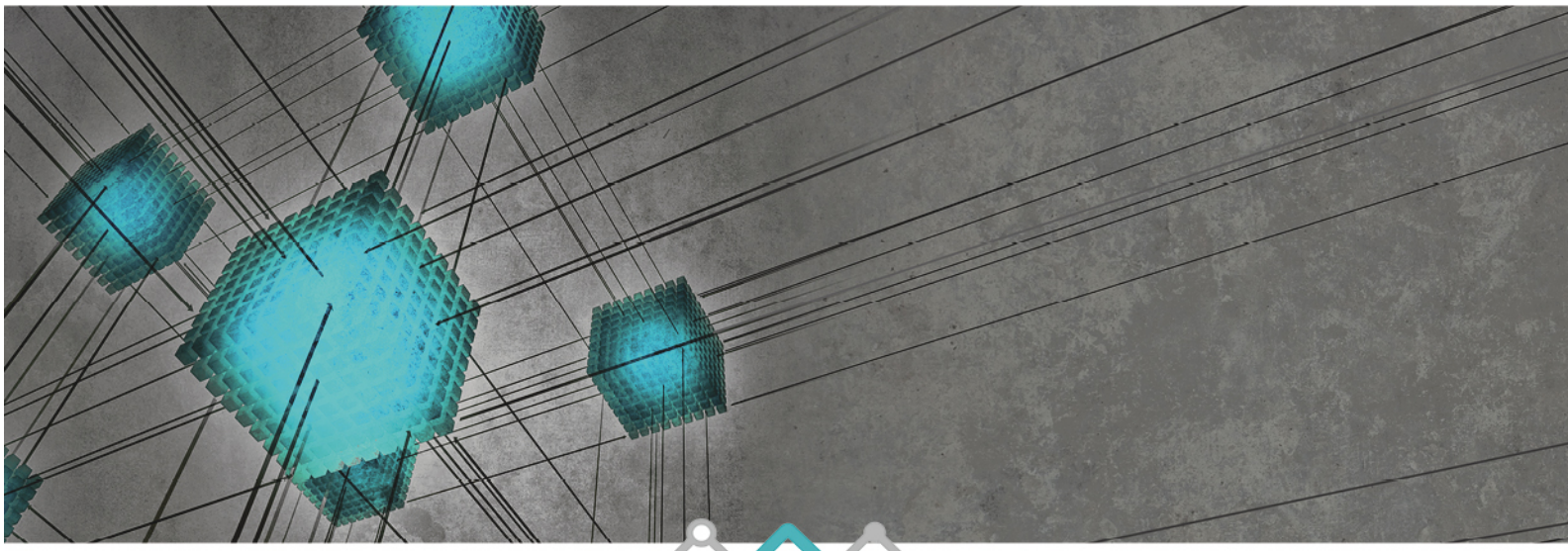

interiot

INTEROPERABILITY
OF HETEREOGENEUS
IOT PLATFORMS.

D4.6

Interoperable IoT Framework API and Tools, Model and Engine v2

June 2018

INTER-IoT

INTER-IoT aim is to design, implement and test a framework that will allow interoperability among different Internet of Things (IoT) platforms.
Most current existing IoT developments are based on ”closed-loop” concepts, focusing on a specific purpose and being isolated from the rest of the world. Integration between heterogeneous elements is usually done at device or network level, and is just limited to data gathering. Our belief is that a multi-layered approach integrating different IoT devices, networks, platforms, services and applications will allow a global continuum of data, infrastructures and services that can enable different IoT scenarios. As well, reuse and integration of existing and future IoT systems will be facilitated, creating a de-facto global ecosystem of interoperable IoT platforms.
In the absence of global IoT standards, the INTER-IoT results will allow any company to design and develop new IoT devices or services, leveraging on the existing ecosystem, and bring them to market quickly.
INTER-IoT has been financed by the Horizon 2020 initiative of the European Commission, contract 687283.

INTER-IoT

Interoperable IoT Framework API and Tools, Model and Engine v2

Version: 1.1
Security: Public
June 31, 2018

The INTER-IoT project has been financed by the Horizon 2020 initiative of the European Commission, contract 687283

Disclaimer

Executive Summary

The aim of Deliverable 4.6, entitled “Interoperable IoT Framework API and Tools, Model and Engine v2”, is to provide accurate documentation of the work done in the last implementation phase of the Interoperability Framework and API Work Package. This deliverable follows previous reports about the reference Metadata Architecture (D4.1, D4.2), design and implementation of the Framework (D4.3) as well as the initial software release (D4.5). From now on, all improvements performed over the components developed in WP4 are tracked through the official INTER-IoT web documentation. Thus, the current deliverable provides information about the final component developments, release features, documentation elaborated, software distribution plan and extensibility of the solutions. It reports the technical work performed in tasks T4.4 and T4.5.

The focus of D4.6 is to report about implemented features, documentation, source code and releases. Firstly, by offering a complete overview of progress since D4.3 and D4.5 and relation with other WPs. Secondly, by explaining the general details of software distribution and documentation, with references to D3.3 where a general set-up is provided. Thirdly, by providing for each INTER-FW/INTER-API component, an introduction with mentions to progress since the release described in D4.5, the list of component features and approaches to extensibility. Finally, the document explains ethical considerations across all WP4 components.

The technical developments in WP4 have three main goals: (1) Enable extension and scalability of the INTER-IoT solution to support present and future applications or more demanding scenarios; (2) Configure, monitor and manage the different layers from a unique interface; (3) Provide and manage a REST-like API to enable the use of interoperable platforms and INTER-IoT features. INTER-FW addresses the points 1 and 2, providing mechanisms to instantiate, configure and manage instances of the interoperability in new and existing scenarios. The third point is covered by INTER-API.

In the same work package, we have developed the Reference Architecture and Reference Meta-Data Model, which are high level technical approaches to the general problem of interoperability of existing IoT Platforms.

INTER-FW provides the user interface tooling to manage the interoperability mechanisms developed in WP3, as well as the global interoperability API, INTER-API. Apart from this, additional cross-layer work has been performed during this time (such as the creation of the Identity Server and its integration with the API Manager).

The APIs exposed in INTER-LAYER (WP3) have been integrated in INTER-FW and INTER-API. INTER-API is a result of the REST-like design of the INTER-IoT layered interoperability stack, providing a unique approach and experience to call the different interoperability APIs developed in INTER-LAYER. In addition to a unified interface, this approach also provides security mechanisms that allow implementation of fine-grained security polices to INTER-FW and subsequently also INTER-LAYER.

The relation between the other work packages (WP3, WP5, WP6, WP7 and WP8) is provided in section 1.2.

This document is the last report about the activities of WP4, even if new software releases will be provided, these will be improvements over original developments carried out during INTER-IoT WP4.

List of Authors

Organisation	Authors	Main contributions
VPF	Pablo Giménez	Provision and implementation of Platforms UX definitions.
UNICAL	Raffaele Gravina	Relation with WP5.
PRO	Miguel Llácer, Miguel A. Llorente, Miguel Montesinos	Coordination of task T4.4. Development of INTER-FW, INTER-API, Identity Manager, integration activities.
TU/e	Tim van der Lee	REST API Monitoring. Deliverable internal review.
XLAB	Flavio Fuart, Gašper Vrhovšek, Dario Gavranovič	Coordination of task T4.5 and leading the compilation of D4.6. Customisation and deployment of the API Request Manager, integration activities, contribution to CMF (Web application) implementation.
SRIPAS	Katarzyna Wasielewska-Michniewska, Paweł Szmeja, Wiesław Pawłowski	Provision and documentation of IPSM API and Semantic Repository definitions, contribution to CMF developments.
RINICOM	Eric Carlson	Provision and documentation of N2N API definitions.
UPV	Eneko Olivares, Andreu Belsa, Jara Suárez de Puga	Provision and documentation of D2D, N2N, AS2AS API definitions. Security-related tasks, integration activities, contribution to INTER-FW and INTER-API Developments.
SABIEN	Gema Ibañez	Ethics. Internal review.

Change control datasheet

Version	Changes	Pages
0.1	TOC, basic structure	5
0.2	Introduction, INTER-API	12
0.3	INTER-FW, Identity manager, Ethics	18
0.4	Section 2, Executive Summary	29
1.0	Version for internal review	31
1.1	Final	32

Contents

List of Figures

List of Tables

Acronyms

API	Application Programming Interface
AS2AS	Application Service Layer Interoperability
CI	Continuous Integration
CLI	Command Line Interface
CMF	Configuration and Management Framework
CRUD	Create, Read, Update and Delete
D2D	Device Layer Interoperability
D3.2	Deliverable 3.3: Methods for Interoperability and Integration v.2
D3.3	Deliverable 3.3: Methods for Interoperability and Integration Final
D4.1	Initial Reference IoT Platform Meta-Architecture and Meta Data Model
D4.2	Final Reference IoT Platform Meta-Architecture and Meta Data Model
D4.3	Deliverable 4.3: Interoperable IoT Framework Model and Engine v1
D4.5	Deliverable 4.5: Interoperable IoT Framework API and Tools v1
D4.6	Deliverable 4.6: Interoperable IoT Framework API and Tools, Model and Engine v2
D7.2	Technical Evaluation and Assessment report
D7.3	Final evaluation report
GUI	Graphical User Interface
INTER-API	INTER-FW and INTER-LAYER exposed as software APIs
INTER-CASE	CASE tool for supporting the Automated Application of INTER-METH
INTER-Domain	INTER-IoT cross use case pilot
INTER-FW	INTER-IoT Interoperable IoT Framework
INTER-Health	INTER-IoT Mobile Health Pilot
INTER-IoT	H2020 Project: Interoperability of Heterogeneous IoT Platforms
INTER-LAYER	INTER-IoT Layer integration tools
INTER-LogP	INTER-IoT Transportation Pilot
INTER-METH	Methodology for the integration of heterogeneous IoT Platforms
INTERMW	INTER-IoT Middleware

IO	Input Output
IPSM	IoT Platform Semantic Mediator
IoT	Internet of Things
KPI	Key Performance Indicator
LDAP	Lightweight Directory Access Protocol
Mnn	Month nn of the INTER-IoT project
N2N	Network Layer Interoperability
OAuth	OAuth is an open standard for access delegation
PDP	Policy Decision Point
QoS	Quality of Service
REST	Representational state transfer
SAML	Security Assertion Markup Language
SSO	Single Sign-On
T4.4	Task 4.4 - Implementation of the IoT Interoperable Framework Engine
T4.5	Task 4.5 - Design and Implementation of the IoT Interoperable Framework APIs and Tools for Programming and Managing Interoperable IoT Platforms
T5.2	Task 5.2 - Definition of a Full-fledged Methodology for IoT Platforms Integration
T5.3	Task 5.3 - Implementation of a CASE tool for supporting the Automated Application of the Methodology INTERMETH
UX	User experience
WP2	INTER-IoT Work Package 2: Requirements and Use Cases
WP3	INTER-IoT Work Package 3: Layer Interoperability
WP4	INTER-IoT Work Package 4: Interoperability Framework API
WP5	INTER-IoT Work Package 5: Methodology for the Integration of IoT Platforms
WP6	INTER-IoT Work Package 6: Integration and Deployment
WP7	INTER-IoT Work Package 7: Evaluation and Assessment
WP8	INTER-IoT Work Package 8: Impact Creation
WSO2	WSO2 is an open source technology provider
XACML	eXtensible Access Control Markup Language

1 Introduction

This deliverable is an evolution of D4.5, *Interoperable IoT Framework API and Tools v1* and final implementation of features defined in D4.3 *Interoperable IoT Framework Model and Engine v1*. It reports about final outcomes of research and development efforts elaborated in the previous documents. The fact that this, third, deliverable reports about implemented features, documentation, source code and releases makes it a self-standing document.

A reader interested in using the software developed in WP4 needs to familiarise himself with this document only. The authors also made extensive effort to publish the information needed for usage and further development in standard approaches of publishing documentation and source code. This means, that the main entry for future users will be GitHub[1] and Read the Docs[2] .

Details about documentation, source and binary distributions are extensively elaborated in D3.3, *Methods for Interoperability and Integration Final*. To avoid repetition the reader is referenced, throughout this report, to relevant sections in D3.3.

On the other hand, a more curious reader, interested in the theoretical background and reasons for specific technological choices, will reach for earlier deliverables of this WP.

To position this report in the project's perspective, an overview of progress since the last release (Section 1.1) and relation with other WPs (Section 1.2) is provided below. Then, the details of Software Distribution are provided in Section 2. In Section 3, we provide, for each WP4 component: an introduction with mentions to progress since the release described in D4.5, the list of component features, approaches to extensibility and finally, a release and distribution plan. Ethical considerations, practically unchanged since the publication of D4.3, are referenced in a separate section. In the conclusion, we summarise the main points of the work performed in this WP.

1.1 Progress since D4.3 and D4.5

In this last phase of WP4 the effort has been devoted to the final implementation of INTER-FW and INTER-API. Based on specifications provided in D4.3 and feedback from Pilots and Open Calls, both WP4 components have been finalized and integrated.

The INTER-IoT Configuration and Management Framework (abbreviated, INTER-FW) has introduced important changes in terms of new features, more complete tabs and better performance. The main new feature is the ability to access a common interface to manage INTER-IoT users across the

[1] https://github.com/interiot
[2] https://readthedocs.org/

different INTER-IoT environments, particularly INTER-FW and INTER-API (API Manager), using a central identity repository (Identity Server) and single sign on (SSO). All the tabs have also been updated with more and more user-friendly features. Remarkable changes are:

- In platforms tabs, now it is possible to manage permissions to platform resources.
- In the gateways tab, the information is displayed in a better way, with more relevant data. Now it is possible managing rules from the framework.
- From the Networks tab, the rules now can be properly managed and monitored.
- In the Services tab, a better management of flows has been introduced, a new node repository has been implemented, performance improvements and feature enhancements have been achieved due integration with open call third parties.
- In the semantics tab, a full redesign has been completed, including a IPSM messages translation tool to support the development of new INTER-IoT enabled applications and services.
- API Management tool has now access to the Identity Server.
- In the Users tab now INTER-IoT users can be managed for all the INTER-IoT web resources in a single place. The permission of these users over the platform resources can also be manipulated.
- A tool to feed the INTER-METH process is also available in form of a web interface and a web service.

As for what INTER-API is concerned, it now integrates all REST API interfaces of INTER-LAYER components. In addition, based on user feedback, a unified REST API has been designed and implemented. This task included documentation and analysis of existing REST APIs, unification of calls in order to fit in a unified REST paradigm across all INTER-LAYERS (naming conventions, resources hierarchy, data structures) and development of so called "mediators" in order to remap the newly defined REST calls to INTER-LAYER back-ends. INTER-API has also been integrated with the INTER-FW identity server, thus allowing centralised identity management and SSO. Additionally, with the development of specific SAML security mediators, fine-grained security policies can be defined (for example, by considering REST segments-resources or specific parameters). Out-of-the box metrics and API monitors have been activated to leverage functionalities provided by underlying WSO2 products.

Source code of both modules is kept in the project's private Gogs repository, until all components are equipped with the correct license texts, at which point they will be moved to GitHub. To support the documentation efforts, all the necessary infrastructure to streamline the creation of comprehensive documentation has been put in place as a WP3 effort. In addition, a process to release binary distributions is in place (Section 2.1), together with the first set of software releases (Section 2.2.2).

At the time of submission of D4.3, main architectural choices and software specifications were made and most of the core WP4 functionalities either developed or under advanced state of development. In this last phase of technical developments, some components have seen further extension of their features and services they offer. These specifics are further elaborated in introductory sections for each component separately.

1.2 Relation with other Work Packages

The development activities performed in WP4 are devoted to the framework and API of the INTER-IoT interoperability approach. This means that within WP4 we integrate all work performed in WP3, providing mechanisms and interfaces to ease the use and the understanding of the multi-layer interoperability operations performed in INTER-IoT. The relation with technical work packages can be described in a very simplified way as WP3 is used by WP4 which is used by WP5.

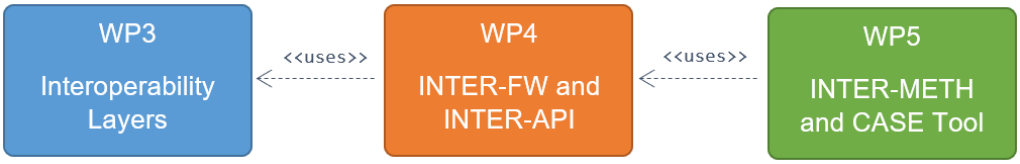

Figure 1: Relation with other Work Packages

Thus, WP3 users will leverage developments of WP4 to manage their interoperability stack built with the INTER-IoT architecture (following the Reference Architecture developed within T4.1 and T4.2). The methodology of defining the different phases on how to build a complete interoperability solution from the problem statement until the maintenance phase of the project is defined in WP5, which relies on WP4 developments for the instantiation and deployment phases. WP6 is also a consumer of WP4 since the development in WP4 is focused in user and developer experience, which is engaged by the three pilots (INTER-Health, INTER-LogP and INTER-Domain) of the project. Finally, WP7 and WP8 also are related to WP4 results as the first uses the results to assess and evaluate the final impact of the developments in WP3 and the second defines business cases and potential markets for the INTER-FW solution.

Although the technical work has been managed to keep as independent as possible, in order to make the most of the research activities planned, the tight relation between the rest of work packages has been a constant in the technical, management and dissemination activities across the project.

1.2.1 Relation with WP3

The relation between WP3 and INTER-FW / INTER-API is based in the consumption of the interoperability mechanisms to create user, developer and administrator interfaces that can help in the tasks of building interoperable IoT platforms or making existing scenarios interoperable.

For INTER-FW, the Configuration and Management framework is based in the low-level APIs provided by each of the interoperability infrastructure layers developed on WP3. As the development of each of the layers has been kept as independent as possible from the others, these APIs are different and sometimes processes of abstraction and homogenization have been made. Thus, INTER-FW acts as a *gluing code* that helps in using and understanding how data is produced and consumed in the different IoT artefacts involved in each deployment. The Software Development Framework provides examples, templates, archetypes and documentation to achieve a better undertake of the WP3 technologies by the stakeholders including developers, data owners and data-driven businesses. Finally, the INTER-IoT User Management developed in WP4 is bound to WP3 as it controls and supervises

the use of the resources accessible in the layers by limiting the access to the INTER-API and other resources.

For INTER-API, the relation is also very tight, as it is based on the APIs provided by the layers, and helps in the usage and administration of the layers.

An important bulk of work performed in WP3, which related to the deployment of a software development framework for INTER-IoT has been extensively reused in WP4. The code repository, binary repositories and other CI tools have been deployed as part of WP3 tasks, but used to their full potential in also in this WP.

1.2.2 Relation with WP5

WP5 is mainly related to WP4 on the instantiation of the configuration and deployment phases.

In particular, T5.2 defines a set of pre-instantiated interoperability-oriented patterns which are tailored on and eventually implemented via the INTER-FW and INTER-API.

T5.3 is mainly devoted to the implementation of the Inter-CASE Tool consisting of different phases. In the *Configuration* phase, the concept of configuration of IoT artefact is perfectly mapped to INTER-FW. Also, the *Deployment* phase is completely implemented by INTER-FW, as it allows to control the different IoT elements on a new or existing IoT scenario.

1.2.3 Relation with WP6

Work Package 6 is, basically, the proof that WP4 is done according the requirements and use cases setup in WP2 and WP3 is ready to be used in real scenarios solving real life problems. In WP2 we have defined the requirements and use cases the INTER-IoT platform must be compliant with. In WP3 all the components were made that will form INTER-IoT. WP6 involves the integration of all the separate parts, which is the first step. In WP4, a Reference Architecture has been defined to be used a common framework to develop IoT interoperability solutions. INTER-FW and INTER-API are the endpoints for IoT Platforms and data owners, third party developers and public authorities. All INTER-IoT components must work together and be coupled to the pilots. Once this first integration step is successful the next step is done, which is proving that the product is made as described in the requirements and that the product works according the use-cases described in WP2.

1.2.4 Relation with WP7

Work Package 7 is the formal evaluation of all aspects of the project. This includes the exploitation, pilots, impact, interoperability and ethical, societal, gender and legal evaluation. In relation to WP3, the technical evaluation will focus on KPIs related to specific pieces of technology developed in INTER-layer. Additionally, the process evaluation will review the system and tools used to control and manage all aspects of WP3 with the outcome being D7.3 where KPIs are reported.

1.2.5 Relation with WP8

Work Package 8 is focused on Impact Creation. In relation to WP4, many of the pieces of technology developed in INTER-FW are whole or parts of products which feature in the INTER-IoT business and exploitation plans. Understanding their capabilities and unique selling points allows for the development of convincing business cases and value propositions. Particularly in the case of INTER-FW and INTER-API, these are the first contact point to most of the stakeholders and to all the potential customers. This makes very relevant the feedback provided by WP8 activities to WP4 developments and the works to improve the UX important for WP8. Additionally, the INTER-FW and INTER-API code base will be open-source and as such will be central to the Freemium business model developed as part of INTER-IoT.

2 Software Distribution

As mentioned in the introduction, WP4 strongly relies in WP3 results regarding source code management and release of documentation. Please refer to D3.3 for these two topics. Binary distributions and releases are specific for WP4, thus we provide them in the following sections.

2.1 Binary distribution

Virtualization of software components developed in WP4 simplifies their deployment and management. Docker is the technology selected to perform the virtualization. Docker ad related technologies are described more in detail in D3.2., section *2.6.2 Virtualization and Clusterisation of Layers*. The main features offered are:

- Dockerization of INTER-FW, Identity Server and INTER-API.
- Creation of Docker compose files to deploy the whole WP4 bundle of software components. The purpose is to offer a complete and functional deployment of the whole framework.
- Private Docker registry to store the Inter-IoT Docker images (see D3.3).
- Eventual usage of Docker tools to manage the containers: Docker Swarm and Portainer.

Thus, the binary release of software artifacts developed in WP4 will be through Docker images, and stored in a private Docker registry. The following rules apply for the WP4 binary distribution:

- Each software release must be marked with a specific commit tag in the code repository as a release.
- For each release a Docker image, tagged with the same version, must be created and pushed to the INTER-IoT private Docker repository server[3].
- Each release must have a specific version of the documentation, and that documentation must contain links to download those binary packages. The documentation should clear enough for a successful installation of each distribution release package.

[3] https://docker-registry.inter-iot.eu

2.2 Releases

2.2.1 Release and distribution plan

The newly developed parts of WP4 components are going to be released open source under the Apache 2.0 license.

The following releases have been planned:

- *Closing of WP4 activities*. This is the main feature release and provides a full implementation of core functionalities as planned for the execution of WP4. This release is internal, thus available to project partners and Open Call projects.
- *License-ready release*. This release will contain all licensing information and will be ready for public distribution.
- *Integration-improved release*. This release will consider lessons learned during the Integration and pilot deployment tasks (WP6).
- *Evaluation-improved release*. This release will consider the results of the technical evaluation results (D7.2) and related improvements to meet the required technical standards.

2.2.2 Release Summary

In this section the release information summary table for each WP4 component at the time of closing WP4 activities is given, including component name, version, source code, documentation, Docker image and dependencies.

INTER-FW	Configuration and Management application	**Version**	1.0.0
		Source Code	https://git.inter-iot.eu/Inter-IoT/framework
		Documentation	https://docs2.inter-iot.eu/docs/framework/latest/
		Docker Image	docker.inter-iot.eu/inter-fw-web:1.0.0
		Dependencies	*Frontend:* Bootstrap-vue – Bootstrap styles in vue; Vue-i18n – Internazionalization; Vis – Network topology diagrams; Xmlbuilder – JSON to XML parsing. *Backend:* Express – Nodejs web framework; Mongoose – mongoDB management; Bluebird – block requests creation; Soap – WSO2 web services Client-oatuh2 authentication and SSO over WSO2
	Identity Manager	**Version**	1.0.0
		Source Code	https://git.inter-iot.eu/Inter-IoT/identity_server
		Documentation	https://git.inter-iot.eu/Inter-IoT/framework
		Docker Image	docker.inter-iot.eu/inter-fw-ids:1.0.0
		Dependencies	WSO2 Dependencies: WSO2 Identity Server; MYSQL Driver (OJDBC 5.7)

Table 1: Summary table of INTER-FW component distribution

INTER-API	REST API Manager	**Version**	2.0.0.
		Source Code	https://git.inter-iot.eu/Inter-IoT/api_manager
		Documentation	https://docs2.inter-iot.eu/docs/framework/latest/
		Docker Image	docker.inter-iot.eu/inter-api:2.0.0
		Dependencies	WSO2 API Manager; WSO2 Identity Server; INTER-LAYER components

Table 2: Summary table of INTER-API component distribution

3 WP4 Software Components

The content of this section provides a specific delta from the previous version of WP4 software release. This section will present the improvement of solutions compared to the beginning of WP4, their implementation status, as well as the steps to follow for the extension of their use in the future.

3.1 INTER-FW Configuration and Management Framework

The INTER-FW configuration and management framework is a web based application that encapsulates the functionalities supporting the instantiation, configuration and management of IoT interoperability solutions in existing or new IoT based solutions. Architecture, requirements, analysis and design have been described in previous documents of the INTER-IoT project (see D4.1 Initial Reference IoT Platform Meta-Architecture and Meta Data Model, D4.2 Final Reference IoT Platform Meta-architecture and Meta-data Model and D4.3 Interoperable IoT Framework Model and Engine v1). This document, which complements the software release, it is only reported the software development progress since last delivery.

The main advances performed during this period are related to the expansion of the functionalities of the initial release. INTER-FW provides a single management place for all the interoperability layers developed in INTER-IoT, as well as a user management tool for authentication and authorization control. During the reported period, all the layers tabs have improved performance, increased the information and quality of the data provided and integrated better the UX to the underpinning interoperability processes. Most of these layers now also include examples and templates to guarantee their extensibility and future readiness.

3.1.1 Release features

In the current release, INTER-FW provides the following changes and improvements:

- Platforms tab
  - Improved integration with INTERMW internal API. In the previous version, some of the features were directly requested to platforms due the interim status of development of the INTERMW. Now as this status is more advanced, all the requests are based on the interoperability layers.

  - Adapted to new INTERMW API. The INTERMW API has changed considerably during this period and new adaptations must have been done to follow the new concepts and to support new features.

- Devices tab
  - Introduced new devices section where all the devices connected to INTER-IoT are represented. The user experience of INTER-FW has been improved by adding a new devices tab were all registered devices are displayed, leveraging new features of INTERMW.
- Gateways tab
  - Improved the information provided in the 'info' section and the overall aspect. The number of parameters displayed and the general look and feel has been improved.
  - Physical gateways now provide information per sensor. Currently, sensors on the gateways are detected and reported via INTER-FW. Information about them (IO type, type, last value and timestamp) is also offered. Further information from the physical gateways is also available.
  - Virtual gateways now can be managed in the INTER-FW. There is a list of virtual gateways (digital twins) and they can be operated via the application.
  - Added rule management (CRUD) and definition. Gateway rules are listed and can be created, read, updated and removed.
- Network tab
  - Network topology representation improved. General look and feel of this tab has been updated.
  - Statistics section working with ports and flows information. Information of this tab has been improved, introducing all the relevant parameters that INTER-IoT virtual network controller can manage.
  - Implemented QoS management with rules setup, queues and meters. This section has been completely implemented, allowing to set up networking rules to manipulate the QoS.
- Services tab
  - Flow repository improved with start, modify, stop and remove options. New operations can be performed with the flows.
  - Node repository implemented. Now the custom INTER-IoT to manage IoT services are listed and basic operation as updating or removing a node are available.
  - Back-end service management improved with contributions of open call participants.
- Semantics tab
  - Complete redesign of the UX.
  - The server where the instance of IPSM is running is now configurable. Now the INTER-FW instance can manage several IPSM instances y completing the 'url' field.

  - Message translation tool provided to support developers and integrators. This tool supports translation of custom messages without the need of using INTERMW components or connecting via CLI to the message broker of IPSM.
  - Improved and adapted to new version of API.
  - Alignment definition tool. The alignments now can be uploaded in INTER-FW.
  - Channel definition tool. The channels now can be defined within INTER-FW.
  - Performance improvements. General changes for a better UX.
- API tab
  - New identity server tab. Now the identity server backed can be accessed from this tab.
- User Management tab
  - Users now are synchronized with the identity server and shared across applications. There is a common repository of users (see 3.2) where the INTER-IoT users are stored and can be managed. INTER-FW provides front-end interfaces for operations with this server.
  - Improved the user model. More user information added.
  - Users now have roles and permission over platforms and IoT Devices.
- CASE tab
  - Created this tab to help the CASE tools definition. This tab helps exporting information from INTER-FW CMF to INTER-CASE tool. A web service has also been provided.

3.1.2 Extensibility

The CMF application is extensible by expanding the contents of the current tabs or creating new ones. The web application is based on well-known technologies such as vue.js [4] , express.js [5] or MongoDB [6]. These technologies are excellently documented and supported by examples across the web, which make them adequate for future-proof developments and evolution of the current INTER-FW interfaces.

On the other hand, the extensibility of INTER-FW is bounded to the extensibility of layers, described in D3.3. As far as the tabs are defined to accomplish in a graphical, user-friendly way the tasks performed by the layers. The tabs features can be expanded by scripting, aggregating or presenting differently the layers features, but the ultimate key for expansion is dependent of the extension of features in the interoperability layers.

[4] https://vuejs.org/
[5] http://expressjs.com
[6] https://www.mongodb.com/

3.2 INTER-FW Identity Manager

The INTER-IoT Identity Manager (also referred as Identity Server) is a new component introduced in the current release with the purpose of centralising and unifying user management between INTER-FW, INTER-API and any other future applications based on INTER-Layer. INTER-IoT user information is stored in a single central repository where their authentication, authorization and accountability details are persisted. The Identity Server has been integrated with INTER-API and INTER-FW, although the interfaces for usage by additional services are available in case of future applications of INTER-IoT.

3.2.1 Release features

The most important features of this newly introduced component that are used in INTER-IoT are:

- *Configuration of single sign-on (SSO).* The Identity Server provides a base for cross-authentication between platforms.
- *Configuration of XACML policies.* The system can be configured with default policies that can be changed in real-time. Those XACML policies affect authorization constraints of INTER-IoT users. These policies are one of the most advanced standards for fine-grained authorization and have a high potential for future applications.
- *Configuration of OAuth.* The Identity Server allows OAuth authentication across all integrated modules. This feature is used in INTER-API (see section 3.3).

3.2.2 Extensibility

The extensibility of the Identity Server is achieved through several extension points:

- Creation and application of new XACML policies. The full process is explained in the INTER-IoT documentation.
- Aggregation of new authenticators and connectors[7] (new identity providers) for identity federation[8]. This could allow users accessing INTER-IoT without having an INTER-IoT account, by leveraging the Identity Server to control and restrict the most sensitive parts of the system. A possible scenario of this extension could be a deployment of INTER-IoT by a public service that allows all users (e.g. the citizens in a Smart City) access read-only information about different publicly owned IoT platforms operating in an area.
- The INTER-IoT use of the identity server has a third expansion point through Entitlement Mediators[9]. An Entitlement Mediator intercepts requests and evaluates actions performed by users against an XACML policy. Identity Server can be used as XACML Policy Decision Point (PDP) where the policy is set. This is also explained in the official documentation, and consists essentially in adding authorization points to the Identity Server for protecting endpoints in the INTER-API and other resources.

[7] https://docs.wso2.com/display/ISCONNECTORS/WSO2+Identity+Server+Authenticators+and+Connectors+Documentation

[8] https://docs.wso2.com/display/IS550/Identity+Federation

[9] https://docs.wso2.com/display/IS510/Configuring+WSO2+ESB+Entitlement+Mediator+with+Identity+Server

3.3 INTER-API

The API Manager solution is described in D4.3, while the first version was provided as part of D4.5. The basic functionality supports access to single INTER-LAYER components and allows basic authentication.

In addition to those features, this new version provides a unified API access to INTER-LAYER, integrates with the Identity Manager (thus whole INTER-FW) and provides fine-grained access control. It also provides full support to Pilots and Open Calls. We refer to this M30 release as INTER-API V2 in the text that follows. The release is tagged as V2 as it contains substantial advancement from the previous version as well as some changes that are not backward-compatible.

Documentation for deployment and usage has been provided as described in section 2.2.2.

3.3.1 Release features

INTER-API is a deployment of the WSO2 API Manager with INTER-IoT API definitions and some extended functionalities through implementation of API mediators.

The INTER API V2 release provides the flowing features specific to INTER-IoT:

- *INTER-FW SSO.* Integration with the INTER-FW Identity server as described in Section 3.2.
- *Fine-grained access control.* A specific extension (mediator) to allow the definition of custom security policies. In this way different controls can be put in place, for example "Allow user X accessing only device Y".

  We have identified the following types of API users with the corresponding set of access and management privileges: INTER-FW core users (full access to all INTER-FW features), INTER-FW front-end users (restricted set of access rights necessary to execute API calls). In addition, a set of access policies, using SAML definitions, is being developed as part of INTER-LogP specific requirements.
- *Unified INTER-LAYER access.* The main feature is unified access to all INTER-LAYER components through a unified REST API endpoint (by means of a custom API mediator).

  The creation of a unified API access through the API manager consisted of several steps. First, we created a API design document for each INTER-LAYER component, which is, in our case, provided through Swagger definitions. Then, those definitions where analyzed with focus on REST best-practice approach and different naming conventions among INTER-Layers. Then, a unified API interface was defined, with specification of mappings to corresponding back-end systems. Finally, a unified OpenAPI definition has been created in addition to a "mediator" module that maps the APIs. The user should then subscribe to the APIs through the API subscription web GUI.
- *Monitoring.* Out-of-the box metrics and API monitors have been activated to leverage functionalities provided by underlying WSO2 products. These include API Analytics[10] and Carbon Metrics[11].

[10] https://docs.wso2.com/display/AM220/Configuring+APIM+Analytics

[11] https://docs.wso2.com/display/AM220/Monitoring+with+WSO2+Carbon+Metrics

3.3.2 Extensibility

The selection of *WSO2 API Manager* as basis for the INTER-IoT INTER API provides several opportunities for customization and extension of features provided by INTER-IoT. The following list addresses those we find most relevant for addressing IoT interoperability tasks:

- *INTER-LAYER REST API evolution.* Each INTER-LAYER component has several extensibility mechanisms that allow adding new functionalities or improvements of existing services. This will inevitably lead to changes in REST API calls exposed by single layers. INTER-API provides API Life-cycle Management and Versioning, so that every change is typically deployed as a prototype for early promotion. After a period of time during which the new version is used in parallel with the older versions, the prototyped API can be published and its older versions can be depreciated[12].With updates to OpenAPI definitions of the Unified INTER-IoT REST API every change is documented and presented to prospective users in a standard format adopted by a majority of modern software development frameworks.
- *Custom mediation sequences*[13]*.* The default mediation flow can be extended with the development of custom mediation sequences. In the case of the Unified INTER-IoT REST API, two Inflow Custom Mediation Extensions have been provided: (1) Integration with Identity Server SAML security policy definitions. This integration allows extension of existing security policies in the Identity Server. (2) Mediation for the mapping of unified API calls to INTER-LAYER and re-mapping of received responses.

  With further implementation of Inflow and Outflow Custom Mediation Extensions users can implement use-case specific APIs to, for example, execute a series of INTER-LAYER calls, merge the results and serve them to users.
- *Custom authorisation policies.* As explained with custom mediation and in the in Section 3.2, custom authorisation policies can be defined to consider REST API segments (resources) and parameters. In this way, very fine-grained access policies can be defined.
- *Integration with corporate User Stores.* In INTER-IoT pilots and tests two different deployments have been used: (1) Self-standing API Manager with build-in identity management and (2) integration with the WSO2 Identity Manager. In principle, WSO2 API Manager can be integrated with several User Store types[14], such as LDAP, Active Directory and custom realms.

The decision to use an existing product as basis for API management has been shown as a very good choice, as it allowed us to easily adapt and extend by taking into account feedback from Pilot and Open Calls developers. It also allows implementation of extensions unforeseen by INTER-IoT requirements and use-case definitions.

[12] https://docs.wso2.com/display/AM220/Create+a+New+API+Version
[13] https://docs.wso2.com/display/AM220/Adding+Mediation+Extensions
[14] https://docs.wso2.com/display/IS410/Configuring+User+Stores

4 Ethics

Ethics is a central consideration to all INTER-IoT planning and development. As requested at the interim review, an ethical advisory board has been established. This board, within INTER-IoT, continuously reviews ethical issues. The aim of the committee is to ensure that ethical considerations and issues are addressed in the conduct of the research and development work undertaken within the project. The committee seeks to support and encourage the process of ethically conducted research to maintain the safety and well-being of participants and researchers to promote ethical values.

Ethical issues related to WP4 outputs have been elaborated in Section 5 of deliverable D4.3. During the development of INTER-FW and INTER-API specifications and technologies defined in that deliverable have been followed. We are thus concluding, that ethical issues have not changed since the previous assessment.

5 Conclusions

In this document, the final release of INTER-FW and INTER-API is provided. This work provides the main features to configure, administer and manage heterogeneous IoT platforms in scenarios where interoperability is a key factor. These solutions are already being tested and validated in the three INTER-IoT pilots (INTER-LogP, INTER-Health and Cross-domain Pilot) and the Open Calls who joined the project in mid-2017.

A full coverage of the use cases, including mechanisms to expand and adapt to specific interoperability scenarios, has been achieved. Different interoperability layers (D2D, N2N, MW2MW, DS2D2 and AS2AS) have been integrated thorough a unified management framework and API access.

The task of further integration and extension is being taken over in WP6, while we expect feedback from evaluation activities (WP7). It will further influence the evolution of modules developed herein. It is the hope of INTER-IoT partners that further engagement with the Open Source community and development of business models in WP8 will secure long-term sustainability of INTER-FW and INTER-API.