Multi-layered Security Technologies

for hyper-connected smart cities

D4.4: M-Sec cloud and data level security - final version

April 2021

Grant Agreement No. 814917

Multi-layered Security technologies to ensure hyper-connected smart cities with Blockchain, BigData, Cloud and IoT

Project acronym	M-Sec
Deliverable	D4.4M-Sec cloud and data level security - final version
Work Package	WP4
Submission date	7 April 2021
Deliverable lead	Aamir Bokhari (YNU)/ Mathieu Gallissot (CEA)
Authors	Aamir Bokhari (YNU), Mathieu Gallissot, Levent Gurgen, Christophe Munilla (CEA), Akira Tsuge (KEIO),
Internal reviewer	Arturo Medela (TST) /Nobukazu Yoshioka, Takafumi Komoto (NII)
Dissemination Level	Public
Type of deliverable	DEM

The M-Sec project is jointly funded by the European Union's Horizon 2020 research and innovation programme (contract No 814917) and by the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN (contract No. 19501).

Version history

#	Date	Authors (Organization)	Changes
v0.1	12 February 2021	Mathieu Gallissot (CEA)	Full ToC and assignments
v0.2	15 March 2021	Akira Tsuge (KEIO)	Section 2.3, 3 content provided
v0.3	16 March 2021	Mathieu Gallissot (CEA)	Section 2 updated
v0.4	17 March 2021	Mathieu Gallissot (CEA)	Section 1, 4 updated
v0.5	18 March 2021	Aamir Bokhari (YNU)	Section 2 updated
v0.6	26 March 2021	Orfefs Voutyras (ICCS)	FGs figures updated
v0.7	1 April 2021	Levent Gürgen, Christophe Munilla (CEA)	sensiNact contributions included
v0.8	06 April 2021	Arturo Medela (TST)	Internal review
v0.9	06 April 2021	Akira Tsuge (Keio)	Common API and interaction with other FG updated
V0.10	06 April 2021	Aamir Bokhari (YNU), Mathieu Gallissot (CEA)	Review and final completions
v0.11	07 April 2021	Nobukazu Yoshioka, Takafumi Komoto (NII)	Internal review
v0.12	07 April 2021	Aamir Bokhari (YNU), Mathieu Gallissot (CEA)	Comments addressed
v0.13	07 April 2021	Mathieu Gallissot (CEA)	Version ready for submission

Table of Contents

List of Tables

List of Figures

Glossary

Acronym	Description	Acronym	Description
AES	Advanced Encryption Standard	HSM	Hardware Security Module
API	Application Programming Interface	MCU	Micro Controller Unit
APPI	Act on the Protection of Personal Information	MPU	Micro Processor Unit
CPU	Central Processing Unit	RSA	Encryption algorithm named after its inventors: Rivest, Shamir, and Adleman
Dx.y	Deliverable y of WP x	PII	Personal Identifiable Information
ECC	Elliptic Curve Cryptography	SSD	Single Shot Multibox Detector
ECDSA	Elliptic Curve Digital Signature Algorithm	SPI	Serial Peripheral Interface
ECH	Elliptic Curve Diffie-Hellman	Tx.y	Task y of WP x
FG	Functional Group	TCG	Trusted Computing Group
GDPR	General Data Protection Regulation	TPM	Trusted Platform Module
GLCIC	Globally and Locally Consistent Image Completion	UC	Use Case
HMAC	Hash-based Message Authentication Code	WP	Work Package

Executive Summary

The work described in this deliverable (D4.4) was carried out in the framework of WP4 – "Multi-layered Security Technologies", and more specifically, in the framework of T4.2 – "Cloud and Data Level Security". The report presents the updated and final version of the document (the first version being D4.3), providing the technical details of the Functional Group and Functional Components related to the Task.

All technical partners involved in this task collaborated and developed the appropriate tools to meet the objectives set out in the project, especially with regard to novel Security aspects in IoT contexts. Every partner focuses on the individual modules that they are responsible for during the implementation phase of WP4 and supports the integration activities of WP2, while following the common Architecture framework set by WP3 in D3.4.

All of the updated versions of the WP4 technical deliverables (D4.2, D4.4, D4.6, D4.8, D4.10) follow the same approach and have the same structure. Section 1 provides an introduction to the scope of this document and its relation with other WPs and Tasks. Sections 2 and 3, which aggregate all the main outcomes of the Task, present extensively the FGs and the Functional Components covered by the Task, by providing an extensive description of the corresponding functionalities, and details related to the API of the FG and its interactions with other FGs of the M-Sec solution. Finally, Section 4 concludes the document.

Regarding the differences between 'D4.3 M-Sec Cloud and Data Level Security – first version' and 'D4.2 M-Sec IoT Security Layer – final version':

- **Section** Erreur ! Source du renvoi introuvable. has remained more or less the same, but the relation of the Task to the corresponding M-Sec risks and threats is added.
- Following the updated version of the M-Sec Architecture, 2 **and 3** focus on the presentation of the Task from an updated FG perspective. As such, the "Hardware-based encryption" and "Software-based Threat Monitoring" have been moved to D4.2, and are replaced by the "SOXFire" and "sensiNact" components (initially presented in D4.9). The "GANonymizer" component remains in this deliverable.
- The Package Information, Installation Instructions, and the Licensing Information are omitted in the new version, but they are replaced by sections related to APIs.
- **Sections2.4 and 3.4** are brand new sections and are a result of the common integration activities between all of the technical Tasks of the project.
- **Section 4** corresponds to Section 5 of the previous version of the document.

All in all, the deliverable is considered to have provided all of the information required to expose the M-Sec technical solutions related to T4.2 as well as the results of the integration and demonstration-related activities.

1. Introduction

1.1 Scope of the document

This document is the final version of demonstrator "M-Sec cloud and data level security" involving developments carried out in Task 4.2 of the M-Sec project. Regarding the previous version of this demonstrator, deliverable 4.3, it introduces changes from WP3 such as the notion of functional groups as well as the risks and threats mitigated by M-Sec technologies.

1.2 Relation to other work packages and tasks

The following figure summarises the relations of this deliverable (and the corresponding task) to other tasks and WPs.

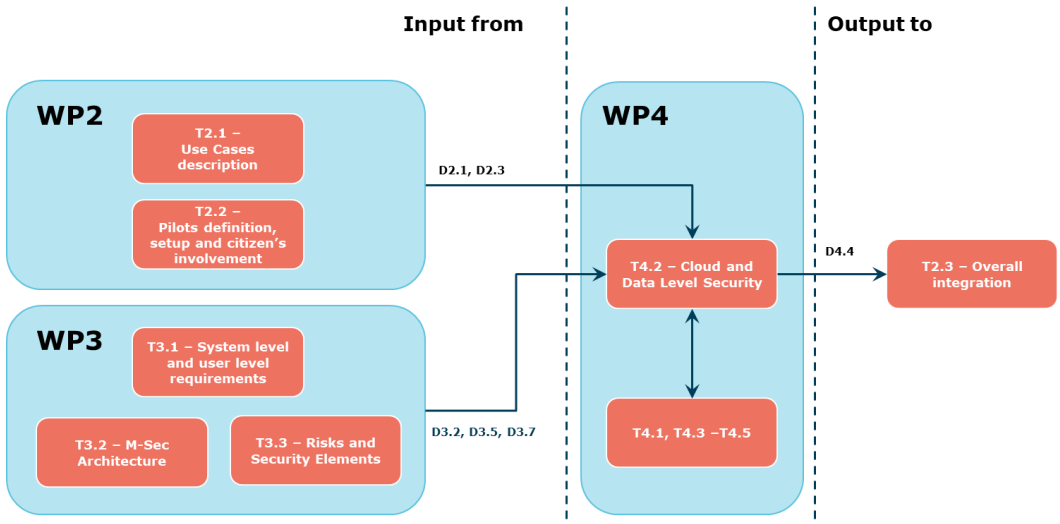

Figure 1. T4.2 and D4.4 relations to other WPs and Tasks

The work done in Task 4.2 is directly related to WP3. T4.2 receives as input system and user requirements from T3.1 and Risks- and Threats-related information from T3.2. Moreover, it follows the common Architectural framework that has been identified in T3.2 for the coordination of all the technical activities. Similarly, the Task receives input from WP2 related to the coverage of the needs of the UCs and the pilots.

Within this very same WP, T4.2 is closely related to the other Tasks that focus on other security layers. Communication between the WP4 tasks leads to an end-to-end security solution. The objective of Task 4.2 is to provide technologies to secure the exchange between data from the IoT layer (T4.1) to a remote registry (T4.3). Concerning Task 4.1, data security is a common problem. The encryption mechanisms depend directly on the nature of the IoT Device and we propose data security methods that rely on both, software and hardware integration developed in Task 4.1. With respect to Task 4.3, we are introducing through our developments means of strong authentication of data via encryption mechanisms. This strong authentication

makes it possible to ensure to a distributed register the source of the data and thus to make a decision on the permanent inscription or not in the register.

Finally, the results of this report are directly provided as input to T2.3 which is focusing on the overall integration activities. Together with the other final deliverables of WP4, D4.2 provides all the information and functionalities required for an integrated security solution.

1.3 Relation to M-Sec Risks

The complete list of potential risks and threats that may affect M-Sec's IoT layer can be checked in Table 1, as extracted from Task 3.3. All of these threats are of Type "Cloud", and Sub-Type "Data Access", "Storage" or "Management". Specific interfaces are provided in Deliverable 3.5 "Risks and security elements for a hyper-connected smart city".

Table 1 shows extracted risks regarding data and cloud which have the highest degree of severity. Three risks mitigation strategies are enabled:

- Add countermeasures with encryption and authentication, using SOXFire and sensiNact to manage flow from devices to applications.
- Reduce risks severity by using anonymization with the GANonymizer.
- Transfer cross-layer management of security to Security Manager.

Table 1: M-Sec T4.2 risks and threats

Threat #	Description	STRIDE Threat Class	M-Sec Asset	Source	Probabi-lity	Critica-lity	Ra-ting	Comments/ Mitigation
Thr.CD.1	Impersonation: A third party uses a false ID to gain access to the cloud	S	SoxFire, sensiNact,	Use Case 2, 3	3	3	9	Authentication centralized using a common authentication system
Thr.CD.5	Data (raw & processed, personal data) stored in the cloud can be read by an intruder	I	SoxFire, sensiNact, GANonymizer	Use Case 2, 3	3	5	15	Use of strong encryption and authentication Anonymise data
Thr.CD.8	Attacker can poison cloud database and/or alters outgoing information	T	SoxFire, sensiNact,	Use Case 2, 3	3	5	15	Use of strong encryption and authentication
Thr.CD.11	Attacker gains knowledge of sensitive exchange data	I	SoxFire, sensiNact, GANonymizer	Use Case 3	3	5	15	Use of strong encryption Anonymise data
Thr.CD.14	Man in the middle attack: a third party puts itself between the entity (e.g. gateway or application) that communicates with the cloud and the cloud itself, without them noticing (so that both actually communicate with the intruder)	S	SoxFire, sensiNact, GANonymizer	Use Case 3	3	5	15	Use of strong encryption Anonymise data
Thr.CD.18	Nobody is responsible for cloud maintenance	Management issue	Security Manager	Use Case 3	3	3	9	Use common security management framework

Thr.CD .19	Nobody is responsible for system management and maintenance (e.g. cloud infrastructure)	Management issue	Security Manager	Use Case 3	3	3	9	Use common security management framework
Thr.CD .23	Attacker changes configurations and prevents proper communication to an actuator	D	SoxFire	Use Case 3	3	3	9	Monitor integrity and security status

2. Secure City Data Access FG

2.1 General Description of the FG

The goal of the Secure City Data Access Functional Group is to act as a bridge between IoT devices and applications. In the manner of IoT platforms, this functional group (middleware layer) includes two parts as shown in Figure 2 below:

- Southbound access which deals with devices related functional groups. It handles various IoT protocols and standards and compiles data in a common model.
- Northbound access which deals with storage FG. It provides access to data using different web services and format given the affinity of each application vendor.

Regarding security, this functional group has a particular role as it bridges two different paradigms: embedded devices on one side with low encryption capabilities and long lifespan, and end-user devices with high computational capabilities and shorter lifespan on the other side. In terms of cybersecurity requirements, these two paradigms use different standards given their ability, risks, and exposure. For example, encryption capabilities differ at least in two aspects: the first one is the complexity of the encryption process due to the computing power and the second one is the freshness of the algorithm used, with few upgrade capabilities on IoT devices. As a result, it is expected for this FG to adapt and adjust the level of security on both sides.

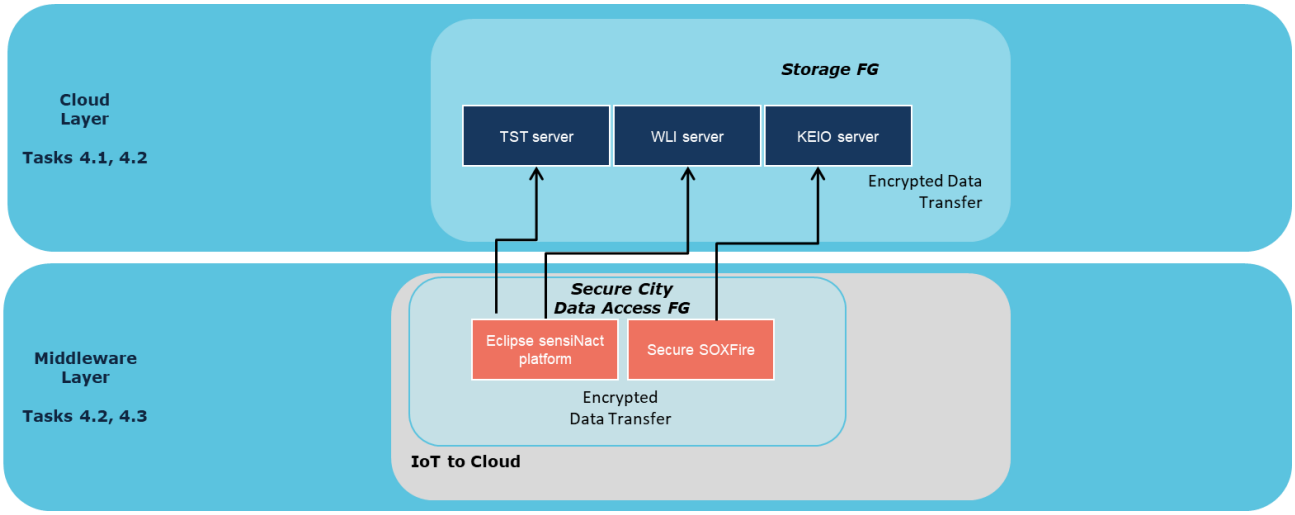

Figure 2. The Secure City Data Access FG and the Storage FG

In the following section, the components of this FG are described in detail. Section 2.4 presents the interactions of these components with components of other M-Sec FGs. Finally, the Annex presents the position of the FG within the whole M-Sec Architecture.

2.2 Components of the FG

SOXFire

Description

SOXFire can provide practical distributed and federated infrastructure for IoT sensor data sharing among various users/organizations in a way that is scalable, extensible, easy to use, and secure with preserving privacy.

SOXFire is based on Carnegie Mellon University's Sensor over XMPP and Openfire which is an XMPP server and provides anonymous subscription/data acquisition function and subscription via federation so that data can be shared with anyone. In addition, we built "Secure SOXFire" with higher security by integrating the Security Management Tool and multiple layers of security at the enterprise level. Secure SOXFire provides the following functions in a highly secure environment.

- Anyone can exchange data with the anonymous subscription function.
- Subscription via federation is possible, providing a scalable federated sensor network architecture for smart cities.

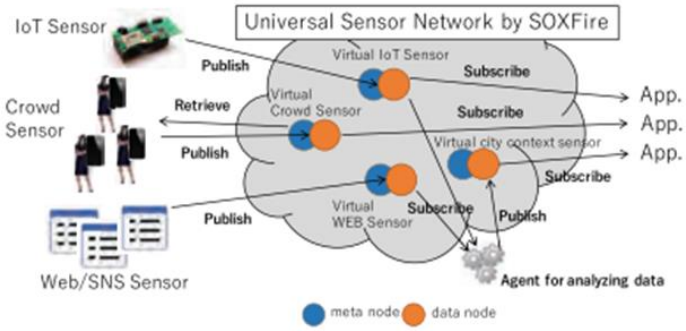

Figure 3. Keio SOXFire

API for SOXFire

XMPP server name: sox.jn.sfc.keio.ac.jp

Authorization: Username/Password

Publish Connection:

Authorization: Username/Password

Device parameter:

- Device: Device name
- Transducer (List): Transducer name and each parameter

Subscribe Connection:

Device parameter:

- Device: Device name

sensiNact

General Description

The sensiNact Gateway developed by CEA and now promoted by spin-off company Kentyou implements the basic blocks for connectivity, service abstraction, device management, virtualization, and remote access. The sensiNact Gateway allows interconnection of different networks to achieve secured access and communication with embedded devices. The sensiNact platform defines a generic service model, as shown in Figure 4, and a set of generic access methods.

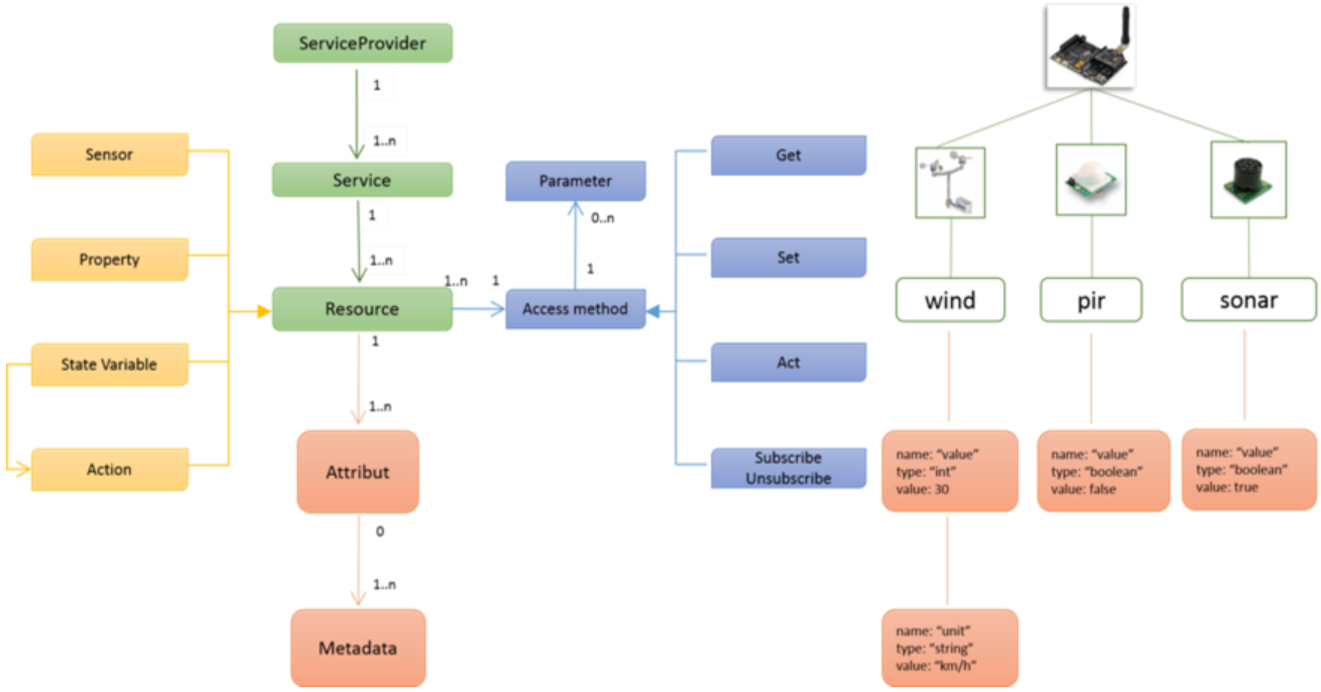

Figure 4. sensiNact service model

The Service provider, Service and Resource triptych is the spine of the model:

- the Service provider is attached to one location
- the Service is attached to one business concept
- and the Resource is attached to one business data.

A Resource, on which apply Access Methods, is a collection of Attributes, characterized by Metadata.

Access Methods allow reading (client/server or publish/subscribe model) to write, and to actuate.

Connectivity

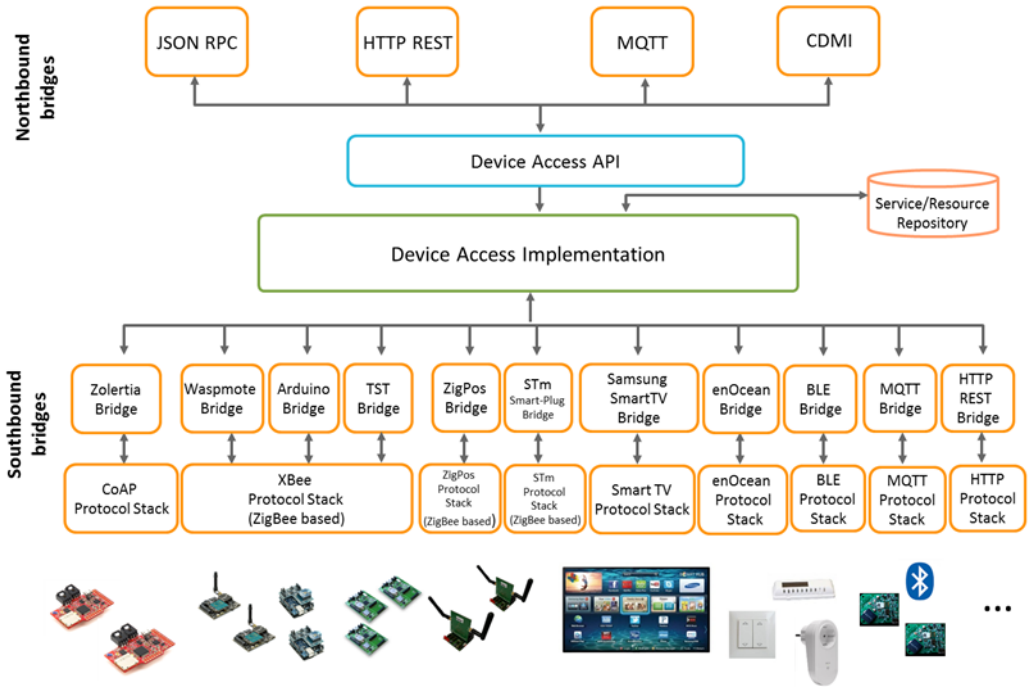

Figure 5. sensiNact gateway northbound and southbound connectivity

On the southbound side, the sensiNact gateway allows coping both with "physical device" protocols and "virtual device" ones, allowing a uniform and transparent access to a BLE network, or an HTTP Restful web service for example (pell-mell a non-exhaustive list of supported protocols: EnOcean, Bluetooth Low Energy, MQTT, CoAP, NGSI, Openhab, SOXFire, etc…). On the northbound side, the sensiNact gateway provides both client/server and publish/subscribe access protocols (MQTT, JSON-RPC, HTTP Restful, NGSI, CDMI, SOXFire, etc…)

Security Components

In sensiNact, three sites allow providing a secured encapsulation of the data relayed by the platform.

- The encryption of data coming from connected counterpart is easily implementable over each southbound bridge;
- The modules signature allows to define which module will be allowed to provide a feature, or a remote system/device connection;
- Finally, an OAuth2 / OpenID intermediation service provides northbound access security

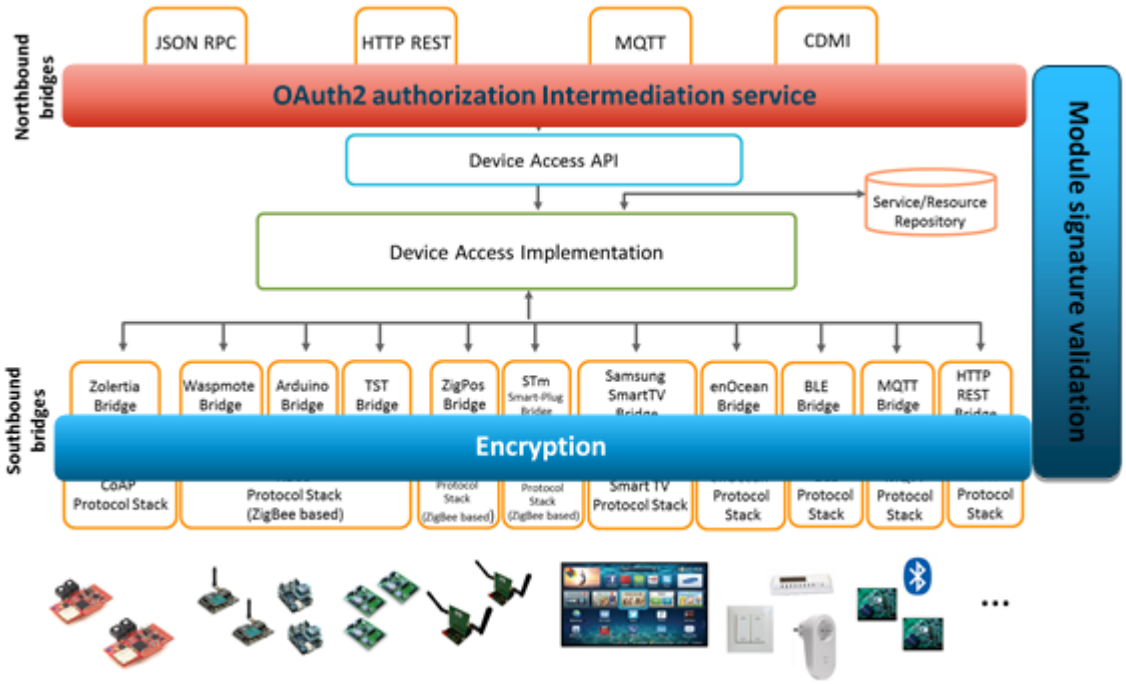

Figure 6. Secured connectivity

Encryption

The modular and generic southbound bridge template allows including a security process between the sensiNact and the connected counterpart, like keys sharing allowing encrypting exchange data without any extra implementation or update of the platform.

Module signature validation

The signature of the modules that compose the platform at build time permits validating that an installed module is allowed to connect to the others and to provide a new feature or secured access to a connected counterpart.

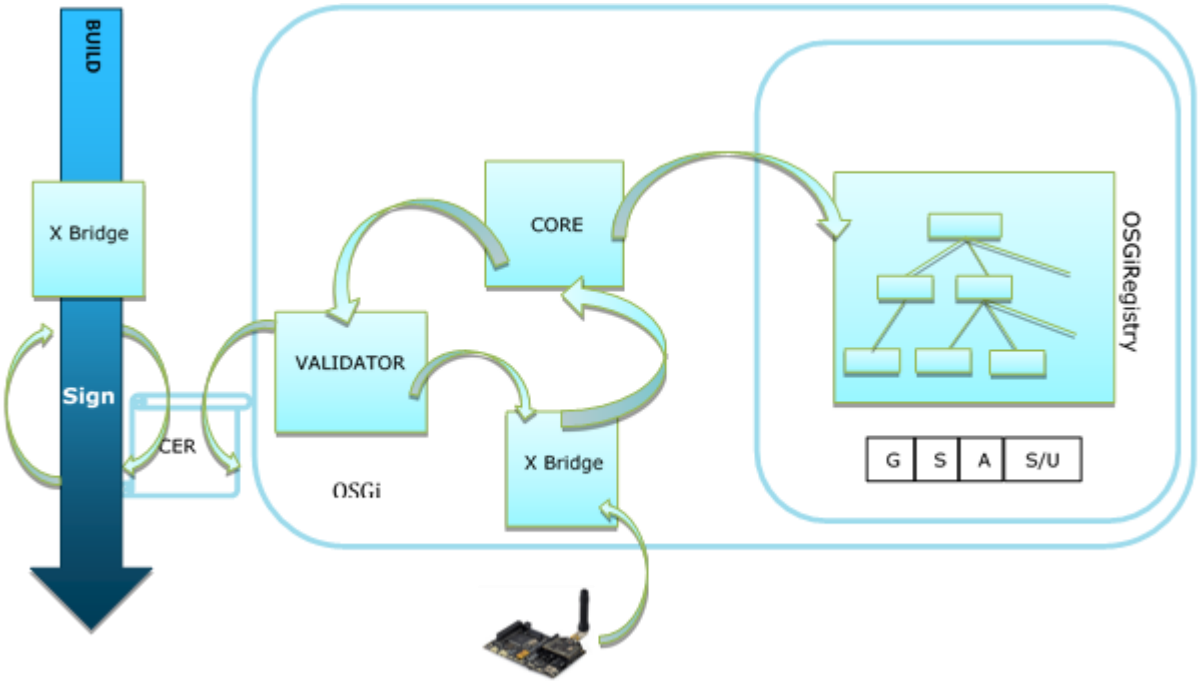

Figure 7. Sign the X bridge

OAuth2 authorization intermediation service

An oAuth2/OpenID filter, implementing, for now, Authorization code and Resource owner password credentials flows, offers a standard northbound secured access solution.

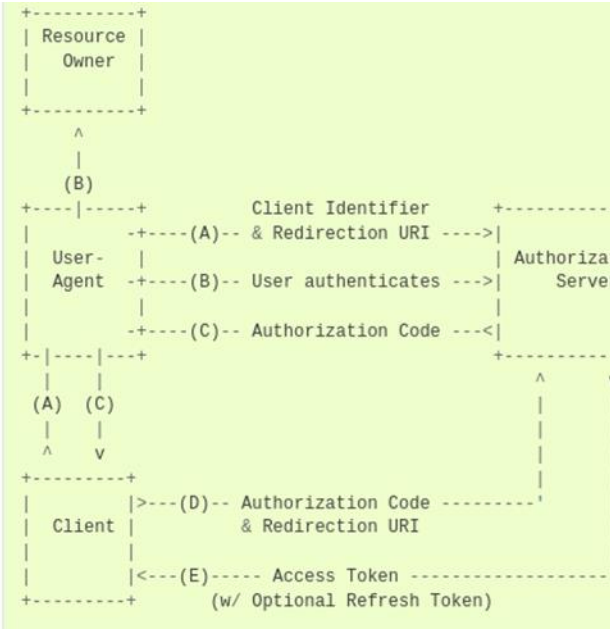

Figure 8. Authorization code flow

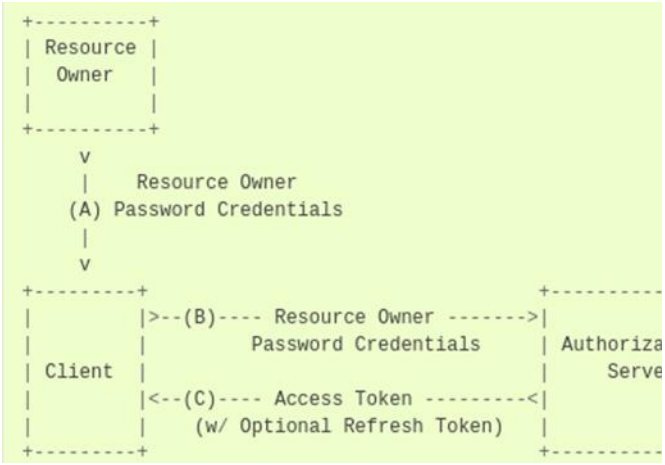

Figure 9. Resource owner password credential flow

API for sensiNact

- **List Service Providers**
  https://<host>:<port>/sensinact/providers

  HTTP Method
  GET

  Headers
  Accept:application/json

  Query
  N/A

  Content
  N/A

  Response
  {"providers":["<provider-identifier-1>", "<provider-identifier-2>", …, "<provider-identifier-n>"],"type":"PROVIDERS_LIST","uri":"/","statusCode":200}

- **Service Provider description**
  https://<host>:<port>/sensinact/providers/<provider>
  or
  https://<host>:<port>/sensinact/<provider>

  HTTP Method
  GET

  Headers
  Accept:application/json

  Query
  N/A

  Content
  N/A

  Response
  {"response":{"name":"<provider>","services":["<service-name-1>","<service-name-2>",…,"<service-name-n>"]},"type":"DESCRIBE_PROVIDER","uri":"/<provider>","statusCode":200}

- **Service description**
  https://<host>:<port>/sensinact/providers/<provider>/services/<service>
  or
  https://<host>:<port>/sensinact/<provider>/<service>

  HTTP Method
  GET

  Headers
  Accept:application/json

Query
N/A

Content
N/A

Response
{“response”:{“name”:”<service>“,”resources”:[{“name”:”<resource-name-1>“,”type”:”<resource-type-1>“},{“name”:”<resource-name-2>“,”type”:”<resource-type-2>“}, …, {“name”:”<resource-name-n>“,”type”:”<resource-type-n>“}]},”type”:”DESCRIBE_SERVICE”,”uri”:”/<provider>/<service>“,”statusCode”:200}

- **Resource description**
  https://<host>:<port>/sensinact/providers/<provider>/services/<service>/resources/<resource>
  or
  https://<host>:<port>/sensinact/<provider>/<service>/<resource>

  HTTP Method
  GET

  Headers
  Accept:application/json

  Query
  N/A

  Content
  N/A

  Response
  {“response”:{“name”:”<resource>“,”attributes”:[{“metadata”:[{“name”:”<metadata-name-11>“,”type”:”<metadata-type-11>“,”value”:<metadata-value-11>},”name”:”<metadata-name-12>“,”type”:”<metadata-type-12>“,”value”:<metadata-value-12>},…,”name”:”<metadata-name-1n>“,”type”:”<metadata-type-1n>“,”value”:<metadata-value-1n>}],”name”:”<attribute-name-1>“,”type”:”<attribute-value-1>“}, …, {“metadata”:[{“name”:”<metadata-name-n1>“,”type”:”<metadata-type-n1>“,”value”:<metadata-value-n1>},”name”:”<metadata-name-n2>“,”type”:”<metadata-type-n2>“,”value”:<metadata-value-n2>},…,”name”:”<metadata-name-nn>“,”type”:”<metadata-type-nn>“,”value”:<metadata-value-nn>}],”name”:”<attribute-name-n>“,”type”:”<attribute-value-n>“}], “accessMethods”:[<access-method-description-1>,<access-method-description-2>,…, <access-method-description-n>],”type”:”<resource-type>“},”type”:”DESCRIBE_RESOURCE”,”uri”:”/<provider>/<service>/<resource>“,”statusCode”:200}

- **GET**
  https://<host>:<port>/sensinact/providers/<provider>/services/<service>/resources/<resource>/GET
  or
  https://<host>:<port>/sensinact/<provider>/<service>/<resource>/GET

  HTTP Method
  GET

  Headers
  N/A

Query
attributeName (optional - value by default)

Content
N/A

Response
{“response”:{“name”:”<attribute-name>“,”type”:”<attribute-type>“,”value”:<attribute-value>,”timestamp”:<last-update-timestamp>},”type”:”GET_RESPONSE”,”uri”:”/<provider>/<service>/<resource>“,”statusCode”:200}*

the response may contain all other non-fixed metadata attached to the targeted attribute

A GET access method can apply on multiple resources if the provider path element is replaced by a valid ldap formatted string filter

- **SET**
  https://<host>:<port>/sensinact/providers/<provider>/services/<service>/resources/<resource>/SET
  or
  https://<host>:<port>/sensinact/<provider>/<service>/<resource>/SET

  HTTP Method
  POST

  Headers
  Accept:application/json
  Content-Type:application/json

  Query
  N/A

  Content
  {“parameters”: [{“name”:”attributeName”,”type”:”string”,”value”:”<attribute-name>“},{“name”:”value”,”type”:”<attribute-type>“,”value”:<attribute-value>}]}
  or
  [{“name”:”attributeName”,”type”:”string”,”value”:”<attribute-name>“},{“name”:”value”,”type”:”<attribute-type>“,”value”:<attribute-value>}]
  or
  [{“name”:”value”,”type”:”<attribute-type>“,”value”:<attribute-value>}] as by default the value attribute is targeted

  Response
  {“response”:{“name”:”<attribute-name>“,”type”:”<attribute-type>“,”value”:<attribute-value>,”timestamp”:<last-update-timestamp>},”type”:”SET_RESPONSE”,”uri”:”/<provider>/<service>/<resource>“,”statusCode”:200}*

  the response may contain all other non-fixed metadata attached to the targeted attribute

- **SUBSCRIBE**
  https://<host>:<port>/sensinact/providers/<provider>/services/<service>/resources/<resource>/SUBSCRIBE
  or
  https://<host>:<port>/sensinact/<provider>/<service>/<resource>/SUBSCRIBE

  HTTP Method
  POST

Headers
Accept:application/json
Content-Type:application/json

Query
N/A

Content
{“parameters”: [{“name”:”attributeName”,”type”:”string”,”value”:”<attribute-name>“},{“name”:”callback”,”type”:”string”,”value”:<callback-url>}]}
or
[{“name”:”attributeName”,”type”:”string”,”value”:”<attribute-name>“},{“name”:”callback”,”type”:”string”,”value”:<callback-url>}]
or
[{“name”:”callback”,”type”:”string”,”value”:<callback-url>}] as by default the value attribute is targeted

Some extra parameters can be used:

“conditions” - the JSON array gathering the JSON formatted description of the constraints applying on the subscription, for example : {“name”:”conditions”,”type”:”array”,”value”:[{“operator”:”<”,”operand”:200, “type”:”int”, “complement”:false}]} meaning that an notification will be send only if the updated value (an integer in the present case) is less than 200

“timeout” - the long duration of the subscription (in milliseconds)

“policy” - the integer identifying the error handling policy to be used (CONTINUE, STOP, ROLLBACK, IGNORE, ALTERNATIVE, LOG)

Response
{“response”:{“initial”:{“name”:”<attribute-name>“,”type”:”<attribute-type>“,”value”:<attribute-value>,”timestamp”:<last-update-timestamp>},”subscriptionId”:”<subscription-identifier>“},”type”:”SUBSCRIBE_RESPONSE”,”uri”:”/<provider>/<service>/<resource>“,”statusCode”:200}*

Notification
{“type”:”CALLBACK”,”callbackId”:”<subscription-identifier>“,”messages”:[{“notification”:{“name”:”<attribute-name>“,”type”:”<attribute-type>“,”value”:<attribute-value>,”timestamp”:<update-timestamp>},”type”:”ATTRIBUTE_VALUE_UPDATED”,”uri”:”/<provider>/<service>/<resource>/<attribute>“}]}*

- **UNSUBSCRIBE**
  https://<host>:<port>/sensinact/providers/<provider>/services/<service>/resources/<resource>/UNSUBSCRIBE
  or
  https://<host>:<port>/sensinact/<provider>/<service>/<resource>/UNSUBSCRIBE

HTTP Method
POST

Headers
Accept:application/json
Content-Type:application/json

Query
N/A

Content
{"parameters": [{"name":"subscriptionId","type":"string","value":"<subscription-identifier>"}]}
or
[{"name":"subscriptionId","type":"string","value":"<subscription-identifier>"}]

Response
{"response":{"message":"unsubscription done"},"type":"UNSUBSCRIBE_RESPONSE","uri":"/<provider>/<service>/<resource>","statusCode":200}

- **ACT**
  https://<host>:<port>/sensinact/providers/<provider>/services/<service>/resources/<resource>/ACT
  or
  https://<host>:<port>/sensinact/<provider>/<service>/<resource>/ACT

  HTTP Method
  POST

  Headers
  Accept:application/json
  Content-Type:application/json

  Query
  N/A

  Content
  []
  The list of parameters is actuation dependent

  Response
  {"response":{"task":"ACT", "start":<actuation-start-timestamp>,"end":<actuation-end-timestamp>,"uri":"/<provider>/<service>/<resource>","status":"(EXECUTED|ABORDED)"},"type":"ACT_RESPONSE","uri":"/<provider>/<service>/<resource>","statusCode":200}

  the response may also contain the triggered array, that gathered JSON formated descriptions of attributes modified by the actuation (it is possible to link ActionResources to StateVariableResources), as well as the potential returned result of the actuation

2.3 API

The programming interface of the FG benefits from the work carried out in the previous BigCloutT project in which a bridge between sensiNact and SOXFire was developed. More information is available on the public deliverable "D4.4 - Integrated BigClouT platform" available on the BigCloutT website[1].

Keio SOX Fire as an MQTT-based Publish / Subscribe model and sensiNact based on HTTP-based Restful architecture have different APIs at present, but they have the same function as a platform for basic IoT data distribution. Therefore, wrapping as a common API is not too difficult. Currently, there is no specific

[1] http://bigclout.eu/

requirement as a pilot, so such a wrapper has not been implemented yet, but we would like to consider a specific implementation as future development.

2.4 Interaction with other FGs

The following Figure 10 presents the interactions of the Secure City Data Access FG with other FGs and components of the M-Sec solution.

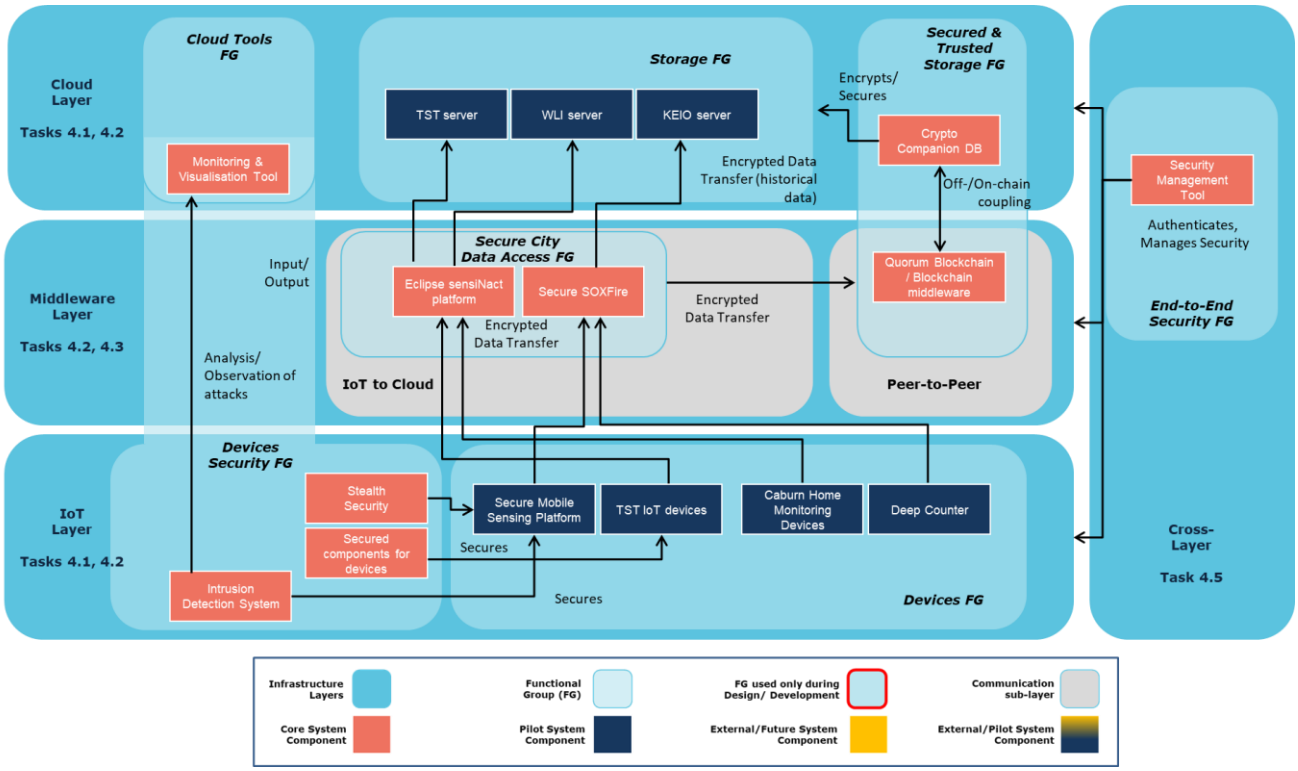

Figure 10. Interaction of the Secure City Data Access FG with other FGs

Interaction with Security manager

sensiNact is integrated with the security manager described in Deliverable 4.10 "*End to end security*" in order to have user authentication using the identity federation module implemented upon keycloak. After the initial configuration, two user entries were defined in a newly created **Exp** organization unit:

- sensiact-admin
- sensinact-user

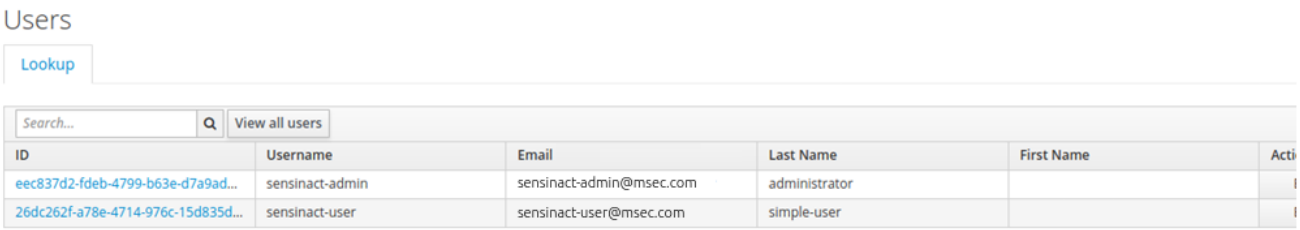

Figure 11. sensiNact test users displayed in the keycloak interface

We cloned the sensiNact's git source code repository:
`git://git.eclipse.org/gitroot/sensinact/org.eclipse.sensinact.gateway.git`

And, in the root directory of the newly created local repository, we built the sensiNact Gateway distribution using *maven*: `mvn clean install`

At the end of the `build process the distribution/generator/target/sensinact` contained the distribution used for the demonstration. **Paths mentioned below refer to this location.**

Configuration

We created the configuration file defining the oauth2 mechanism `./cfgs/sensinact-security-oauth2.config` and we defined the content as below:

```
discoveryURL=https://localhost:8443/auth/realms/test/.well-known/openid-configuration
certsURL=https://localhost:8443/auth/realms/test/protocol/openid-connect/certs
client_secret=31e1e909-e37a-48d5-8c76-74d7ca8e3b60
client_id=testClient
slider=admin:GET:/sensinact/slider/*
```

We next configured the HTTP service provided by the *felix* framework in `conf/config.properties`:

```
org.apache.felix.https.port=8899
org.apache.felix.http.debug=true
org.apache.felix.http.jettyEnabled=true
org.apache.felix.http.whiteboardEnabled=true
```

We also had to configure the OAuth2 configuration file in `conf/config.properties`:

```
org.eclipse.sensinact.security.oauth2.config=cfgs/sensinact-security-oauth2.config
```

and to copy the necessary modules:

```
cp load/simulation/slider-2.0-SNAPSHOT.jar ./bundle/
cp load/rest/*.jar ./bundle/
cp ../../../../platform/sensinact-security/sensinact-security-oauth2/target/*.jar ./bundle/
```

Finally, by launching sensiNact we were able to test all system allowing to configure the security: `./sensinact`

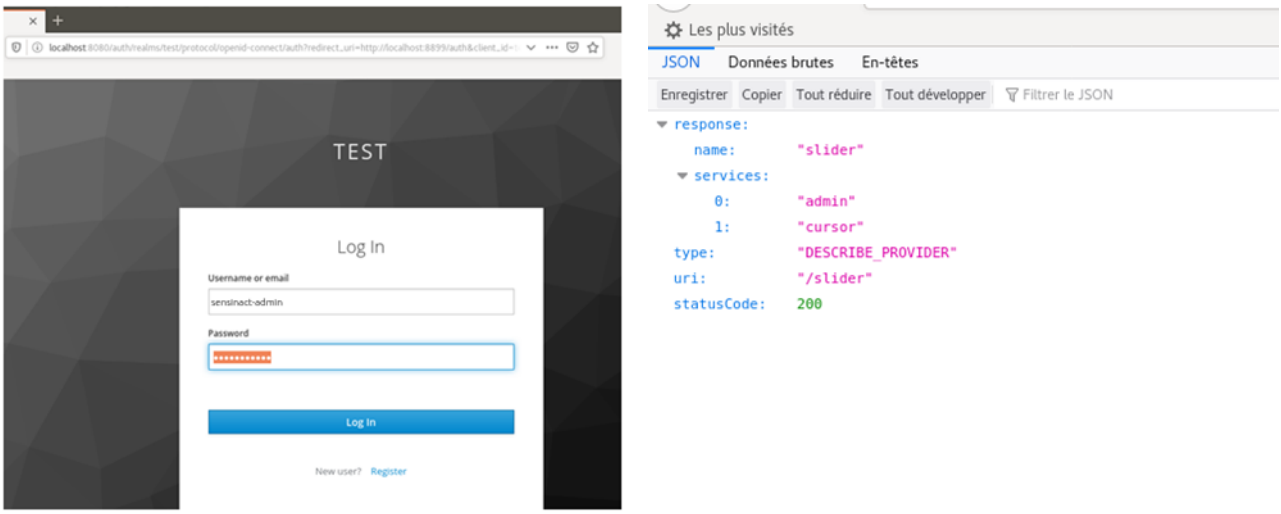

Figure 12. login as sensiNact-admin

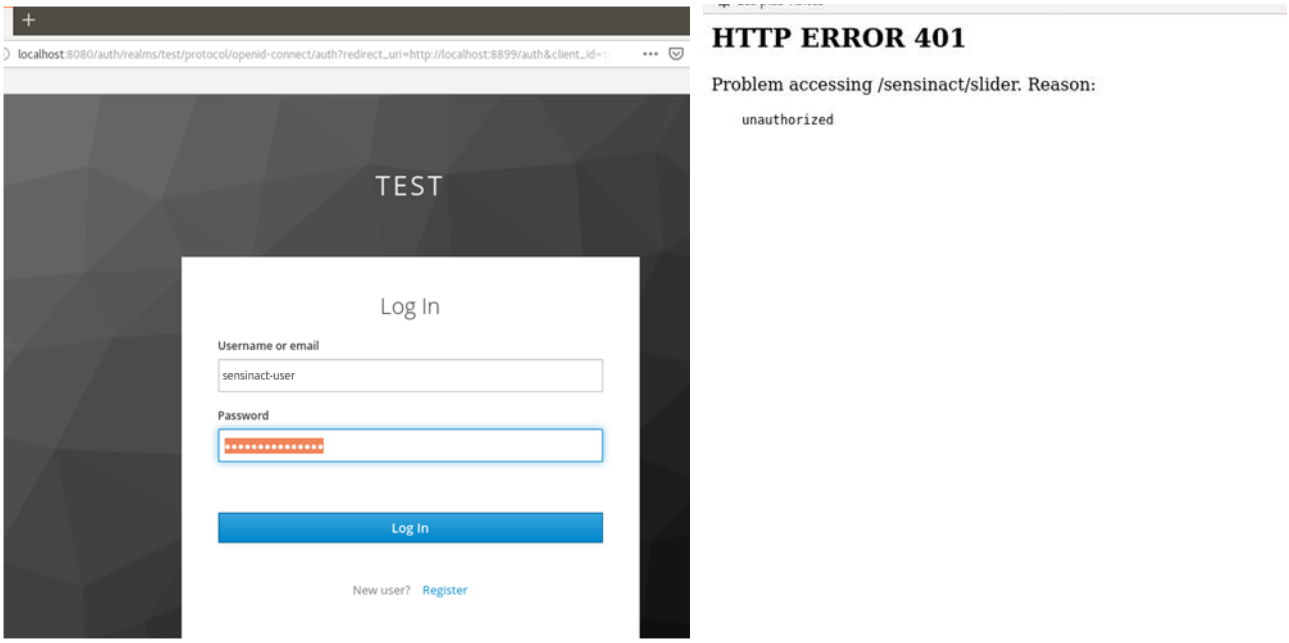

Figure 13. login as sensiNact user

Such integration enables to benefit from other layer's security, for example when a user or device is being revoked from the system.

Interaction with Devices Security

As an interaction with Devices Security FG, there is an example of integration between an IoT device (sensor) and Keio SOX Fire, which is a data distribution platform, in IoT sensing by a cleaning vehicle in UC3. The IoT devices security FG is on the Southbound. The data is securely transferred across the cloud using encryption

protection. For example, in case of Use Case 3, the data is securely transferred across the cloud using XMPP protocol that has a built-in encryption support in it.

The IoT devices security FG is on the Southbound. The data is securely transferred across the cloud using encryption protection. For example, in case of Use Case 3, the data is securely transferred across the cloud using XMPP protocol that has built-in encryption support in it.

Interaction with Secured and Trusted Storage FG and IoT Marketplace FG

As interaction with Secured and Trusted Storage FG, we considered the integration between Keio SOX Fire and IoT Marketplace. As a technical verification of the pilot, this time, the architecture was adopted as a prototype to pass the data from Keio SOXFire to IoT MarketPlace via the SOXFire – IoT MarketPlace Bridge. Ideally, it will be necessary to make it a common API with IoT Marketplace like a wrapper with a common API with SensiNact, and we would like to consider it as future development.

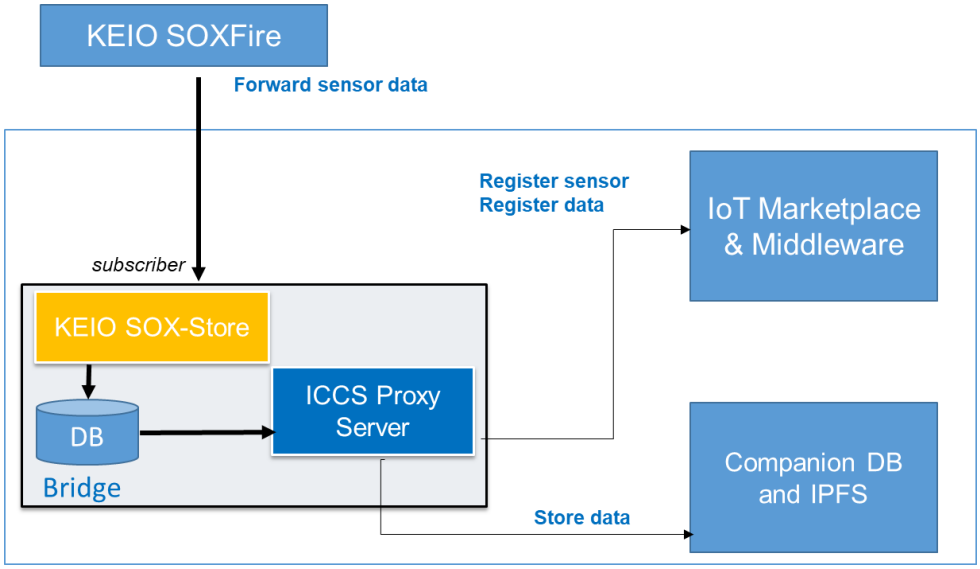

Figure 14.Integration of SOXFire with Trusted Storage FG

Figure 14 shows the proposed integration between SOXFire and two components of the Secured and Trusted Storage FG. In this figure, SOXFire receives data from sensors and act as a broker to send these data to a process called SOX-Store.

3. Privacy Management Tool FG

3.1 General Description of the FG

The privacy management functional group aims to address the requirements from GDPR & APPI standards for the privacy of personal identifiable information (PII) by anonymizing any such identifiable information from the video camera images being captured by the IP cameras mounted on smartphone applications or IoT devices.

In the following section, the components of the FG are described in detail. Section 1.2 presents the interactions of these components with components of other M-Sec FGs. Finally, the Annex presents the position of the FG within the whole M-Sec Architecture.

3.2 Components of the FG

GANonymizer

In situations where video data is used in various IoT application use cases such as smart cities, personal information is often a problem. GANonymizer is a technology that automatically deletes personal information contained in such videos using AI technology. GANonymizer is an imaging processing tool to remove (make transparent) the pedestrians and cars recorded in the driving record videos. Its objective is to avoid the privacy leakage when distributing and utilizing the videos. The GANonymizer is developed using deep learning-based object detection techniques and is currently designed to run securely on the KEIO's secured mobile sensing platform for use case 3 & 4, for addressing the privacy issue.

By using GANonymizer, the background image can be automatically filled as if there was no PII object, instead of a general mosaic method. GANonymizer has the following features.

- Automatically detect objects related to personal information or PII.
- In addition to deleting the target object, a background image is automatically generated as if the object did not exist.

Figure 15: GANonymizer Test Results

The results of applying GANonymizer are shown in Figure 15.

- Upper row images are original video images
- Lower row images are after removing privacy objects, like cars and human objects.

GANonymizer consists of two parts of neural networks. In order to detect the target objects from the input image, which might violate the GDPR/APPI privacy requirements, we adopt the deep neural networks: Single Shot Multibox Detector (SSD). And in order to generate a more natural image, we adopt Globally and Locally Consistent Image Completion (GLCIC), which is one of the most successful models in image completion.

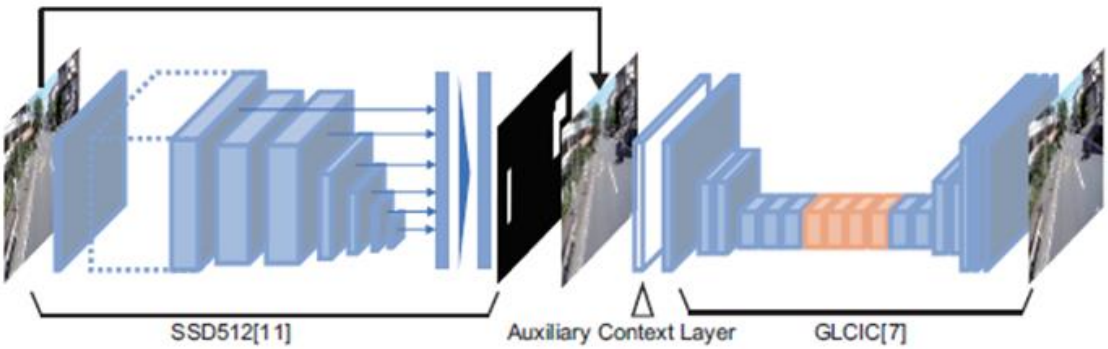

Figure 16. GANonymizer

Single Shot multibox Detector (SSD)

SSD is one of the popular models that can detect the object with high accuracy. Especially, we select SSD512 which is the variant SSD model and performs higher than any others. Since the target objects are general and those are contained in PascalVOC dataset, we use the model weights which are trained by PascalVOC.

Globally and Locally Consistent Image Completion (GLCIC)

After the target objects are detected, GANonymizer replaces the area where the target objects exist as if there are no objects. There are a lot of completion methods using computer vision technology. We adopt the in-painting methods which adopt the deep neural networks to succeed in generating the images more realistic and natural.

GLCIC is based on Generative Adversarial Networks (GAN) and consists of three networks: the completion network, a local discriminator network, and a global discriminator network. Since GLCIC requires an image and corresponding binary mask for its input, GANonymizer creates the mask based on the bounding boxes which are the outputs from SSD512. Then GLCIC reconstructs the mask part of the input image based on the whole image and is trained by the procedure of GAN. The local discriminator assesses the quality of the mask part of the input image which is completed by the completion network. Simultaneously, the global discriminator assesses the quality of the entire image which is completed by the completion network. The training is terminated when the discriminator networks cannot distinguish between the original input image and the image which is reconstructed by the completion network, that is when the completion network becomes able to reconstruct the mask part of the input image realistically and naturally.

In terms of object removal, it is significant to naturally reconstruct masks based on the various background of images, Hence, for our GLCIC, we apply the model trained with the places dataset, which contains the pictures of the various place, so that it can reconstruct the mask more naturally.

3.3 API

Endpoint : POST https://notus.ht.sfc.keio.ac.jp/image

Request

Headers

- Authorization : Username/Password for Basic Authentication
- Content-Type : application/json

Body (json)

- { “image”: “${base64 encoded image}” }

Response

Body (json)

- { “image”: “${base64 encoded image}” }

Example

Request image / Response image

4. Conclusion

In this document, we present two functional groups which enhance the security of data between the devices and their respective back-ends in complementary ways. At first, the Secure City Data Access functional group implements a security function that reduces the risks regarding *man-in-the-middle* attacks or other kinds of risks that are more and more sensitive in complex distributed IoT architectures. Finally, we present the privacy management FG that removes any individual or cars on video streams that can be captured in the public space.

At this stage, the security technologies demonstrated in this deliverable are under integration with the use cases as described in Table 2.

Table 2: Demonstrators and its correlation with Use Case Pilots

Demonstrator	Use Case Pilots	Purpose
sensiNact	Use Case 1 & 2	Provide embedded security layer to IoT devices
SOXFire	Use Cases 3 & 4	Deliver a reliable and trustworthy sensor data to authorized users via secured & trusted storage FG
Image Processing Tool for Video Privacy (GANonymizer)	Use Case 3 & 4	Enable privacy on the video feeds from IP cameras

Annex

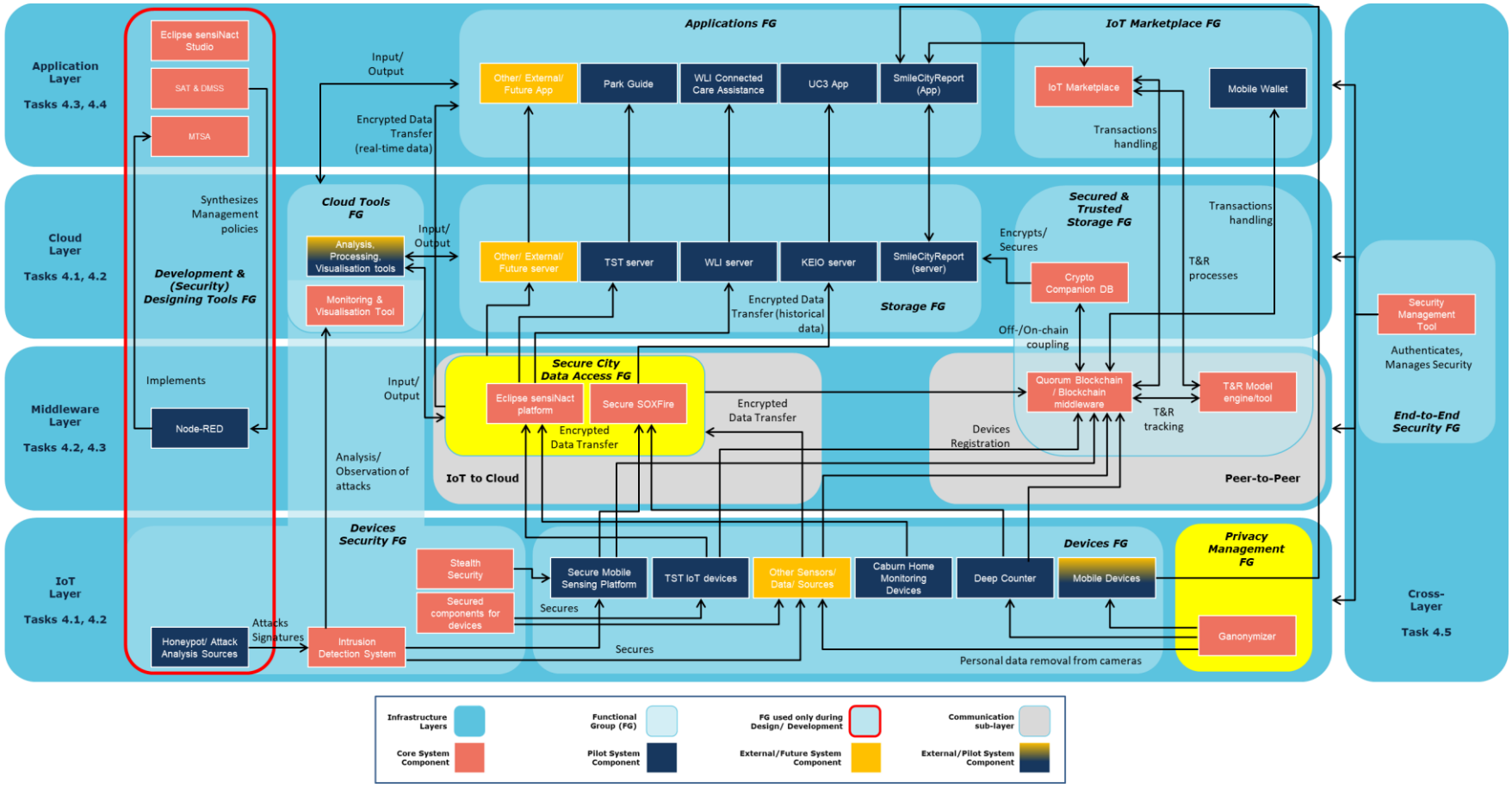

Figure 17. The M-Sec Architecture (T4.2 FG in yellow)