IRIS

Integrated and Replicable Solutions for Co-Creation in Sustainable Cities

Project Acronym:	IRIS
Project Full Name:	Integrated and Replicable Solutions for Co-Creation in Sustainable Cities
Grant Agreement:	No 774199
Project Duration:	5 ½ years (starting 1 October 2017)

Deliverable 4.8

Second round of software updates on CIP components standards and processes

Work Package:	WP4: City Innovation Platform (CIP)
Task:	T4.6 Software updates on CIP components, standards and processes
Lead Beneficiary:	CERTH
Due Date:	30 September 2022 (M60)
Submission Date:	31 March 2023 (M66)
Deliverable Status:	Final
Deliverable Style:	Report
Dissemination Level:	PU
File Name:	D4.8 Second round of software updates on CIP components standards and processes.pdf

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 774199

Authors

Surname	First Name	Beneficiary
Vanmeulebrouk	Bas	CIV
Roux	Stéphane	NCA
Alldritt	Noel	GOT
Nordström	Camilla	GOT
Tsarchopoulos	Panagiotis	CERTH
Isaioglou	George	CERTH
Tsompanidou	Eleni	CERTH
Hamlatzis	Iosif	CERTH
Petropoulos	Stefanos	CERTH
Giagoudis	Ioannis	CERTH
Piperagas	Grigorios	CERTH
Zestanakis	Panagiotis	CERTH
Pramagioulis	Dionysios	CERTH

In case you want any additional information or you want to consult with the authors of this document, please send your inquiries to: irissmartcities@gmail.com.

Reviewers

D4.8 was an update of D4.7. For this reason it was only reviewed by the PMT.

Version History

Version	Date	Modifications made by
0.6	15/10/23	CERTH prepared the update of D4.7 and distributed it to UTR, NCA and GOT
0.7	1/12/2023	GOT added information regarding updates and lessons learned
0.8	10/1/2023	UTR added information regarding updates and lessons learned
0.9	17/1/2023	GOT added information regarding updates and lessons learned
1.0	24/3/2023	CERTH consolidated the contributions of the LH cities and wrote lessons learned and conclusions

Disclaimer

Executive Summary

This document, the deliverable D4.8, is related to task T4.6: Monitoring of CIP operation in LH cities and continuous upgrades.

The IRIS City Innovation Platform (CIP) collects, manages and exchanges data to support the IRIS demonstrations, as well as, to support the development of new applications and services in the IRIS Lighthouse (LH) cities (Utrecht, Nice Côte d'Azur and Gothenburg). The CIP and its components can handle large volumes of data from different sources.

Based on the CIP Reference Architecture (D4.4), standards and essential components (D4.5), and the Data Governance Plan (D4.3), each LH city has deployed, in month 30, a fully operational local instance of the CIP. The process was documented in the deliverable D4.6: Integration of CIP in LH Cities. D4.6 describes for each LH city the final CIP architecture, the particular implementation of the required technical components (i.e. Context Broker, Data Catalog, Historical Data, API Management, Load Balancing, Identity and Access Management, etc.), the available Data Models and the implementation of the local Data Market.

The operation of the CIP in the LH cities (T4.6) supports the local demonstrations in data-related tasks (e.g. collection, management, analysis, exchange, etc.). After its initial deployment, CIP had to be upgraded and its components had to be fine-tuned to cope with the particularities of the demonstration in the different LH cities. Moreover, the critical role of the CIP in the operation of many demonstrations and the monitoring data collection made it necessary to monitor its continuous operation. For this reasons, during task T4.6, the WP teams in Utrecht, Nice and Gothenburg a) developed additional functionalities in the local CIP installations to meet the new requirements from the IRIS demonstrations, b) updated some CIP components based on software bugs found or the availability of new versions of these components, c) implemented tools for monitoring the health and performance of the CIP, and d) implemented of processes for continuous deployment of upgrades/updates. In addition, CERTH developed an external tool to monitor the health and performance of the CIPs that deliver monitoring data automatically.

The design, development and operation of CIP in three Lighthouse Cities offered many lessons (Utrecht, Nice Côte d'Azur and Gothenburg). From the technical perspective, it is strongly recommended that a CIP should be widely interoperable with most if not all components coming from projects and vendors that use and promote open-source software so that it is easier to implement upgrades as soon as they become available. A CIP should also ensure compatibility with existing IT infrastructure and pay close attention to security aspects.

The development of CIP is more than an IT issue. It is also business development and it is important to bring businesses on board right at the start of CIP implementation to assign information owners and ensure the willingness to pay for the service. This is because stakeholders with potentially competing interests are involved in the CIP. Therefore, the CIP must be able to accommodate all these different requirements.

Data sovereignty is essential as data is the most valuable raw material. The challenge is to ensure that the data is handled correctly and is accessible and owned by the correct parties. Federated solutions can address legal issues by separating data and accessing it through a federated interface with well-defined contracts.

There are multiple open standards available for use. However, few of them are actively developed and used for collaboration between cities. Within IRIS we have seen that FIWARE offers an extensive collection of generic enablers for CIP functionality. The FIWARE Smart Data Models allow for the harmonization of APIs and data models and compliance with Minimal Interoperability Mechanisms of the Open and Agile Smart Cities Association.

In Nice the 3Vs (volume, variety, and velocity) are defining properties of a CIP and posed challenges in handling the data produced. The platform needed to be able to manage the large volume of data and handle its variety and quick velocity. To meet these challenges, the CIP architecture must be scalable and able to standardize data and have monitoring tools to ensure data quality and availability. The biggest challenge in moving from a demonstrator to an operational platform is ensuring high availability and organizing the technical and maintenance teams to maintain its operational condition.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

Abbreviation	Definition
ADE	Application Domain Extension
API	Application Programming Interface.
BeNext	Dutch company that collects home energy consumption data
BIM	Building information model
CIM	City Information Model
CIP	City Innovation Platform
CKAN	Comprehensive Kerbal Archive Network. An open-source open data portal for the storage and distribution of open data.
CSV	Comma-separated Values
CPU	Central Processing Unit
DC	Data Center
DCAT-AP	Data Catalogue Vocabulary -Application Profile
DMZ	Demilitarized Zone
DPA	Data Protection Authority
DPO	Data Protection Office
EIP-SSC	European Innovation Partnership on Smart Cities and Communities
ELT	Extract, Transform and Load
EU	European Union

EV	Electric Vehicle
FIWARE	Future Internet-ware. A curated framework of Open Source Platform components to accelerate the development of Smart Solutions
FME	Feature Manipulation Engine
GDPR	General Data Protection Regulation
GML	Geography Markup Language
IAM	Identity & Access Management
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LoRaWAN	Low-power, wide-area network
LTE	Long Term Evolution
MQTT	Message Queue Telemetry Transport,
NGSI	Next Generation Service Interface. An information model and API for publishing, querying and subscribing to context information.
OCPI	Open Charge Point Interface
OGC	Open Geospatial Consortium
OUP	Open Urban Platforms (also working group of EIP-SCC)
PbD	Privacy by Design
PEAR	Privacy Enhancing ARchitecture
PET	Privacy Enhancing Technology
PIA	Privacy Impact Assessment
PMRM	Privacy Management and Reference Model and Methodology
PRIPARE	Preparing Industry to Privacy-by-design by supporting its Application in Research
RAM	Random Access Memory
RDF	Resource Description Framework
REST	Representational State Transfer
SAREF	Smart Applications REFerence
SLA	Service Level Agreement
STH	Short Term History
TLS	Transport Layer Security
USEF	Universal Smart Energy Framework
WMS	Web Map Service
WP	Work Package
XML	Extensible Mark-up Language

Copyright

1 Introduction

1.1 Prerequisites

To fully understand the content of this document and benefit the most from it, it is recommended to read first the deliverables D4.5: Implementation and integration of core CIP components" (IRIS Project, 2019) and D4.6: Integration of CIP in LH Cities (IRIS Project, 2020). The deliverable D4.5 describes the technologies and technical components necessary for the deployment of the IRIS City Innovation Platform (CIP). Moreover, it contains all the information about the configuration management and deployment procedures required to implement or integrate the CIP's generic APIs, data models and FIWARE platform components. D4.5 ensures that CIP's system components offer the data collection functionality needed from the IRIS integrated solutions. Based on D4.5, the three IRIS Lighthouse (LH) cities (Utrecht, Nice Côte d'Azur and Gothenburg) integrated the core CIP components to create a local instance of CIP to support the local demonstrations. Thus, the deliverable D4.6 presents the integration of CIP in the IRIS LH cities. It describes how each LH city used the reference CIP architecture and the available components to deploy a fully operational CIP. Each LH city presents the final architecture and how each technical component (i.e. Context Broker, Data Catalog, Historical Data, API Management, Load Balancing, Identity and Access Management, etc.) has been implemented. Moreover, each LH city presents the available Data Models and the implementation of the local Data Market. If someone wants more information about the IRIS CIP, the following deliverables cover its functional and technical aspects:

- D4.2: Document describing the functional aspects of the architecture in accordance with the Functional & Technical requirements "interoperable and open solutions, standards and new business models" (IRIS Project, 2018).
- D4.4: Document translating the use cases of the Lighthouse Cities into a technical architecture as a reference for each implementation of the CIP within Lighthouse and Follower cities (IRIS Project, 2019).

1.2 Scope, objectives and expected impact

Deliverable D4.8 is related to task T4.6 Monitoring of CIP operation in LH cities and continuous upgrades. The task T4.6 includes:

- The development of new functionalities in the local CIP installations to meet the new requirements from the IRIS demonstrations.
- The update of CIP components based on software bugs found or the availability of new versions of these components.
- The implementation of tools for monitoring the health and performance of the CIP.
- The implementation of processes for continuous deployment of upgrades/updates.

The operation of the CIP in the LH cities (T4.6) supports the local demonstrations in data-related tasks (e.g. collection, management, analysis, exchange, etc.). After its initial deployment, the local CIPs had to be upgraded and its components had to be fine-tuned to cope with the particularities of the demonstration in the different LH cities. Moreover, the critical role of the CIP in the operation of many demonstrations and the monitoring data collection made it necessary to monitor its continuous operation.

D4.7 was an intermediary document aiming to inform about the evolution of the IRIS activities in WP4 City Innovation Platform. Moreover, the document facilitated the knowledge sharing between the local development teams but also with the IT personnel of the Fellow Cities. Emphasis was given in the update of architecture, data models and APIs, data market, and use cases.

The final deliverable of the task, D4.8 (this document) contains the outcomes of the task and presents all the activities starting from the installation of the CIP in the LH cities (M30) until the end of the project (M66). Therefore, D4.8 includes the content of D4.7 in terms of use cases and technical details but additionally contains the lessons learned from the development and operation of CIP in the three IRIS LH cities (Utrecht, Nice Côte d'Azur and Gothenburg).

1.3 Contributions of partners

The IRIS three Lighthouse cities developed the majority of the components presented in D4.8. CERTH coordinated the process and implemented a tool that continuously monitor the health and performance of the CIPs regarding the provisioning of monitoring data.

1.4 Relation to other activities

As CIP supports many measures in the LH cities, D4.8 related to the activities of WP5, 6 and 7: Utrecht / Nice / Gothenburg Lighthouse City demonstration activities. Moreover, as CIP is also used to automatically collect monitoring data from a number of demonstrations and to deliver them to KPI tool, D4.8 is related to D9.7 Report on evaluation and impact analysis for integrated solutions. Finally, the D4.8 was used in the knowledge transfer activities of WP8: Replication by Lighthouse regions, Follower cities, European market uptake.

1.5 Structure of the deliverable

The deliverable is structured around LH cities. Each LH city has a dedicated chapter to present the updates in architecture, APIs and data models, data market, use cases, tools implemented to monitor health and performance, and processes implemented for the continuous deployment of upgrades/updates.

- Chapter 1 contains the introduction.
- Chapter 2 presents Utrecht's implementation.
- Chapter 3 presents Nice's implementation.
- Chapter 4 presents Gothenburg's implementation.
- Chapter 5 presents the tool developed by CERTH to check the availability of monitoring data.
- Chapter 6 presents the lessons learned in the three LH cities.
- Chapter 7 contains the conclusions and recommendations.

2 Status of the CIP in Utrecht

2.1 Introduction

To support the IRIS demonstration in Utrecht, a number of data services have been developed. These data services collect data from various heterogeneous sources, store them in a time series or short term history database and publish them using multiple standardised API's. These API's are available in a data marketplace. The following use cases have been implemented using those data services.

- Monitor legal and illegal use of parking spots for charging electric vehicles, both with and without traffic signs ("Laden zonder bord");
- Monitor energy consumption in apartment buildings;
- Monitor energy consumption by smart street lighting.

The monitoring of battery usage from a battery installed to store energy from PV panels is a work in progress.

2.2 Final architecture

Figure 1 depicts the software architecture for the IRIS Utrecht City Innovation Platform (CIP). It consists of four layers.

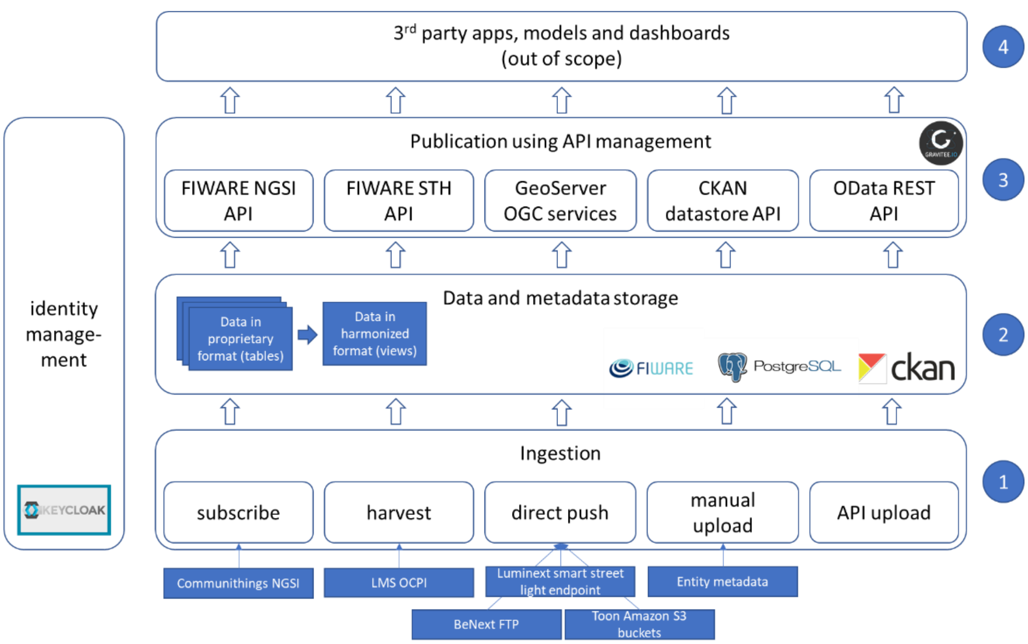


Figure 1 - IRIS Utrecht CIP architecture

1. **Ingestion layer.** Data can be ingested into CIP using a number of methods.

    a. CIP can subscribe, for instance, to a queue or a ContextBroker. This method is used to ingest the parking spot information from the Communithings parking sensors which have been installed;

b. CIP can periodically harvest data from 3rd party API's using a scheduled task. This method is used to collect information from EV charging stations from an OCPI endpoint;

c. Data can be pushed directly to CIP. An endpoint is defined for the data from the Luminext smart street lighting. The energy consumption information from the Toon smart thermostats is uploaded to Amazon S3 buckets, where it is picked up by CIP as soon as the upload finishes. The same method is applied for the BeNext energy consumption data, but then using an FTP upload instead of Amazon S3 buckets.

d. Data can be manually uploaded to CIP by navigating to a certain address and selecting a file (for instance a CSV, Excel or text file). This method is used to maintain static metadata regarding the entities (devices/stations) providing dynamic data.

e. Data can also be uploaded using an API, for instance, by ETL tools such as FME. This method is not being using within the frame of IRIS Utrecht.

2. **Data and metadata storage layer.** Data is stored "as is", in proprietary format. For publication purposes, data is transformed into a harmonised format using views. This allows us to support multiple output formats using the same raw data, modify the output format, apply this modification to historical data, and support multiple versions of the same dataset.

3. **Publication layer**. Data are published using appropriate, harmonised data models in different APIs. Whereas the API standardises communication between client and server (the envelope), the data model standardises the contents of the communication between client and server (the contents of the envelope). These data models and APIs are discussed in more detail in the next paragraph. The APIs rely on the views which select the raw data in a harmonised format. The APIs are available via an API management layer which acts as the marketplace and provides the starting point for including APIs in the FIWARE Biz ecosystem;

4. **External apps and platform layer.** The API's can be used to create apps, models, dashboards, etc. Developers take out a subscription on an API in the marketplace and can subsequently use the API within the boundaries provided by the plan they subscribed to. Creating those apps, models, and dashboards is beyond the scope of the IRIS Utrecht project. The IRIS project provides the data services which can be used to create apps, models and dashboards.

2.3 Data Models and APIs

The following data models are applied when publishing data using APIs.

The "Laden zonder bord" use case relies on two data sources: data from parking spots and data from EV charging stations. The **ParkingSpot data model** is used to publish information regarding parking spots. Figure 2 provides the mapping. All mandatory fields are present. Some optional fields are missing in the IRIS Utrecht dataset, but this is not a problem for the data model or using the data for the intended use cases.

id	Mandatory	Unique identifier.	From station metadata
type	Mandatory	Entity type. It must be eq	Dataset constant
dataProvider	Optional	Specifies the URL to infor	Dataset constant
dateModified	Mandatory	Last update timestamp o	From measurement
dateCreated	Mandatory	Entity's creation timestar	From station metadata
name	Optional	Name given to the weath	From station metadata
location	or address	Location of the weather	From station metadata
address	or location	Civic address of the weat	From station metadata
dateObserved	Mandatory	The date and time of this	From measurement
source	Optional	A sequence of characters	Dataset constant
refDevice	Optional	A reference to the device	-
refPointOfInterest	Optional	A reference to a point of	From station metadata
description	Optional	Description about the pa	-
status	Mandatory	Status of the parking spo	From measurement
timestamp	Optional	Timestamp which reflect	From measurement
width	Optional	Width of the parking spot	-
length	Optional	Length of the parking spo	-
refParkingGroup	Optional	Group to which the parki	From measurement
refParkingSite	Mandatory	Parking site to which the	From measurement
category	Mandatory	Category(ies) of the parki	From measurement
In line with FIWARE model		Either a mandatory or optional field which is present or a missing optional field	
Not in line with FIWARE model		Missing mandatory field	
Not in FIWARE model		Not in FIWARE model, present in data, useful to add	

Figure 2 - ParkingSpot data model mapping

The Communithings sensors have been installed in parking spots for disabled persons and parking spots for charging electric vehicles. They can be distinguished using the information in the category field. The category is either "ONSTREET - DISABLED" (for the parking spots for disabled persons) or "ONSTREET - ELECTRICALVEHICULE" for the parking spots for charging electric vehicles. Furthermore, the sensors installed in the parking spots for disabled persons do not have a GPS and thus, as a consequence, no location, just an address. This is not an issue from a data model perspective since either location or address must be present. The sensors in the parking spots for charging electric vehicles have a GPS and thus both a location and an address.

Since all parking spots for electric vehicles come in groups of two or four, they can also be modelled using the ParkingGroup data model (see Figure 3).

id	Mandatory	Unique identifier.	From station metadata
type	Mandatory	Entity type. It must be eq	Dataset constant
dataProvider	Optional	Specifies the URL to infor	Dataset constant
dateModified	Mandatory	Last update timestamp o	From measurement
dateCreated	Mandatory	Entity's creation timestar	From station metadata
name	Optional	Name given to the weath	From station metadata
location	or address	Location of the weather	From station metadata
address	or location	Civic address of the weat	From station metadata
dateObserved	Mandatory	The date and time of this	From measurement
source	Optional	A sequence of characters	Dataset constant
refDevice	Optional	A reference to the device	-
refPointOfInterest	Optional	A reference to a point of	From station metadata
description	Optional	Description about the pa	-
status	Mandatory	Status of the parking spo	From measurement
timestamp	Optional	Timestamp which reflect	From measurement
In line with FIWARE model		Either a mandatory or optional field which is present or a missing optional field	
Not in line with FIWARE model		Missing mandatory field	
Not in FIWARE model		Not in FIWARE model, present in data, useful to add	

Figure 3 - ParkingGroup data model mapping

This is useful for the specific use case since it alleviates us from coming up with an exact match between parking spots and sockets of an EV charging station. Instead, we can simply compare the number of available parking spots in the parking group with the number of available sockets in the EV charging station to determine the status of the combination and label it as legal or illegal.

The EVChargingStation data model is used to publish information regarding EV charging stations (see Figure 4). The information from the EV charging stations is not publicly available since it is sensitive information. Access is not granted automatically.

			Utrecht
			LMS/Lomboxnet
id	Mandatory	Unique identifier.	From station metadata
type	Mandatory	Entity type. It must be eq	Dataset constant
dataProvider	Optional	Specifies the URL to infor	Dataset constant
dateModified	Mandatory	Last update timestamp o	From measurement
dateCreated	Mandatory	Entity's creation timestar	From station metadata
name	Optional	Name given to the weath	From station metadata
location	or address	Location of the weather	From station metadata
address	or location	Civic address of the weat	From station metadata
dateObserved	Mandatory	The date and time of this	From measurement
description	Optional	Description of this chargi	From station metadata
image	Optional	A URL containing a photo	-
capacity	Mandatory	The total number of vehi	From station metadata
allowedVehicleType	Mandatory	Vehicle type(s) which can	From station metadata
socketType	Mandatory	The type of sockets offer	From station metadata
socketNumber	Optional	The total number of sock	From station metadata
availableCapacity	Optional	The number of vehicles w	From measurement
amperage	Optional	The total amperage offer	From station metadata
voltage	Optional	The total voltage offered	From station metadata
openingHours	Optional	Opening hours of the cha	-
status	Optional	Status of the charging sta	From measurement
areaServed	Optional	Area served by this charg	-
owner	Optional	Charging station's owner	Dataset constant
operator	Optional	Charging station's operat	Dataset constant
network	Optional	The name of the Networl	Dataset constant
contactPoint	Optional	Charging station contact	Dataset constant
chargeType	Optional	Type(s) of charge when u	Dataset constant
acceptedPaymentMetho	Optional	Accepted payment methc	Dataset constant
In line with FIWARE model		Either a mandatory or optional field which is present or a missing optional field	
Not in line with FIWARE model		Missing mandatory field	
Not in FIWARE model		Not in FIWARE model, present in data, useful to add	

Figure 4 - EVChargingStation data model mapping

For the information regarding energy consumption, there is no FIWARE Smart Data Model available. Therefore, an EneryConsumptionObserved data model was created using essential elements from other data models. This model contains the amount of energy consumed, the amount of energy produced by the PV panels, the amount of energy received by the apartment building/delivered by the energy supplier and the amount of energy delivered by the apartment building/received by the energy supplier.

The following API's are available:

The harmonised data are available using a number of API's. The code has been set up in such a fashion that adding additional API's is relatively simple.

- The FIWARE NGSI API (ContextBroker). This API provides access to the current state of entities and allows users to subscribe to updates from those entities;
- The FIWARE Short Term History (STH) API provides access to historical information from entities. Using the FIWARE NGSI API, users can learn which entities are available and which attributes those entities support. Subsequently, the STH API can be used to access historical information for those entities/attributes. The STH API can also be used to aggregate historical information for entities/attributes;
- GeoServer OGC services provide access to the information using the Web Map Service (WMS) and Web Feature Service protocols of the Open Geospatial Consortium. Although primarily targeted towards spatial information, these protocols can also be used to access alpha numeric information. And although these protocols are somewhat old fashioned (they do not adhere to REST principles yet), there is a lot of software available which can work with these services;
- The CKAN datastore API provides read/write access to data in the time series/STH database. There are several ETL tools (for instance FME) that provide connectors for this API;
- The OData REST API allows, for instance, Tableau to connect to the information in the time series database. The municipality of Utrecht uses Tableau to create dashboards.

2.4 Data market

API's are made available via an API management component Gravitee. Gravitee allows data providers to publish their API's, define plans for those API's and monitor usage of their API's. Data users can use Gravitee to test API's, read the documentation and take out subscriptions in API's.

Gravitee consists of three components:

1. The management portal (see Figure 5 for a screenshot of the front page) is used by administrators to configure API's and plans and inspect usage statistics. This look and feel of this management portal is not yet tailored towards the project's corporate identity;
2. Data users use the portal to subscribe to API's (see Figure 6 for a screenshot of the frontpage). The look and feel of this portal is tailored towards the project's corporate identity. The portal is accessible via https://tst-gravitee-iris-portal.dataplatform.nl/ (in test), https://acc-gravitee-iris-portal.dataplatform.nl/ (in acceptance) and https://gravitee-iris-portal.dataplatform.nl/ (in production). The test and acceptance environments may contain changes that have not been deployed to production yet;
3. The gateway is a non-visual component that is used to connect apps, models and dashboards too. Software is not connected to the API directly but via this gateway. The gateway validates and logs the request before forwarding it to the actual API.

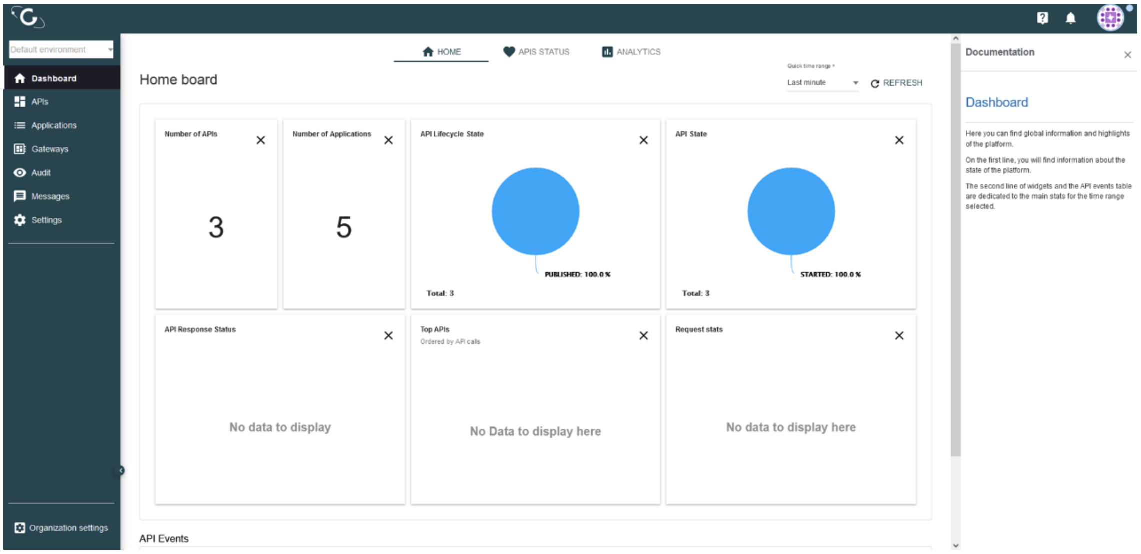


Figure 5 - Frontpage of the Gravitee management portal

Figure 6 - IRIS API management portal for data users

For most API's, three plans have been created[1]:

- A sandbox plan which can be used without registration and subscription. This plan allows users to experiment with the API without having to take out a subscription. This plan allows for a limited number of requests;
- A data user plan for users who want to actually use the API. The restriction on the number of requests is increased significantly. Depending on the nature of the data, subscriptions are automatically approved or need the approval of the administrator;
- A data provider plan for users who are allowed to also upload data. Subscriptions to this plan always require approval by the administrator.

[1] For the EV charging station API, the sandbox plan is absent since this API discloses nonpublic data.

2.5 Use cases update

2.5.1 Laden zonder bord

Description

Usually, parking sports designated for charging electrical vehicles are equipped with a traffic sign stating that parking is only permitted when charging an electrical vehicle. Installing these traffic signs requires a permit and installation and maintenance of these traffic signs requires resources. In addition, it leads to additional clutter in public space and as a consequence more resistance to charging stations for electrical vehicles than strictly necessary. Purpose of "Laden zonder bord" or "Charging without traffic sign" was to investigate what would happen if such a traffic sign was not installed with parking spots designated for charging electrical vehicles. Would a parking spot without a traffic sign attract more illegal use (parking without charging an electrical vehicle) than a parking spot equipped with a traffic sign?

To be able investigate this, 40 parking spots were equipped with a parking sensor which could determine whether a car was parked at the parking spots. Of those 40 parking spots, 20 were equipped with a traffic sign stating that it was only permitted to park there if you were charging an electrical vehicle and 20 were not equipped with such a traffic sign. The idea was to combine the data from the parking sensors with the data from the charging stations (payment information) to determine legal and illegal use for both categories.

To be able to answer the question at hand, the following data streams have been used:

- Data from the Lomboxnet EV charging stations is published using OCPI Open Charge Point Interface (NKL, 2020). From the OCPI specification, the Charge Detail Records (CDR) are used to obtain the information regarding usage of the charging station. A CDR is the description of a concluded charging session. An agent was created to translate CDR information from OCPI into the NGSI compliant EVChargingStation Smart Data Model.
- Data from the Communithings parking spot sensors are published using an Orion ContextBroker. This context broker uses the NGSI compliant ParkingSpot Smart Data Model to publish updates. As a consequence, no conversion was needed.

Lessons learned

For this use case, a lot of attention was required to prevent undesirable use of the data. Detailed information regarding the use of charging stations would allow charging station providers (competitors of the company who installed the charging stations used for this project) to "cherry pick" locations to install new charging stations, leading to an uneven spread of charging stations across the city. Therefore, unauthorized access to the raw underlying data had to be prevented and the results had to be presented in such a way that it becomes impossible to derive the success of individual charging stations from the results.

In addition, analysing the data was not as straightforward as anticipated. The data from the parking sensors required a lot of post processing to be able to derive actual parking actions from them. There were numerous really short (one second) parking actions which had to be eliminated since these are not actual parking actions. After removal of those, there were still a couple of parking sports which showed strange patterns. Some of these could be explained after a field visit. They were for instance caused by wrong installation, or a pile of sand on top of the sensor. But not all strange patterns could be explained, making us distrust the results.

Conclusion

There was no large difference legal/illegal use in the data between parking spots with or without a traffic sign. However, due to the amount of inexplicable strange patterns found for some of the parking spots this conclusion cannot be drawn with much certainty (Figure 7).

Laadlocatie		Parkeermomenten (aantal)		Parkeertijd (in uren)		Percentage onjuist (%)	
Adres	Projectopstelling	Aantal totaal	Aantal onjuist	Uren totaal	Uren onjuist	Parkeermomenten	Parkeertijd
Bekkerstraat (hoek Griftkade 15)	Zonder bord	518	355	7.488,85	5.329,97	69%	71%
Broeder Alamstraat 2	Zonder bord	598	395	6.286,60	3.048,61	66%	48%
Bufaatdreef 1	Zonder bord	491	221	7.303,94	3.739,07	45%	51%
Kortrijkpolder 8	Zonder bord	476	141	6.211,94	2.687,85	30%	43%
Livingstonelaan	Zonder bord	622	341	12.837,18	9.687,47	55%	75%
Nieuwe Koekoekstraat 102 (J Bekastraat)	Zonder bord	478	313	4.273,31	1.907,31	65%	45%
Nieuwe Pijlsweerdstraat 61	Zonder bord	527	511	773,34	656,33	97%	85%
Van der Duijnstraat 1	Zonder bord	528	379	4.693,81	2.754,03	72%	59%
Westerdijk hoek D van Mollemstraat	Zonder bord	670	300	8.780,54	3.795,45	45%	43%
Wolter Heukelslaan 72 linkerduo	Zonder bord	639	502	9.063,49	6.748,27	79%	75%

Figure 7 - Example "Laden zonder bord" results

2.5.2 Energy consumption

Description

To be able to determine the effect of energy saving measures taken during the renovation of a couple of apartment buildings in the Kanaleneiland neighbourhood, data regarding energy consumption were collected from various sources. To be able to measure energy consumption, the following variables must be known:

- The amount of energy delivered from the network to the connection (both electricity and gas);
- The amount of energy generated by the PV panels (electricity);
- The amount of energy returned from the connection to the network (electricity).

Data from smart thermostats (so called Toon smart thermostats) which measure energy delivered from the to the connection and energy returned from the connection to the network were combined with data from BeNext which measures the energy generated by PV panels. Unfortunately, not all apartment buildings with PV panels were equipped with the BeNext devices the measure energy generation. As a consequence, data from a building equipped with these devices had to be used to estimate energy generation for similar buildings nearby without those devices.

The following data streams have been used for this use case:

- Data from apartment buildings in Kanaleneiland are collected using Toon smart thermostats. Quby - the company responsible for these smart thermostats - collects the data from individual households, aggregates the data at apartment building level and uploads them to an Amazon AWS S3 bucket;
- Data from individual households in the apartment building in Overvecht are collected by BeNext. They provide an API to access these data at household level. For our project, data at household level is never stored. Data is aggregated to apartment building level upon arrival in the platform.

Lessons learned

For this use case, it was extremely important to take privacy aspects into consideration. Providing access to energy consumption data from individual households had to be prevented at all costs. Therefore, data were aggregated at apartment building level. Since apartment buildings are renovated as a whole, this allows us to investigate the effect of energy saving measures at apartment building level. Comparing different apartments (for instance apartments in the top left or top right corner with apartments in the middle) is not possible since that would allow you to link energy consumption data to a limited number of households.

Due to these privacy restrictions, it was not possible to estimate energy generation from a building using data from another building. Not all apartments within both buildings were equipped with a Toon smart thermostat and not all PV installations were activated. To be able to do the estimation, we needed to know which apartments had both a smart thermostat and PV panels, but this information was not available.

Furthermore, in this use case it was important to come up with proper definitions of "deliver" and "return" since these terms have different meanings from different perspectives. And it is very important to make sure all energy consumption is taken into consideration. Apartment buildings have centralized installations which also consume energy which must be taken into consideration.

And for this use case we had to work around the difference in time resolution from different data sources. Data regarding electricity were available at 15-minute intervals, whereas data regarding gas were only available at 60-minute intervals. The Short-Term History API as defined by the FIWARE STH Comet generic enabler provided us with a standardized API to accommodate for these differences. However, this status of this API is a bit uncertain.

Conclusion

The data collected for this use case provides useful insights into the effect of energy saving measures.

2.5.3 Battery

Description

A battery has been installed in the Kanaleneiland neighbourhood to store energy generated by PV panels. This battery allows energy storage when energy production is high but consumption is low. This energy can be used to reduce peak loads in the energy network when production is low and demand is high. Data resulting from operating this battery could be used to simulate the optimal behaviour of this battery on the energy market.

Lessons learned

Obtaining access to this data source has not been accomplished yet. There is no official access point to these data for which we can get a login. There is an unsupported API that we might be able to access with someone else's credentials, but whether that is a neat solution is debatable. Providing access to data in a readable format should be part of the tendering process when purchasing one or more devices which collect data.

Conclusion

Since access to these data was unavailable at the time of writing, a conclusion regarding this use case cannot be drawn.

2.5.4 Smart street lighting

Description

In the Kanaleneiland neighbourhood, smart street lighting has been installed. In order to save energy, these streetlights are only turned on when it is actually needed. In addition, a smart zebra crossing was installed. This zebra crossing is also equipped with lights to warn drivers for the presence of the crossing to increase safety, but is only turned on when traffic is detected. Data from these street lights are collected for two purposes: for inclusion in a digital twin (a 3D model of the Kanaleneiland neighbourhood) and to be able to calculate the amount of energy being saved because of the smart street lights.

Lessons learned

Obtaining access to these data was not difficult from a technical perspective. But obtaining all the information needed for our use case was somewhat more difficult than anticipated since providing the data for our project was not a primary concern for both the company providing the hard and software infrastructure for the smart street lights and their customer at the municipality. For instance, locations of the smart street lights were needed to include the data in the digital twin. The municipality provided us with the locations of the street lights, but those could not be linked to the data coming from the street lights. Solving this issue was not high on their priority list.

Furthermore, a lot of people really like the smart zebra crossing. But due to financial constraints, the municipality will not install more of these smart zebra crossings soon because of the cost associated with installation and maintenance costs. It is important to be clear about this upfront. Doing a prototype or a demonstration project may not be that difficult, but scaling a solution to a whole neighbourhood or municipality may not be feasible despite a successful demonstration project.

Conclusion

The data collected for this use case provides valuable insights into the energy consumption of street lights.

2.5.5 Data for calculated the KPIs

CERTH has created a tool to calculate the IRIS project’s KPIs (KPI tool). This tool relies on the CKAN datastore API. Figure 8 contains a screenshot of these data in the CKAN user interface.

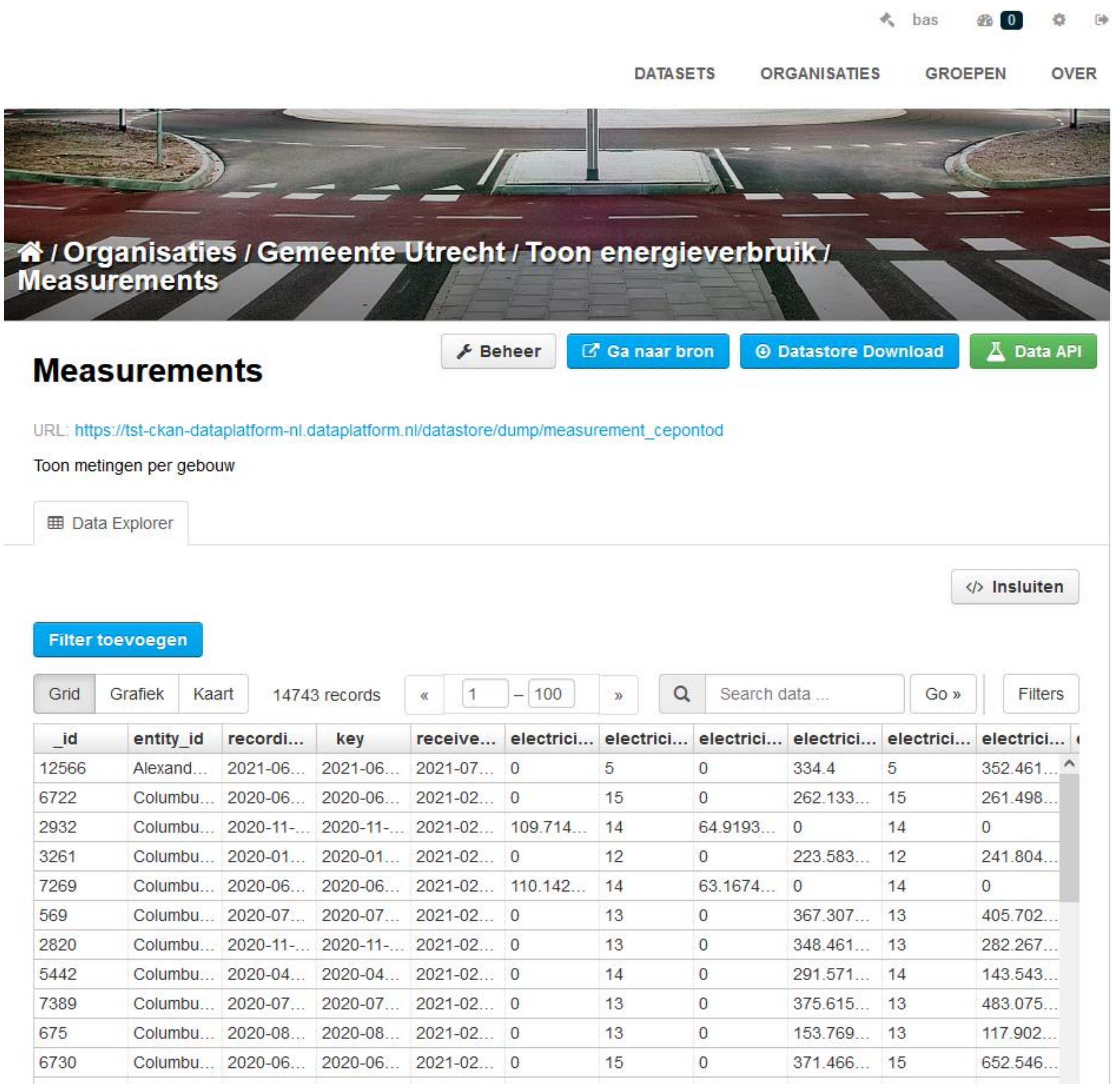


Figure 8 - Raw smart thermostat data in CKAN datastore

2.6 Tools implemented to monitor health and performance of CIP

The usage of CIP is monitored using the Gravitee API management software. Both data providers and data users can log in to see usages statistics for the API's they are offering (in the case of data providers) or using (in the case of data users). Data providers can see usage and performance statistics per application, plans and path in the API (Figure 9). In addition, data providers receive statistics on response times and response codes.

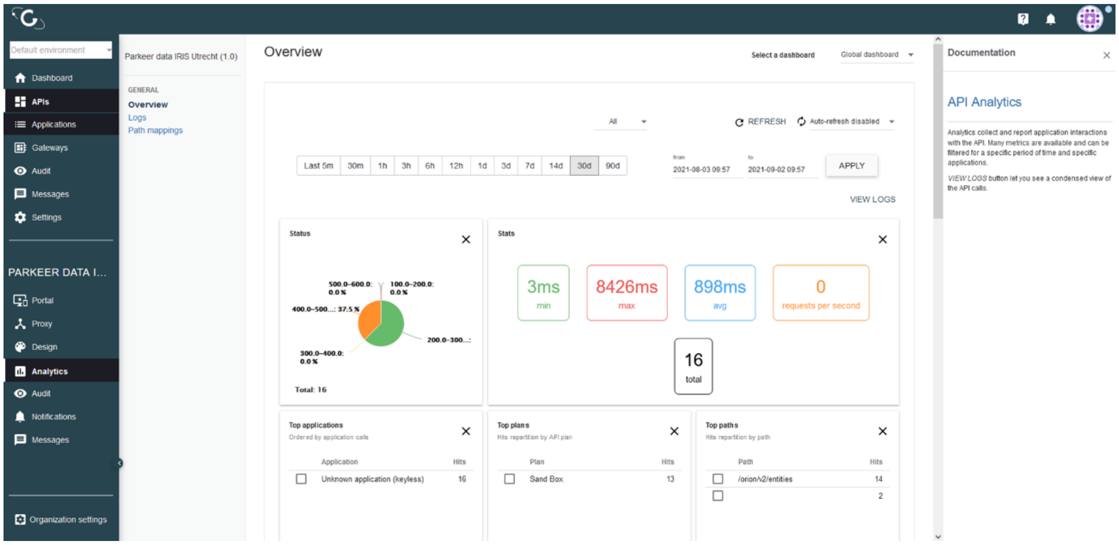


Figure 9 - Gravitee API analytics for data providers

The health and performance of CIP is monitored using Zabbix. Figure 10 presents the Zabbix dashboard. One of the IRIS processes has been manually (intentionally) stopped so to get a warning in the Zabbix dashboard.

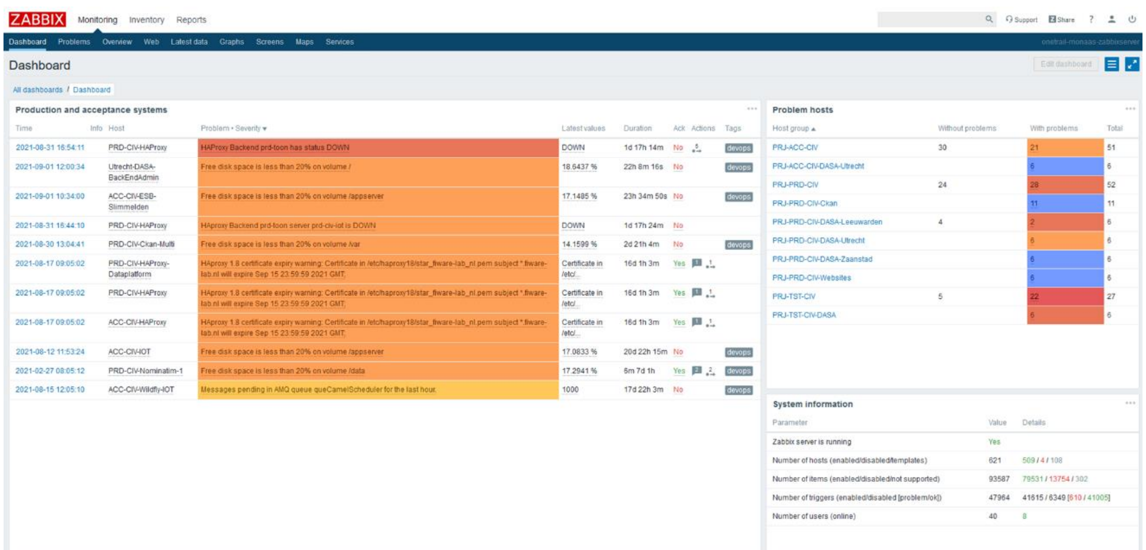


Figure 10 - Zabbix health and performance monitor

Zabbix monitors:

- If processes are running;
- Disk usage. If a disk is full, a service cannot function, for example, because it cannot write log files or data in the database anymore;
- Open files. If there are too many files open at the same time, this is usually an indication of a problem, and at some point, a server will stop functioning since the maximum number of open files has been reached;

- CPU and RAM usage. If there is not enough CPU and/or RAM available, processes tend to run in a sub optimal fashion;
- Queues. Queues act as "to do" lists. If these "to do" lists are not being emptied, something is wrong;
- TLS Certificate validity. If TLS certificates expire, the system cannot function properly. Therefore, warnings are sent out one month before certificates expire, giving us enough time to renew the certificate.

In case of a problem, Zabbix sends out a message to administrators who can resolve the issue. Zabbix reports problems at different levels: a first warning is sent if only 20% disc space is left, a next higher level warning is sent if there is only 10% disc space left, etc.

2.7 Process implemented for continuous deployment of upgrades

CIP is deployed in a test, acceptance and production environment. The test environment is not accessible from the outside world and is used for internal testing purposes. Once a new version passes all tests, it is deployed on the acceptance environment where others can evaluate the new version. Finally, once the new version has been approved, it can be deployed to the production environment. Deployment to production usually takes place during the Civity's maintenance window on Monday evening. Ansible scripts are available to deploy new versions of the software in the three different environments.

2.8 Data platform usage - tool usage observation

To obtain access to CIP, users need to register (unless they are just experimenting with the API's using the sandbox plans). Subsequently, they have to take out a subscription on an API to be able to use it without restriction. Whether these subscriptions are automatically activated depends on the nature of the API. For some APIs, interested users have to explain why they want to access the API. It is then up to the administrator to grant or deny access. See Table 1 for an overview.

Table 1 - Type of access to the APIs in Utrecht CIP

API	Access granted automatically
Parking spots	Yes
EV Charging stations	No
Energy consumption	Yes
Battery usage	To be decided (probably no)

At the time of writing, a challenge is being published to invite interested parties to develop applications based on the IRIS data services. This will provide us with usage statistics of the different services.

3 Status of the CIP in Nice

3.1 Introduction

The Nice Côte d'Azur Métropole has relied on previous WP4 deliverables, particularly D4.4 and D4.5 to deploy and configure the CIP, which will serve as a platform to support the IRIS demonstrators. In order to support the demonstrators developed in Nice, we focused on the main components of the CIP architecture in order to allow:

- Recording, updating and consulting data produced by city sensors and technical systems and components;
- The historicisation of these same data;
- API management.

When setting up the platform, we paid particular attention to availability, scalability, reliability and security.

The NCA CIP will be deployed in the 3 data centres owned and managed by the NCA Directorate of Information Systems (DSI).

The hardware infrastructure is based on bare metal servers (Cisco and a few remaining HP). The servers are blades in enclosures, 3 enclosures per cabinet. The machines are managed with vSphere and VMWare ESX as virtualisation layer, vCenter, vMotion. There is a redundant SAN (EMC), dispatched in main and secondary DC. It also exists a redundant NAS system for low I/O file storage.

The different components will run on virtual machines and be deployed on the 3 data centres to guarantee availability, scalability and reliability. Any of the components will be scalable by adding some instances of containers to scale up the service.

3.2 Basic component update

At the beginning of the IRIS project, the version of the Orion context broker deployed was version 2.0.0 (Figure 11). This version was the one mentioned as stable by the FIWARE community. Since then, new versions of the Orion context broker have been released by the FIWARE community. They often bring new features but also fix bugs. That's why we decided to update this component, which represents the heart of the platform. In order to perform the update, we used the procedure described in the chapter "Process implemented for continues deployment of upgrades".

Following this procedure, we have updated the Orion context broker from version 2.0.0 to version 2.3.0 (Figure 12). Keeping the various components up to date is necessary because it makes it possible to correct bugs and potential security vulnerabilities beyond the new features.


```
{
  "orion": {
    "version": "2.0.0",
    "uptime": "0 d, 7 h, 11 m, 22 s",
```

```
    "git_hash": "485128e135f4225040841f5ab3b85d42cfe68f55",
    "compile_time": "Fri Sep 28 09:56:56 UTC 2018",
    "compiled_by": "root",
    "compiled_in": "4852e5bea506",
    "release_date": "Fri Sep 28 09:56:56 UTC 2018",
    "doc": "https://fiware-orion.readthedocs.org/en/2.0.0/"
  }
}
```


Figure 11 - Orion version before upgrading


```
{
  "orion": {
    "version": "2.3.0",
    "uptime": "300 d, 22 h, 45 m, 29 s",
    "git_hash": "764f44bff1e73f819d4e0ac52e878272c375d322",
    "compile_time": "Tue Nov 5 09:38:37 UTC 2019",
    "compiled_by": "root",
    "compiled_in": "38ab37448d3a",
    "release_date": "Tue Nov 5 09:38:37 UTC 2019",
    "doc": "https://fiware-orion.rtfd.io/en/2.3.0/"
  }
}
```


Figure 12 - Orion version after upgrading

3.3 Final architecture

The platform's scalability using containers allows to virtually support any number of concurrent users and devices by deploying more instances of services, providing the provisioning of enough computing nodes as virtual machines or bare metal servers to absorb the load. By starting with a simple configuration on 3 nodes, we will evaluate performance through some simple use case scenarios, with a monitoring system that stores history. We will use the public DMZ (Demilitarized Zone) and the private DMZ to distribute the different components to ensure security (Figure 13).

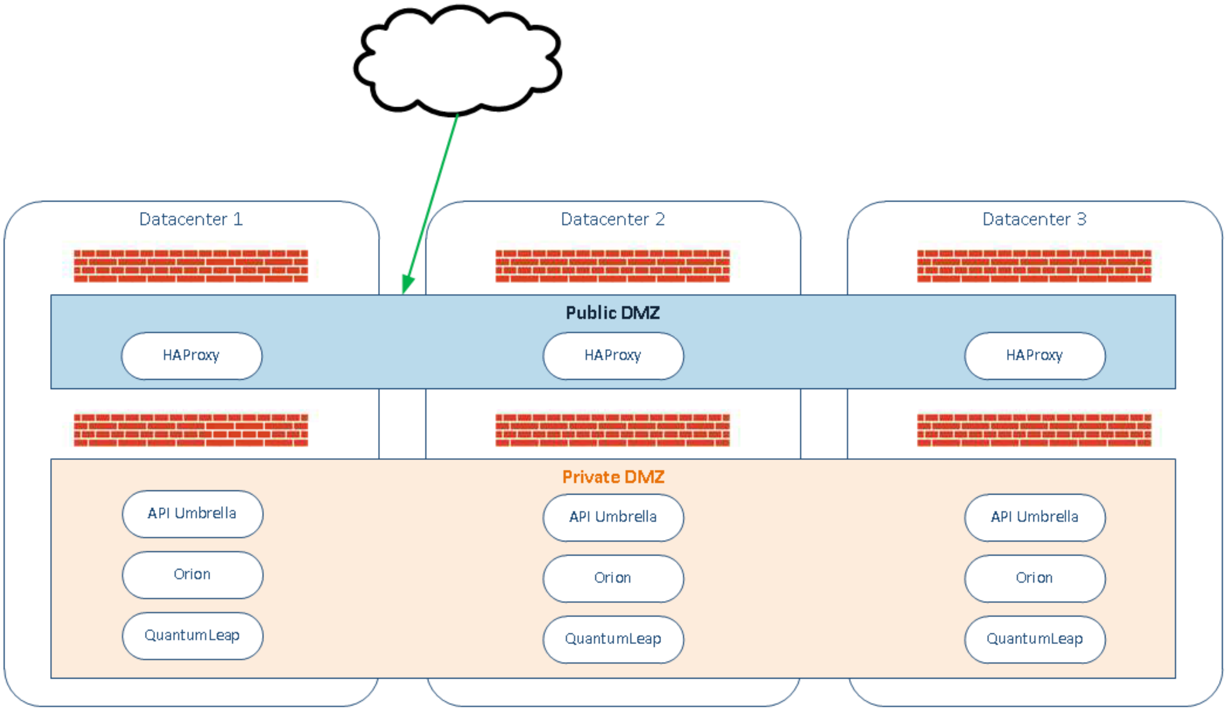


Figure 13 - NCA Architecture

3.4 Data models and APIs

Many data models are used to support the different demonstrators. They are all from the FIWARE Foundation except for two of them that was specifically developed. The description of these datasets is available by following the link in each API:

- Air Quality Observation: observation of air quality conditions at a certain place and time
- Battery Storage: measurement of the remaining energy capacity in a battery
- District Heating And Cooling: measurement of the energy of a district heating and cooling
- EV Charging Station: public charging station supplying energy to electric vehicles
- KPI: Key Performance Indicator
- Noise Level Observation: observation of those acoustic parameters that estimate noise pressure levels at a certain place and time
- Parking Offstreet: Off-street parking
- Photovoltaic: Measurements of a photovoltaic panel
- Three Phase AC Measurement: measurement of the electrical energy consumed by an electrical system that uses Alternating Current
- Traffic Flow Observation: observation of traffic flow conditions at a certain place and time
- Water Observation: represent the parameters of flow, level and volume of water observed
- Weather Observation: observation of weather conditions at a certain place and time

3.5 Data Market

The data is made available via an API management component (Umbrella API). API Umbrella is an open-source API management platform for web services. The FIWARE API structure integrates the Umbrella API as a basic proxy technology to access FIWARE APIs. Once the data models have been identified, we can configure the API Management to create the different APIs necessary for IRIS solutions (Figure 14, Figure 15).

Now we can use APIs to send the data to the CIP, and consumers periodically can easily use the APIs and the standardised models to consume the CIP data.

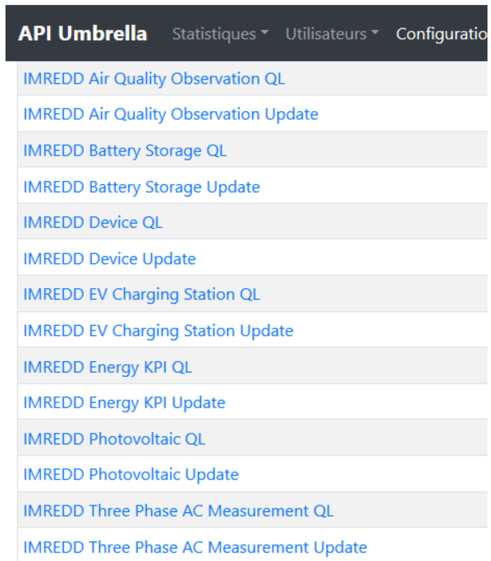
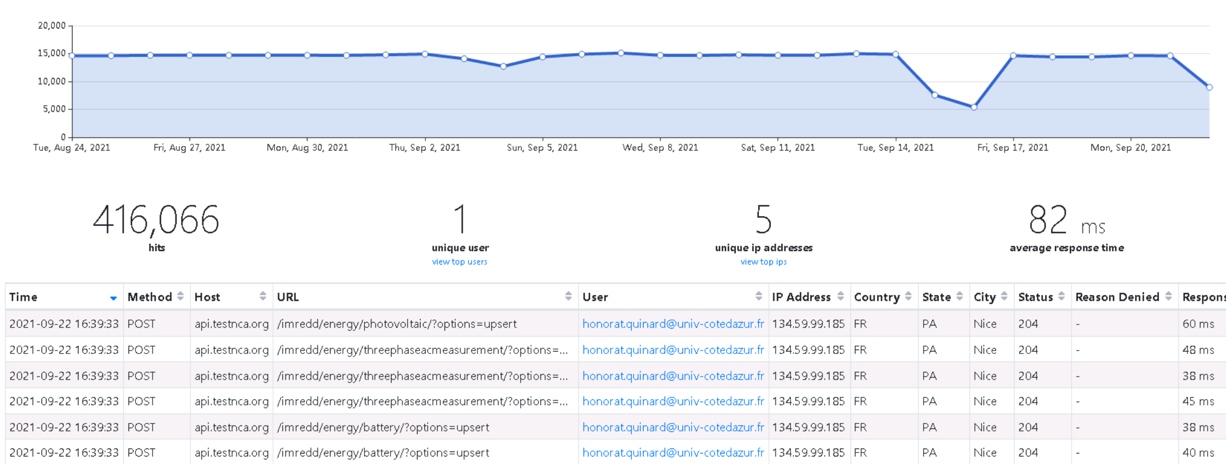


Figure 14 - NCA APIs configuration



Time	Method	Host	URL	User	IP Address	Country	State	City	Status	Reason Denied	Respons
2021-09-22 16:39:33	POST	api.testnca.org	/imredd/energy/photovoltaic/?options=upsert	honorat.quinard@univ-cotedazur.fr	134.59.99.185	FR	PA	Nice	204	-	60 ms
2021-09-22 16:39:33	POST	api.testnca.org	/imredd/energy/threephaseacmeasurement/?options=...	honorat.quinard@univ-cotedazur.fr	134.59.99.185	FR	PA	Nice	204	-	48 ms
2021-09-22 16:39:33	POST	api.testnca.org	/imredd/energy/threephaseacmeasurement/?options=...	honorat.quinard@univ-cotedazur.fr	134.59.99.185	FR	PA	Nice	204	-	38 ms
2021-09-22 16:39:33	POST	api.testnca.org	/imredd/energy/threephaseacmeasurement/?options=...	honorat.quinard@univ-cotedazur.fr	134.59.99.185	FR	PA	Nice	204	-	45 ms
2021-09-22 16:39:33	POST	api.testnca.org	/imredd/energy/battery/?options=upsert	honorat.quinard@univ-cotedazur.fr	134.59.99.185	FR	PA	Nice	204	-	38 ms
2021-09-22 16:39:33	POST	api.testnca.org	/imredd/energy/battery/?options=upsert	honorat.quinard@univ-cotedazur.fr	134.59.99.185	FR	PA	Nice	204	-	40 ms

Figure 15 - NCA API Umbrella (API usage)

The Management portal accessible via https://iot.nicecotedazur.org/apis (Figure 16) allows users to obtain the necessary information to use APIs in their projects.

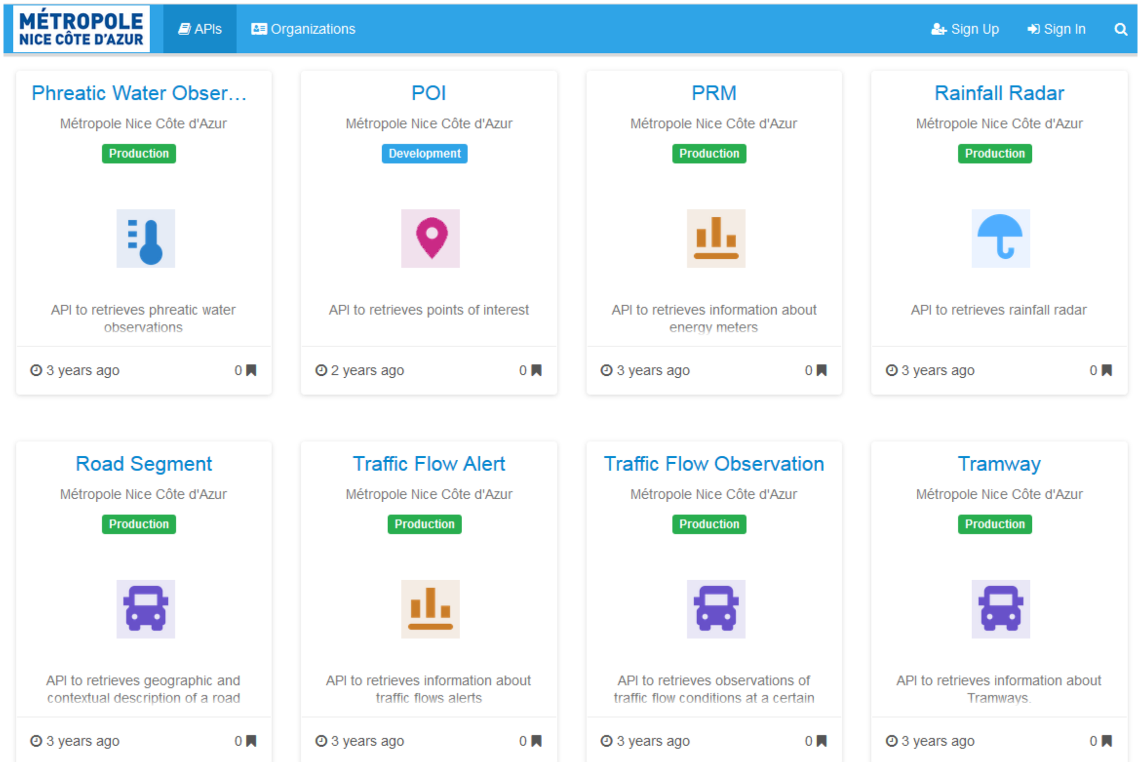


Figure 16 - NCA API Portal

The documentation associated with each of the APIs allows presenting the data models used and enables developers to get started with the API.

3.6 Use cases update

CIP datasets have different uses:

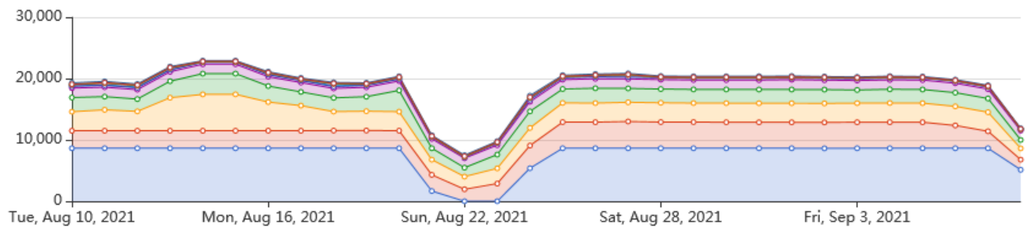
- Use of datasets to produce a new one (use of air quality data, weather observation data to create new air quality data);
- Presentation of data on a dashboard (energy management of the IMREDD building).

3.7 Tools implemented to monitor health and performance of CIP

CIP data are accessible through APIs. To be able to use them, an authorisation key is required (api_key). For this we use the FIWARE component (API Umbrella), which allows the management of APIs. API management is the process of creating and publishing web APIs, enforcing their usage policies, controlling access, collecting and analysing usage statistics, and reporting on performance, but also nurturing the subscriber community to make it grow. API Umbrella is the core API manager, and it provides these functionalities: API Keys, Rate limiting, Analytics, Caching. In addition, it allows unifying the APIs we need to manage.

Analytics / Health performance monitoring

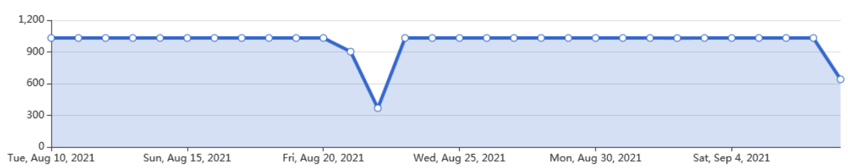
API Umbrella allows understanding how API is being used with rich analytics about API requests. For example, viewing high-level summary data or drilling down into the specifics with a flexible analytics querying interface in the admin tool (Figure 17, Figure 18).



All Hosts / api.nicecotedazur.org / nca

Path
api.nicecotedazur.org/nca/metric/
api.nicecotedazur.org/nca/parking/
api.nicecotedazur.org/nca/environment/
api.nicecotedazur.org/nca/weather/
api.nicecotedazur.org/nca/energy/
api.nicecotedazur.org/nca/lora/
api.nicecotedazur.org/nca/traffic/
api.nicecotedazur.org/nca/device/
api.nicecotedazur.org/nca/transportation/
api.nicecotedazur.org/nca/carsharing/

Figure 17 – 1st example of API Umbrella Analytics



29,775 hits	1 unique user (view top users)	1 unique ip address (view top ips)	386 ms average response time

Time	Method	Host	URL	User	IP Address	Country	Status	Response Time
2021-09-08 16:50:03	GET	api.nicecotedazur.org	/nca/weather/observation/histo/WeatherObserved:ASS...	julien.:	81.200.43.74	FR	200	142 ms
2021-09-08 16:50:03	GET	api.nicecotedazur.org	/nca/weather/observation/histo/WeatherObserved:ASS...	julien.:	81.200.43.74	FR	200	173 ms
2021-09-08 16:50:02	GET	api.nicecotedazur.org	/nca/environment/water/observation/histo/WaterObse...	julien.:	81.200.43.74	FR	200	154 ms
2021-09-08 16:50:02	GET	api.nicecotedazur.org	/nca/environment/water/observation/histo/WaterObse...	julien.:	81.200.43.74	FR	200	113 ms
2021-09-08 16:44:03	GET	api.nicecotedazur.org	/nca/weather/observation/histo/WeatherObserved:ASS...	julien.:	81.200.43.74	FR	200	147 ms
2021-09-08 16:44:02	GET	api.nicecotedazur.org	/nca/weather/observation/histo/WeatherObserved:ASS...	julien.:	81.200.43.74	FR	200	166 ms
2021-09-08 16:44:02	GET	api.nicecotedazur.org	/nca/environment/water/observation/histo/WaterObse...	julien.:	81.200.43.74	FR	200	118 ms

Figure 18 – 2nd example of API Umbrella Analytics

3.8 Process implemented for continuous deployment of upgrades

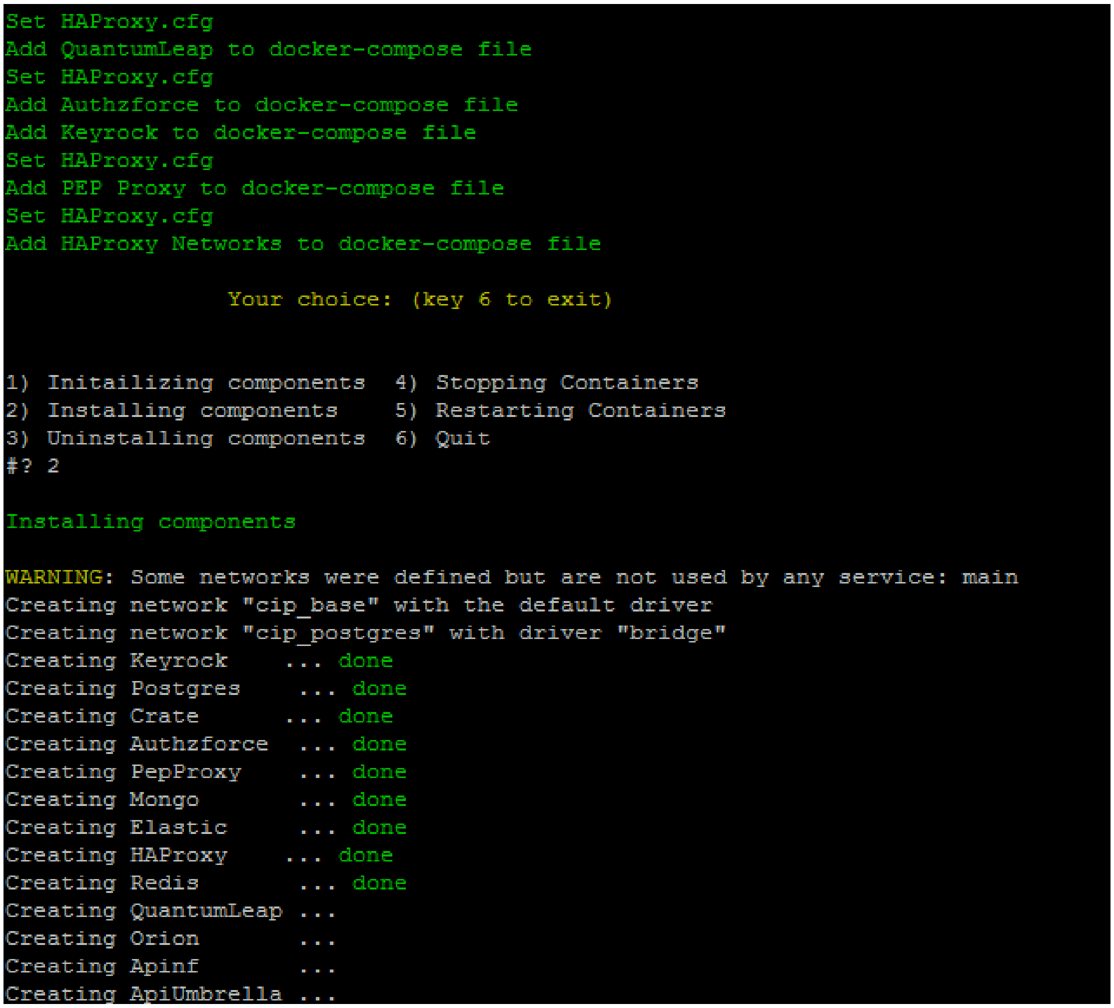
The Nice Côte d'Azur Metropolis has three identical environments in terms of components, servers and networks. The first, called the test environment, allows teams to test new components but also to test new versions of existing components. At the end of this test phase, a procedure and a package are delivered to the teams in charge of the validation and acceptance of the delivery. To do this, the team uses a recipe environment that allows it not only to validate the delivery but also to validate that there are no bugs or regressions following this update. At the end of this validation, if the delivery is accepted, the team uses the production environment to set up the new functionality or the new update.

The update of the components is carried out by modifying the configuration file "settings.json". The file contains an "images" parameter, which allows defining the version of the components. When you want to update the version of a component, you just have to change the reference of the image used ().

```
"images": {
   "apinf_image": "apinf/platform:0.54.0",
   "apiumbrella_image": "nrel/api-umbrella:0.15.1",
   "authzforce_image": "fiware/authzforce-ce-server:release-8.1.0",
   "bizecosystemapis_image": "fiware/biz-ecosystem-apis:v7.6.0",
   "bizecosystemcharging_image": "fiware/biz-ecosystem-charging-backend:v7.6.1",
   "bizecosystemlogicProxy_image": "fiware/biz-ecosystem-logic-proxy:v7.6.1",
   "bizecosystemrss_image": "fiware/biz-ecosystem-rss:v7.6.0",
   "comet_image": "fiware/sth-comet:2.4.0",
   "crate_image": "library/crate:3.0.5",
   "cygnus_image": "fiware/cygnus-ngsi:1.9.0",
   "elasticsearch_image": "elasticsearch:2.4",
   "grafana_image": "grafana/grafana:5.4.0",
   "haproxy_image": "haproxy:1.7.11",
   "keyrock_image": "fiware/idm:7.6.0",
   "iotagentJSON_image": "fiware/iotagent-json:latest",
   "iotagentUL_image": "fiware/iotagent-ul:latest",
   "mongo_image": "mongo:3.4.17",
   "mysql_image": "mysql:5.7",
   "orion_image": "fiware/orion:2.3.0",
   "pepProxy_image": " fiware/pep-proxy:7.5.1",
   "perseocore_image": "fiware/perseo-core",
   "perseofe_image": "fiware/perseo",
   "postgres_image": "postgres:10.2",
   "quantumleap_image": "smartsdk/quantumleap:0.5",
   "redis_image": "redis:latest"
}
```

Figure 19 – An example of a component's settings file

To start the upgrade procedure, it is necessary to execute the following script "./cip/setup.sh" and choose the second option: "Installation of CIP components".



```
Set HAProxy.cfg
Add QuantumLeap to docker-compose file
Set HAProxy.cfg
Add Authzforce to docker-compose file
Add Keyrock to docker-compose file
Set HAProxy.cfg
Add PEP Proxy to docker-compose file
Set HAProxy.cfg
Add HAProxy Networks to docker-compose file

                Your choice: (key 6 to exit)


1) Initailizing components  4) Stopping Containers
2) Installing components    5) Restarting Containers
3) Uninstalling components  6) Quit
#? 2

Installing components

WARNING: Some networks were defined but are not used by any service: main
Creating network "cip_base" with the default driver
Creating network "cip_postgres" with driver "bridge"
Creating Keyrock    ... done
Creating Postgres    ... done
Creating Crate      ... done
Creating Authzforce  ... done
Creating PepProxy    ... done
Creating Mongo       ... done
Creating Elastic     ... done
Creating HAProxy    ... done
Creating Redis       ... done
Creating QuantumLeap ...
Creating Orion       ...
Creating Apinf       ...
Creating ApiUmbrella ...
```

Figure 20 – An example of a component's upgrade procedure

3.9 Data platform usage - tool usage observation

CIP data are accessible through APIs. In order to be able to use them, an authorisation key is required (api_key). Users must request an API key by filling out a form and specifying the use they want to make of it. If the request is accepted and after signing an agreement, they receive by email their key that they can use immediately.

Table 2 - Availability of aggreements for NCA APIs

API	Agreement
Air Quality Observation	Yes
Battery Storage	Yes
District Heating And Cooling	Yes
EV Charging Station	Yes
KPI	Yes
Noise Level Observation	Yes
Parking Off-street	Yes

Photovoltaic	Yes
Three Phase AC Measurement	Yes
Traffic Flow Observation	Yes
Water Observation	Yes
Weather Observation	Yes

4 Status of the CIP in Gothenburg

4.1 Introduction

Gothenburg aspires to become a Smart City, where data from different domains can be combined to create a more vivid picture of a city. This will create new benefits for citizens, as it is the prerequisite for data-driven innovation/AI. Data sharing is central. To achieve data sharing for a Smart City, some key components have been identified: a City Information Model (CIM), which is the data model of the City data and a City Information/Innovation Platform(CIP) which is the platform used to share the data.

The earlier work done by different organisations within the City on CIM and CIP has led to a centralised project being initiated to establish a common service for CIP for all of the City. The project to establish the CIP as a service was started in the being of 2021 and is expected to have the first part of the service CIP in place during 2022. Work on CIP is expected to continue in different phases until 2025.

The CIM and the CIP of Gothenburg should not only handle static data in the form of 3D models of buildings, roads, schools, bridges, etc., but they should also be able to deal with real-time information from sensors used to measure temperature, traffic, soil moisture, bathing temperature, indoor climate, air quality, etc. Within the different areas of responsibility, the City of Gothenburg wants to create a living digital representation of the city. Therefore, a lot of data or just high-quality data within a sector is not enough. The data also needs to be in a format that makes it possible for the different systems to "talk" to each other and be scalable.

To better understand the challenges with sensor data, the City has been doing tests with technical connectivity platforms, particularly LoRaWan and 5G, to collect sensor data. For Gothenburg, the CIP will be fed with data from multiple data sources, where the first will be our IoT platform. CIP will make data available to more consumers than the original sensor/IoT solution. In other words, Gothenburg will create interoperability through CIP, maximise the value of our data by making it available to multiple consumers, make it possible to combine data with 3D models and create the prerequisites for data-driven innovation with AI, BI, etc. with standardised data models where data is interconnected. This is also an important part of the City of Gothenburg's Open Data initiative.

To meet this challenge, Gothenburg is working in both a top-down and bottom-up approach. The top-down approach is to define the requirements for CIP as a service, which will be built up of microservices, and specify how the microservices will be deployed supported and paid for. The bottom-up approach looks at different technologies to see which ones are the most capable and available for the city. In practice, these mean the city needs to test different technologies to understand better how these technologies can be used to deliver the desired service described in the requirements in the top-down approach.

The first microservices that were implemented were soil moisture for green areas and bathing water temperature services. These have come a long way and was published during 2022. A third service is in progress, i.e. Fresh water to private houses. They are described as use cases in more detail in chapter 4.3 in this document.

4.2 Gothenburg CIP Architecture

Establishing the desired service of CIM and CIP, along with the required support organisations, is expected to take several years. This is due to the complexity of the task and the size of the City with all its different organisations. We will be working in an agile manner to implement CIP, and because of this, there will not be a common detailed architecture for Gothenburg CIP in place from the beginning. This will be developed over time.

In the last few years, Gothenburg city has been involved in several projects to test and evaluate the CIM and CIP concept according to the FIWARE principles and has acquired knowledge from projects done in cities like Utrecht. The IRIS project has provided a lot of valuable input. The use cases with sensor data in chapter 4.3 are the latest input. Another example is the work done in IRIS task 7.6 when implementing the CIM pilot, where the sharing of building project data (BIM) was the focus. The architecture used for the CIM pilot is described in detail in the IRIS deliverables D4.6 and D7.8. In the first phase of IRIS, Gothenburg also did a simple POC (Proof of Concept) on sharing parking data using FIWARE technology.

The following projects are other examples of tests that have contributed to the CIP knowledge base in Gothenburg:

- **Nordic Way 2 project**: In this project, the City of Gothenburg tested sharing of dynamic environmental zones using FIWARE technology.
- **LoVIoT project**: the focus for this project was surveillance of air and water from an environmental perspective. Among other things, information about air quality and the temperature was measured and collected. The data was shared using a platform based on FIWARE technologies.
- **SCOREwater project**: The focus of this project was that several cities would use the same technology and format (e.g. FIWARE) to acquire and publish environmental sensor data, such as air quality.

The City of Gothenburg has developed an overview of the architechture (Figure 21) that will be used for the CIP as a service which is shown below. This architecture will be used for different applications and be applied over several years. This will most likely be a hybrid environment, supporting cloud-based services and microservices and internally hosted services to deal with sensitive data, combined with the implementation of different technologies and connectivity. This is due to the fact that the technology is rapidly changing and the legal requirements for data sharing are not mature.

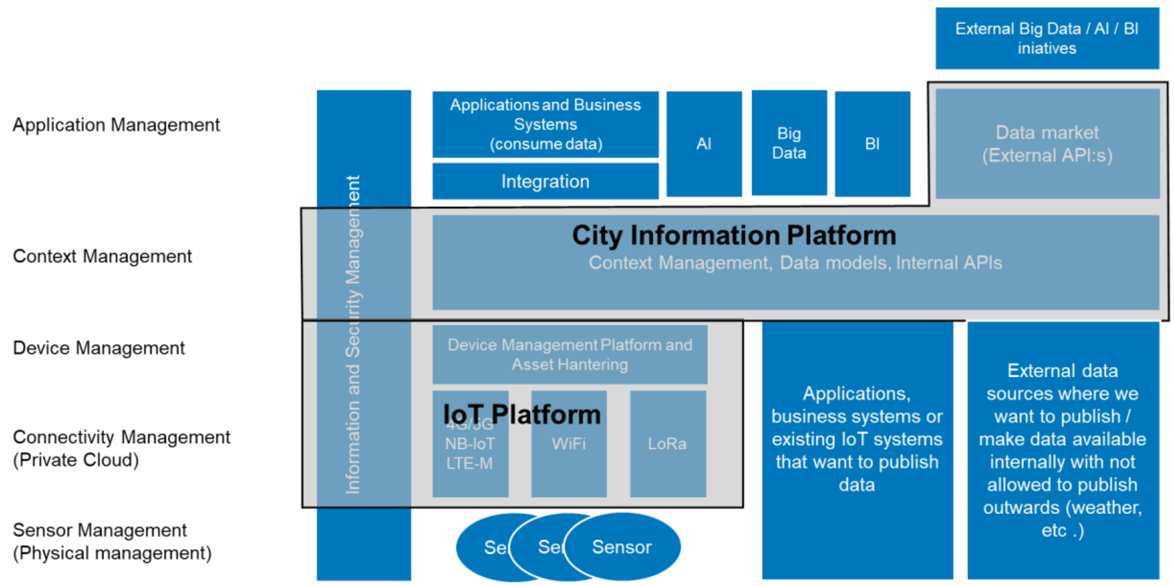


Figure 21 - Gothenburg CIP architecture

4.3 Gothenburg CIP IoT demonstration Use Cases

4.3.1 Overview

Municipal employees have a high workload, but several tasks can be automated using sensor technology. Examples of these are checking soil moisture to know if a green space should be irrigated, whether an adult diaper should be changed, if there is a leak in a municipal building, etc. Unfortunately, today a lot of time and money is spent on these tasks.

Cheap sensors create an opportunity for a municipality to save time and money by moving from scheduled work practices to a needs-driven work practice where sensors can indicate whether maintenance needs to be done or not.

The technology is relatively simple, with battery-powered sensors that communicate wirelessly and last up to 10 years. Wireless communication can include several types of wireless communication, e.g., 5G with NB-IoT/LTE-M, LoRaWAN and WiFi. Using sensors will require new ways of working, changes in contracts and business development to be able to take advantage of the new technology.

In addition, data is generated in quantities that have never existed before. Data can allow us to act in a more proactive way. There are new business models and working methods for many businesses, from inspection tasks performed by people to sensors that monitor things all the time.

Several organisations in the City of Gothenburg are already up and running with this concept and have purchased platforms to be able to take advantage of sensors, but this is in silos. This hinders the data to be used by others and could also lead to vendor lock-in. Because of this, the City and its organisations believe that an IoT Platform should be offered as a service within the city along with associated expertise instead having to procure systems and consultants where a lock-in effect can occur.

To move on, the City needed some valuable use cases, and the Smart Sustainable Parks and Bathing Water Temperature use cases, described below, were identified.

A problem that was identified early was the need for Scalable IT Infrastructure that can deal with large amounts of data in a controlled and secure manner. Gothenburg has to a certain extent, an ability to deal with scalable IT infrastructure today with virtualisation of servers but has now realised that knowledge within Docker/Container/Kubernetes IT infrastructure is a requirement when scaling up data sharing and creating a CIP. Technology such as FIWARE is built upon Docker/Container concepts. This meant we needed to involve our IT production department and address this knowledge gap. We needed to look at how this technology can be supported in production, so it was decided to do some "bottom-up" tests together with the IT-production department to understand the new demands CIP as a service will have on IT production and support.

4.3.2 Smart Sustainable Parks Use Case

If trees or green areas are not watered enough, plants will die, and the time and money used to grow these plants will go to waste. This could also happen if the green areas are given too much water. Besides this, the municipality wastes freshwater, which is a valuable resource and costs money. Today, the maintenance of green areas and parks are based on time schedules. If cheap sensors can indicate when watering is needed, money can be saved by using the data to work on a "need" basis and leaving a scheduled work practice. Predictive maintenance can be made, which will make it possible to plan routes adjusted to when maintenance is needed. In combination with other data, such as soil type, temperatures, and weather information, an application can be developed that suggests the best-suited plants for a specific plantation or park.

These types of sensor-based systems used to maintain green areas exist and can be purchased today. But historically, when purchasing a sensor-based system, this has often led to a "silo" system where the data is only used for one purpose and is not easily shared. This will also lead to vendor lock-in, as the municipality might find themselves trapped having to keep on buying more sensors that can only be used in the "silo system". If the data from the sensors is instead shared in an interoperable way, via CIP, data from many different sensors can be consumed by the same application.

The data could then also be used for other purposes. For example, building projects could use the soil moisture data to see if their construction will affect the water level and the plants/parks in a negative way. If plants are killed because of construction work, the construction companies could be forced to pay a heavy fine, so they are interested in not negatively affecting surrounding green areas. Another interesting application, which could use the soil moisture data, could be a warning application for flooding.

4.3.3 Bathing Water Temperature Use Case

Swedish people love to swim and are highly interested in the bathing temperatures of the water. Gothenburg has many places where it is possible to swim due to its closeness to the sea and many nice lakes that are well suited to swimming. The interest in bathing temperatures is rising as there is an ongoing trend for winter bathing. Swimming, in general, is an activity that the City believes will positively affect the citizens' wellbeing, thus something the City wants to support. Also, lovely places to swim makes Gothenburg an attractive city to live in and visit. Before this project, bathing water temperatures were measured and registered manually several times a day during the summer months. The temperatures are provided to the citizens through an app and through the city homepage. This is a highly appreciated service.

Using sensors to measure the temperatures and automatically get the temperatures registered through digital techniques, the manual labour and the costs are expected to be much less. Furthermore, the temperatures can be measured more often and all year without any additional effort. These new possibilities and increasing demand for winter bathing have led to the decision to start measuring and publishing bathing water temperatures all year round, using sensors and the CIP to share the data. The service will also be increased so that the temperatures are measured more frequently.

If the bathing temperatures are shared through the CIP, they can be used in other ways than just being visible on the homepage and app of the City of Gothenburg. For example, the bathing temperature information is requested by wellness/swimming/sports associations that want to publish this information on their own homepages. The Swedish Agency for Marine and Water management could also use this data to calibrate their data models to predict water temperatures. Other uses for the bathing water temperatures could be input to research done by environmental or climate scientists. If the sensors are also upgraded to measure pollutions, for instance, this could be very interesting.

4.3.4 Fresh water to private houses

The Municipality have started to use the CIP and IoT platforms to retrieve and publish information received from fresh water smart meters. This information is not only used to bill the customer but also indicate if there is leakage, risk for freezing or water is running in the wrong direction. These functions together with metering, can not only give citizens better control over the water usage but also identify leaks at an early stage. Today roughly 20% of all produced fresh water does not reach the consumer.

4.4 Technical specifications (hardware & software)

The data to be measured by sensors and shared for the two use cases are soil moisture level and water temperature.

4.4.1 Connectivity

The City of Gothenburg has identified that there are different types of connectivity with distinct advantages and disadvantages. The decision is to offer three different types:

- LoRaWAN
- NB-IoT (Narrow Band IoT)/LTE-M through 5G
- WIFI

The City aims to support the best selection of sensors available on the market. LoRaWAN is well established in Sweden today. NB-IoT/LTE-M will be further enabled through the ongoing roll-out of 5G, where the plan is that Gothenburg will have 5G coverage by the end of the year. 5G enables massive connectivity, with over a million sensors per square kilometre. LoRaWAN and WIFI have lower theoretical roofs for the numbers within a physical area. The City of Gothenburg has a well-established WIFI network that covers all municipal buildings and several areas within the city and is considered very stable. Because of that, it was decided to test the other two types of connectivity to get more input.

For the Smart Sustainable Park use case, it was decided to use LTE-M sensors to measure soil moisture and for the bathing water temperature use case, it was decided to use LoRaWAN sensors. When the project was started, the City's telematics supplier could not offer NB-IoT, but this will be available by the end of the year. As it turned out, the LTE-M sensors could not be made to function properly. So new sensors will replace these using LoRaWAN to make it work.

Regarding the 5G LTE-M sensors, to measure soil moisture, work is ongoing, but so far, they have not been made to work in a scalable environment. This is due to several reasons: NB-IoT has existed for several years and several suppliers deliver NB-IoT sensors. The ambition from the start was to use these for soil moisture measurement. Then it was discovered that Gothenburg's supplier of telematics could not deliver NB-IoT. The City had to find a supplier that could supply LTE-M sensors. The supplier of telematics has accelerated the roll-out of NB-IoT, upon request from the City of Gothenburg. That is expected to be finished by the end of the year 2021. Then the City of Gothenburg might want to try these sensors instead. Gothenburg's supplier of telematics does not offer eSIM cards or device management of the sensors, only connectivity and SIM cards. Authentication of SIM cards is done manually today, where each SIM card needs to be entered by hand. Due to security reasons, the sensors need to be segmented into an APN which requires a unique username and password for the APN. Besides this, each sensor has its own IP address. The sensor’s IP address needs to be opened through the firewall between the backend and the APN for the sensor to send its data to the backend system. These tasks are done by hand and need to be automated to deal with large numbers of sensors. On top of this, there is no asset or device management of the sensors in place, which is a prerequisite so that a support organisation can offer support to the business areas. As such, this limits the usage of LTE-M and NB-IoT sensors in a production environment.

The challenges above create a barrier for the city's ambition to roll out a hundred thousand of sensors. The City of Gothenburg has had dialogue with several suppliers of IoT/CIP platforms, to investigate how suppliers handle large-scale rollouts. Not many have more than a couple of thousand sensors. The market for 5G sensors is deemed to be rather immature. Tools and routines are missing for large-scale roll-outs of sensors.

Regarding LoRaWAN, which has existed in Sweden for a while now, there is a much more mature connectivity solution. The connection to the bathing water temperature sensors has been very successful. To continue the tests with soil moisture measurements, we will use LoRaWan instead. The same sensors can be bought with different connectivity, so this should not be a problem.

4.4.2 Technical Architecture

FIWARE is built using Docker container technology. Even though Gothenburg is familiar with virtualisation of servers, the production department was unfamiliar with docker containers. The Docker container concept is fundamental to FIWARE and especially when having scalable IT infrastructures. Scalable IT infrastructure is a prerequisite when dealing with a large amount of data.

The first thing done was to create a Docker container environment together with the production department, which could host the different FIWARE components. This was achieved using Red Hat OpenShift.

Figure 22 below presents a screenshot from the OpenShift environment and the components used for the use cases.

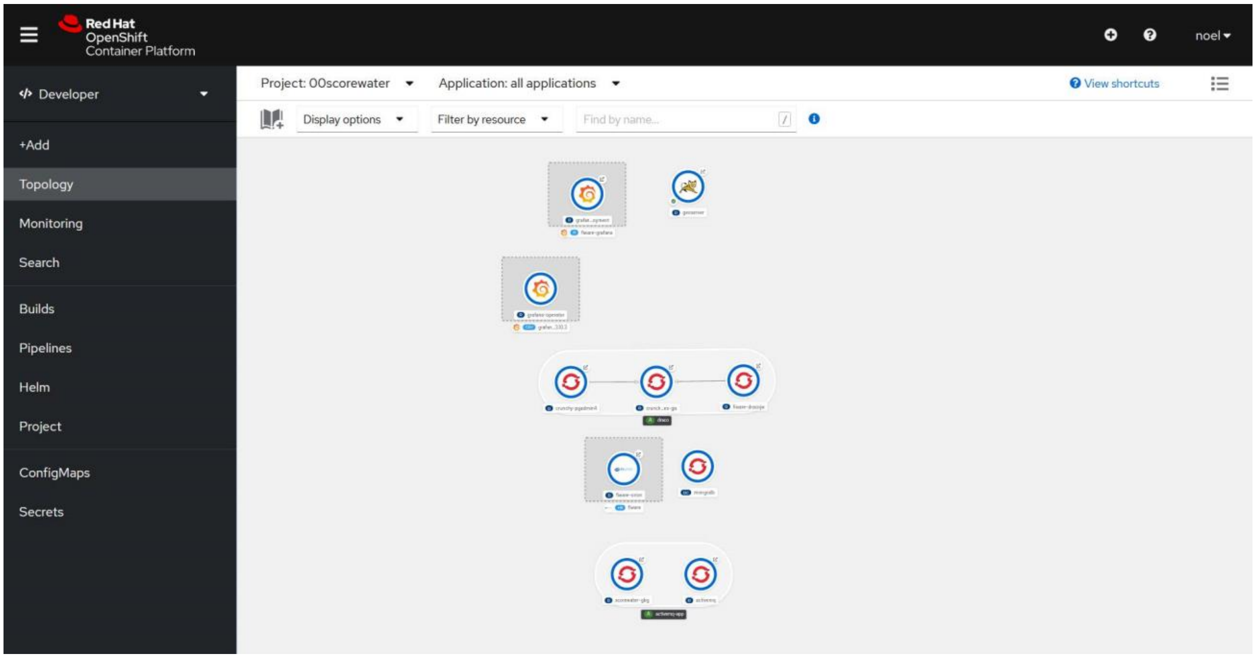


Figure 22 - Red Hat OpenShift environment

Figure 23 below presents the current evolving IT architecture for the test of the use cases. The City considers CIP as both internal data sharing as well as external data sharing. The primary focus when it comes to internal data sharing is to allow for horizontal integration and interoperability, which is the core of the Smart City concept.

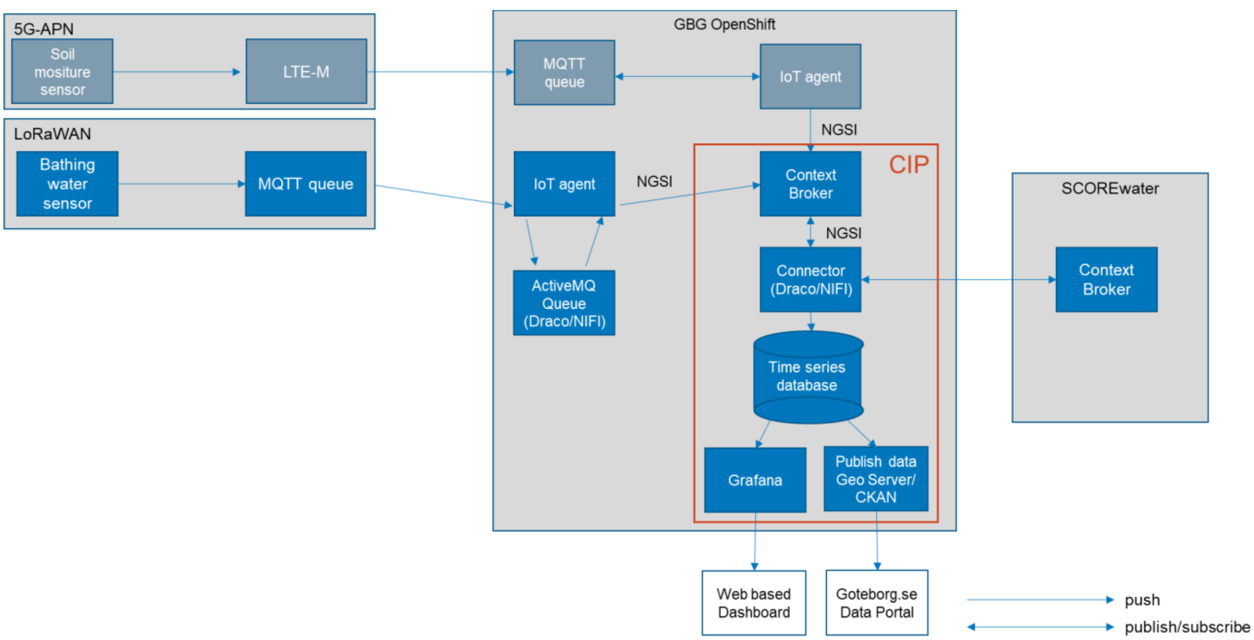


Figure 23 - IT-Architecture for use cases Smart Sustainable Parks and Bathing Temperature

Below is a description of the different components in Figure 23:

MQTT, Message Queue Telemetry Transport, is a Message Queue to handle incoming payloads (hexadecimal strings) from the sensors.

ActiveMQ Queue is used by the IoT Agent to buffer payloads from the sensors to be converted later.

The Iot Agents converts the buffered sensor-specific payload data to a FIWARE Smart City data model. In the case of bathing water temperatures, the hexadecimal payload is converted to a WaterQualityObserved object according to the Smart City data model. This is published to the FIWARE Context Broker using the NGSI API.

The Context Broker (FIWARE) is used for internal data sharing together with the Connector

The Connector (Draco) is used to create persistent data. The Connector connects to the Context Broker and saves the water quality observed data in a PostGIS database (a time-series database). It could also connect to the Context Broker of the SCOREwater project and receive data on air quality that could be published.

The Time Series database is a PostGIS database that saves the data in a time sequence.

The Grafana component is used to visualise the data in the time series database through a dashboard. This could be used by support personnel to monitor the data. See Figure 24 below.

Figure 24 - Graph over bathing water temperatures in the Grafana dashboard

GeoServer/CKAN can be used to publish the data. Currently, only GeoServer is used to publish the data as geodata. In the future CKAN can be used also. In the future the data can be published on Goteborg.se data portal and then be used as described in the use case descriptions.

No end-user applications exist, as the data is not published yet, but as described in the use case description above, there are intended applications that will use the data.

Figure 25 below presents a detailed image of how the six different temperature sensors of the bathing temperature use case communicate with the IoT Agent.

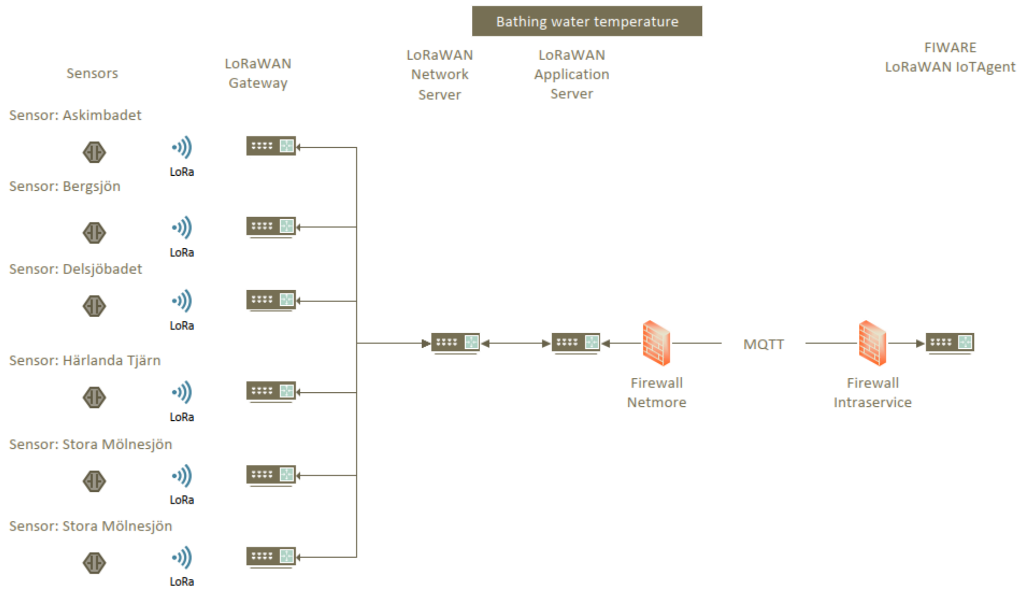


Figure 25 - Bathing temperature sensors way of communicating temperature data with the IoT Agent

4.4.3 Data models

When implementing the use cases, it has been noticed that there are many discussions about which data models are to be used. This could be OneM2M, SAREF, OGC SSN/SOSA, sector specific data models such as healthcare or FIWARE smart city data models, among others. As always, developers are looking for one data model to apply for all domains, but this is not thought to be the correct approach. It is a matter of perspective of whom decides what data model to use. A vertical integration for a specific business area such as healthcare or a horizontal integration to combine data from several different sources such as schools, parks, roads, and healthcare. This is an age-old question of interoperability. In essence, this may mean that an IoTAgent must publish data in more than one format: The first format is specific to the vertical business system, the second format can achieve horizontal integration and interoperability. The second could be a FIWARE smart city data model. An application that consumes data must subscribe to data in at least two formats in the same way.

4.4.4 CIP/IoT platform setup

The Docker container concept of FIWARE is fundamental when having scalable IT infrastructures. To create the Docker container environment using Red Hats OpenShift, support was received from a Red Hat FIWARE technical expert, who is also Red Hat's member of FIWARE technical committee. This was beneficial both to the City of Gothenburg and to Red Hat to learn about the challenges a city has when establishing a FIWARE Docker container environment. Besides this, we also received support from the SCOREwater project in replicating the IoT agent. It became evident that there is a large threshold for a city to work with docker container FIWARE components. This means that despite only taking a few hours to configure the components, it took weeks to understand the Docker container environment of FIWARE in the Open Shift environment. This should not be underestimated. We have also had difficulties acquiring consulting support through the City of Gothenburg's existing IT frame agreements with consultancy companies. Despite the challenges, we believe that this is a very good way to move forward with a CIP. Our ambition is to package the implementation of the bathing water temperature use case in Gothenburg as a microservice using FIWARE and make it available in GITHUB.

The focus of these use cases has been to create a scalable technical infrastructure used to support the use cases. It has not been on implementing end-user applications. That the data can be used in many ways when shared is in essence obvious. What is not obvious and discussed enough is the barriers to creating production environments and the challenges with public tendering and open-source platforms that we have encountered. The question we think that should be asked is why so few CIP projects go into production and why so few cities succeed in establishing a CIP based on FIWARE.

4.5 Governance & Planning of activities

Governance

Intraservice is the department in the City of Gothenburg that is responsible for City wide common IT services for City of Gothenburg. This ranges from laptops, business systems, to mobile phones. A project to establish a City Information Platform as a service for the entire city was nominated by Intraservice and has started during 2021. This is expected to take several years to complete. Park and Nature is the City organisation that maintains the City's parks and green areas. It is the City department responsible for the business areas in the use cases.

Figure 26 below shows the expected governance model for the different areas of responsibility when implementing an IoT platform.

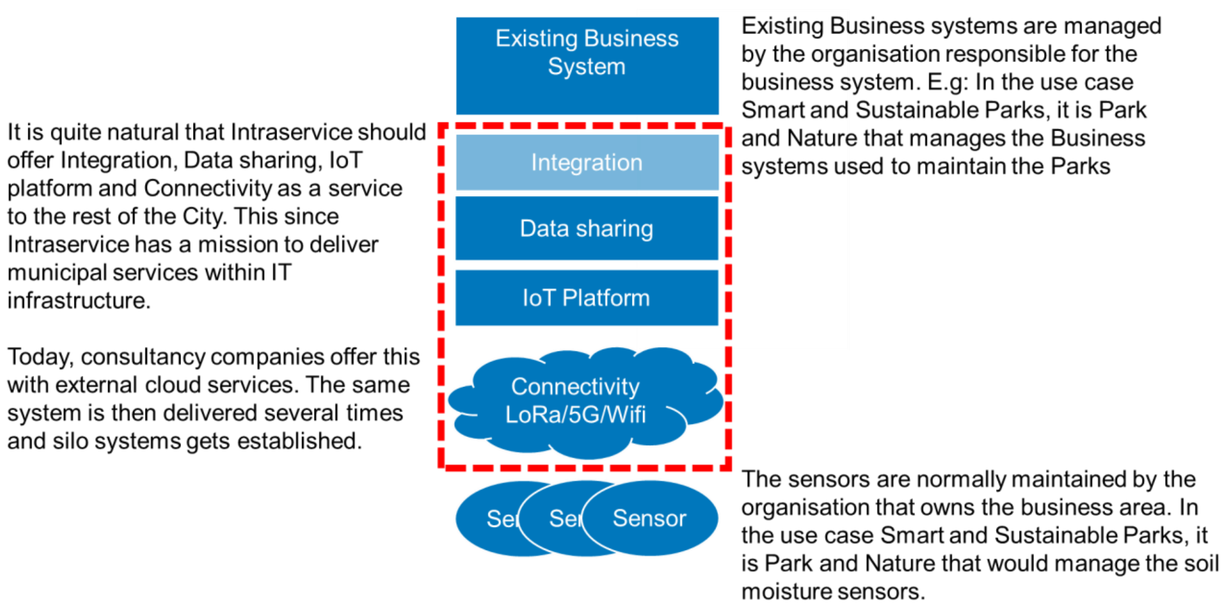


Figure 26 - Governance model IoT platform and use case Smart Sustainable Parks

In the figure, the example from the use case: Smart Sustainable Parks is given. Park and Nature would be responsible for the Existing Business System to maintain parks and the sensors to collect the soil moisture data. Intraservice would be responsible for the Integration, Data Sharing, IoT Platform, and the Connectivity.

The scope of the Intraservice project to establish CIP as a service does not include the organisational requirements or the prerequisites to take advantage of the CIP service. Some examples of these are how an organisation should classify its data, as well as having the resources required to work with the question. Also, when the project to establish the service of CIP was started, it was assumed that each city department or organisation could decide if they wanted to use CIP or not. The City's executive office has now started a project called "Data as a strategic resource" and is expected to establish policies and guidelines for the different authorities and companies within the City in respect to data sharing. CIP is expected to be part of the policies to implement data sharing.

Intraservice has about 40 business cases in the pipeline where a selection of these business cases will be nominated for implementation in city’s CIP during 2023.

Commissioning

There are two perspective of the commissioning. The first which is the City-wide implementation of CIP as a service that will encompass the different activities within the city. The second perspective is the short-term activity to publish the sensor data in the use cases. This second perspective will be incorporated in the commissioning for CIP as a Service.

The time plan for CIP as a service is expected to run until the end of 2025 and will consist of several commissioning steps with the first one expected in 2022. During 2021 and partially in 2022, we are building the requirements specification for CIP as a service. These include not only technical requirement but also the organisational requirements, new roles, and interfaces to existing support and production organisations within the City of Gothenburg.

From the second perspectives point of view, we have built an environment together with the production department during 2021. This includes sensors, connectivity, back-end system including the Context broker, dashboards, and Geo-Servers to publish the data in a docker container environment. From May until now, the sensors have been generating data and the data has been retrieved via an IoT Agent and then published through the Context Broker. In addition, a subscriber of the Context Broker has saved this information to a PostGIS database, which is published internally through the Geo Server. At this moment in time, we are hoping to make the Geo Server available publicly and expect this to happen during September or October.

5 External monitoring of CIP

CERTH, the developer and operator of the IRIS KPI tool, monitors the health and performance of the CIPs that deliver monitoring data automatically. CERTH created monitoring scripts in the Postman API platform[2]. The monitoring scripts perform tests by calling the APIs of the CIPs that provide monitoring data and validate the responses automatically. When one of these tests fail, the script sends a notification by email or through the available integrations with tools like Slack. The monitors are configured to run every hour. A graph or console presents the results of the CIP's responses.

On the graph, we can see how the APIs have performed over time. Each monitor run is represented by a bar in the graph. Below in Figure 27, Figure 28 and Figure 29, every bar shows the time and date that the monitor ran and got a response. The upper section charts the CIP's average response time for each run, while the lower section visualizes the number of failed tests for each run. The user can hover over each run individually to view the exact response time and failed requests' per cent. A green or a red bar appears if we have a succeeded or failed response. Next to the green or red status bars is the number of requests per monitor.

In addition, the user can use filters to identify recurring patterns in the monitoring runs by selecting particular requests. Last but not least, he can navigate through past run results to review what happened at one specific point in time by choosing a particular date.

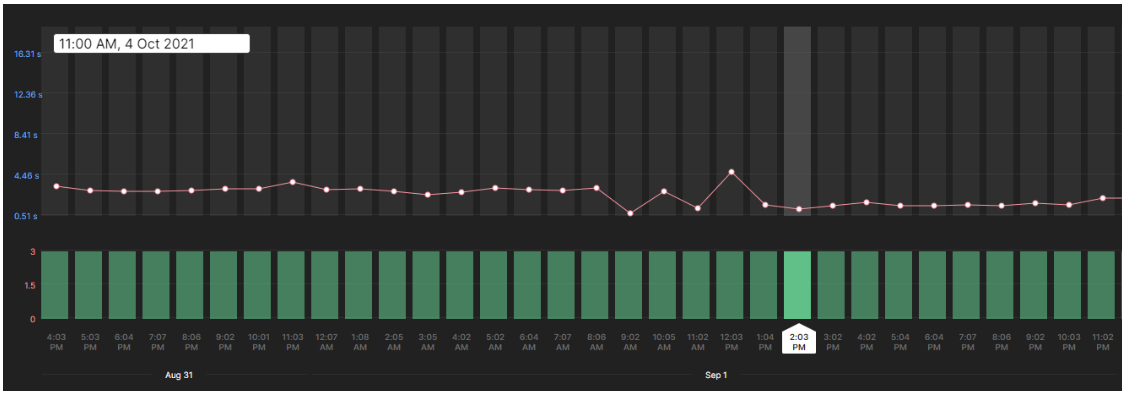


Figure 27 - External monitoring of Utrecht CIP.

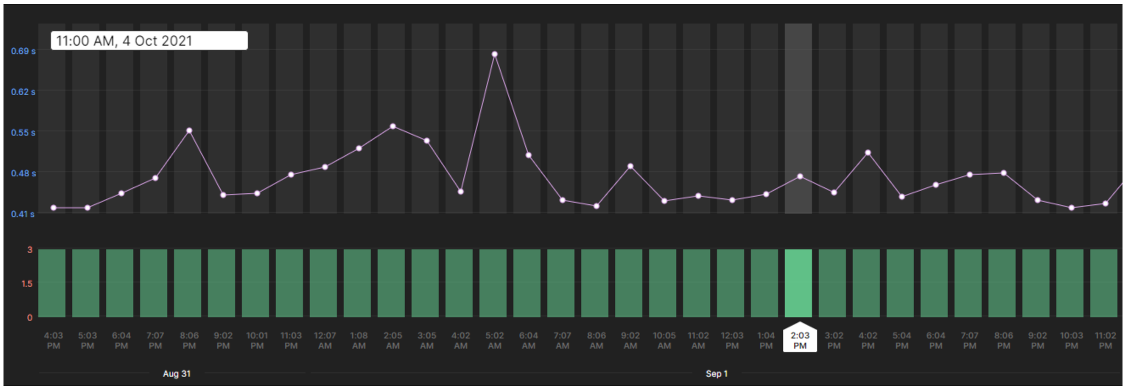


Figure 28 - External monitoring of Nice CIP.

[2] https://www.postman.com/product/what-is-postman/

Figure 29 - External monitoring of all CIPs

6 Lessons Learned

6.1 Lessons learned in Utrecht

The City Innovation platform has been used to create different data services in Utrecht. From these use cases, different lessons have been learned. These can be sub-divided into lessons learned regarding technical, governance and standards aspects.

6.1.1 Technical

The software components of the FIWARE ecosystem required to be able to run an instance of the City Innovation Platform are available and do work. Using the FIWARE ContextBroker, organisations can implement OASC Minimal Interoperability Mechanism (MIM) 1 (Living-in.EU, 2022). The Smart Data Models facilitate MIM2 and the components of the FIWARE/TMForum business API ecosystem support MIM3. Other FIWARE generic enablers or open source alternatives for those generic enablers provide additional functionality to operate a reliable City Innovation Platform.

It is important to set-up proper monitoring of the different systems to identify hick-ups in data streams at an early stage, otherwise problems have to be detected by end-users. As a consequence, only problems in frequently used data streams will be identified (and fixed).

Existing IT infrastructures are both an opportunity and a threat. They already contain components which may be re-used for the City Innovation Platform such as a single sign on solution or a database management system. This reduces the investments needed to get a City Innovation Platform up and running. On the other hand, if the existing IT infrastructure relies on proprietary standards and software, it may incompatible with the open standards and open source software from the FIWARE ecosystem.

6.1.2 Governance

Governance aspects should have received more attention. When setting up a data service, different public and private partners are involved with different stakes. It is important to provide the different parties involved with the confidence they can exercise control over their contribution to the City Innovation Platform. They should be able to act (for instance by withdrawing access to a certain dataset) in case of illegal use. The different public and private partners should constitute a trust network with a governing body such as a municipality which keeps an eye on the network and steps in in case of issues, now and in the future.

6.1.3 Standards

While open standards are by no means perfect, in the City Innovation platform they do serve their purpose. Because of open standards, it is possible to combine data streams from different silos and verticals. And the open standards allow for replication of use cases from one environment to another. Whereas the FIWARE NGSI standard standardizes the API's used and the Smart Data Models harmonize the contents of the dataset, open standards regarding authentication and authorization have not been applied.

6.2 Lessons learned in Nice

The 3Vs (volume, variety and velocity) are three defining properties or dimensions of a City Innovation Platform.

Volume: The amount of data in the data platform is far greater than the volumes usually processed and is constantly growing.

Variety: Data is by definition heterogeneous and often in unstructured formats. This is the reason why they must be standardized.

Velocity: Data is generated in much shorter time frames and it must be possible to collect and use it almost in real time.

The 3Vs are the biggest challenge for the CIP. Its architecture must allow it to respond to the volume and velocity of the data produced. Through its APIs and data models, the CIP must respond to data heterogeneity.

It is therefore necessary to ensure the availability of the CIP, to set up a platform to respond to the load, both the volume of requests but also a number of important requests in the same period of time. For this, the architecture must be able to meet scalability requirements.

But the biggest difficulty is to ensure the quality and availability of the data. Indeed, it happens that the data is no longer transmitted and stored in the CIP. This has the effect of penalizing or even interrupting the service that uses this data. For this, it is necessary to set up monitoring and alert tools to ensure a quality service. This also involves defining a service contract between producers on the one hand and consumers on the other.

In conclusion, moving from a demonstrator to an operational platform is the most important challenge, it requires the implementation of a highly available architecture, supervisory tools but above all the organization both from a technical point of view and the teams in charge of maintaining the platform in operational condition.

6.3 Lessons learned in Gothenburg

6.3.1 Technical

Interoperability – The market is immature and moving quickly. Interoperability allows for a black box approach as well as combining out of the box systems with highly customized existing systems. Interoperability needs to be a central strategy as well as being supplier neutral. NGSI-LD is central in this respect. This is important not only externally but also internally.

IT Architecture – Use IT architecture to separate those who consume data from those who publish data. This allows old and new systems to be combined. This is part of interoperability but a different perspective.

Horizontal/Vertical Integration – Quite often there is good vertical integration within a business area (e.g. transport, healthcare, etc) but due to the fact that a city has many different business area the challenge can be how to achieve a horizontal integration between the different areas. Be aware that the data models used in the vertical don't have to be the same in the horizontal. Think that interoperability must work in both directions (vertical/horizontal).

Open Source - This is a good way step in without being locked into an agreement with a supplier for several years. Should an open source product not be up to the challenge it can be replaced without have to deal with legal issues and agreements. Open Source requires competence and this might need to be procured. Open Source does not mean killing the market it means procuring competence instead of a product. Quite often there are more companies willing to supply competence that a product for a specific area so in some respects this will increase competition if dealt with correctly. This can be a strategy to deal with a market or products that have not reached maturity or to share results with other cities.

Container Technology – Containers and Kubernetes is a great way to create reusable components and share them. This can also offer the capability to host your own services (i.e. private cloud). This technology is central when it comes to microservices such as Fiware. The capability to deal with this technology needs to be acquired and we can say from experience this not an easy step.

6.3.2 Governance and onboarding process

Information classification – To know how to share and have control over data then it needs to be classified. This is a starting point and nothing can be done with the data until this process is completed for a data set. This has been a big problem with amongst other things BIM data as ownership and classification has not been dealt with.

Data sovereignty – Data is the most important raw material – The big question being how to ensure that we own our data (content, format and accessibility) and that the data is handled correctly. Are we locked into a supplier ecosystem and are the citizens data been dealt with in a democratic way?

Federated solutions –Legal issues may require that data is separated and that the which may seem counter intuitive, but the access is through a federated interface with well-defined contracts. Containers can be a good way to achieve federated solutions.

IT Architecture and Competitive Tendering – Use IT architecture to create the prerequisites for competitive tendering of the various parts e.g. use data brokers/middleware (e.g. Context Brokers) to separate those who consume data from those who publish data. Combined with Interoperability this allows vendors to offer solutions to more cities as they know what to expect and in turn should increase competition and reduce lock in to a small number of suppliers.

Business development – This is not an IT issue or a digitization issue. It's business development and someone will have to pay for the service and the CIP. Bring people from the business with from the start, information classification, information owners and ensure the willingness to pay. Often the driving factor is integration, use of sensors and future use of AI/BI.

AI / BI – Many cities wish to benefit from technologies like AI and BI but to do this a city needs a CIP/UDP to have control over the data.

EIF4SCC and Data Spaces – These initiatives as well as those that the IRIS have developed are well on their way to be policy so it can be good to be aware of them and start talking about them internally.

6.3.3 Standards

Standards – There are many standards on the market but the few that can be actively developed and used to collaborate between the cities. Smart City Data models are a good example of how this can be done and many initiatives like OASC, GAIA-X, mm point to these.

Collaborate – Find other municipalities and organizations that are in the same phase and interact with them. IRIS, Open and Agile Smart Cities, Living In EU (living-in.eu) address many of the questions and challenges a city encounters with its CIP/UDP.

7 Conclusions & recommendations

The IRIS City Innovation Platform (CIP) is operational in the three LH cities (Utrecht, Nice Côte d'Azur and Gothenburg) and supports the local demonstrations by collecting, managing and exchanging data. During task T4.6, the three LH cities improved the functionality of the local CIPs by developing additional functionalities to meet the new requirements from the IRIS demonstrations. The improvements are reflected in updating architecture, data models and APIs, data market, and use cases. In addition, the LH cities updated come CIP components based on software bugs found or the availability of new versions of these components. Finally, they implemented tools for monitoring the health and performance of the CIP and processes for continuous deployment of upgrades/updates. In addition, CERTH developed an external tool to monitor the health and performance of the CIPs that deliver monitoring data automatically. T4.6 also supported the knowledge exchange between the LH cities' development teams and the Fellow Cities' IT personnel and the relevant stakeholders.

The two-fold purpose of the IRIS CIP was to provide an infrastructure for collecting monitoring data to calculate the project's Key Performance Indicators, and to support various use cases/data services. Some examples use cases are:

- Monitoring E-Mobility with LoRa network: Gain insight into the (in-efficient) usage of parking bays and charging infrastructure by measuring the illegal use of EV charging bays with parking sensors. (UTR)
- Smart Street Lighting with multi-sensoring: Enhance data-driven district policies aimed at reducing/minimizing problems by using data collected through these multi-sensors on lamp posts. (UTR)
- Fighting Energy Poverty: Give the tenants of housing corporation Bo-Ex control over and/or a better understanding of their energy bills, resulting in reduced energy bills and increased tenants' disposable income. (UTR)
- Sensors data collection in air quality: Upgrade the city's air quality mapping model with hourly forecast and real-time information by using air quality data from sensors combined with traffic data. (NCA)
- BIM/CIM data display: Demonstrate the capacity of the multi-scale BIM and its ability to integrate real-time "hot" data from the CIP at the urban and building scale. (NCA)
- Data control and monitoring for Smart e-mobility: Implement a "Smart Charging" management system that operates EVCIs and electric carsharing fleets over a city while testing various related use cases. (NCA)
- Smart Sustainable Parks: Determine whether a green area needs to be watered. The focus has been to go from timetable-based work practices to a need-based one. (GOT)
- Bathing Water Temperature: Provide real-time information on water temperature in lakes using sensors so people who love to swim to be informed. (GOT)

The design, development, and operation of CIP in three Lighthouse Cities offered many lessons and recommendations. To summarize:

Technical:

- Interoperability needs to be a central strategy, as well as being supplier neutral. It aims for a black-box approach and combines out-of-the-box systems with highly customized existing systems.
- CIP should be compiled using software components from different open-source projects and vendors, allowing for the possibility of re-using existing components and trying out alternatives.
- CIP should ensure compatibility with existing IT infrastructure.
- Open source can increase competition and decrease development time. However, cities to benefit from open source might need to procure the required competence.
- Security aspects require more attention.

Governance and onboarding process

- The development of CIP is more than an IT issue. It is business development, and someone will have to pay for the service and the CIP. So bring people from the business from the start, use information classification, assign information owners and ensure the willingness to pay.
- Stakeholders with potentially competing interests are involved in the CIP. Therefore, the CIP must be able to accommodate all these different requirements.
- Data sovereignty is essential as data is the most valuable raw material. The challenge is to ensure that the data is handled correctly and is accessible and owned by the correct parties. Federated solutions address legal issues by separating data and accessing it through a federated interface with well-defined contracts.
- Competitive Tendering can increase competition by using data brokers/middleware and creating interoperability. This will allow vendors to offer solutions to more cities, reducing lock-in to a small number of suppliers.

Standards

- There are multiple open standards available for use. However, few of them are actively developed and used for collaboration between cities.
- FIWARE offers an extensive collection of generic enablers for CIP functionality.
- FIWARE Smart Data Models allow for the harmonization of APIs and data models and compliance with Minimal Interoperability Mechanisms of the Open and Agile Smart Cities Association.

8 Reference

IRIS Project. (2018). *D4.2 Functional & technical requirements for integrated, interoperable and open solutions, standards and new business models.* Retrieved 9 1, 2021, from https://irissmartcities.eu/sites/default/files/documents/d4.2_functional_and_technical_requirements_for_integrated_interoperable_and_open_solutions_standards_and_new_business_models.pdf

IRIS Project. (2019). *D4.4 Document with technical solution reference architecture for CIPcomponents.* Retrieved 9 1, 2021, from https://irissmartcities.eu/system/files/private/irissmartcities/d4.4_document_with_technical_solution_reference_architecture_for_cip_components_1.2.pdf

IRIS Project. (2019). *D4.5 Implementation and integration of core CIP components.* Retrieved 9 1, 2021, from https://irissmartcities.eu/system/files/private/irissmartcities/d4.5-implementation_and_integration_of_core_cip_components.pdf

IRIS Project. (2020). *D4.6 : Integration of CIP in LH Cities.*

Living-in.EU. (2022). *Technical specifications for the MIMs Plus version 5 final MIMs Plus.* Retrieved from https://living-in.eu/mimsplus

NKL. (2020). OCPI Open Charge Point Interface 2.2.