



*Smart system of renewable energy storage based on **IN**tegrated **EV**s and **bA**tteries to empower mobile, **D**istributed and centralised **E**nergy storage in the distribution grid*

Deliverable n°:	<b>D8.1</b>
Deliverable name:	<b>Cloud based flexibility management system: Flexibility Cloud, phase 1</b>
Version:	<b>1.0</b>
Release date:	<b>28/02/2018</b>
Dissemination level:	<b>Public</b>
Status:	<b>Approved</b>
Author:	Mark Fisher, Jon Egil Spetaas, Stig Ødegaard Ottesen, Terje Lundby – eSmart Systems AS



**Document history:**

Version	Date of issue	Content and changes	Edited by
0.0	01/02/2018	First completed draft version	Mark Fisher, Jon Egil Spetaas, Stig Ødegaard Ottesen and Terje Lundby
0.5	13/02/2018	Updated draft version	Jon Egil Spetaas, Terje Lundby
1.0	20/02/2018	Peer-reviewed version. To be submitted.	Terje Lundby

**Peer reviewed by:**

Partner	Reviewer
GreenFlux	Michel Bayings/Lennart Verheijen
Lyse	Oddbjørn Sørseth

**Deliverable beneficiaries:**

WP / Task
WP7/T7.2
WP8/T8.3, T8.4, T8.5, T8.6
WP10/T10.2

## Table of contents

<b>Executive summary .....</b>	<b>6</b>
<b>1 Introduction.....</b>	<b>7</b>
<b>2 INVADE main processes and their relations.....</b>	<b>8</b>
<b>3 Functionality .....</b>	<b>9</b>
3.1 Master data and configurations	10
3.1.1 Overview	10
3.1.2 Network, area and zone management	12
3.1.3 Site Management	20
3.1.4 Asset management	28
3.1.5 External System Cross Referencing management	38
<b>4 Data Repository .....</b>	<b>41</b>
4.1 Overview	41
4.2 Business Entities	41
4.3 Business Entity Connections	48
4.4 Assets	49
4.4.1 General Asset Properties	49
4.4.2 Area Asset	51
4.4.3 Resource Asset	52
<b>5 Platform Architecture .....</b>	<b>56</b>
5.1 Overview	56
5.2 Changes to the architecture detailed in D4.1 Concept Design	57
5.2.1 Influx Time Series DB	57
5.3 Interfaces	57
5.4 Flexibility Could Security	58

5.5	Data Management APIs	58
5.5.1	Overview	58
5.5.2	External Facing Platform Integration API for Asset management	58
5.5.3	Entity Cross referencing to External systems	59
<b>6</b>	<b>GDPR.....</b>	<b>61</b>

**Abbreviations and Acronyms**

<b>Acronym</b>	<b>Description</b>
AAD	Azure Active Directory
API	Application programming interface
BRP	Balance Responsible Party
DSO	Distribution System Operator
EV	Electric Vehicle
FC	Flexibility Cloud
FCS	Flexibility Cloud Software
FES	Front End Systems
FO	Flexibility Operator
IIP	Integrated INVADE Platform
GDPR	General Data Protection Regulation
NA	Not Applicable
PV	Photovoltaic
SM	Smart Meter
TBD	To Be Determined
ToU	Time-of-Use
TSO	Transmission System Operator
WP	Work Package

## Executive summary

In the INVADE project, a central delivery is the cloud-based Integrated INVADE platform, which will be used by the Flexibility Operator to manage flexibility from flexibility providers and offer this to flexibility customers.

The first version of the Integrated INVADE platform will be delivered in June 2018 and implemented at pilots during the autumn 2018. The fundamental parts of the Integrated INVADE platform, namely the asset management, asset control and data repository, is presented in this delivery. It will be followed up with a second delivery, which will focus on the business intelligence in the Integrated INVADE platform. This delivery is due in June 2018.

This deliverable is structured as follows: First an overview of the main processes which are needed in the Flexibility Cloud is presented. Then, the functionality of the main process Master data and configuration is described. Further, the data repository and the key entities related to asset management is shown, and finally an updated version of the platform architecture is presented.

A main finding in the document is that the management of the fundamental data is a major part of the Integrated INVADE platform, and it is crucial that this is designed to work efficiently.

# 1 Introduction

This document contains the deliverable *D8.1 Cloud based flexibility management system: Flexibility Cloud, phase 1*. According to the DoA, the deliverable covers three major parts: asset management, asset control and data repository, as implemented in the tasks *T8.1 Implementation of the Flexibility Cloud data architecture* and *T8.2. Implementation of the Flexibility Cloud control architecture*.

The document builds on the already existing eSmart platform, which was partly developed in the H2020 EMPOWER project, and input from INVADE deliverables: D4.1, D4.2, D5.3, D7.1 and D10.1.

The main purpose with the current deliverable is to document the basic parts of the Flexibility Cloud (as defined above). The intended audience of the deliverable is mainly the INVADE project partners to whom the implementation of the flexibility management system is relevant.

As stated above, D8.1 contains the basic parts of the Flexibility Cloud. The document will be followed by *D8.2 Cloud based flexibility management system: Flexibility Cloud, phase 2*, which will be delivered in month 18 covering the tasks *T8.3 Implementation of the Flexibility Cloud flexibility management algorithms, functions and monitoring and control dashboards* and *T8.4 Integration of Electric Vehicles in the Flexibility Cloud*. This means that D8.1 contains the fundamental parts, while D8.2 contains the business intelligence.

The rest of this document is organized as follows: Chapter 2 contains an overview of the business processes that the Flexibility Cloud will support. Chapter 3 describes the functionality covered, while Chapter 4 contains the data repository and chapter 5 details about the overall platform architecture.

## 2 INVADE main processes and their relations

The main objective with the Integrated INVADE Platform (IIP) is to support the Flexibility Operator (FO) delivering flexibility services from the flexibility vendors (the prosumers) to the flexibility customers. An overview of the different flexibility customers, flexibility services and the different pilots are listed in Table 1, which is copied from *D4.2 INVADE architecture of pilots*.

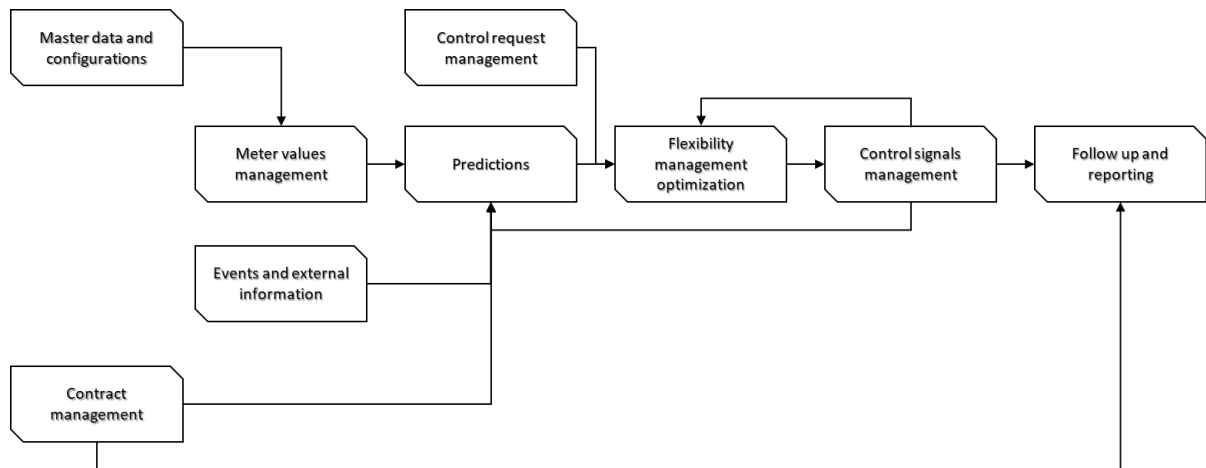
**Table 1: Flexibility services to be used in each pilot**

Flexibility customer	Flexibility services INVADE	Norwegian pilot	Dutch pilots	Bulgarian pilot	German pilot	Spanish pilot
DSO	Congestion management	N	Y	N	Y	Y
	Voltage / Reactive power control	N	Y	N	Y	Y
	Controlled islanding	N	N	N	TBD	Y
BRP	Day-ahead portfolio optimization	N	Y	TBD	N	TBD
	Intraday portfolio optimization	N	Y	TBD	N	Y
	Self-balancing portfolio optimization	N	Y	TBD	TBD	Y
Prosumer	ToU optimization	Y	Y	Y	Y	TBD (phase 2)
	kWmax control	Y	Y	Y	Y	TBD (phase 2)
	Self-balancing	Y	Y	Y	Y	TBD (phase 2)
	Controlled islanding	TBD	N	TBD	Y	N

It is, however, not finally decided which services that will be in phase 1 and 2. A working assumption is that the prosumer services for ToU optimization, kWmax control and Self-balancing plus the DSO service Congestion management will be in phase 1. These are the only services that will be covered in this document. Notice that the term *phase* is a bit confusing, since it is used in two different contexts in the project. *Phase 1* in the table above relates to what will be implemented at the pilot sites in month 18, while *Phase 1* in the name of the current deliverable is the first part of what will be delivered in month 18. For this reason, we will avoid using the term *phase* in the rest of this document.

To support the four flexibility services listed above, a set of what we can denote *main processes* will be needed in the Flexibility Cloud. Each main process can be seen as a business process or an overall function. Figure 1 shows an overview of the main processes and how they are linked together.





**Figure 1. Overview of Flexibility Cloud main processes**

Each main process will be further split into sub-processes.

Note that the figure shows all main functions that will be covered in D8.1 and D8.2 together. D8.1 contains (parts of) the main processes Master data and configurations.

### 3 Functionality

This chapter describes the functionality for the related processes. Each main process is described with an overview, that gives an overall description of the content, including the purpose, the split into sub-processes and, if relevant, a flow chart or sequence diagram. Then, each sub-process is further described according to the following structure:

- Purpose
- Involved roles. Describes which roles (entities, systems and different types of users)
- Preconditions. Things that must be in place before initiating the process
- Main steps/list of user stories. Defines the steps at an overall level. Also lists the user stories (will be further detailed later)
- Functions. Describes any functionality in terms of calculation, validation, algorithm etc
- User interfaces. Lists user interfaces that are needed to support the functionality

### 3.1 Master data and configurations

#### 3.1.1 Overview

The purpose with Master data and configurations is to enter and maintain all fundamental data, meaning data that are static or that are changed rarely (could also be denoted structure data). Examples of such data are information about assets, sites, areas and grid related information.

The main process is split into sub-processes as illustrated in Figure 2.

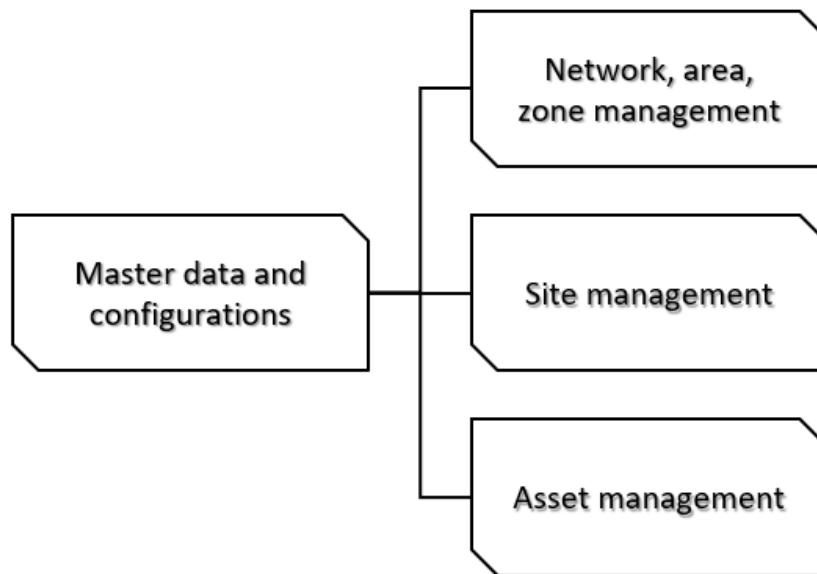


Figure 2. Main process and sub-processes for Master data and configurations

##### 3.1.1.1 Definitions

The following diagram shows a typical topology that can be configured, between areas, sites and resource assets.

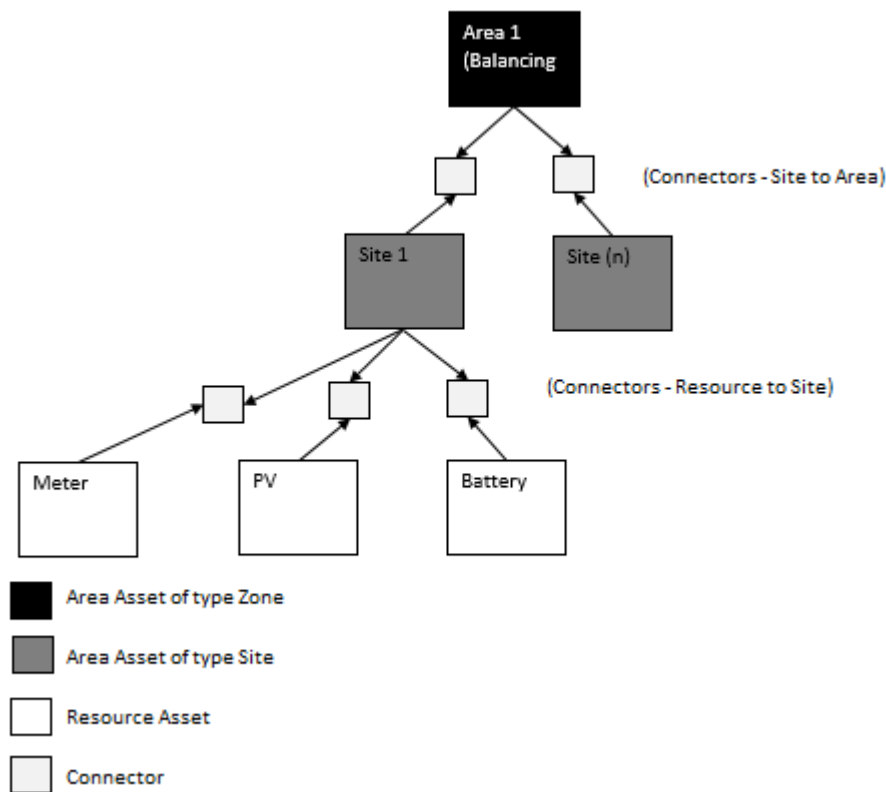


Figure 3. Relationships between Area, Sites and Resource Assets

## Definitions

Within the Flexibility Cloud (FC) data repository, there is a generic data type called Asset that is used to model entities that have common attributes, traits or relationships. For the purpose of the pilots, the generic Asset entity type will be used to model Areas, Sites And resource assets.

**Area** is defined as to be a grouping entity with an Area Type classification. Typical area type classifications could be DSO Zone, Balancing Area, Flexibility Zone, Weather, Market. The exact area types required for each pilot set up is configurable and will be determined later in the D8.2 delivery.

Typically, the area will represent a grouping where we want to have summarized data and collect data that can be utilized by the relations of what sites are within the area, for

example to find the weather forecast for a site based on its connection to a Weather area.

**Site:** is defined to be a grouping entity with an Area type classification of Site. Typical site type classifications could be a House, Building, Charging Site, Industrial Facility, Car Park and probably others. The exact site types required for each pilot set up is configurable and will be determined later in the D8.2 delivery.

The site will represent a grouping where we want to have summarized data for a set of resource assets that can be managed collectively within the site, and where we can manage any mixture of generation, consumption, storage, and charging and related flexibility collectively. Each site generally will be configured with one Main Meter (where the main meter meters the net exchange with the grid, i.e. the purchase and the sales).

**Resource Asset:** is defined to be a representation a physical or virtual asset (or collection of assets) that have a classification type of generation, consumption, storage, charging or monitoring. See section “Resource Asset” for further details about each type of resource that can be modelled. Examples of resource assets include PV, Battery, Meter, Water heaters, Wind turbines.

### **Site to Area Connections**

In general, all site assets will be connected to Areas (via asset connector of type Area), validation ensures a Site has only one active connection to one Area (of a particular classification type) over any period of time.

### **Resource to Site Connections**

In general, all Resource assets will be connected to a site (via asset connector of type Site). Validation ensures a resource asset has only one active connection to one Site over any period of time.

## **3.1.2 Network, area and zone management**

### **3.1.2.1 Purpose**

The purpose of this process is to enter, update and remove information about areas and their connectivity to sites. Here an area will be of a certain configured area type such as DSO Zone, Balancing Area, Flexibility, Weather Area, Market (Price Area) for instance.

This provides an aggregation point for summarizing data. Validation will ensure a site can only have one active connection to one area of each type over the connection period. E.g. A site can be connected to both a Flexibility Area & a Weather Area over the same period but not to more than one Weather area and one Flexibility area over the same period of time.

#### 3.1.2.2 Involved roles

Some of the user stories involve a FO user role, while some are automated and hence involve only system roles.

#### 3.1.2.3 Preconditions and Assumptions

The input-data that are going to be input must be ready at a pre-defined format.

#### 3.1.2.4 Main steps/list of user stories

There are two main ways to enter new information: 1) Manually by FO user or 2) Automatic by import. The manual process will typically be like this:

1. The FO receives (outside the Flexibility Cloud) all necessary parameters for one or several areas
2. The FO opens a screen and enters the parameters
3. The parameters are validated and stored

The automated process will be like this:

1. An external System or FO receives (outside the Flexibility Cloud) all necessary parameters for one or several areas
2. The external System or FO calls the FC Data Management API providing an Area or list of Areas and all necessary parameters, and control options to save the areas
3. The parameters are validated and stored
4. The system returns to the caller details about the data saved, including the internal system keys of the areas stored and any error information related to the supplied area details, if the area could not be stored

The following user stories are covered:

1. Enter new area, via UI
2. Update area, via UI

3. Remove area, via UI
4. Reinstate area, via UI
5. Create, Update, Delete & Reinstate area, via API
6. Connect Area to External System, via UI

#### 3.1.2.5 Enter new area, via UI

##### **Purpose**

To capture the definition of a new area, including all its properties and optionally its connectivity to sites within the area.

##### **Description**

As an operator of the system I want to be able to

- Add new areas with its related properties to the system.
- Manage Site to Area connections (optionally)
  - Connect Sites to Areas (of different types)
  - Manage the Connectivity to Areas in terms of valid from/to dates (optional) and its status (active/deleted)

Note: Area connectivity to Sites can also be managed when a Site is created/amended in the associated use case under Site Management.

##### **Roles**

FO configures a new area into the system.

##### **Preconditions and Assumptions**

- Any sites to be connected to the area have already been defined in the system

##### **Acceptance Criteria**

- Connectivity to (child) Sites is validated to ensure there are no active connections to the same area type with overlapping date ranges
- On supply of valid data, the area is persisted to the systems data repository, else get an appropriate error message

### 3.1.2.6 Update area, via UI

#### **Purpose**

To capture amendments to the definition of an existing area, including all its properties and optionally its connectivity to sites within the area.

#### **Description**

As an operator of the system I want to be able to

- Locate the area to be amended via a set of search criteria
- Amend the areas related properties.
- Manage Site to Area connections (optionally)
  - Connect Sites to Areas (of different types)
  - Manage the Connectivity to Areas in terms of valid from/to dates (optional) and its status (active/deleted)

Note: Area connectivity to Sites can also be managed when a Site is created/amended in the associated use case under Site Management.

#### **Roles**

FO reconfigures, enhances the configuration of an existing area in the system.

#### **Preconditions and Assumptions**

- Any Sites to be connected to the area have already been defined in the system

#### **Acceptance Criteria**

- Connectivity to (child) Sites is validated to ensure there are no active connections to the same area type with overlapping date ranges
- On supply of valid data, the area is persisted to the systems data repository, else get an appropriate error message

### 3.1.2.7 Remove area, via UI

#### **Purpose**

To indicate an area is deleted.

Note: the system will only tag entities as deleted in the system. This will restrict them from being shown in the UI unless the user sets a system option to Show Deleted data.

### **Description**

As an operator of the system I want to be able to

- Locate the area to be deleted via a set of search criteria
- Mark the area as deleted (by setting its entity status to Deleted).

### **Roles**

FO decides when an area is to be marked as deleted, additionally they have the option to phase out the use of an area by setting its valid from and to dates as an alternative.

### **Preconditions and Assumptions**

- None

### **Acceptance Criteria**

- Area is persisted to the systems data repository with entity status of Deleted.

#### 3.1.2.8 Reinstate deleted area, via UI

### **Purpose**

To indicate an area is Active.

Note: the system will only tag entities as deleted in the system. This will restrict them from being shown in the UI unless the user sets a system option to Show Deleted data. If an area has been deleted in error, it can easily be reactivated via this user story.

### **Description**

As an operator of the system I want to be able to

- Locate the area to be reinstated via a set of search criteria (that includes the retrieval of deleted areas)
- Mark the area as Active (by Setting its entity status to active).
- When reactivating an area, the normal validation that is applied on adding/editing an area will be verified.



Note: For Deleted areas it is useful to see all the areas data (as per area amendment), hence the reactivation is achieved by using the same UI form as for Site amendment, with the entity status being available for amendment.

## Roles

FO decides if to reactive an area.

## Preconditions and Assumptions

- None

## Acceptance Criteria

- Connectivity to (child) Sites is validated to ensure there are no active connections to the same area type with overlapping date ranges
- On supply of valid data, the area is persisted to the systems data repository (with entity status Active), else get an appropriate error message

### 3.1.2.9 Create, update, delete & reinstate Area, via API

#### Purpose

To capture the definition of a new area, amendment and deletion of existing area (via system to system integration or via external load of data), including all its properties and optionally its connectivity to the sites within the area.

It provides the same basic functionality as user stories “Enter new area, via UI”, “Update area, via UI”, “Remove area, via UI”, “Reinstate deleted area, via UI” but via a system API interface. Additionally, the cross referencing of the Area entity to any external Area representation with a different external key is handled in this user story.

#### Description

As an integrator of data into the system I want to be able to

- Add, Amend, Delete and reinstate areas with their related properties to the system.
- Manage Site to Area connections (optionally)
  - Connect Sites to Areas (of different types)

- Manage the Connectivity to Areas in terms of valid from/to dates (optional) and its status (active/deleted)
- Manage the Connectivity to External System (cross referencing) (optionally)
  - Connect External systems primary key to the FC Area key, thus enabling the client to provide the look up of the area to the API either with the external keys or FC internal keys.

### **Roles**

- System to System integration
- Pilot Site set up via scripting

### **Preconditions and Assumptions**

- Any Sites to be connected to the area have already been defined in the system

### **Acceptance Criteria**

- Connectivity to (child) Sites is validated to ensure there are no active connections to the same area type with overlapping date ranges
- On supply of valid data, the area is persisted to the systems data repository, else get an appropriate error message
- On supply of valid data, the area is persisted to the systems data repository and the area FC reference is returned, else get an appropriate error message
- The API must support partial data upload, via control parameters the message supports options to indicate what data is being provided. Thus, providing an API that can support both full reconfiguration of an area or partial amendment to an area, e.g. To Add a new site connected to the area and not supplying all the other data.

### **API interface**

- The management of areas will be via the Asset management API defined in section “Asset management”.

### 3.1.2.10 Cross reference area to External System, via UI

#### **Purpose**

To cross reference an area in the FC via its internal unique key to an external systems unique key. This enable the API to accept data with external system references and to manage the look ups to the FC internal entity via the cross referencing.

See user story in section “External System Cross Referencing management” for details.

### 3.1.2.11 Functions

Validations:

1. Name is mandatory, Valid From Date < Valid To date (both optional)
2. Site can only be associated to one Area over any period of time for a particular area connection type. For example, can be associated to one weather area and one DSO area, concurrently, but if associate to two DSO areas these must be for different date ranges.

### 3.1.2.12 User interfaces

User interface	Short description
Entity Overview	Provides a list of all entities in the system, with search criteria to find entities of a certain type e.g. Sites, Areas, Resource Assets (Generation, Load, Charging, Storage, Monitoring), additional elastic search criteria will be available to search across properties e.g. for a string of characters.
Area management	Provides a single screen for the management of areas (create, update, delete, undelete), including <ul style="list-style-type: none"> <li>▪ its properties</li> <li>▪ its entity status (Active, deleted)</li> <li>▪ its site connections to areas</li> </ul>

### 3.1.3 Site Management

#### 3.1.3.1 Purpose

The purpose of this process is to enter, update and remove information about sites and their connectivity to Areas and connectivity to (Resource) Assets within the site. Here a site is a location where resource assets are grouped together, typically each site will have one main meter (where the main meter meters the net exchange with the grid, i.e. the purchase and the sales at the site) and a set of resources that are of types Generation, Load, Charging, Storage or Monitoring.

Each site will typically be connected to areas that represent a collection of sites such as DSO Zones, Flexibility Areas, Weather Areas. This provides an aggregation point for summarizing data.

Example of sites could be a House, Building, Charging Site, Car Park.

Typically, sites will be connected via contracts to contract partners, like DSOs and Electricity suppliers and the Customer, that is outside the scope of this user story.

#### 3.1.3.2 Involved roles

Some of the user stories involve a FO user role, while some are automated and hence involve only system roles.

#### 3.1.3.3 Preconditions and Assumptions

The input-data that are going to be input must be ready at a pre-defined format.

#### 3.1.3.4 Main steps/list of user stories

There are two main ways to enter new information: 1) Manually by FO user or 2) Automatic by import. The manual process will typically be like this:

1. The FO receives (outside the Flexibility Cloud) all necessary parameters for one or several sites
2. The FO opens a screen and enters the parameters
3. The parameters are validated and stored

The automated process will be like this:

1. An external System or FO receives (outside the Flexibility Cloud) all necessary parameters for one or several sites

2. The external System or FO calls the FCS Data Management API providing a Site or list of Sites and all necessary parameters, and control options to save the sites
3. The parameters are validated and stored
4. The system returns to the caller details about the data saved, including the internal system keys of the sites stored and any error information related to the supplied site details, if the site could not be stored

The following user stories are covered:

1. Enter new site, via UI
2. Update site, via UI
3. Remove site, via UI
4. Reinstate site, via UI
5. Create, Update, Delete & Reinstate Site, via API
6. Connect Site to External System, via UI

#### 3.1.3.5 Enter new site, via UI

##### **Purpose**

To capture the definition of a new site, including all its properties and optionally its connectivity to areas and connectivity to the resource assets at the site.

##### **Description**

As an operator of the system I want to be able to

- Add new sites with its related properties to the system.
- Manage Site to Area connections (optionally)
  - Connect Sites to Areas (of different types)
  - Manage the Connectivity to Areas in terms of valid from/to dates (optional) and its status (active/deleted)
- Manage the Connectivity to Resource Assets at the site (optionally)
  - Connect resource Assets to Site
  - Manage the Connectivity to resource Assets in terms of valid from/to dates (optional) and its status (active/deleted)

Note: Resource Assets connectivity to Sites can also be managed when an Asset is created/amended in the associated use case under Asset Management

### **Roles**

FO configures a new site into the system.

### **Preconditions and Assumptions**

- Any Areas the site is to be connected to have already been defined in the system
- Any Resource Assets to be attached to the site have already been defined in the system

### **Acceptance Criteria**

- Connectivity to (parent) Areas is validated to ensure there are no Active connections to the same area type with overlapping date ranges
- Connectivity to (child) Resource Assets is validated to ensure there are no Active connections to the same Asset type with overlapping date ranges, on this site or any other site.
- On supply of valid data, the site is persisted to the systems data repository, else get an appropriate error message

#### 3.1.3.6 Update site, via UI

### **Purpose**

To capture amendments to the definition of an existing site, including all its properties and optionally its connectivity to areas and connectivity to the resource assets at the site.

### **Description**

As an operator of the system I want to be able to

- Locate the site to be amended via a set of search criteria
- Amend the sites related properties.
- Manage Site to Area connections (optionally)
  - Connect Sites to Areas (of different types)

- Manage the Connectivity to Areas in terms of valid from/to dates (optional) and its status (active/deleted)
- Manage the Connectivity to Resource Assets at the site (optionally)
  - Connect resource Assets to Site
  - Manage the Connectivity to resource Assets in terms of valid from/to dates (optional) and its status (active/deleted)

Note: Resource assets connectivity to Sites can also be managed when an Asset is created/amended in the associated use case under asset management

## Roles

FO enhances the configuration of an existing site in the system.

## Preconditions and Assumptions

- Any Areas the site is to be connected to have already been defined in the system
- Any Resource Assets to be attached to the site have already been defined in the system

## Acceptance Criteria

- Connectivity to (parent) Areas is validated to ensure there are no Active connections to the same area type with overlapping date ranges
- Connectivity to (child) Resource Assets is validated to ensure there are no Active connections to the same Asset type with overlapping date ranges, on this site or any other site.
- On supply of valid data, the site is persisted to the systems data repository, else get an appropriate error message

### 3.1.3.7 Remove site, via UI

#### Purpose

To indicate a site is deleted.

Note: the system will only tag entities as deleted in the system. This will restrict them from being shown in the UI unless the user sets a system option to Show Deleted data.

**Description**

As an operator of the system I want to be able to

- Locate the site to be deleted via a set of search criteria
- Mark the site as deleted (by Setting its entity status to Deleted).

**Roles**

FO decides when a site is to be marked as deleted, additionally they have the option to phase out the use of a site by setting its valid from and to dates as an alternative.

**Preconditions and Assumptions**

- None

**Acceptance Criteria**

- Site is persisted to the systems data repository with entity status of deleted.

**3.1.3.8 Reinstate deleted site, via UI****Purpose**

To indicate a site is Active.

Note: the system will only tags entities as deleted in the system. This will restrict them from being shown in the UI unless the user sets a system option to Show Deleted data. If a site has been deleted in error, it can easily be reactivated via this user story.

**Description**

As an operator of the system I want to be able to

- Locate the site to be reinstated via a set of search criteria (that includes the retrieval of deleted sites)
- Mark the site as Active (by Setting its entity status to Active).
- When reactivating a site, the normal validation that is applied on adding/editing a site will be verified.

Note: For Deleted sites it is useful to see all the Sites data (as per site amendment), hence the reactivation is achieved by using the same UI form as for Site amendment, with the entity status being available for amendment.



**Roles**

FO decides if to reactive a Site.

**Preconditions and Assumptions**

- None

**Acceptance Criteria**

- Connectivity to (parent) Areas is validated to ensure there are no Active connections to the same area type with overlapping date ranges
- Connectivity to (child) Resource Assets is validated to ensure there are no Active connections to the same Asset type with overlapping date ranges, on this site or any other site.
- On supply of valid data, the site is persisted to the systems data repository (with entity status Active), else get an appropriate error message

**3.1.3.9 Create, update, delete & reinstate Site, via API****Purpose**

To capture the definition of a new site, amendment and deletion of existing site (via system to system integration or via external load of data), including all its properties and optionally its connectivity to areas and connectivity to the resource assets at the site.

It provides the same basic functionality as user stories “Enter new site, via UI”, “Update site, via UI”, “Remove site, via UI”, “Reinstate deleted site, via UI” but via a system API interface. Additionally, the cross referencing of the Site entity to any external Site representation with a different external key is handled in this user story.

**Description**

As an integrator of data into the system I want to be able to

- Add, Amend, Delete and reinstate sites with their related properties to the system.
- Manage Site to Area connections (optionally)
  - Connect Sites to Areas (of different types)
  - Manage the Connectivity to Areas in terms of valid from/to dates (optional) and its status (active/deleted)

- Manage the Connectivity to Resource Assets at the site (optionally)
  - Connect resource Assets to Site
  - Manage the Connectivity to resource Assets in terms of valid from/to dates (optional) and its status (active/deleted)
- Manage the Connectivity to External System (cross referencing) (optionally)
  - Connect External systems primary key to the FCS Site key, thus enabling the client to provide the look up of the site to the API either with the external keys or FCS internal keys.

### **Roles**

- System to System integration
- Pilot Site set up via scripting

### **Preconditions and Assumptions**

- Any Areas the site is to be connected to have already been defined in the system
- Any Resource Assets to be attached to the site have already been defined in the system

### **Acceptance Criteria**

- Connectivity to (parent) Areas is validated to ensure there are no Active connections to the same area type with overlapping date ranges
- Connectivity to (child) Resource Assets is validated to ensure there are no Active connections to the same Asset type with overlapping date ranges, on this site or any other site.
- On supply of valid data, the site is persisted to the systems data repository and the Site FCS reference is returned, else get an appropriate error message
- The API must support partial data upload, via control parameters the message supports options to indicate what data is being provided. Thus, providing an API that can support both full reconfiguration of a Site or partial amendment to a site, e.g. To Add a new Site connector and not supplying all the other data.

## API interface

- The management of sites will be via the Asset management API defined in section “Asset management”, in particular this user story SaveAsset and SaveAssetList methods, that accept a JSON representation of the Site Asset.

### 3.1.3.10 Cross reference site to External System, via UI

#### Purpose

To cross reference a Site in the FC via its internal unique key to an external systems unique key. This enable the API to accept data with external system references and to manage the look ups to the FC internal entity via the cross referencing.

See user story in section “External System Cross Referencing management” for details.

### 3.1.3.11 Functions

Validations:

1. Name is mandatory, Valid From Date < Valid To date (both optional)
2. Site can only be associated to one Area over any period of time for a particular area connection type. For example, can be associated to one weather area and one DSO area, concurrently, but if associate to two DSO areas these must be for different date ranges.
3. When linking Resource Asset to Sites a resource Asset can only be associated to one active Site over any period of time.

### 3.1.3.12 User interfaces

User interface	Short description
Entity Overview	Provides a list of all entities in the system, with search criteria to find entities of a certain type e.g. Sites, Areas, Resource Assets (Generation, Load, Charging, Storage, Monitoring), additional elastic search criteria will be available to search across properties e.g. for a string of characters.
Site management	Provides a single screen for the management of Sites (create, update, delete, undelete), including

	<ul style="list-style-type: none"> <li>▪ its properties</li> <li>▪ its entity status (Active, deleted)</li> <li>▪ its connections to areas and resource assets</li> </ul>
--	---

### 3.1.4 Asset management

#### 3.1.4.1 Purpose

The purpose of this process is to enter, update and remove information about resource assets and their connectivity to sites. Here a site is a location where resource assets can be grouped together, typically each site will have one main meter (where the main meter meters the net exchange with the grid, i.e. the purchase and the sales at the site) and a set of resources that are of types Generation, Load, Charging, Storage or Monitoring.

Each resource relevant to the pilot will be added to the system and connected to its related site.

#### 3.1.4.2 Involved roles

Some of the user stories involve a FO user role, while some are automated and hence involve only system roles.

#### 3.1.4.3 Preconditions

The input-data that are going to be input must be ready at a pre-defined format.

#### 3.1.4.4 Main steps/list of user stories

There are two main ways to enter new information: 1) Manually by FO user or 2) Automatic by import. The manual process will typically be like this:

1. The FO receives (outside the Flexibility Cloud) all necessary parameters for one or several resource assets
2. The FO opens a screen and enters the parameters
3. The parameters are validated and stored

The automated process will be like this:

1. An external System or FO receives (outside the Flexibility Cloud) all necessary parameters for one or several resource assets
2. The external System or FO calls the FC Data Management API providing a resource asset or list of assets and all necessary parameters, and control options to save the assets
3. The parameters are validated and stored
4. The system returns to the caller details about the data saved, including the internal system keys of the assets stored and any error information related to the supplied asset details, if the asset could not be stored.

The following user stories are covered:

1. Enter new resource asset, via UI
2. Update resource asset, via UI
3. Remove resource asset, via UI
4. Reinstate resource asset, via UI
5. Create, Update, Delete & Reinstate resource asset, via API
6. Connect resource asset to External System, via UI

#### 3.1.4.5 Enter new resource asset, via UI

##### **Purpose**

To capture the definition of a new resource asset, including all its properties and optionally its connectivity to sites.

##### **Description**

As an operator of the system I want to be able to

- Add new resource assets with its related properties to the system.
- Resource assets are classified in terms of resource types, so when defining a resource asset the resource type will be selected (Generation, Load, Storage, Charging or Monitoring). Then depending on the resource type a set of properties associated with that resource type will be captured (see Data Repository section “Resource Asset” for details of the particular properties for each resource type).

- Resource assets can additionally be controllable, if it is controllable an additional set of the properties related to its controllability are captured (see Data Repository section “Resource Assets that are Controllable” for details).
- Resource assets can additionally support monitoring, if the resource support monitoring (captures metering data), an additional set of the properties related to its metering/monitoring are captured (see Data Repository section “Resource Assets supporting Monitoring” for details).
- Resource assets can model a physical resource or can be a virtual to model for instance all load at a site where each individual physical load device is not monitored separately. In the case of modelling a physical device:
  - a set of properties related to its installation details can be captured. These properties are not relevant to the Invade pilots so will not be detailed further in the document.
  - a set of properties related to how it is communicated with can be captured. These properties are not relevant to the Invade pilots so will not be detailed further in the document.
- Manage resource to Site connections (optionally)
  - Connect resource assets to sites (only one active connection will be allowed over any period of validity)
  - Manage the Connectivity to sites in terms of valid from/to dates (optional) and its status (active/deleted)

Note: Resource Assets connectivity to Sites can also be managed when a Site is created/amended in the associated use case under Site Management.

## **Roles**

FO configures a new resource Asset into the system.

## **Preconditions and Assumptions**

- Any Sites the resource asset is to be connected to have already been defined in the system

**Acceptance Criteria**

- Connectivity to (parent) Site is validated to ensure there are no Active connections from the same resource Asset with overlapping date ranges, to any site.
- On supply of valid data, the resource asset is persisted to the systems data repository, else get an appropriate error message

**3.1.4.6 Update resource asset, via UI****Purpose**

To capture amendments to the definition of a resource asset, including all its properties and optionally its connectivity to sites.

**Description**

Will support the same functionality as enter new Resource asset, described in the previous section but in addition:

- As a starting point must be able to locate the Asset to be amended via a set of search criteria
- Switching Resource type, e.g. changing resource from a Load resource to a generation resource, would only normally be done if the wrong category was selected by error when adding the asset. If data such as generation data was already captured, and you switched it to a load type, then we need to protect against an inconsistent data model. To achieve this the Resource type selection will be disabled on edit, if time series data has been captured.

**Roles**

FO reconfigures, enhances the configuration of an existing resource Asset in the system.

**Preconditions and Assumptions**

- Any Sites the resource asset is to be connected to have already been defined in the system

**Acceptance Criteria**

- Connectivity to (parent) Site is validated to ensure there are no Active connections from the same resource Asset with overlapping date ranges, to any site.
- On supply of valid data, the resource asset is persisted to the systems data repository, else get an appropriate error message

**3.1.4.7 Remove resource asset, via UI****Purpose**

To indicate a resource Asset is deleted.

Note: the system will only tag entities as deleted in the system. This will restrict them from being shown in the UI unless the user sets a system option to Show Deleted data.

**Description**

As an operator of the system I want to be able to

- Locate the resource asset to be deleted via a set of search criteria
- Mark the resource asset as deleted (by Setting its entity status to deleted).

**Roles**

FO decides when a resource asset is to be marked as deleted (e.g. misconfigured), additionally they have the option to phase out the use of a resource asset by setting its valid from and to dates as an alternative.

**Preconditions and Assumptions**

- None

**Acceptance Criteria**

- Resource asset is persisted to the systems data repository with entity status of deleted.



### 3.1.4.8 Reinstate resource asset, via UI

#### **Purpose**

To indicate a resource asset is Active.

Note: the system will only tag entities as deleted in the system. This will restrict them from being shown in the UI unless the user sets a system option to Show Deleted data. If a resource asset has been deleted in error, it can easily be reactivated via this user story.

#### **Description**

As an operator of the system I want to be able to

- Locate the resource asset to be reinstated via a set of search criteria (that includes the retrieval of deleted resource assets)
- Mark the resource asset as Active (by Setting its entity status to active).
- When reactivating a resource asset, the normal validation that is applied on adding/editing a resource asset will be verified.

Note: For Deleted assets it is useful to see all the asset data (as per resource asset amendment), hence the reactivation is achieved by using the same UI form as for resource asset amendment, with the entity status being available for amendment.

#### **Roles**

FO decides if to reactive a Resource Asset.

#### **Preconditions and Assumptions**

- None

#### **Acceptance Criteria**

- Connectivity to (parent) Site is validated to ensure there are no Active connections from the same resource Asset with overlapping date ranges, to any site.
- On supply of valid data, the resource asset is persisted to the systems data repository (with entity status Active), else get an appropriate error message

#### 3.1.4.9 Create, update, delete & reinistate reosurce asset, via API

##### **Purpose**

To capture the definition of a new resource asset, amendment and deletion of existing resource asset (via system to system integration or via external load of data), including all its properties and optionally its connectivity to sites.

It provides the same basic functionality as user stories “Enter new resource asset, via UI”, “Update resource asset, via UI”, “Update site, via UI”, “Remove resource asset, via UI”, “

Reinstate resource asset, via UI” but via a system API interface. Additionally, the cross referencing of the resource asset entity to any external asset representation with a different external key is handled in this user story.

### **Description**

As an integrator of data into the system I want to be able to

- Add, Amend, Delete and reinstate resource assets with their related properties to the system.
- Resource assets are classified in terms of resource types, so when defining a resource asset, the resource type will be selected (Generation, Load, Storage, Charging or Monitoring). Then depending on the resource type, a set of properties associated with that resource type will be captured (see Data Repository section “Resource Asset” for details of the particular properties for each resource type).
- Resource assets can additionally be controllable, if it is controllable an additional set of the properties related to its controllability are captured (see Data Repository section “Resource Assets that are Controllable” for details).
- Resource assets can additionally support monitoring, if the resource support monitoring (captures metering data), an additional set of the properties related to its metering/monitoring are captured (see Data Repository section “Resource Assets supporting Monitoring” for details).
- Resource assets can model a physical resource or can be a virtual to model for instance all load at a site where each individual physical load device is not monitored separately. In the case of modelling a physical device:
  - a set of properties related to its installation details can be captured. These properties are not relevant to the Invade pilots so will not be detailed further in the document.
  - a set of properties related to how it is communicated with can be captured. These properties are not relevant to the Invade pilots so will not be detailed further in the document.
- Manage resource to site connections (optionally)
  - Connect resource assets to sites (only one active connection will be allowed over any period of validity)

- Manage the Connectivity to sites in terms of valid from/to dates (optional) and its status (active/deleted)
- Manage the Connectivity to External System (cross referencing) (optionally)
  - Connect External systems primary key to the FC resource asset key, thus enabling the client to provide the look up of the resource asset to the API either with the external keys or FC internal keys.

### **Roles**

- System to System integration
- Pilot resource assets set up via scripting

### **Preconditions and Assumptions**

- Any site that the asset is to be connected to have already been defined in the system

### **Acceptance Criteria**

- Connectivity to (parent) site is validated to ensure there are no Active connections from the same resource Asset with overlapping date ranges, to any site.
- On supply of valid data, the asset is persisted to the systems data repository and the resource asset FC reference is returned, else get an appropriate error message
- The API must support partial data upload, via control parameters the message supports options to indicate what data is being provided. Thus, providing an API that can support both full reconfiguration of a resource asset or partial amendment to an asset, e.g. To Add a new site connector and not supplying all the other data.

## API interface

- The management of resource assets will be via the Asset management API outlined in section “Data Management APIs”.

### 3.1.4.10 Cross reference resource asset, via UI

#### Purpose

To cross reference a Resource Asset in the FC via its internal unique key to an external systems unique key. This enable the API to accept data with external system references and to manage the look ups to the FC internal entity via the cross referencing.

See user story I section “External System Cross Referencing management” for details.

### 3.1.4.11 Functions

Validations:

1. Name is mandatory, Valid From Date < Valid To date (both optional)
2. Resource Asset can only be associated to one active Site over any period of time.

### 3.1.4.12 User interfaces

User interface	Short description
Entity Overview	Provides a list of all entities in the system, with search criteria to find entities of a certain type e.g. Sites, Areas, Resource Assets (Generation, Load, Charging, Storage, Monitoring), additional elastic search criteria will be available to search across properties e.g. for a sting of characters.
Resource Asset management	Provides a single screen for the management of resource assets (create, update, delete, undelete), including <ul style="list-style-type: none"> <li>▪ its properties</li> <li>▪ its entity status (Active, deleted)</li> <li>▪ its connections to sites</li> </ul>

### 3.1.5 External System Cross Referencing management

#### 3.1.5.1 Purpose

Enable external systems to cross reference entities in the external system to entities within the FC.

#### 3.1.5.2 Involved roles

FO user role, additionally External System cross referencing can be automatically handled when receiving entity data via the APIs, the handling of which is specified in the individual use case for updating data via the API.

#### 3.1.5.3 Preconditions

The input-data that are going to be input must be ready at a pre-defined format.

#### 3.1.5.4 Main steps/list of user stories

There are two main ways to enter entity cross referencing information: 1) Manually by FO user or 2) Automatic by import. The manual process will typically be like this:

1. The FO receives (outside the Flexibility Cloud) all necessary parameters for one or several entities (areas, sites or Resource Assets)
2. The FO locates the equivalent entity in the FC system by search for then entity to be cross referenced
3. The FO opens a screen and enters the cross-referencing parameters
4. The parameters are validated and stored

The automated process will be handed via the use cases that create, update, delete or reinstate an entity, which have already been described in the previous sections. In summary:

1. When supplying the entities key the client can specify the entity type, external system name and the external system key (alternatively they can supply the FC internal key if desired, in which case no cross referencing will be performed).
2. The supplied entity type, external system and external key is looked up to see if a cross reference exists in the FC for that entity, if it does not then it will be a new entity to be created, if it does exist it will be an existing entity to be amended/deleted or reinstated depending on the data provided.

3. If it is a new system cross reference that is persisted to the data repository as part of saving the entity.
4. The system returns to the caller details about the data saved, including the internal system key of the entity stored and any error information related to the supplied entity details, if the entity could not be stored.

The following user stories are covered:

Cross reference FC entity to External System entity, via UI

### 3.1.5.5 Cross reference FC entity to External System entity, via UI

#### **Purpose**

To cross reference any FC entity via its internal unique key to an external systems unique key. This enable the API to accept data with external system references and to manage the look ups to the FC internal entity via the cross referencing.

Note: the API from “¡**Error! No se encuentra el origen de la referencia.**Asset management” section supports automatic cross referencing on data creation. However, you can additionally use the UI to set up cross referencing.

#### **Description**

As an operator of the system I want to be able to

- Locate the entity by entity type (Site, Area, Resource Asset) to be cross referenced via a set of search criteria. Note the entity types list that can be cross referenced will be at least Site, Area, Resource Asset, but others will also be available for cross referencing.
- Manage entity cross reference to External System(s)
- Cross reference the entity in the Flexibility Cloud to one or more external systems that may have a different unique key to represent the entity
  - The cross reference will consist of an External System Identifier(Name), The External System Key, the FC Entity Type (Site, Area, resource Asset) and the FC unique key (GUID), Entity Status (Active or deleted)
  - Add Cross reference to an External System
  - Remove Cross reference to an External System (set its entity status to deleted)

- Reinstate Cross reference to an External System (set its entity status to active)

### Roles

FO can use this when the same entity has been set up in FO and another system and there is no common key, so the mapping between the systems can be established.

### Preconditions and Assumptions

- Entity to be cross referenced exists in FC and an External System

### Acceptance Criteria

- Cross reference is persisted to the external system in the FC data repository.

#### 3.1.5.6 User interfaces

User interface	Short description
External System Cross Referencing Overview	Provides a list of all entities in the system, with search criteria to find entities of a certain type e.g. Sites, Areas, Resource Assets, and the external system, additional elastic search criteria will be available to search across properties e.g. for a string of characters.
System Cross Referencing management	Provides a single screen for the management of cross reference data between entities in the FC and external systems (create, update, delete, undelete), including <ul style="list-style-type: none"> <li>▪ its properties</li> <li>▪ its entity status (Active, deleted)</li> </ul>



## 4 Data Repository

### 4.1 Overview

The overview below is a logical representation of the key entities related to asset management required by the Invade project. It represents a subset of the entries modelled within the platform.

Key: 1. blue boxes are the business entities. 2. grey boxes are interfaces (common attributes) that an asset can inherit.

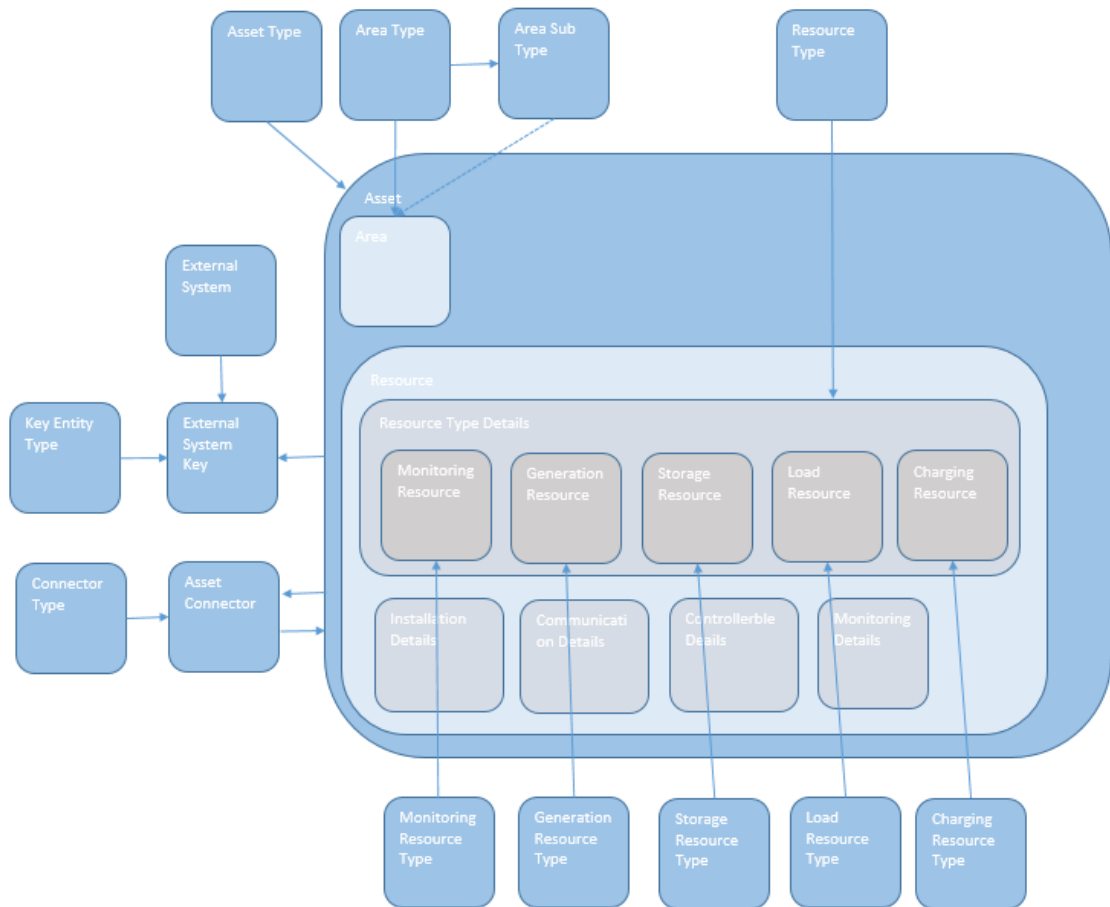


Figure 4. Logical Entity Modelling/Relationship Diagram

### 4.2 Business Entities

The key business entities relevant to the Asset management aspects of the pilot studies modelled within the platform are detailed below.

Entity	Type of	New	Description
Asset Entity Type			<ul style="list-style-type: none"> <li>▪ The classification of what the asset represents. For Invade we will utilize the following types: Area, Resource. Each of these types is defined next.</li> <li>▪ List of entity types is controlled by the system</li> </ul>
Area Type		Yes	<ul style="list-style-type: none"> <li>▪ A Classification of an Area Asset. E.g. DSO, Balancing, Flexibility, Weather, Market, Site</li> <li>▪ List of area types is configurable within the system</li> <li>▪ For invade we will model a hierarchy of Resource connected to parent area of type Site, and sites connected to parent areas such as DSO Zones, Weather, Flexibility depending on the pilot requirements. This hierarchy provides different aggregation and data collection levels.</li> </ul>
Area Sub Type		Yes	<ul style="list-style-type: none"> <li>▪ A Sub classification of an Area type. For a site examples are House, Building, Charging Site, Car Park,</li> <li>▪ List of area sub types is configurable within the system</li> </ul>

Resource Type		<ul style="list-style-type: none"> <li>▪ A classification of Resource Asset. For Invade we will utilize the following types: Generation, Resource, Load, Storage, Charging, Metering.</li> <li>▪ A Resource Asset may be a representation of a physical asset such as a PV, a battery or a Meter, or it could be a generic representation of a set of physical devices such as a generic Load.</li> <li>▪ Each Resource Type has a Classification of the known representation types defined next (Monitoring Resource Type, Generation resource Type, Storage resource Type, Load resource Type, Charging Resource Type)</li> <li>▪ List of resource types is controlled by the system</li> </ul>
Monitoring Resource Type		<ul style="list-style-type: none"> <li>▪ A classification for the monitoring resources type, i.e. what the resource represents.</li> <li>▪ For Invade will include Meter, Sensor, Other Monitor</li> <li>▪ List is controlled by the system</li> </ul>
Generation Resource Type		<ul style="list-style-type: none"> <li>▪ A classification for the Generation resources type, i.e. what the resource represents.</li> <li>▪ For Invade will include PV, Virtual Generation</li> </ul>

			<ul style="list-style-type: none"> <li>▪ List is controlled by the system</li> </ul>
Storage Resource Type			<ul style="list-style-type: none"> <li>▪ A classification for the Storage resources type, i.e. what the resource represents.</li> <li>▪ For Invade will include Battery, EV, Virtual Storage</li> <li>▪ List is controlled by the system</li> </ul>
Load Resource Type			<ul style="list-style-type: none"> <li>▪ A classification for the Load resources type, i.e. what the resource represents.</li> <li>▪ For Invade will include Water Heater, Floor Heater, Space Heater, Virtual Load</li> <li>▪ List is controlled by the system</li> </ul>
Charging Resource Type			<ul style="list-style-type: none"> <li>▪ A classification for the Charging resources type, i.e. what the resource represents.</li> <li>▪ For Invade will the classification could be Charge Point, Virtual Charging, Vehicle to Grid, Vehicle to Home, Vehicle to Building</li> <li>▪ List is controlled by the system</li> </ul>
Area (zone)	Asset	Yes	<ul style="list-style-type: none"> <li>▪ Relevant for DSO, Balancing, Flexibility, Weather, Market services for example</li> <li>▪ Provides a way to group Site assets into Areas (via Sites being connected to areas)</li> <li>▪ Provide an aggregation point for the related sites</li> </ul>

			<ul style="list-style-type: none"> <li>▪ Provides a logical place to capture additional data for the area, for example to attach Market price data to a Market Price Area</li> </ul>
Area(Site)	Asset		<ul style="list-style-type: none"> <li>▪ A logical representation of assets at a particular “location” that can be related to a customer and collectively assigned to a contract.</li> <li>▪ Provides a way to group Resource assets into Sites (via the resource Assets being connected to the Site)</li> <li>▪ Provide an aggregation point for the related sites</li> </ul>
Resource	Asset		<ul style="list-style-type: none"> <li>▪ A logical representation of an asset or set of assets, that will be connected to a particular Site</li> <li>▪ Specifically, depending on the assets Resource Type (Generation, Load, Storage, Charging) a set of Additionally properties are captured.</li> <li>▪ If resource is Controllable e.g. battery can be controllable, then it inherits a set of properties relevant to its controllability.</li> <li>▪ If the Asset supports monitoring (metering), then it inherits a set of properties relevant to monitoring/metering.</li> <li>▪ If the resource represents a physical asset, typically we can</li> </ul>

		<p>collect a set of Installation properties</p> <ul style="list-style-type: none"> <li>▪ Also if the resource represents a physical asset, typically we can collect a set of communication properties</li> <li>▪ Each resource Asset will also be classified by its Resource Representation Type (defined Above), this is useful for querying e.g answering questions such as what was generated by PV for example)</li> </ul>
Asset		<ul style="list-style-type: none"> <li>▪ In the Data Repository, an Asset is defined as any entity that can be modelled in a generic way, as per the Overview section above. Hence the generic asset does not just cover physical things such as devices but can also represent logical things such as sites and area or a collection of physical devices of a certain resource type for instance.</li> <li>▪ All Asset entities will have a set of common properties</li> <li>▪ The platform supports a large number of asset types, only the ones relevant to the pilots will be further documented in this repository.</li> <li>▪ Depending on the asset Type additional properties will be</li> </ul>

			<p>captured, for example see the definition of resource asset above.</p> <ul style="list-style-type: none"> <li>▪ Assets can be connected to other assets via the Asset Connector (with reason and from/to dates)</li> <li>▪ Each Asset can have any number of time series associated/linked to it, this will be detailed further in D8.2.</li> </ul>
Asset Connector			<ul style="list-style-type: none"> <li>▪ Connects Assets to Assets over a time period and for a reason (connector type).</li> <li>▪ Note validation ensures Assets will be connected correctly based on the connection type. E.g. if connection type is site then the parent connection must be an asset of type site and the child must be a Device Asset.</li> </ul>
Connect Type			<ul style="list-style-type: none"> <li>▪ Indicates the reason entities are connected, i.e. The type of connection (e.g. Parent)</li> </ul>
External System			<ul style="list-style-type: none"> <li>▪ Represents a system external to the IIP that it is communicating with</li> </ul>
External System Key			<ul style="list-style-type: none"> <li>▪ Holds a link to an external system key (unique identifier) to the platform unique key of an entity (Guid). The particular platform entity referenced is indicated by the link to Key Entity Type</li> </ul>

			<ul style="list-style-type: none"> <li>▪ Enables each entity exposed to external systems to be cross referenced to external systems unique identifiers.</li> </ul>
Key Entity Type			<ul style="list-style-type: none"> <li>▪ Indicates the platform entities that can be cross referenced externally e.g. Area (by type), Asset</li> </ul>

### 4.3 Business Entity Connections

Section “3.1.1.1 Definitions” provides an overview of how Areas, Sites and resource Assets are related to one another (this is supported in the logical data model via the Asset Connect entity).

In addition to the details provided in the modelling section, as the relationship between any asset entity to any other asset entity is modelled in a generic way additional relationship can be configured if required. However, at this stage we do not feel this is necessary within the FC solution as the internal communications between controllers and devices will be handled by other parties within the pilots, but if it is needed for any reason, it is handled within the data model. For example, the following types of relationships can be modelled:

- Meter to device relationships e.g. where a meter meters a particular device. Validation ensures for an Asset Meter Connection type the child device can only be connected to one parent meter asset over any period of time.
- Gateway to device relationship e.g. where a gateway controls set of devices. Validation ensures for an Asset Gateway Connection type the child device can only be connected to one parent meter asset over any period of time.



## 4.4 Assets

### 4.4.1 General Asset Properties

The key business properties relevant to the pilots held on the Asset entity are detailed below.

Property	Type	New	Description
Key	<i>guid</i>		<i>Unique key of Asset (system generated)</i>
Name	<i>string</i>		<i>Friendly name of asset</i>
Description	<i>string</i>		<i>Description of the asset</i>
Valid From	<i>Datetime</i>		<i>Optional</i>
Valid To	<i>Datetime</i>		<i>Optional</i>
Address	<i>reference</i>		<i>Reference to the address of the asset (optional)</i>
Participant	<i>reference</i>		<i>Reference to the participant</i>
ExternalKeys	<i>Set of references</i>		<i>Set of references to External systems identifiers for the asset. Enables each external system to be associated to the asset with a different unique key</i>
AssetProperties	<i>Set of asset properties</i>		<i>Set of asset properties (classified by Asset Property Type). Enables each asset to have an arbitrary set of properties attached identified by the attributes property type</i>
Resource Properties	<i>Set of Properties related to the resource type</i>		<p><i>Only for assets of Type Resource:</i></p> <p><i>Depending on the type of resource a set of properties will be captured on the asset related to the Resource type (e.g LoadResource, StorageResource, ProductionResource, ChargingResource).</i></p> <p><i>These additional properties are detailed in the following sections.</i></p>

Controllable Properties	<i>Set of Properties related to resource assets that can be controlled</i>		<i>These additional properties are detailed in the following sections.</i>
Monitoring Properties	<i>Set of Properties related to resource assets that support monitoring (metering)</i>		
Installation Properties	<i>Set of Properties related to resource assets installation</i>		<i>Only applicable to representations of physical assets, where installation details are captured, it is not envisaged these will need to be captured for the pilots.</i>
Communication Properties	<i>Set of Properties related to resource assets communication</i>		<i>Only applicable to representations of assets that can be communicated with, it is not envisaged these will need to be captured for the pilots as the communication to the devices will be handled by one of the partners in the pilot.</i>
EntityStatus	<i>enum</i>		<i>Active, Deleted</i>

Set of AssetConnectors	Reference		Set of references to Assets (with valid from/to and connection type (e.g Area, Site) for the Asset. Links an asset to other assets. Note validation will typically ensure an asset can only connect to one other asset “parent” of a certain type over any period of time. Typical example for the Invade project is a Resource Asset being linked to a parent Site and Sites being linked to Areas (of different types).
------------------------	-----------	--	---

#### 4.4.2 Area Asset

Is an Asset type with additional properties.

For the purpose of the Pilots the Area entity will be used to model DSO zones, or areas where flexibility needs to be provisioned, or areas where market price data is collected or where weather is reported.

Additionally, sites will be modelled as an Area of sub type Site, this takes advantage of the generic grouping functionality for areas, hence the platforms asset data model supports many variations of asset groupings/hierarchies through the asset to asset connections.

##### 4.4.2.1 Properties

The key business properties relevant to the pilots held on the Area entity are detailed below.

Property	Type	New	Description
AreaType	Reference		Reference to the Type of Area e.g. DSO/Flexibility Zone, Weather, Market, Site

##### 4.4.2.2 Area Site Asset

A site is a location with a main meter. A site can represent a dwelling, a house, a commercial building, an industrial facility, a charging station and probably other types.

Is an Area Asset type with additional properties.

For the purpose of the Pilots the Site will be a location where a collection of resource assets can be grouped for the purpose of modelling together.

#### 4.4.2.3 Properties

The key business properties relevant to the pilots held on the Site entity are detailed below.

Property	Type	New	Description
AreaSubType	Reference		Reference to the Type of site e.g. House, Parking lot, Office, Building, Charging Station
MaximumLoad	double		Optional maximum load for the site

#### 4.4.3 **Resource Asset**

All resource assets have the following additional properties:

Property	Type	New	Description
IsMetered	bool		Indicator as to whether the resource is being metered (monitored) i.e if readings are being taken for the resource, if true the properties under section "Resource Assets supporting Monitoring" will be captured
IsControllable	bool		Indicator as to whether the resource is being controllable, if true the properties under section "Resource Assets that are Controllable" will be captured

##### 4.4.3.1 Load Resource

Assets that are of a Load resource type have the following additional properties:

Property	Type	New	Description
LoadResourceType	enum		Type of load resource, see definition in Business Entities section
InstalledMaxCapacity	double		

#### 4.4.3.2 Generation Resource

Assets that are of a Generation Resource type have the following additional properties:

Property	Type	New	Description
GenerationResourceType	enum		Type of generation resource, see definition in Business Entities section
InstalledMaxCapacity	double		

#### 4.4.3.3 Storage Resource

Assets that are of a Storage resource type have the following additional properties:

Property	Type	New	Description
StorageResourceType	enum		Type of storage resource, see definition in Business Entities section
InstalledMaxCapacity	double		In kWh
InstalledMinCapacity	double		In kWh, normally zero
MaxChargingPower	double		In kW
EfficiencyOfCharging	double		Factor between (0-1)
MaxDischargingPower	double		In kW
EfficiencyOfDischarging	double		Factor between (0-1)

#### 4.4.3.4 Charging Resource

Assets that are of a Charging resource type have the following additional properties:

Property	Type	New	Description
ChargingResourceType	<i>enum</i>		<i>Type of charging resource, see definition in Business Entities section</i>
InstalledMaxCapacity	<i>double</i>		
MaxChargingPower	<i>double</i>		<i>In kW</i>
MaxDischargingPower	<i>double</i>		<i>In kW</i>
ChargingOnly	<i>bool</i>		True implies supports charging only, else also support discharging
Controllable Properties	<i>Set of Properties related to whether the resource is controllable</i>		<i>Depending on whether the resource is controllable an additional set of properties will be captured on the asset related to how the asset can be controlled. These additional properties are detailed in the next section.</i>
Monitoring Properties	<i>Set of Properties related to whether the resource supports monitoring</i>		<i>Depending on whether the resource supports monitoring (metering) an additional set of properties will be captured on the asset related to the metering. These additional properties are detailed in the next section.</i>

#### 4.4.3.5 Monitoring Resource

An Asset resource of type monitoring will support the additional properties specified in section “Resource Assets supporting Monitoring”. Additionally, any Asset resource can be configured to support monitoring as sometimes the monitoring is imbedded in the resource asset type.

#### 4.4.3.6 Resource Assets that are Controllable

Assets that are controllable will have the following additional properties.

Property	Type	New	Description
ControllableType	<i>enum</i>		<i>None, OnOff, Continuous</i>
FlexibilityType	<i>enum</i>		<i>None, Disconnectable, Reducible, ShiftableVolume, ShiftableProfile, ShiftableProfileButInterruptible</i>

#### 4.4.3.7 Resource Assets supporting Monitoring

Assets that support monitoring will have the following additional properties.

Property	Type	New	Description
SupportsMonitoring	<i>bool</i>		<i>If true, the following properties can be captured.</i>
MonitoringResourceType	<i>enum</i>		<i>e.g Meter, Sensor</i>
MeasurementType	<i>enum</i>		<i>Energy, Temperature, Heat</i>
StartValue	<i>double</i>		
IsMainMeter	<i>bool</i>		<i>For the pilots each site will be configured to have one main meter for measurement type Energy, that meters the net exchange with the grid, i.e. the purchase and the sales.</i>

#### 4.4.3.8 Resource Assets that have Installation Details

Only applicable to representations of physical resource assets, where installation details are captured, as previously stated it is not envisaged these will need to be captured for the pilots so will not be detailed further in this document.

4.4.3.9 Resource Assets that have Communication Details

Only applicable to representations of resource assets, which can be communicated with electronically, as previously stated it is not envisaged these will need to be captured for the pilots (within this platform as will be handled by the other partners) so will not be detailed further in this document.

## 5 Platform Architecture

### 5.1 Overview

The deliverable "D4.1 INVADE Concept Design" contains an overview of the Flexibility Cloud Software Architecture (FCS) in "Section 3" and "Annex 2 – Azure components and microservices in FCS" of the same document provides an overview of the Azure cloud-based architecture utilized by the platform.

For the purpose of clarity, the overall software architecture for the platform is shown below:

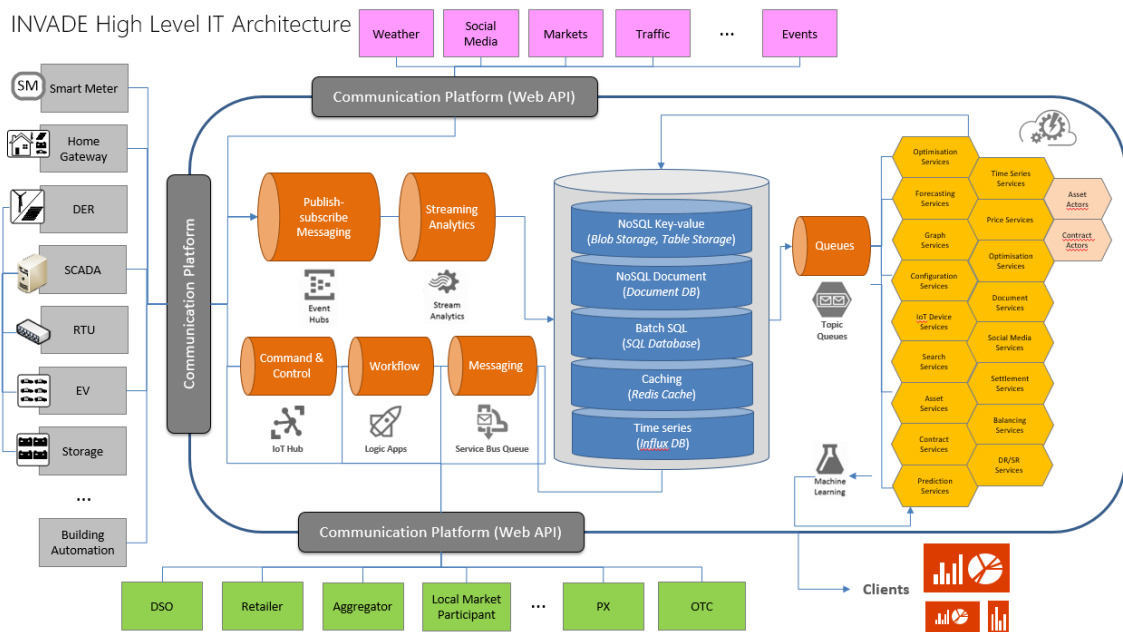


Figure 5. FC Architecture Overview



## 5.2 Changes to the architecture detailed in D4.1 Concept Design

### 5.2.1 Influx Time Series DB

Time series data within the platform will now be stored in the Influx Time series DB (previously stored in Blob storage). This database has been specially designed and optimized to work with time series data. It can collect, normalize, correlate and aggregate data over 100s of data sources. It has analysis and reporting and machine learning capabilities and is a highly scalable solution.

## 5.3 Interfaces

The following interface methods are supported:

- Data management APIs (for configuration of Assets, topology and time series data)
- Event Hub processing, FCS utilizes Azure Event Hub for high-frequency messages, such as real-time data.
- Batch Processing of files from Front End Systems (FES), for processing file-based interfacing of high-frequency data.
- Publishing to External End Points, FCS will have the ability to configure external end points where messages can be published to, typically the endpoint can be a Queue, topic queue, external Api.
- Applications for presentation, management and control of the data and activities within the platform
  - Desktop UI for updating time series values when manual operations are required to the data. Such operations may be required after receiving notifications from validations for missing or poor-quality metering data.
  - Web based UI for all configuration, manual maintenance and reporting in the form of dashboards.
  - Mobile App based UI for client reporting on load, generation, charging and storage data. Also, for other mobile client operations such as booking of EV-charging.

## 5.4 Flexibility Could Security

- Applications utilize Azure Active Directory(AAD). It makes it possible to integrate with a customer's own Azure AD. Meaning that if a customer is already using Office365 or have an on-premise Active Directory they can sync to an Azure Active Directory, they get single sign on.
- APIs will also be able to utilize AAD B2C for headless authentication.
- EventHub uses SAS tokens. For our IoT devices we have a management application that handles the issuing of SAS tokens.

## 5.5 Data Management APIs

### 5.5.1 Overview

An Asset management API has been identified to support the set of use cases identified in the use cases in this document. The API enable the definition of the master data required for the pilots to be loaded into the platform externally from using the platforms UI.

The API is an external facing restful web API service. It will be a façade to the internal platform services providing a consistent externally facing Api for management of data.

### 5.5.2 External Facing Platform Integration API for Asset management

API	Description
Asset Management	<ul style="list-style-type: none"> <li>▪ Supports retrieval, creation, amendment and deletion of Assets (Areas, Sites and Resources)</li> <li>▪ Additional supports the management of connections between assets. For the purpose of the pilots it will be to support connections between Areas and Sites and Sites and Resources.</li> </ul>

- |  |   |
|--|---|
|  | <ul style="list-style-type: none"><li>▪ The actual API definition is not part of the scope of this document, it is document as part of the D7.2 delivery.</li></ul> |
|--|---|

### 5.5.3 Entity Cross referencing to External systems

Internal to the platform each main entity will have a unique key (Guid), from an integration perspective the same entity in another platform may have its own unique identifier. The external facing Api will support the maintenance of data by the client of the Api providing either the platform key (entity Guid) or by providing a system name and its unique identifier. Internally the platform will hold a mapping table between each external system's key and the platform key for each entity.

This cross-referencing system also supports the ability to cross reference entities to common standardized coding schemes e.g. if we were dealing with country entity we could cross reference a country to its ISO ALPHA-2-Code, ISO ALPHA-3-Code, and its ISO NUMERIC code, via this cross-referencing system.

An example is provided below:

Entity	Platform Key (Guid)	External System Name	External Key
Area Site Asset	3794C7B1-3B18-40B2-829A-FF09AD8948E5	External System 1	1-344534457
Area Site Asset	3794C7B1-3B18-40B2-829A-FF09AD8948E5	External System 2	Abcd
Resource Asset	E117387B-30E8-44A1-8C21-536D995672CD	External System 1	243456886

The external cross referencing allows data to be passed to/from the platform via the Api with the entity keys the external system is familiar with (i.e. the platform holds cross referencing to the external systems). Alternatively, as on entity creation in the platform the platforms entity key is returned to the caller, then the external system can hold references to the platforms entity keys if it prefers.

## 6 GDPR

The platform utilizes the Microsoft cloud-based solution Azure for security, storage and processing of data. To ensure that privacy is in accordance with the applicable regulations and the new General Data Protection Regulation (GDPR), a separate agreement has been signed with Microsoft that requires Microsoft to comply with the Privacy / GDPR Terms.

Additionally, eSmart is working on establishing and implementing GDPR compliant routines when it comes to:

- Customer consent for data storage and usage
- Data life cycle management and what will happen at the end of customer relationships
- Routine for reporting events in relation to data to the customer and to the National Authority, which is Datatilsynet in Norway

This is work in progress since the implications of the new regulations is not fully known yet, and the new regulations have not been approved by law. The established routines will therefore be more closely described in D8.2.