24,000 * Y) + (20,000 * X) €

Regarding the second architecture, it does indeed meet all the requirements of the MIM 1, achieving a minimum level of interoperability through the use of the NGSI standard from FIWARE and with components in the architecture such as QuantumLeap or PostgreSQL.

FIWARE, as well as other components of the architecture like QuantumLeap, are open source. The data storage costs of PostgreSQL are not considered in this comparison as they are highly dependent on the amount of data, and in the previous architecture, there is a database assumed to be similar in cost, making the comparison of little use.

- 0 €

In this architecture, an operator is needed for each provider and each consumer who is capable of understanding how to make REST requests using GET, PUT, etc., and who understands the data identifiers of each provider in the architecture. This work has also been tested from Libelium, and a computer engineer with a basic/intermediate level can easily perform these functions in half a working day if estimated very negatively, meaning it would take much less time to perform such functions.

- 27.500 * 0.5 * (X + Y) €

For the cost of this architecture, the price of launching and maintaining in the cloud is not taken into account, for example, the implementation of a similar architecture has been carried out, and the maintenance cost is around €2,000 monthly, but it also depends a lot on the size of the architecture. Launching it would be somewhat more expensive, taking into account the training of the workers needed to understand the architecture at an internal level, how it works in terms of code, etc. Again, due to the difficulty of the subject, the novelty, and the lack of examples of real cases along with the fact that in the previous architecture, these types of costs are not considered due to their relevance in relation to the Management of Context Information MIM 1, in this analysis, these types of costs are not estimated, being considered similar for both architectures and not affecting the comparison.

Coste de la arquitectura 2: 0 + (27.500 * 0.5 * (X + Y) €

After this analysis, it can be clearly seen that the cost of both architectures focused on the Management of Context Information, among other things, is dependent on the number of participants (both providers and consumers) in the architecture—the more there are, the more expensive it is.

This translates to the first architecture requiring more understanding of the data and manual connections between them, in such a way that an operator has to be aware of the meaning of the data, its relationship with other data, from which provider it comes, etc. In the second architecture, the data are interconnected, the architecture allows for requests in a machine-understandable manner and returns data with context traceability, and the operator does not need to translate those data, nor understand them, nor collect them from a database manually.

In summary, if a new consumer or provider is added to architecture 2, they only need to know how to use the NGSI mechanism to interact with all the other participants of the architecture and understand the context of the information without manual connections and wastage of time in communications with new providers or new consumers. The problem with the first architecture is that the more participants there are, the more complex the management of context data by the participants will be because the architecture does not reach the minimum level of interoperability defined in the MIM 1 requirements, thereby increasing the total cost. The solution in the second architecture is that this management of context data is done within the same architecture and through links in the data, and not manually by the participants, allowing the architecture to scale and add participants increasing the total cost to a much lesser extent in comparison with the first architecture.

6 Conclusion

6.1 Conclusion and cost appraisal

Recalling the European data strategy, the goal is to create a single market that ensures global competitiveness and data sovereignty. To enter this global single market, participants must comply with the minimum level of interoperability, which, in the case of MIM 1, allows access to information and data from different sources within the global market, understanding, managing, sharing, and using them comprehensively and integrated with different solutions.

This access and use of all data sources within the global market allow estimating the economic dimensions unlocked by an architecture simply by meeting the minimum interoperability level of MIM 1. This, in a way, allows economically evaluating the impact that the RabbitMQ architecture would have compared to the architecture that uses the NGSI-LD mechanism and complies with the requirements of MIM 1.

In section "2.1.2 The Value of Data," it is explained that according to the EU Data Market Monitoring Tool (2020), the EU Data Market Study estimates that data will reach a value of 550 billion Euros in the EU 27 in the most realistic scenario. With an estimated value of data of 550 billion euros, and with a common data space containing all that data, an architecture that complies with the requirements of MIM 1, and thus minimally interoperable in the data interaction part for sharing and use, would allow harnessing the value of all that data.

For these reasons, it is essential to have a certification procedure to determine if an architecture complies with the requirements of MIM 1. However, leveraging the value of data is not as straightforward and immediate if the architecture complies with the requirements of MIM 1. It is important to remember that a common data space is a federated ecosystem and that for an architecture to achieve the minimum level of interoperability, it is not enough to comply with the part of data context interaction, but there are nine other MIMs currently covering other levels of interoperability necessary to be part of that desired ecosystem for creating a global data market.

Therefore, without forgetting the limitations, a certification procedure has been developed for the compliance of an architecture with the requirements in the implementation ITU Y.MIM of MIM 1, tested for two architectures of the same pilot project and very relevant use case for the CitCom.AI project. Libelium plays a fundamental role in the development of Task 3 of CitCom.AI TEF, specifically in the development of deliverable 3.1, in which, among other things, it analyses how compliance with the requirements of MIM 1 should be in relevant architectures of Use Cases of the CitCom.AI project, such as Aarhus Living Lab.

7 References:

[1]: https://www.itu.int/en/ITU-T/about/Pages/default.aspx

[2]: https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/how-to-participate/org-details/999999999/project/101100728/program/43152860/details

[3]: https://www.compute.dtu.dk/english/news/nyhed?id=9121fce9-b6db-45d9-947a-a14f7471fada

[4]: Otto, B. (2022). The Evolution of Data Spaces. In: Otto, B., ten Hompel, M., Wrobel, S. (eds) Designing Data Spaces: https://link.springer.com/chapter/10.1007/978-3-030-93975-5_1

[5] European Commission, Directorate-General for Communications Networks, Content and Technology, Cattaneo, G., Micheletti, G., Glennon, M. (2020). The European data market monitoring tool: key facts & figures, first policy conclusions, data landscape and quantified stories: d2.9 final study report, Publications Office. https://data.europa.eu/doi/10.2759/72084

[6] https://wikis.ec.europa.eu/display/jrcdataspaceswiki/1.3++Definitions

[7]: Open Platforms and Large-Scale Pilots in Digitising European Industry (2021): "Design Principles for Data Spaces". Available on: https://doi.org/10.5281/zenodo.5244997 [Consulted on: 10/01/2024]

[8]: Reiberg, Niebel, Kraemer (2022), 'What is a Data Space?', Gaia-X Hub Germany, White Paper 1/2022.

[9]: Data Space Business Alliance (2023): *Data Space Business Alliance Technical Convergence V2*. [Available on:] https://data-spaces-business-alliance.eu/

[10]: Nagel, L. (2021): "The Magic of Data Spaces Now". *International Data Spaces*. Available on: "https://internationaldataspaces.org/the-magic-of-data-spaces-now/" [Consulted: 02/11/2023]

[11]: 2 Core concepts and roles - Glossary - Data Spaces Support Centre (dssc.eu)

[12]: https://dssc.eu/space/Glossary/176554052/2.+Core+Concepts

[13]: European Commission, New European Interoperability Framework Promoting seamless services and data flows for European public administrations, EIF brochure, ISBN 978-92-79-63756-8. https://ec.europa.eu/isa2/sites/isa/files/eif_brochure_final.pdf. Last accessed: february, 2024.

[14]: European Commission, Directorate-General for Communications Networks, Content and Technology, Directorate-General for Digital Services, (2021) Proposal for a European Interoperability Framework for Smart Cities and Communities (EIF4SCC). Publications Office of the European Union. https://data.europa.eu/doi/10.2799/816559

[15]: Arnold, G. W., Wollman, D. A., FitzPatrick, G., Prochaska, D., Holmberg, D., Su, D. H., ... & Boynton, P. A. (2010). NIST framework and roadmap for smart grid interoperability standards, release 1.0.

[16]: Bass, L., Clements, P., Kazman, R. (2012). Software Architecture in Practice. Reino Unido: Pearson Education.

[17]: Serrano, M., Barnaghi, P., Carrez, F., Cousin, P., Vermesan, O., & Friess, P. (2015). Internet of Things iot semantic interoperability: Research challenges, best practices, recommendations and next steps. *IERC: European Research Cluster on the Internet of Things, Tech. Rep*.

[18]: https://www.bing.com/ck/a?!&&p=5e17721869e08a48JmltdHM9MTcwMTY0ODAwMCZpZ3VpZD0zMGU1ODE5NC04NjQ5LTY2MjUtMGJhYi05MTU1ODdlMTY3YWQmaW5zaWQ9NTE4Ng&ptn=3&ver=2&hsh=3&fclid=30e58194-8649-6625-0bab-915587e167ad&psq=CADDIA+PROGRAMME&u=a1aHR0cHM6Ly9ldXItbGV4LmV1cm9wYS5ldS9MZXhVcmlTZXJ2L0xleFVyaVNlcnYuZG8_dXJpPUNPTToxOTkzOjAwODQ6RklOOkVOOlBERg&ntb=1

[19]: https://eur-lex.europa.eu/EN/legal-content/summary/electronic-interchange-of-data-between-administrations-ida-programme.html

[20]: https://eur-lex.europa.eu/EN/legal-content/summary/the-idabc-programme-2005-2009.html#KEYTERMS

[21]: COM(2008)583 https://www.bing.com/ck/a?!&&p=fe18d111a16e2d63JmltdHM9MTcwMTY0ODAwMCZpZ3VpZD0zMGU1ODE5NC04NjQ5LTY2MjUtMGJhYi05MTU1ODdlMTY3YWQmaW5zaWQ9NTE4Mw&ptn=3&ver=2&hsh=3&fclid=30e58194-8649-6625-0bab-915587e167ad&psq=COM(2008)583&u=a1aHR0cHM6Ly9ldXItbGV4LmV1cm9wYS5ldS9sZWdhbC1jb250ZW50L0VOL0FMTC8_dXJpPUNPTToyMDA4OjU4MzpGSU4&ntb=1

[22]: https://www.bing.com/ck/a?!&&p=4b7ea7d5f187f54cJmltdHM9MTcwMTY0ODAwMCZpZ3VpZD0zMGU1ODE5NC04NjQ5LTY2MjUtMGJhYi05MTU1ODdlMTY3YWQmaW5zaWQ9NTI0MQ&ptn=3&ver=2&hsh=3&fclid=30e58194-8649-6625-0bab-915587e167ad&psq=The+Interoperability+Solutions+for+European+Public+Administrations+(ISA)+programme+(2010-+2015)&u=a1aHR0cHM6Ly9wdWJsaWNhdGlvbnMuZXVyb3BhLmV1L3Jlc291cmNlL2NlbGxhci9jODBjN2ZmMS0xNGU3LTQzZDYtYTU2Zi05OWJmMjU0YzMzZjkuMDAwMS4wMi9ET0NfMQ&ntb=1

[23]: https://www.bing.com/ck/a?!&&p=d29cc1b2af6de0ddJmltdHM9MTcwMTY0ODAwMCZpZ3VpZD0zMGU1ODE5NC04NjQ5LTY2MjUtMGJhYi05MTU1ODdlMTY3YWQmaW5zaWQ9NTE5NQ&ptn=3&ver=2&hsh=3&fclid=30e58194-8649-6625-0bab-915587e167ad&psq=The+Interoperability+Solutions+for+European+Public+Administrations+(ISA)+programme+(2010-+2015)&u=a1aHR0cHM6Ly9lYy5ldXJvcGEuZXUvaXNhMi9pc2EyX2VuLw&ntb=1

[24]: https://data.europa.eu/en/news-events/news/interoperable-europe-act

[25]: https://digital-strategy.ec.europa.eu/en/policies/simpl

[26]: https://digital-strategy.ec.europa.eu/en/activities/digital-programme

[27]: https://digital-strategy.ec.europa.eu/en/faqs/testing-and-experimentation-facilities-tefs-questions-and-answers

[28]: https://digital-strategy.ec.europa.eu/en/activities/work-programmes-digital

[29]: https://citcom.ai/

[30]: https://oascities.org/

[31]: https://docs.google.com/document/d/1nC5wSulop4Ig5xWtZ4nsmfbMs4mfc1eQ/edit?usp=sharing&ouid=114211767408858427688&rtpof=true&sd=true

[32]: https://mims.oascities.org/

[33]: Fu, G., Zhang, Y., & Yu, G. (2020). A fair comparison of message queuing systems. *IEEE Access*, *9*, 421-432.

[34]: Ionescu, V. M. (2015, September). The analysis of the performance of RabbitMQ and ActiveMQ. In *2015 14th RoEduNet International Conference-Networking in Education and Research (RoEduNet NER)* (pp. 132-137). IEEE.

[35]: Nannoni, N. (2015). Message-oriented middleware for scalable data analytics architectures.

John, V., & Liu, X. (2017). A survey of distributed message broker queues. *arXiv preprint arXiv:1704.00411*.

[36]: https://living-in.eu/news/mims-plus-version-60-approved

[37]: Wikipedia contributors. (2024, January 18). API. In Wikipedia, The Free Encyclopedia. Retrieved 11:23, January 25, 2024, from https://en.wikipedia.org/w/index.php?title=API&oldid=1196841027

[38]: Berners-Lee, T., Fielding, R., & Masinter, L. (2005). *Uniform resource identifier (URI): Generic syntax* (No. rfc3986).

[39]: Lange, K. (2016). The little book on REST services. *Dostupno na: https://www. kennethlange. com/books/The-Little-Book-on-REST-Services. pdf [4. svibnja. 2020]*.

[40]: Context Information Management (CIM) ETSI Industry Specification Group (ISG) (2023): *GS CIM 009 - V1.7.1 - Context Information Management (CIM); NGSI-LD API*, Published in: https://www.etsi.org/standards-search

[41]: https://www.ogc.org/

[42]: What is data modelling? IBM. Consulted on: [30/01/2024]. Available on: https://www.ibm.com/topics/data-modeling?mhsrc=ibmsearch_a&mhq=data%20modeling

[43]: https://www.itu.int/rec/T-REC-Y.2060-201206-I

[44]: FIWARE-NGSI v2 Specification. ETSI Industry Specification Group (ISG) (2018): *FIWARE-NGSI v2 Specification*, Published in: https://fiware.github.io/specifications/ngsiv2/stable/

[45]: https://dssc.eu/

[46]: https://data-spaces-business-alliance.eu/

[47]: https://living-in.eu/

[48]: https://www.ds4sscc.eu/

[28] = [49]: https://digital-strategy.ec.europa.eu/en/activities/work-programmes-digital

[50]: https://digital-strategy.ec.europa.eu/en/policies/strategy-data

[51]: https://commission.europa.eu/index_en

[52]: https://sdgs.un.org/

[53]: https://digital-strategy.ec.europa.eu/en/activities/testing-and-experimentation-facilities

[54]: https://digital-strategy.ec.europa.eu/en/library/results-new-european-data-market-study-2021-2023

[55]: https://learn.opengeoedu.de/en/opendata/vorlesung/merkmale/5star

[56]: https://www.agrifoodtef.eu/

[57]: https://tefhealth.eu/home

[58]: https://ai-matters.eu/

Figure 4: Adservio (2021): *What is a Data Lake*. Available on: https://www.adservio.fr/post/data-lake-and-its-benefits-in-data-management [Consulta: 07/11/2023]

Figure 5: European Interoperability Timeline: https://ec.europa.eu/isa2/sites/default/files/european-interoperability-timeline.jpg

Figure 32: insights from: https://oascities.org/minimal-interoperability-mechanisms/

8 ANEXO 1

1 - The **European Commission** acts as the administrative powerhouse of the European Union (EU). The European Commission is responsible for proposing legislation, implementing EU policies, and managing the Union's budget. It also ensures compliance with EU laws and decisions.

The European Commission is actively engaged in projects associated with Data Spaces, contributing to its overarching agenda of promoting digitalization and innovation driven by data within the European Union. Aiming to ensure that Europe's digital transformation is fully realized by 2030, the European Commission set the **European Digital Decade** strategy, which ambitions and objectives are: a digitally skilled population and highly skilled digital professionals, secure and sustainable digital infrastructures, digital transformation of businesses and digitalisation of public services.

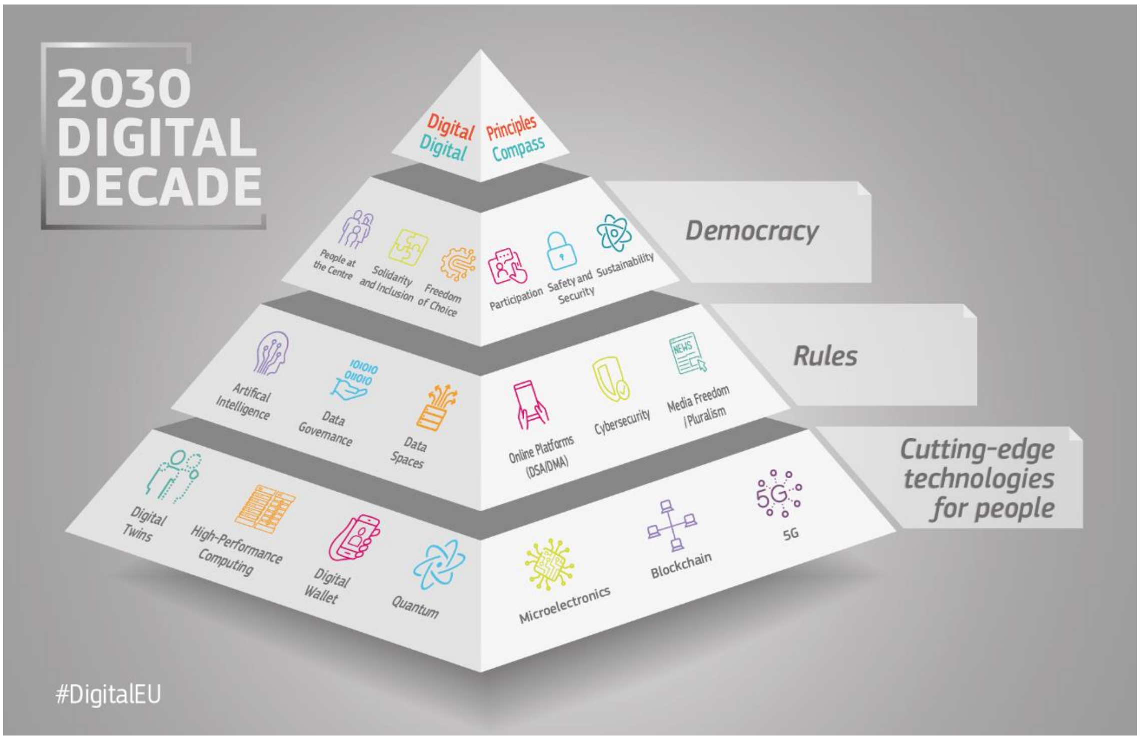


Figure: from the official website of the European Commission, European Digital Decade:

https://digital-strategy.ec.europa.eu/en/policies/europes-digital-decade#tab_1

[Consulted: 05/12/2023]

The concept of Data Spaces aligns with the EU's vision for a digital single market and the European Data Strategy, which aims to harness the potential of data while ensuring trust and privacy.

However, the EU Commission has already taken a few steps before, for example with the General Data Protection Regulation [1] (2016) creating a solid framework for digital trust, or other initiatives that have fostered the development of the data economy like the Cybersecurity Act [2] (2019) and the Open Data Directive [3] (2019).

[1] Regulation (EU) 2016/679

[2] Regulation (EU) 2019/881

[3] Directive (EU) 2019/1024

The following figure is aiming to depicture the structure explained in this point 1:

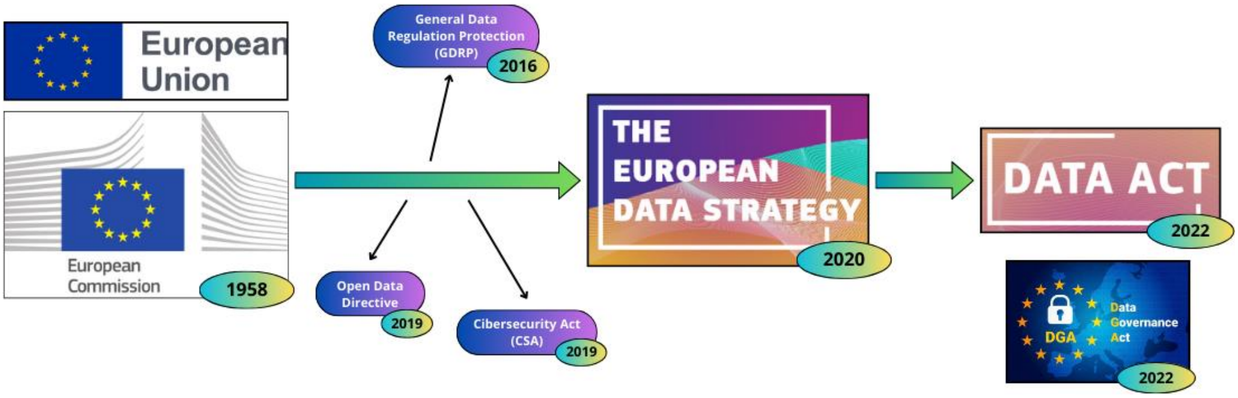


Figure XX: Author of this work.

1.1 - In February 2020, the European Commission announced the **European Strategy for Data**, aiming at creating a single market for data to be shared and exchanged across sectors efficiently and securely within the EU. The European strategy for data aims at creating a single market for data that will ensure Europe's global competitiveness and data sovereignty. Common European data spaces will ensure that more data becomes available for use in the economy and society while keeping the companies and individuals who generate the data in control.

https://digital-strategy.ec.europa.eu/en/policies/strategy-data

1.2 - A key pillar of the European strategy for data, the **Data Act**. Data Act is a Regulation on harmonised rules on fair access to and use of data. It seeks to increase trust in data sharing, strengthen mechanisms to increase data availability and overcome technical obstacles to the reuse of data. The Data Act was adopted by the Commission on 23 February 2022

https://digital-strategy.ec.europa.eu/en/policies/data-governance-act

1.3 – **The Data Governance Act** also supports the set-up and development of common European data spaces in strategic domains involving both private and public players, in sectors such as health, environment, energy, agriculture, mobility, finance, manufacturing, public administration and skills. The Data Governance Act entered into force on 23 June 2022, and it is applicable since September 2023.

https://digital-strategy.ec.europa.eu/en/policies/data-governance-act

2 - The **Open and Agile Smart Cities** (OASC), OASC was born in 2010, with another name: "Connected Smart Cities Network" (CSCN), it was an informal group of EU project coordinators in the area of "smart cities", and the OASC network was launched in March 2015, co-founded by Martin

Brynskov. The OASC realised that innovation resulting from publicly funded projects usually hit a dead-end soon after the funding stopped.

It is an initiative aimed at promoting open and collaborative standards in the development of smart cities. OASC works to establish a common framework that enables interoperability among technological solutions implemented in cities, thus facilitating the creation of more efficient and sustainable urban environments. The mission is to unite cities and communities around the world to build a global market for solutions, services, and data based on the needs of cities and communities. What this means is that a global market is needed to make it possible to share solutions between cities that are facing similar challenges.

3 - **Data Spaces Business Alliance** (DSBA). DSBA is a joint endeavour started by four organizations (Gaia-X, Big Data Value Association, FIWARE Foundation, and International Data Spaces Association) enabling the creation and operation of interoperable data spaces based on European values. DSBA is an organization that focuses on advancing and promoting data sovereignty, secure data exchange, and trusted data ecosystems. It aims to create a framework and ecosystem that enables secure and controlled data sharing among businesses and organizations. The alliance aims to empower businesses with the ability to retain control over their data while enabling secure and privacy-preserving data exchange and collaboration with trusted partners.

3.1 - The **International Data Spaces Association** (IDSA) is on a mission to create the future of the global, digital economy with International Data Spaces (IDS), a secure, sovereign system of data sharing in which all participants can realize the full value of their data. The IDSA provides in its webpage an image of how a data space could be:

https://internationaldataspaces.org/why/data-spaces/ [Consulted on: 28/11/2023]

3.2 - **FIWARE** aims to provide a set of open standards and components to enable the development of portable and interoperable smart solutions in a faster, easier, and affordable way, which facilitates the creation of digital ecosystems where different actors and applications can collaborate and share data efficiently.

https://www.fiware.org/about-us/

3.3 - **Gaia-X** is an initiative that develops, based on European values, a digital governance that can be applied to any existing cloud/ edge technology stack to obtain transparency, controllability, portability and interoperability across data and services. Gaia-X is French and German initiative and it comes from the private sector.

https://gaia-x.eu/

3.4 - **The Big Data Value Association** (BDVA) was established to support the Big Data Value ecosystem. It aims to boost European Big Data Value research, development, and innovation, and to foster a positive impact on the economy and society.

This image is not a general data space, it is only an example of a Data Space from the Internacional Data Space Alliance and it may give the reader one image to understand that there is one provider and one consumer, which are exchanging data within an environment that has a standard vocabulary, a broker, identity provider… these are rules and policies that both provider and consumer accept to be part of the Data Space. For better understanding, go to the webpage and mouseover to discover the different parts of it.

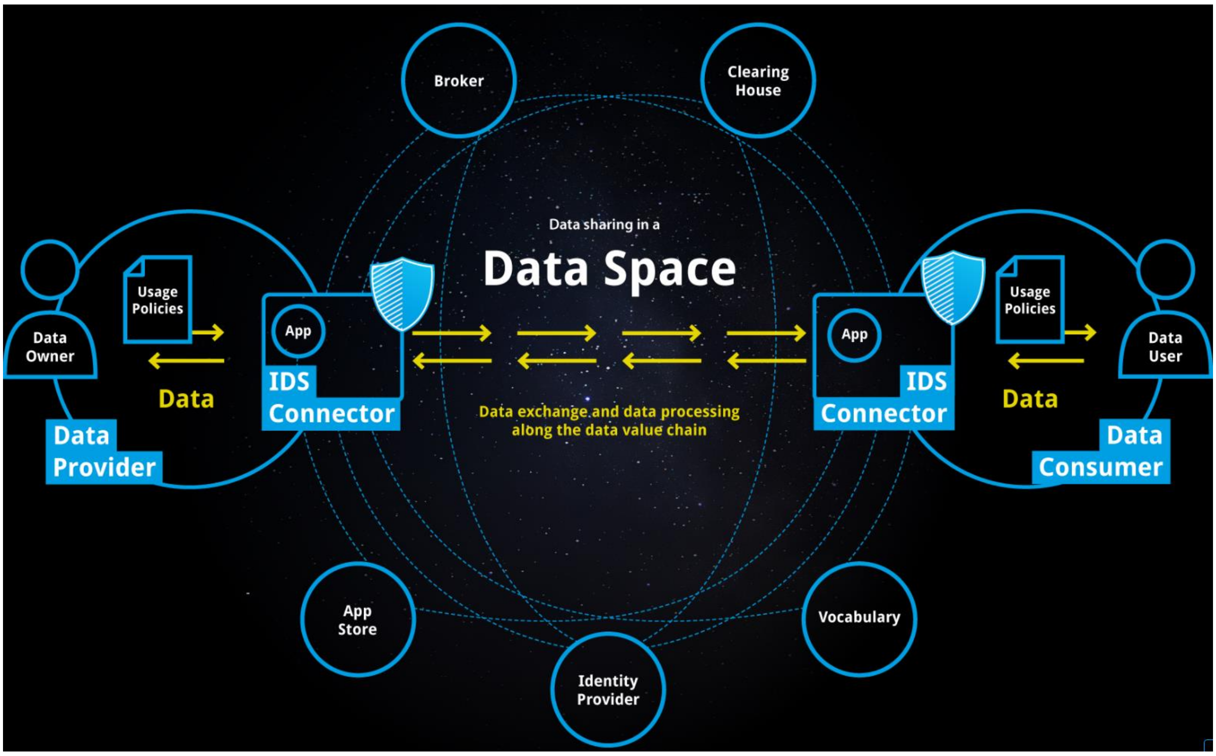
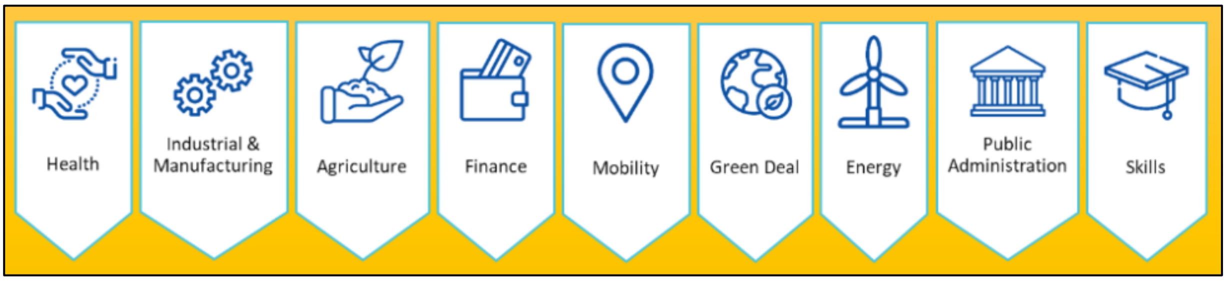


Figure X: Data Space Components

4 - **Data Space Support Centre**. Funded by the European Commission as part of the Digital Europe Program, the Data Spaces Support Centre aim to contribute to the creation of common data spaces, that collectively create a data sovereign, interoperable and trustworthy data sharing environment, to enable data reuse within and across sectors, fully respecting EU values, and supporting the European economy and society. Their vision is to explore the needs of data space initiatives, define common requirements and establish best practices to accelerate the formation of sovereign data spaces as a crucial element of digital transformation in all areas.

https://dssc.eu/space/Mission/175308804/Mission+and+Vision

The centrepiece of the European Data Strategy is the concept of “data spaces”, for which the Commission defined nine initial domains:



Nowadays, the Data Spaces Support Centre recognises eleven domains: cultural heritage, EUCAIM, Interoperability / Energy, Language, Agriculture, Green, Manufacturing, Mobility, Skills, Smart Cities, Tourism DATES, Tourism DSFT and Health.

https://dssc.eu/space/DC/27983886/Community+of+Practice

5 - **Aligning Reference Architectures, Open Platforms and Large-Scale Pilots in Digitising European Industry** (**OPEN DEI)** project aims to detect gaps, encourage synergies, support regional and national cooperation, and enhance communication. Open DEI focuses on ensuring that digital transformation efforts are unified and that best practices are shared among different sectors and regions.

Open DEI activities are focused on promoting digital skills and fostering innovation in key sectors like healthcare, agriculture, manufacturing, and energy.

https://www.opendei.eu/

6 – **Living-in.EU**: Living in EU is an initiative that was born in 2019. Its main objective is the digital transformation of Europe, supporting cities and regions to advance in their digital transformation according to European values. The initiative focuses on ensuring that cities and regions can reuse and scale existing digital solutions.

Living in EU promotes interoperability through minimum interoperability mechanisms and reference architectures to guide the implementation of open urban data platforms, enabling a reliable and seamless flow of data between the platform, its data sources, and connected services.

https://living-in.eu/

9 ANEXO 2

PYTHON CODE 3.12 (Visual Code)

“HELLO WORLD” (06/11/2023)

This tutorial contains all the steps needed to install RabbitMQ, install Docker, and build and run in your computer the “Hello World!” program using Visual Studio Code application. Is based on the tutorial of RabbitMQ available at the webpage: https://www.rabbitmq.com/getstarted.html

This webpage tutorial explains different ways to do it, using python, java, ruby… These options are very interesting but in this tutorial is used Visual Code Studio because of the knowledge of the author and the easier way to see the results more visually in order to compare the real scenarios presented in the thesis with the RabbitMQ concept structures. To achieve this program conversion the author has been based on a series of YouTube videos from the channel JumpstartCS (2022). Available on: https://youtube.com/playlist?list=PLalrWAGybpB-UHbRDhFsBgXJM1g6T4IvO&si=qk9RMgw_g9udflrG

Before starting with the “Hello World!” tutorial is necessary to install Docket, run it in the computer, and understand some concepts related to Docker, RabbitMQ and Visual Studio Code.

Install Docker

https://docs.docker.com/desktop/install/windows-install/

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.

Docker provides the ability to package and run an application in a loosely isolated environment called a container.

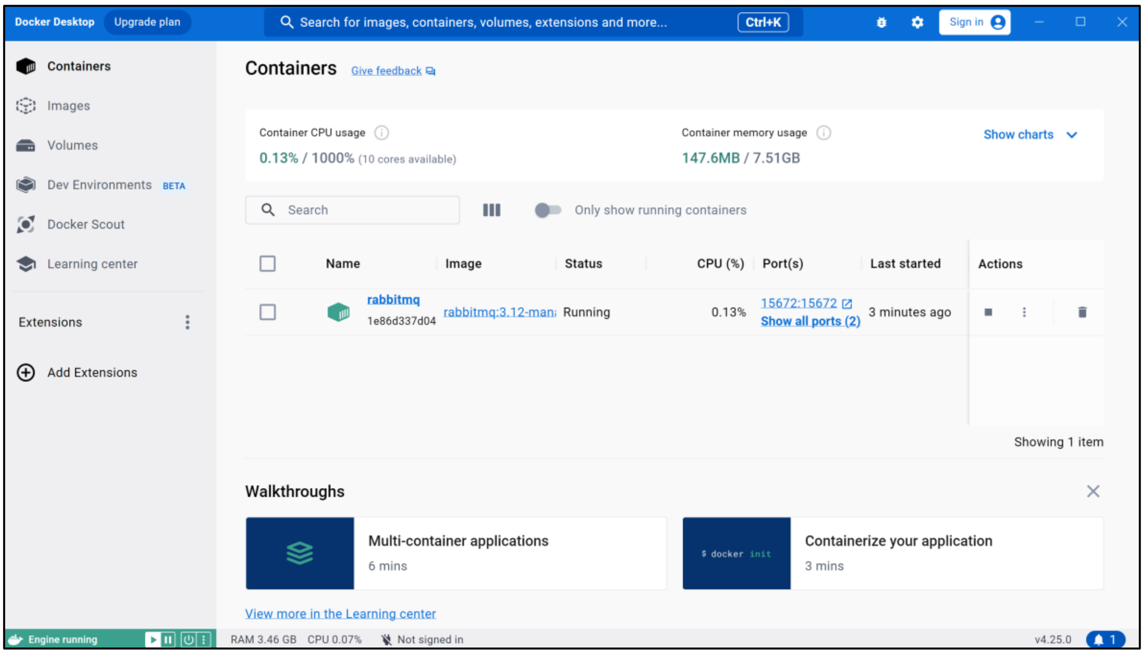
Second Step

Open CMD in your computer.

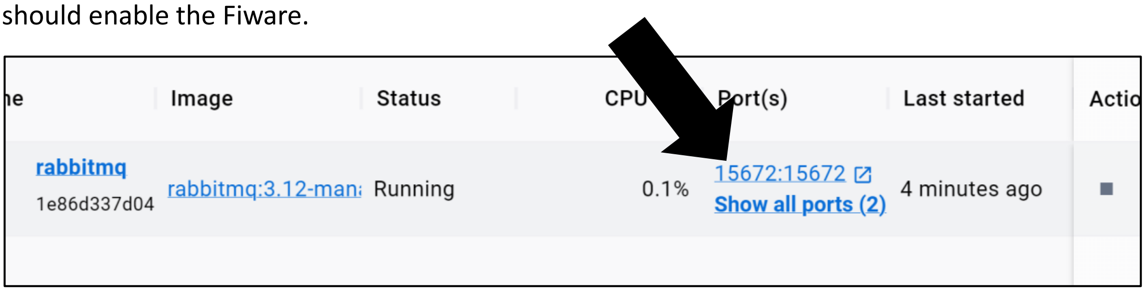
```
docker run -it --rm --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3.12-management
```

With the Docker installed, copy this line in the CMD. With this, your computer downloads RabbitMQ from Docker.

Now, you can see in the Containers (in Docker) that RabbitMQ is running; see Figure 1.



It is important to have rabbitmq running in port: 15672. (it is done by default) In case it is not you should enable the Fiware.



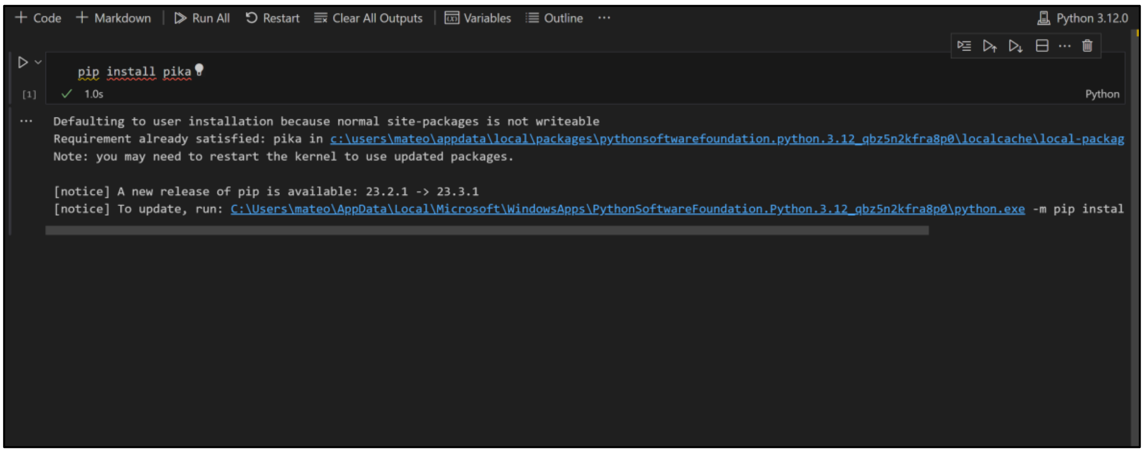
Open Folder

This work is done in Visual Studio Code.

First create a Folder, in this case the name is “RabbitMQ” for working on it.

Work in Visual Studio Code

Install Pika




```
pip install pika
```

Now, Pika is installed.

Connection with RabbitMQ server

```
import pika

connection = pika.BlockingConnection(pika.ConnectionParameters('localhost'))
channel = connection.channel()
```

We're connected now to a broker on the local machine - hence the localhost.

THEN JUST FOLLOW THE STEPS IN THE WEB.

https://www.rabbitmq.com/tutorials/tutorial-one-python.html

Tutorial YouTube (another version)

https://www.youtube.com/watch?v=j3tDM_hi90g&ab_channel=TheCoderCaveesp

Tutorial on VS Code (same version) Hello world! Tutorial

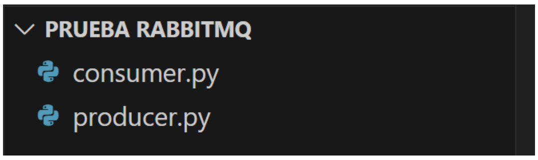
https://youtu.be/kwQDpHcM4HM?si=0FIMx37JGvW-a7g7

Hello world! Tutorial

This work is in Visual Studio Code version 1.84.1 (user setup)

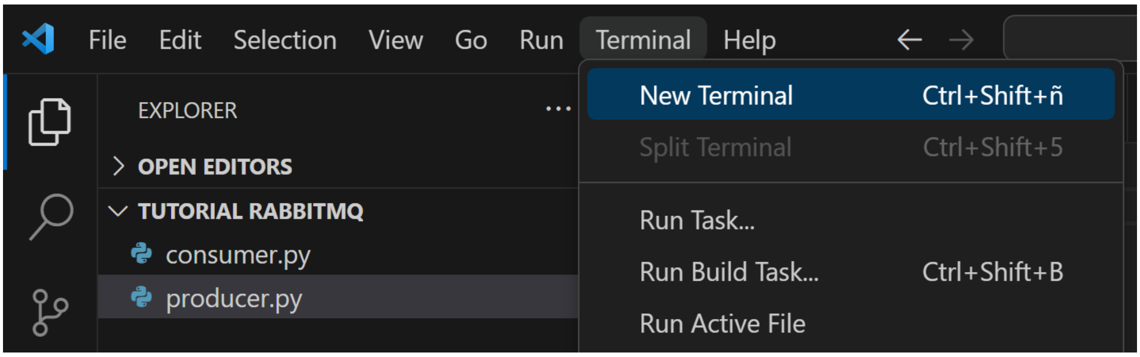
First, we create a Folder in our computer in this case the name is PRUEBA RABBITMQ

Now, we create two files, one consumer (receiver) and one producer (sender).



Now is necessary to install Pika (we already have pika in our computer (Docker))

To install pika, open Terminal / New Terminal



Then write on the terminal:

```
pip install pika
```

Now pika is installed.

Write the following code in producer.py

```
import pika

connection_parameters = pika.ConnectionParameters('localhost')

connection = pika.BlockingConnection(connection_parameters)
```

```
channel = connection.channel()

channel.queue_declare(queue='Queue 1')

message = "Hello this is my first message"

channel.basic_publish(exchange='',
                      routing_key="'Queue 1",
                      body=message)

print(f"sent message: {message}")

connection.close()
```

Now, you have created the **producer** which is going to send a message "Hello WORLD" to the queue named "Queue 1" through the channel created with the connection (this is more advanced and is not necessary how it works now).

And write the following in consumer.py

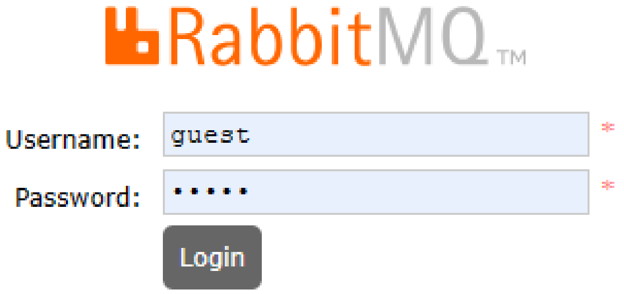
```
import pika

def on_message_received(ch, method, properties, body):
    print(f"received neew message: {body}")

connection_parameters = pika.ConnectionParameters('localhost')

connection = pika.BlockingConnection(connection_parameters)

channel = connection.channel()

channel.queue_declare(queue=''Queue 1')

channel.basic_consume(queue=''Queue 1',
                      auto_ack=True,
                      on_message_callback= on_message_received)
```

```
print("Starting consuming")
channel.start_consuming()
```

Now, you have created the **consumer** which is going to receive a message “Hello WORLD” to the queue named “Queue 1” through the channel created with the connection (this is more advanced and is not necessary how it works now).

Before run anything, first we are going to check the RabbitMQ Management page: http://localhost:15672/

Here we have to write username and password, by default both are “guest”.



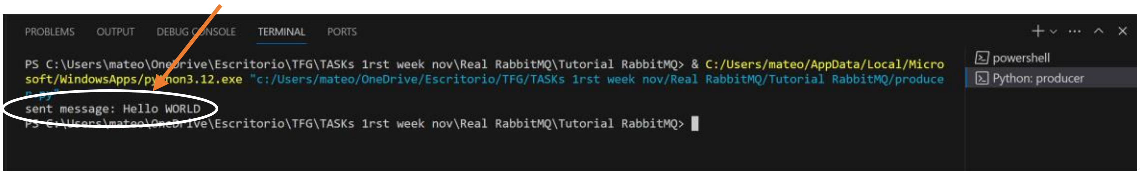
At this page we can see how is working our RabbitMQ.

We want to se the Queues so we go to the “Queues and Streams” page:

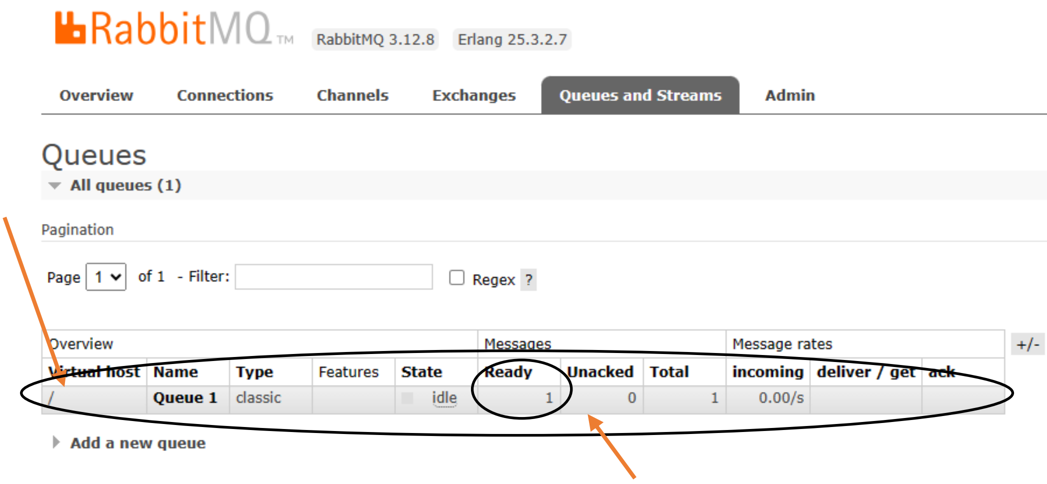
As you can see, there isn’t a Queue because we don’t run the producer.py yet. Let’s see it.

Now we want to run the producer.py: In the producer.py, we go to the arrow and select “Run Python File in Dedicated Terminal” as you can see in the image below:

And now, you can see in the terminal that message.

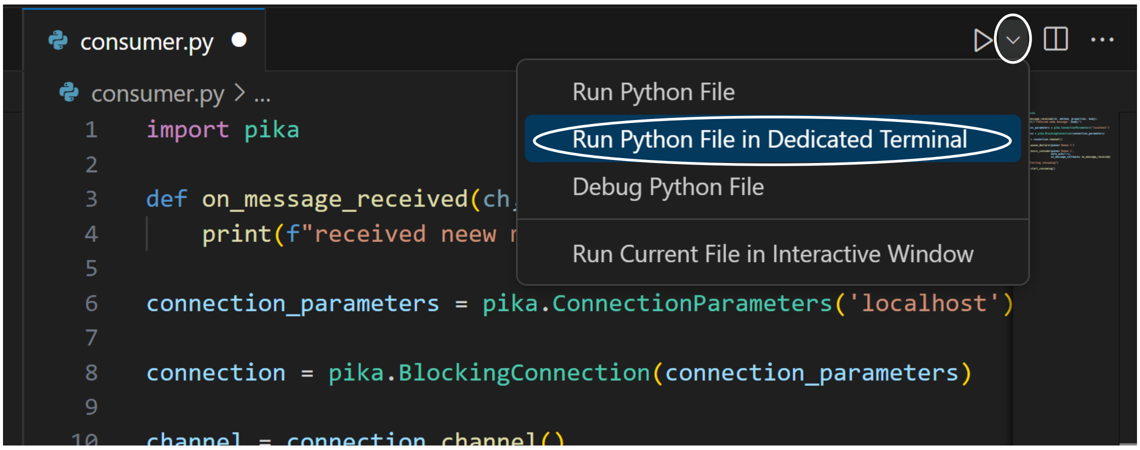


To see what happened we return to the RabbitMQ management page, and you can see that “Queue 1” has been created:

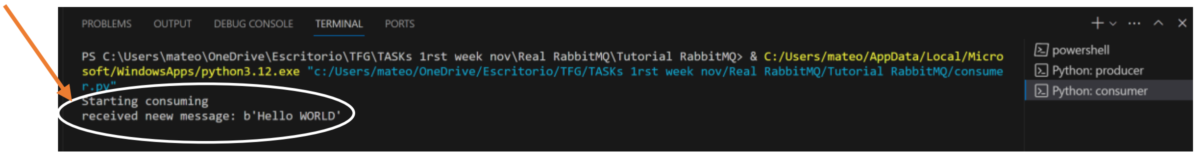


Also here, there is on messages, one message Ready, this is because we don’t run the consumer yet, so anyone has received the message, and it is stored in the “Queue 1”.

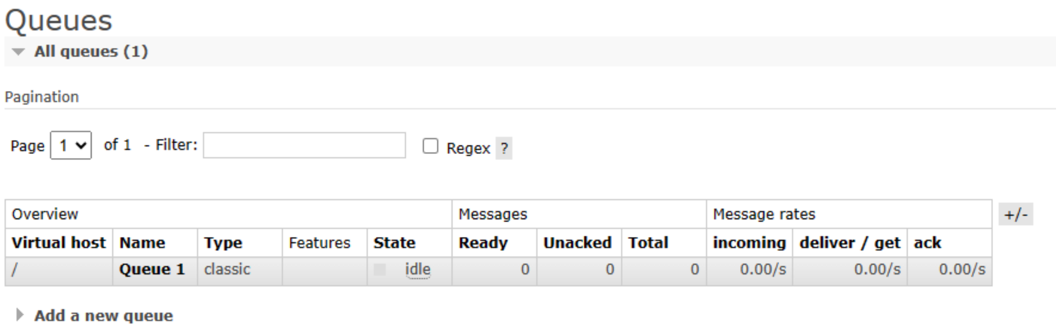
At this point, we are going to run consumer.py and see what happens. We run it the same way as the producer:



When it runs, you can see the that the message we sent from the producer arrives:



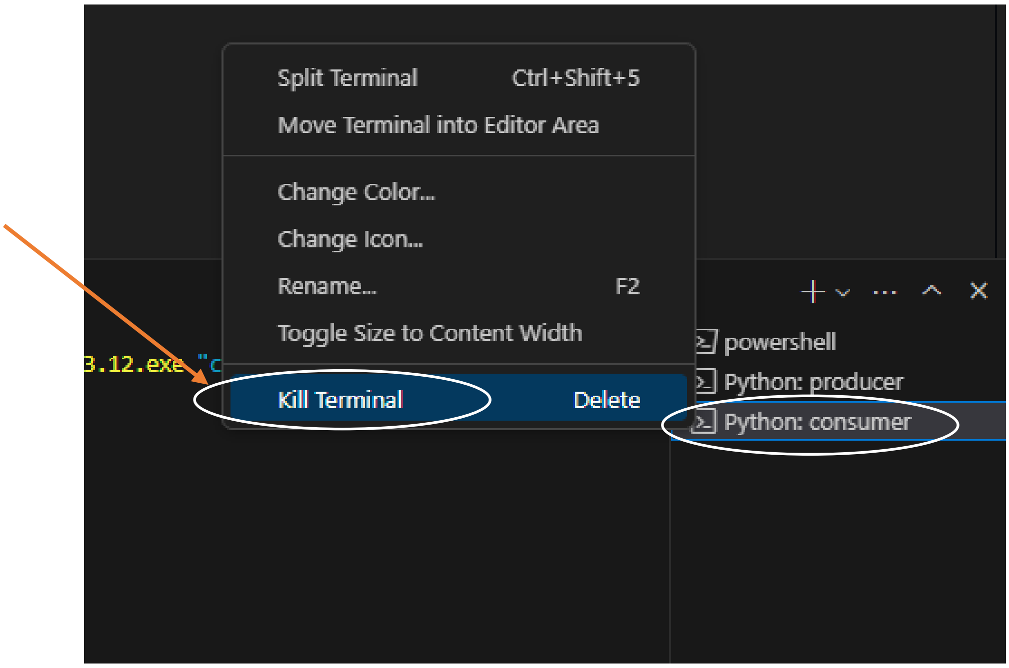
Now, the message has been received by the consumer, so let's see what happens in the RabbitMQ page:



Overview					Messages			Message rates		
Virtual host	Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack
/	Queue 1	classic		idle	0	0	0	0.00/s	0.00/s	0.00/s

The message is not in the Queue 1 because it has been received by the consumer.py so now the Queue 1 is empty.

The last thing is how closing the consumer.py or producer.py. The easiest way to do this is "Kill Terminal".



In this case, consumer.py terminal is closed, this means that it is not receiving messages anymore until it is opened again. Hence if producer sends messages now, all messages are going to the Queue 1 and waiting until consumer.py has being opened.

The Hello WORLD tutorial ends here.