



*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>D8.4</b>
Deliverable name:	<b>Cloud based flexibility management system: Flexibility Cloud, phase 3</b>
Version:	<b>1.0</b>
Release date:	<b>14/02/2019</b>
Dissemination level:	<b>Public</b>
Status:	<b>Submitted</b>
Author:	Jon Egil Spetaas, Mark Fisher, Susann Kjellin Eriksen, Hoang Minh Tran, Lasya Priya Kotu, Terje Lundby – eSmart Systems AS



**Document history:**

Version	Date of issue	Content and changes	Edited by
0.1	08/01/2019	First draft version	Jon Spetaas
0.5	04/02/2019	Ready for peer-review	Jon Spetaas
0.6	11/02/2018	Updated from Badenova peer-review	Jon Spetaas
0.7	13/02/2018	Updated from Albena peer-review	Jon Spetaas
1.0	14/02/2018	Peer-reviewed	Jon Spetaas Susann Kjellin Eriksen

**Peer reviewed by:**

Partner	Reviewer
Badenova	Fabian Burkard/Malte Thoma
Albena	Dimitar Stanev

**Deliverable beneficiaries:**

WP / Task
WP4 / T4.5, T4.6
WP7 / T7.2
WP8 / T8.3, T8.4, T8.5, T8.6
WP9 / T9.4
WP10 / T10.2 – T10.7

## 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>10</b>
3.1 Contract Management	10
3.1.1 Overview	10
3.1.2 Contract Management user stories	14
3.2 Meter values management	24
3.2.1 Overview	24
3.2.2 Receive Meter Values	25
3.3 Events and external information	28
3.3.1 Overview	29
3.3.2 Events and external information user stories	30
3.4 Calculator and scheduling	33
3.4.1 Overview	33
3.4.2 Calculations	37
3.4.3 Schedules	37
3.5 Predictions Management	38
3.5.1 Overview	38
3.5.2 Predictions management user stories	40
3.6 Flexibility Management Optimization	43
3.6.1 Overview	43
3.6.2 Calculate optimal flexibility schedule	48
3.7 Follow up and reporting	51
3.7.1 Overview	51
3.7.2 Follow up and reporting user stories	52
3.8 Control signals management	56

3.8.1	Overview	56
3.8.2	Control signals management user stories	57
3.9	Flexibility Services Management	59
3.9.1	Overview	59
3.9.2	Flexibility Messages user stories	60
<b>4</b>	<b>Integrated INVADE Platform - Web UI .....</b>	<b>65</b>
4.1	Overview	66
4.1.1	List view	66
4.1.2	Detailed screen	66
4.1.3	Time series values	67
4.2	Security	68
4.2.1	Tenants	68
4.2.2	User configuration	68

**Abbreviations and Acronyms**

<b>Acronym</b>	<b>Description</b>
API	Application programming interface
B2C	Business to Consumer (security model)
BRP	Balance Responsible Party
DSO	Distribution System Operator
EV	Electric Vehicle
FC	Flexibility Cloud (The platform)
FO	Flexibility Operator
GUID	Global Unique IDentifier
IIP	Integrated INVADE Platform
NA	Not Applicable
PV	Photovoltaic
TBD	To Be Determined
ToU	Time-of-Use
TSO	Transmission System Operator
VAT	Value Added Tax
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 fundamental parts of the Integrated INVADE platform, namely the asset management, asset control and data repository were presented in D8.1.

The first version of the business intelligence in the Integrated INVADE platform was described in D8.2 and delivered in June 2018. This deliverable covered Contract Management, Meter Values Management, Events And External Information, Calculator And Scheduling, Predictions Management, Flexibility Optimization Management, Follow Up And Reporting, and finally described the Control Signals Management.

In this deliverable, a second version of the business intelligence is presented. Since many of the sections described in D8.2 are revised during the implementation of the second version of the system, this document contains not only the additional functionality, but also all sections from the D8.2. Thus, this document overrules all descriptions in D8.2.

This deliverable is structured as follows: First an overview of the main processes covered in the IIP is presented. Then, the functionality of each of the main processes is described. Finally, a description of user interfaces in the Integrated INVADE Platform as well as the security model is provided.

# 1 Introduction

This document contains the deliverable D8.4 “Cloud based flexibility management system: Flexibility Cloud, phase 3”. In this deliverable, the implementation of the advanced flexibility management operations is presented. The intended audience of the deliverable is mainly the INVADE project partners to whom the implementation of the flexibility management system is relevant.

The document builds on the already delivered D8.2 “Cloud based flexibility management system: Flexibility Cloud, phase 2”, which explained the implementation of the simplified flexibility management operations. D8.2 was delivered in M18.

The document also builds on the already existing eSmart platform, which was partly developed in the H2020 EMPOWER project, and input from work performed in other INVADE work packages: D4.1, D4.2, D4.3, D5.3, D5.4, D7.1, D7.2 and D10.1.

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 a general description of user interfaces and how security is implemented in the client.

## 2 INVADE main processes and their relations

The main objective of 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.

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	N	Y
BRP	Day-ahead portfolio optimization	N	N	N	N	N
	Intraday portfolio optimization	N	N	N	N	N
	Self-balancing portfolio optimization	N	N	Y	N	Y
Prosumer	ToU optimization	Y	Y	Y	N	N
	kWmax control	Y	Y	Y	N	N
	Self-balancing	Y	Y	Y	Y <sup>1</sup>	N
	Controlled islanding	N	N	N	N	N

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

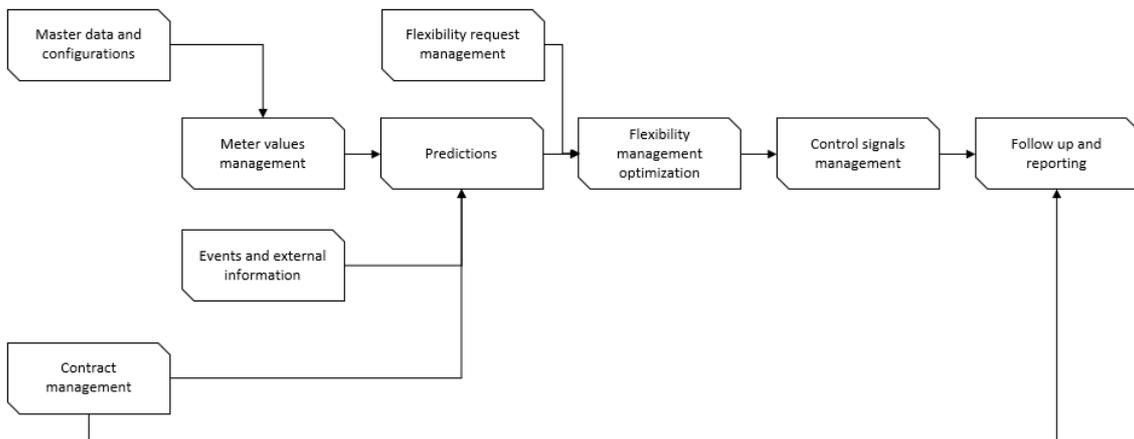
BRP services Day-ahead portfolio optimization and Intraday portfolio optimization will not be tested by pilots. However, the BRP request for flexibility will be handled in the same way by the Integrated INVADE Platform as for Self-balancing portfolio optimization.

Controlled islanding will not be tested in this project.

To support the flexibility services listed above, a set of *main processes* will be needed in the IIP. 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.

---

<sup>1</sup> Prosumer self-balancing service is going to be supplied by SMA outside the scope of the INVADE project.



**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 are covered in D8.1, D8.2 and D8.4 together. D8.1 contains the main processes master data and configurations, the remaining processes are described in this document.

## 3 Functionality

This chapter describes the functionality for the related processes. Each main process is described with an overview, which 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) are involved in the user stories.
- Preconditions. Describes what data elements and previous processes 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)
- User interfaces. Lists user interfaces that are needed to support the functionality

### 3.1 Contract Management

This section replaces the section 3.1 Contract Management in D8.2.

#### 3.1.1 Overview

In general, a contract is the link between a Contract owner, a Customer, a Contract Product and locations with all controllable and non controllable assets.

This section describes how to configure a flexibility contract between an FO and a prosumer, and the contracts required to offer aggregated flexibility to external parties such as DSO or BRP.

The links between data objects in Contract management are described in Figure 2.

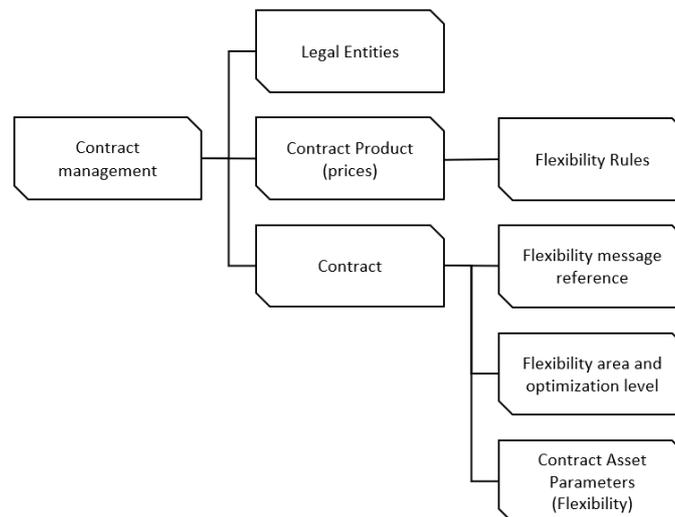


Figure 2: How data is linked in Contract Management

## Definitions

**Legal Entity** is the general term in the IIP defining any partner to a contract. Both the Contract owner and the Contract Customer are legal entities described with different entity roles. We strive to keep the Contract platform as general as possible with any type of legal entity as both contract owner and contract customer. Relevant legal entity types in the IIP are FO, Prosumer, DSO and BRP. FO will be the pilots' role in contract and should in addition to Flexibility Operator have the Legal Entity role Contract Owner. Prosumer, DSO and BRP should have the Legal Entity role Counterparty for Contract, in addition to DSO for DSO's and BRP for BRP's.

**Contract Products** forms the pricing of the contract. Multiple lines of pricing are possible, and it is the combination of lines that gives the total price for the contract.

Combining Amount Type and Related Contract Type describes the elements in each line.

The Prosumer Optimization contract or DSO/BRP flexibility request contract (in this document referred to as Flexibility Contract) may have Contract Product elements from energy and grid contracts as parameters for optimization even if no energy or grid contracts are to be settled with the FO.

Some pricing examples:

- Energy contract elements
  - Variable consumption and Spot price with mark-up – energy consumption

- Variable production and Spot price with multiplier – surplus energy delivered back to the grid
- Fixed monthly fee
- Grid contract elements
  - Variable consumption and Fixed fee per kWh – energy fee
  - Variable tiered consumption and fixed fee – subscribed energy and overconsumption
  - Max Consumption and variable fee – demand charge tariff calculated from the hour with the highest energy per month
- Flexibility contract
  - Control Plan Activation – Awarding prosumers for their flexibility according to fees setup for each flexible device
  - Delivery Notice – FO awarding prosumers for sales or charging BRP and DSO for their purchase of flexibility
  - Fixed fee – FO charging BRP or DSO for having available flexibility

**Flexibility Rules** describes overall parameters for optimization. The set of flexibility rules regulates how, when and how often the optimization should be executed. Flexibility Rules are defined on the Contract Product which may be used for several contracts. Thus, a limited number of these configurations are required.

The parameters are:

- Resolution – Time resolution for the results from optimization algorithm
- Optimization Frequency – How often is the optimization executed
- Optimization Horizon in Resolution intervals – How long is the optimization planning ahead
- Optimization History in Resolution intervals – How long is the history basis for optimization
- Optimization Strategy – Alternatives are Prosumer\_Optimization, DSO\_Optimization and BRP\_Optimization. Combinations will occur, but then the different roles will have separate contracts. E.g. a prosumer optimization contract for a set of households combined with a DSO optimization contract between FO and DSO.

**Flexibility Message** Reference is the reference for messages to and from external parties requesting flexibility. These messages are described in delivery D7.2, section 4.4 Message management API.

**Message Types** – describes which messages to produce and properties for each type. Messages will be event based or scheduled on a rolling or day ahead/within day basis. Message types to be tested:

- **Baseline Energy Notice** – The base line use of energy for an Asset (Area) associated to a Flexibility contract, prior to any BRP or DSO Flexibility request acceptances for that Area and contract.

The message details will consist of the Net Energy PTU (Per time unit) forecasted per Area. This is the baseline that will be used as a basis for flexibility acceptance, in terms of evaluation of delivered flexibility.

- **Flexibility Request** – A Request for Flexibility for an Asset (Area) associated to a Flexibility contract
- **Flexibility Offer** – An Offer of Flexibility for an Asset (Area) associated to a Flexibility contract, based on receipt of a Flexibility Request
- **Delivery Note** – An indication of the settlement based on FlexibilityOfferAcceptance

**Contract - Types** sets the definition of what parameters are required on a contract. Energy, Grid, Service and Flexibility forms the main types of contract.

The Contract types are:

- **Energy Contract** - Retail and other types of trade with energy between any two legal entities. For prosumer this contract will have elements of prices and taxes for both purchase of energy and sales of surplus energy back to the grid.
- **Grid Contract** – Contract between any legal entity and DSO. For prosumer this contract type will have elements of prices and taxes with DSO.
- **Service Contract** – In use when FO sells or buys commodities and services. This contract type will be calculated as part of the settlement in IIP version 2. The contract type does not influence optimization.
- **Prosumer Optimization** - Flexibility services contract between FO and prosumer. Connects the prosumer to its sites and resources with parameters required for optimization.

- BRP Optimization – Flexibility request management contract between FO and BRP. This contract must specify what messages to exchange and any parameters to these.
- DSO Optimization – Flexibility request management contract between FO and DSO. This contract must specify what messages to exchange and any parameters to these.

**Flexibility area and optimization level** describes which level the contract covers in the asset topology level and which level is optimized. An example is a contract that regulates a prosumer site (flexibility area) and optimizes on site level (optimization level). Another example is the large-scale public parking where the same contract parameters are valid for all the postal codes (defined as a flexibility area of all postal codes) and the contract is optimized per postal code. Each postal code will be defined as a site (optimization level) with all the EV charge points connected to it.

As a simplification, an assumption is made that a BRP area will fully enclose a DSO area. A DSO area will have one or several prosumer sites.

**Contract Assets** is a set of default parameters for each type of controllable and non-controllable assets within the flexibility area of the contract. These defaults may be updated for the specific contract or for single assets and stored to the contract asset repository.

### 3.1.2 Contract Management user stories

#### 3.1.2.1 Purpose

The purpose of this process is to enter, update and remove information about a contract, legal entities, a contract product and the parameters required for optimizing the contracts flexibility.

#### 3.1.2.2 Involved roles

The user stories involve a FO user role.

#### 3.1.2.3 Preconditions and Assumptions

Relevant input data must be ready at pre-defined format.

### 3.1.2.4 Main Steps/List of user stories

New information will be entered manually by FO user. The process will typically be like this:

1. The FO negotiates all necessary parameters with the Contract partner
2. The FO opens a screen and enters the parameters
3. The parameters are validated and stored

The following user stories are covered:

1. Enter new Legal entity
2. Update Legal entity
3. Remove Legal entity
4. Reinstate deleted Legal entity
5. Enter new Contract Product
6. Update Contract Product with Energy and Grid prices
7. Remove Contract Product
8. Reinstate deleted Contract Product
9. Enter new Contract
10. Update Contract
11. Remove Contract
12. Reinstate deleted Contract

### 3.1.2.5 Enter new Legal Entity

#### **Purpose**

The purpose of this process is to enter new Legal Entities as Flexibility Operator, Prosumer, DSO or BRP in the system

#### **Description**

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

- Add a new Legal Entity with all required information.
- Mark the Legal Entity with the one or more roles e.g. Prosumer, DSO, BRP etc.

**Roles**

FO configures a new role into the system.

**Preconditions and Assumptions**

Relevant information about the role is available

**Acceptance Criteria**

New Legal Entity marked are ready for Contract for flexibility for its household.

**3.1.2.6 Update Legal Entity****Purpose**

To capture amendments to a Legal Entity

**Description**

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

- Locate the Legal Entity via a set of search criteria.
- Amend Legal Entity properties.

**Roles**

FO reconfigures, enhances the definition of a Legal Entity in the system.

**Preconditions and Assumptions**

Relevant information about legal entity is available.

**Acceptance Criteria**

Legal Entity is stored with new properties.

**3.1.2.7 Remove Legal Entity****Purpose**

To indicate a Legal Entity 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 Legal Entity via a set of search criteria.
- Mark the Legal Entity as deleted (by setting its entity status to Deleted).

### **Roles**

FO decides when a Legal Entity is to be marked as deleted.

### **Preconditions and Assumptions**

No valid contract with the Legal Entity.

### **Acceptance Criteria**

Legal Entity is stored with entity status of deleted.

#### 3.1.2.8 Reinstate deleted Legal Entity

### **Purpose**

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

**Roles**

FO decides if to reactive a Legal Entity.

**Preconditions and Assumptions**

None

**Acceptance Criteria**

On supply of valid data, the Legal Entity is persisted to the systems data repository (with entity status Active), else get an appropriate error message.

**3.1.2.9 Enter new Contract Product*****Purpose***

The purpose of this process is to enter a new Contract Product to use for flexibility contracts.

**Description**

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

- Add a new Contract Product for prosumer optimization.
- Add the required Contract Product elements.
- Add the Flexibility rules for the set of contracts which this Contract Product will be used.

**Roles**

FO configures a new Contract Product into the system.

**Preconditions and Assumptions**

Relevant information about Contract Product is available.

**Acceptance Criteria**

New Contract Product for use in prosumer flexibility contract is stored to the system.

### 3.1.2.10 Update Contract Product with Energy and Grid prices

#### **Purpose**

To capture additional Contract Product elements for Energy and Grid prices to a Contract Product.

#### **Description**

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

- Locate the Contract Product via a set of search criteria.
- Amend Contract Product with elements of Energy and Grid price properties.

#### **Roles**

FO reconfigures, enhances the definition of a Contract Product in the system.

#### **Preconditions and Assumptions**

Relevant information about Energy and Grid contract with prosumer is available.

#### **Acceptance Criteria**

Contract Product is stored with new Contract Product elements.

### 3.1.2.11 Remove Contract Product

#### **Purpose**

To indicate a Contract Product 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 Contract Product via a set of search criteria.
- Mark the Contract Product as deleted (by setting its entity status to Deleted).

#### **Roles**

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

### **Preconditions and Assumptions**

No valid contract with prosumers using the Contract Product.

### **Acceptance Criteria**

Contract Product is stored with entity status of Deleted.

#### 3.1.2.12 Reinstate deleted Contract Product

### **Purpose**

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

### **Roles**

FO decides if to reactive a Contract Product.

### **Preconditions and Assumptions**

None

### **Acceptance Criteria**

On supply of valid data, the Contract Product is persisted to the systems data repository (with entity status Active), else get an appropriate error message.

### 3.1.2.13 Enter new Contract

#### **Purpose**

The purpose of this process is to enter a new flexibility contract and all required parameters.

#### **Description**

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

- Add a new Contract for prosumer optimization
- Have the Contract Asset defaults added to the Contract

#### **Roles**

FO configures a new Contract into the system.

#### **Preconditions and Assumptions**

Relevant information for the flexibility contract have been negotiated with the prosumer.

#### **Acceptance Criteria**

New flexibility contract is stored to the system.

### 3.1.2.14 Update Contract

#### **Purpose**

To capture specific Contract Asset parameters to a flexibility contract.

#### **Description**

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

- Locate the Contract via a set of search criteria.
- Amend a Contract Asset default with parameters specific for this prosumer to the Contract.

**Roles**

FO reconfigures, enhances the definition of a Contract in the system.

**Preconditions and Assumptions**

Relevant information about prosumers Contract Assets is available.

**Acceptance Criteria**

Contract is stored with new Contract Asset parameters.

**3.1.2.15 Remove Contract****Purpose**

To indicate a Contract 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 Contract via a set of search criteria.
- Mark the Contract as deleted (by setting its entity status to Deleted).

**Roles**

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

**Preconditions and Assumptions**

None

**Acceptance Criteria**

Contract is stored with entity status of Deleted.

**3.1.2.16 Reinstate deleted Contract****Purpose**

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

### Roles

FO decides if to reactive a Contract.

### Preconditions and Assumptions

None

### Acceptance Criteria

On supply of valid data, the Contract is persisted to the systems data repository (with entity status Active), else get an appropriate error message.

#### 3.1.2.17 User interfaces

User interface	Short description
Legal Entity	Provides a single screen for the management of Legal Entities (create, update, delete and undelete)
Contract Product	Provides a single screen for the management of Contract Product (create, update, delete and undelete), including <ul style="list-style-type: none"> <li>▪ its properties</li> <li>▪ its entity status (Active, deleted)</li> </ul>

	<ul style="list-style-type: none"> <li>▪ its flexibility parameters to contract optimization</li> </ul>
Contract	<p>Provides a single screen for the management of Contract (create, update, delete and undelete), including</p> <ul style="list-style-type: none"> <li>▪ its properties</li> <li>▪ its entity status (Active, deleted)</li> <li>▪ its parameters to Contract Assets</li> </ul>

## 3.2 Meter values management

This section is not updated in this document but copied from D8.2 for consistency.

### 3.2.1 Overview

Meter values are imported through API methods described in D7.2 Specification of the CP-API, either as meter readings or energy values and stored to assets in the form of time series.

Meter readings may be received with unstructured or unaligned time stamps. On arrival the readings are first aligned to fixed time resolution of 15 minute. Any missing values in intervals between meter readings will be estimated by interpolating between the existing values.

The delta between meter readings are then calculated and stored as the energy consumed or produced in the time interval. Values received as different phases will be aggregated to one energy value for the resource.

The main process is split into sub-processes as illustrated in Figure 10.

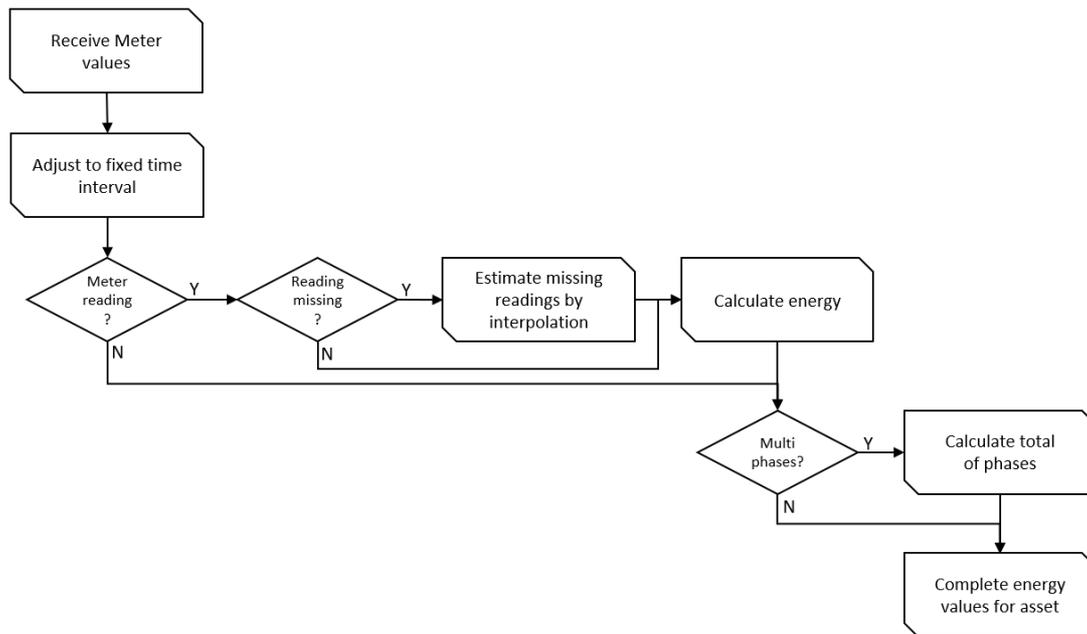


Figure 3. Main process and sub-processes for Meter values management

### 3.2.2 Receive Meter Values

#### 3.2.2.1 Purpose

The purpose of this process is to handle values imported to resources, transformation to energy in usable time segments and estimation of missing values.

#### 3.2.2.2 Involved roles

All processes are automated and hence involve only system roles.

#### 3.2.2.3 Preconditions

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

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

The automated process of receiving meter values will be like this:

1. An external System or FO (outside the IIP) sends all necessary parameters for meter values.
2. The external System or FO calls the IIP Data Management API providing meter values to an asset or list of assets and all necessary parameters to save the meter values.
3. The meter value and its time stamp is validated and stored.

4. The system returns to the caller details about the data saved and any error information related to the supplied meter values if the meter value could not be stored.

The following user stories are covered:

1. Receive meter values as energy values and align to fixed time intervals.
2. Receive meter values as meter reading.
3. Receive meter values as 3 phased meter readings.
4. Receive meter values as energy values when value in previous time segment is missing.

### 3.2.2.5 Receive meter values as energy values and align to fixed time intervals

#### **Purpose**

To capture new energy values, or updates to existing energy values, when values are received with start time and end time different from the time series fixed time intervals.

#### **Description**

As a supplier of meter values of the system I want to be able to

- Align meter values to fixed time intervals
- Add or update energy values to a prosumer resource

#### **Roles**

System to system integration

#### **Preconditions and Assumptions**

The prosumer resource and its topology have already been defined in the system.

#### **Acceptance Criteria**

On supply of valid data, the energy values are stored to the systems time series repository.

### 3.2.2.6 Receive meter values as meter reading

#### **Purpose**

To capture new energy values or updates to existing energy values from meter readings.

#### **Description**

As a supplier of meter values of the system I want to be able to

- Add or update energy values to a prosumer resource as delta from meter readings.

#### **Roles**

System to system integration.

#### **Preconditions and Assumptions**

The prosumer resource and its topology have already been defined in the system.

#### **Acceptance Criteria**

On supply of valid data, the energy values are calculated as delta from current and previous meter reading and stored to the systems time series repository.

### 3.2.2.7 Receive meter values as 3 phased meter readings

#### **Purpose**

To capture new energy values or updates to existing energy values from 3 phased meter readings.

#### **Description**

As a supplier of meter values of the system I want to be able to

- Add or update energy values to a prosumer resource calculated as the delta for each of the three phases.
- Add or update energy values to a prosumer as the sum of energy for the three phases.

#### **Roles**

System to system integration

### **Preconditions and Assumptions**

The prosumer resource and its topology have already been defined in the system.

### **Acceptance Criteria**

On supply of valid data, the energy values are calculated as delta from current and previous meter reading and stored to the systems time series repository for each of the three phases, and total energy as a sum of the three phases.

#### **3.2.2.8 Receive meter values as energy values when value in previous time segment is missing**

### **Purpose**

To capture new energy values or updates to existing energy values as an interpolation between current and previous value.

### **Description**

As a supplier of meter values of the system I want to be able to

- Add or update energy values to a prosumer resource as an interpolation between current and last known value when previous energy values are missing.

### **Roles**

System to system integration.

### **Preconditions and Assumptions**

The prosumer resource and its topology have already been defined in the system.

### **Acceptance Criteria**

On supply of valid data, the energy values are calculated as delta from current and last known energy value and stored to the systems time series repository.

## **3.3 Events and external information**

This section is not updated in this document but copied from D8.2 for consistency.

### 3.3.1 Overview

The purpose with events and external information is to capture additional data as parameters for predictions and optimization. Weather, prices and charge point connections are imported for IIP version 1, but additional parameters may easily be added later. Charge point meter values are imported in Meter Values Management, but the derived status saying that a charger is connected is treated as an event and thus, described in this section.

Weather data is requested from a third-party supplier of weather services as observations and forecasts. Weather observation history is, together with resource asset energy values history, used for training predictions models. This trained model is then, together with weather forecasts, used for prediction of future energy values of resource assets.

Weather observations and forecasts include:

- Temperature
- Cloudiness
- Solar radiation

Prices are imported from the pilots or as a request from a third-party supplier of day ahead prices.

Third-party data is requested on a configurable schedule. E.g. on the 5<sup>th</sup> minute of every 15 minute interval. In IIP version 1 the schedules are configured by eSmart. Typically, the day ahead prices will be requested once a day after they are settled. The weather information will be requested on a schedule with higher frequency, e.g. every hour or every 15 minutes. These schedules are configurable per system tenant and will be discussed with each pilot separately.

Charge point connection history is a status time series derived from the arrival of charge point energy values. Charge points will deliver values when a charger is connected to an EV and values will be blank when a charger is not connected. Status will have the value 1 when the charge point is connected and the value 0 when not connected.

When new information is stored to the above-mentioned time series the prediction service for that series will be triggered automatically.

The process is illustrated in Figure 4.

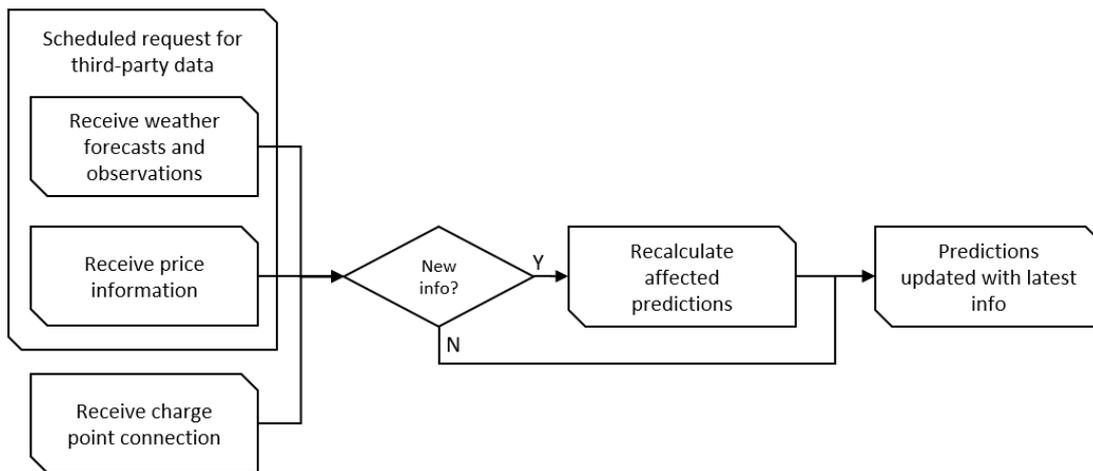


Figure 4: Events and external information process

### 3.3.2 Events and external information user stories

#### 3.3.2.1 Purpose

The purpose of this process is to handle values imported as weather data in the form of forecasts and observations, prices and to derive the charge point status from charge point energy values.

#### 3.3.2.2 Involved roles

All processes are automated and hence involve only system roles.

#### 3.3.2.3 Preconditions

The input-data must be ready at a pre-defined format.

End points for suppliers of third-party data are configured and ready for scheduled requests.

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

The automated process for requesting third-party data will be like this:

1. A schedule triggers a request for third-party data when the specified time in the series of fixed time intervals occur.
2. The system requests data from the third-party system from a configured end point in the correct format.
3. The external data elements and time stamps are validated and stored to the correct time series in the system.

The process for charge point status derived from arrival of charge point energy values will be like this:

1. A charge point delivers energy values, as described in section 3.2.2 Receive Meter Values.
2. Charge points status will be stored with the value 1 when an energy value exists on the charge point energy values time series, and the value 0 if the energy value does not exist.

The following user stories are covered:

1. Scheduled request for weather observations and forecasts.
2. Scheduled request for day ahead prices.
3. Derive charge point status from charge point energy values.

#### 3.3.2.5 Scheduled request for weather observations and forecasts

##### **Purpose**

To capture new weather observations and forecasts from a third-party supplier of weather services.

##### **Description**

As an FO I want the system to

- Request weather observations and forecasts to the system.

##### **Roles**

System to system integration

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

End points for suppliers of third-party data are configured and ready for scheduled requests.

##### **Acceptance Criteria**

Weather observations and forecasts are stored to the system's time series repository.

### 3.3.2.6 Scheduled request for day ahead prices

#### **Purpose**

To capture new day ahead prices from a third-party supplier of energy prices.

#### **Description**

As an FO I want the system to

- Request day ahead prices to the system.

#### **Roles**

System to system integration

#### **Preconditions and Assumptions**

End points for suppliers of third-party data are configured and ready for scheduled requests.

#### **Acceptance Criteria**

Day ahead prices are stored to the system's time series repository.

### 3.3.2.7 Derive charge point status from charge point energy values

#### **Purpose**

To derive charge point status from charge point energy values as input to charge point predictions.

#### **Description**

As an FO I want the system to

- Derive charge point status from charge point energy values as input to charge point predictions.

#### **Roles**

System to system integration

#### **Preconditions and Assumptions**

New energy values for a charge point have arrived in the system.

### Acceptance Criteria

Charge point status is stored to the system's time series repository.

## 3.4 Calculator and scheduling

This section replaces the section 3.4 Calculator and scheduling in D8.2.

### 3.4.1 Overview

This section describes how to support IIP processes with calculations and schedules.

In general, all arriving values stored in time series will trigger a mechanism starting any dependent calculations. This is called aggregation or calculation via model.

Some processes are time dependent and will be triggered at specified intervals.

#### 3.4.1.1 Calculations

All transformations, alignments, aggregations or events to time series values are made by calculations. Main types of calculations are:

- Delta derived as energy between current and previous meter reading.
  - Any missing meter time interval in between two existing readings will be estimated by linear interpolation between the existing values.
- Observation time will be aligned to fixed intervals
- Accumulated sub-meters to resource. E.g. 3 phased meter readings
- Site aggregations for specific datatypes. E.g. metered consumption
- Prediction service triggered by new values via prediction model. Models are described in 3.5 Predictions Management.

Each type of time series can have a list of calculation linked. The calculation meta data for a time series type will contain:

- QueueName
- ResultTimeSeriesType
- ResultTimeSeriesResolution

When a value is imported or stored to the time series repository, the change will trigger the new value to put an item in the specified queue(s) defined in the time series type.

In IIP version 1 these configurations are setup by eSmart. More calculations will be added in IIP version 2.

The main process is split into sub-processes as illustrated in Figure 5.

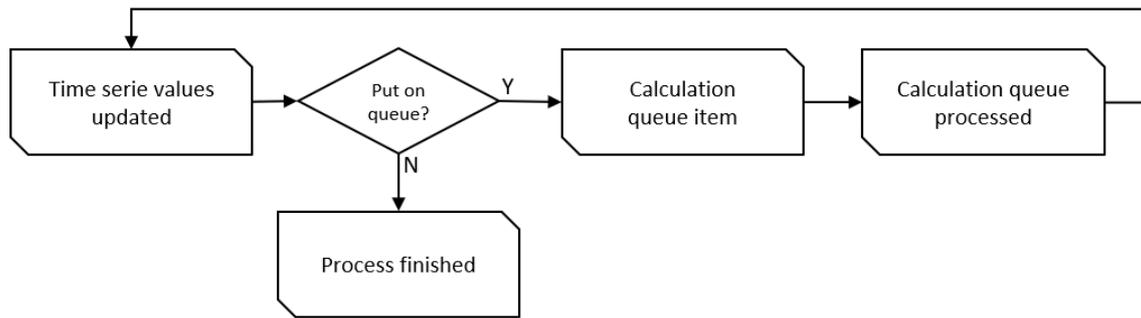


Figure 5. Main process of calculations to updated time series values

The calculations required in IIP are listed in Table 2

Source time series type	Queue Name	Result time series type	Via Model (for predictions)
Consumption Meter Reading	DeltaFromMeterReading	Consumption	
Production Meter Reading	DeltaFromMeterReading	Production	
Consumption	AccumulateTimeSeries	Net Energy	
Production	AccumulateTimeSeries	Net Energy	
Energy Level Meter Reading	ConvertTimeSeriesToResolution	Energy Level	
Charging Meter Reading	DeltaFrom MeterReading	Charging Energy	
Discharging Meter Reading	DeltaFromMeterReading	Discharging Energy	
Consumption Phase1 Meter Reading	DeltaFromMeterReading	Consumption Phase1	
Consumption Phase2 Meter Reading	DeltaFromMeterReading	Consumption Phase2	

Consumption Phase3 Meter Reading	DeltaFromMeterReading	Consumption Phase3	
Charging Meter Reading	DeltaFromMeterReading	Charging Energy	
Production Phase1 Meter Reading	DeltaFromMeterReading	Production Phase1	
Production Phase2 Meter Reading	DeltaFromMeterReading	Production Phase2	
Production Phase3 Meter Reading	DeltaFromMeterReading	Production Phase3	
Charging Phase1 Meter Reading	DeltaFromMeterReading	Charging Energy Phase1	
Charging Phase2 Meter Reading	DeltaFromMeterReading	Charging Energy Phase2	
Charging Phase3 Meter Reading	DeltaFromMeterReading	Charging Energy Phase3	
Discharging Phase1 Meter Reading	DeltaFromMeterReading	Discharging Energy Phase1	
Discharging Phase2 Meter Reading	DeltaFromMeterReading	Discharging Energy Phase2	
Discharging Phase3 Meter Reading	DeltaFromMeterReading	Discharging Energy Phase3	
Discharging Meter Reading	DeltaFromMeterReading	Discharging Energy	
Consumption Phase1	AccumulateTimeSeries	Consumption	
Consumption Phase2	AccumulateTimeSeries	Consumption	
Consumption Phase3	AccumulateTimeSeries	Consumption	
Production Phase1	AccumulateTimeSeries	Production	
Production Phase2	AccumulateTimeSeries	Production	
Production Phase3	AccumulateTimeSeries	Production	
Discharging Energy Phase1	AccumulateTimeSeries	Discharging Energy	

Discharging Energy Phase2	AccumulateTimeSeries	Discharging Energy	
Discharging Energy Phase3	AccumulateTimeSeries	Discharging Energy	
Charging Energy Phase1	AccumulateTimeSeries	Charging Energy	
Charging Energy Phase2	AccumulateTimeSeries	Charging Energy	
Charging Energy Phase3	AccumulateTimeSeries	Charging Energy	
Consumption Charging Energy	SiteAggregation	Metered Consumption	
Production Discharging Energy	SiteAggregation	Metered Consumption	
MainMeter - Consumption MainMeter – Production (-) Site – Metered Consumption (-) Site – Metered Production (+)	SiteAggregation	Residual Energy	
Consumption	ModelForecast	Forecasted Consumption	Via Model
Production	ModelForecast	Forecasted Production	Via Model
Charging Energy	ModelForecast	Forecasted Charging Energy	Via Model
Charging Energy	ModelForecast	Forecasted Charging Point State	Via Model
Charging Point State	ModelForecast	Forecasted Charging Point State	Via Model
Residual Energy	ModelForecast	Forecasted Residual Energy	Via Model
Spot Price	Copy imported spot price to following day Forecasted Spot Price	Forecasted Spot Price	

Table 2: List of calculations in the IIP

### 3.4.1.2 Schedules

A schedule is a triggering mechanism which is started at pre-defined times at pre-defined time intervals. Frequency and time in the interval are configurable and may be altered for each pilot individually. In IIP version 1 these configurations are setup by eSmart.

All schedules are mentioned in other processes but are listed in Table 3.

Process	Frequency	Time in the interval
Request weather forecast and observations	Every hour	Start of hour
Request day ahead prices	Every day	When prices are available. Usually around 13:00 local time
Prediction model training	Every day	At 00:00 (midnight)
Optimization service	Every 15 minutes	10 minutes before the regulating period starts.

Table 3 - Processes triggered by schedule

### 3.4.2 Calculations

Calculation user stories are described in other sections in this document:

- 3.2.2.5 Receive meter values as energy values and align to fixed time intervals
- 3.2.2.6 Receive meter values as meter reading
- 3.2.2.7 Receive meter values as 3 phased meter readings
- 3.2.2.8 Receive meter values as energy values when value in previous time segment is missing

### 3.4.3 Schedules

Schedule user stories are described in other sections in this document:

- 3.5.2.5 Schedule training of prediction model
- 3.3.2.5 Scheduled request for weather observations and forecasts
- 3.3.2.6 Scheduled request for day ahead prices
- 3.6.2.5 Calculate optimal flexibility schedule

### 3.5 Predictions Management

This section replaces the section 3.5 Prediction Management in D8.2.

#### 3.5.1 Overview

The IIP prediction service has two separate services for training prediction models and calculate predictions. The main aim is to develop a generic training and prediction service for each appliance group.

The prediction model training uses full history of metered values and weather data, while the prediction service uses shorter history along with the trained models.

The main process of predictions management is split into sub-processes in Figure 6.

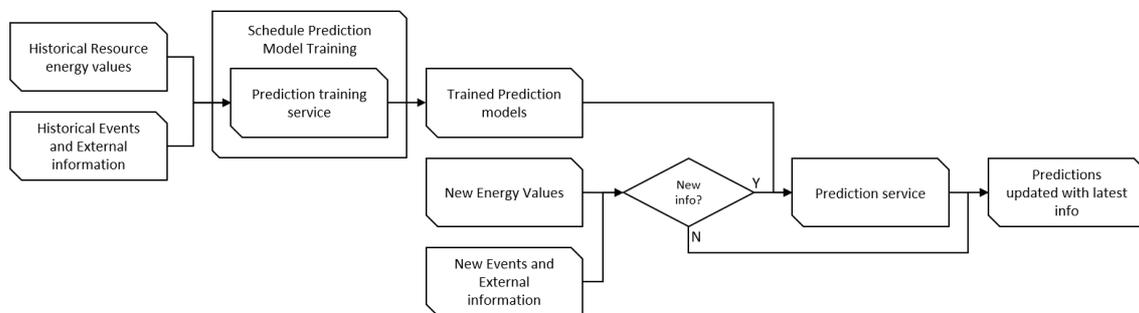


Figure 6: Predictions management overview

##### 3.5.1.1 Prediction training service

Prediction models are trained using sci-kit learn library available in python on a Kubernetes cluster built on Azure platform. Training and prediction web services are developed for each resource group. As the training process takes more time, it is done offline on scheduled time intervals. The prediction models need to be retrained periodically.

##### 3.5.1.2 Prediction service

In the IIP version 1 all predictions will be triggered event-based when new values are imported to the metered time series. The input to the data-driven techniques are the features extracted from the historical data. The machine learning models predict the future values from input time series. Each resource group needs different feature combinations for making predictions. The feature combinations are selected after analyzing the load consumption/generation time series.

### 3.5.1.3 Prediction and training input data

The prevailing parameters for training and prediction services for all the assets present in all the pilot areas, as well as the output time series from each model, are described in Table 4. This list will however be refined during the project.

Source	Input time series	Output time series
Residual Energy	Residual Energy Charge Point Status GUID of the nearest weather station Temperature (Actual and Forecast) Cloudiness (Actual and Forecast) Solar Radiation (Actual and Forecast)	Forecasted Residual Energy
Charge Point status	Charging Energy Charge Point Status GUID of the nearest weather station Temperature (Actual and Forecast) Cloudiness (Actual and Forecast) Solar Radiation (Actual and Forecast)	Forecasted Charge Point Status (connect and disconnect times)
Charging Energy	Charging Energy Charge Point Status GUID of the nearest weather station Temperature (Actual and Forecast) Cloudiness (Actual and Forecast) Solar Radiation (Actual and Forecast)	Forecasted Consumption (required charging)
Production	Production GUID of the nearest weather station Temperature (Actual and Forecast) Cloudiness (Actual and Forecast) Solar Radiation (Actual and Forecast)	Forecasted Production
Consumption (Electric space heating and water heaters)	Consumption GUID of the nearest weather station Temperature (Actual and Forecast)	Forecasted Consumption

	Cloudiness (Actual and Forecast)	
	Solar Radiation (Actual and Forecast)	

**Table 4 - Input and output parameters to prediction service**

### 3.5.2 Predictions management user stories

#### 3.5.2.1 Purpose

The purpose of this process is to train the prediction models and recalculate affected predictions based on new energy values or events and external information.

#### 3.5.2.2 Involved roles

All processes are automated and hence involve only system roles.

#### 3.5.2.3 Preconditions

Historical resource energy values, historical events and external information exists.

Prediction model training schedule is configured in the system.

New energy values or events and external information are received.

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

The automated process of training prediction models will be like this:

1. A schedule triggers the model training service with the correct parameters for the specified model.
2. The prediction model training service runs the training process from updated input values to the model.
3. The trained model is stored to the prediction model Azure blob storage.

The automated process of recalculating affected predictions is like this:

1. New values regarding energy values or events and external information will trigger the prediction service for the relevant resource.
2. The recalculated prediction is returned from the prediction service and stored to the correct time series in the system.

The following user stories are covered:

1. Schedule training of prediction model.
2. Calculate site residual prediction.

3. Calculate PV prediction.
4. Calculate charge point status prediction.

#### 3.5.2.5 Schedule training of prediction model

##### **Purpose**

To train prediction model as basis for predictions.

##### **Description**

As an FO I want the system to

- Train prediction model on configured time intervals.

##### **Roles**

The IIP

##### **Preconditions and assumptions**

Historical resource energy values and historical events and external information exists.

Prediction model training schedule is configured in the system.

##### **Acceptance criteria**

Trained prediction model stored to the prediction service Azure blob storage.

#### 3.5.2.6 Calculate Site residual energy prediction

##### **Purpose**

To calculate site residual energy prediction as basis for Flexibility Optimization.

##### **Description**

As an FO I want the system to

- Trigger calculation of site residual energy prediction from the arrival of new energy values.
- Prediction service returns the site residual energy prediction.

**Roles**

System to system integration.

**Preconditions and Assumptions**

Required parameters to the prediction model of site residual energy are filled with values.

Residual energy is calculated for historic periods

**Acceptance Criteria**

Site residual energy predicted values are stored to the systems time series repository.

**3.5.2.7 Calculate PV prediction****Purpose**

To calculate PV prediction as basis for Flexibility Optimization.

**Description**

As an FO I want the system to

- Trigger calculation of PV prediction from the arrival of new energy values.
- Prediction service returns the calculated PV prediction.

**Roles**

System to system integration

**Preconditions and Assumptions**

Required parameters to the prediction model of PV all have values.

**Acceptance Criteria**

PV predicted values are stored to the systems time series repository.

**3.5.2.8 Calculate charge point status prediction****Purpose**

To calculate charge point prediction as basis for Flexibility Optimization.

**Description**

As an FO I want the system to

- Trigger calculation of charge point prediction from the arrival of new energy values.
- Prediction service returns the calculated charge point prediction.

**Roles**

System to system integration

**Preconditions and Assumptions**

Required parameters to the prediction model of charge point all have values.

**Acceptance Criteria**

Charge point values are stored to the systems time series repository.

## 3.6 Flexibility Management Optimization

### 3.6.1 Overview

The optimization models are developed by UPC in Barcelona and NTNU in Trondheim, as part of T5.4, and integrated as a part of the optimization service in the IIP. The optimization service is designed to handle the flexibility services at prosumer level in phase 2 of the IIP. In IIP phase 3, the BRP and DSO flexibility request management is added.

The service is developed in Python and uses Pyomo package developed by Sandia Lab to build the optimization model. The optimization model is then solved using standard solvers such as open source GLPK or commercial Gurobi. The optimization service is expected to trigger every 15 minutes (time-based triggering) and requires the inputs explained in detail in section 3.6.1.1.

The main process is split into sub-processes as illustrated in Figure 7.

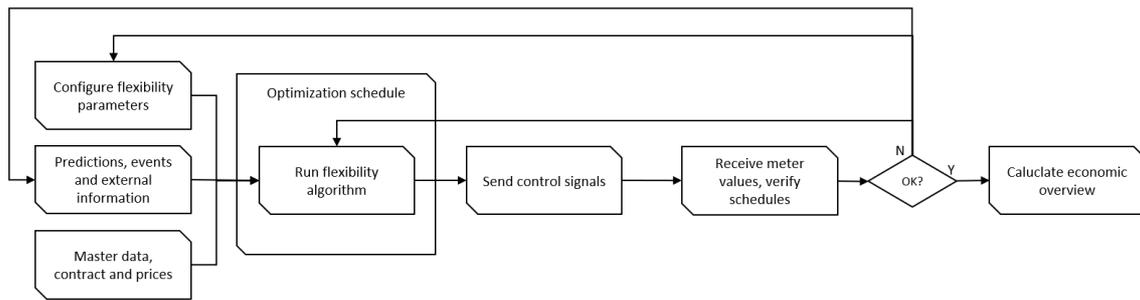


Figure 7: Flexibility Management Optimization process

The output of the optimization service gives a control schedule for configured horizon and intervals for all the flexible devices present in the site and for the whole site. In the Dutch pilot, an optimal capacity will be returned in levels where a capacity constraint is sent as parameter to the optimization service.

The flexibility request management for DSO and BRP will be exposed through separate messages and is described in section 3.9 Flexibility Services Management.

### 3.6.1.1 Optimization input parameters

In this section, the input parameters required for the optimization service are listed. The required input parameters are divided into five categories: predictions data, metered data, fixed technical characteristics, flexibility contracts and contract prices.

The flexibility request management will require the relevant messages to be passed into the model as basis for further optimization. The process is described in section 3.9 Flexibility Services Management.

Additional parameters will be added for the advanced battery model as well as for the advanced models for water heating and space heating.

Parameter name	Unit	Description
<b>Predictions Data</b>		
Generation (PV panels)	kWh	Predicted production
Load (water heaters, space heating)	kWh	Predicted consumption
Charging (EV)	kWh	Predicted consumption
Charge point status	Int	Predicted status
Inflexible load	kWh	Predicted site total consumption – (minus) Predicted total of flexible devices consumption

<b>Metered Data</b>		
Generation (PV panels)	kWh	Received meter data
Load (water heaters, space heating)	kWh	Received meter data
Charging (EV)	kWh	Received meter data
Charge point status	kWh	Received meter data
Charge point status	Int	Derived from received meter data
Inflexible load	kWh	Site total consumption - (minus) flexible devices consumption
Battery	kWh	Received meter data charging and discharging Received state of Charge
<b>Fixed Technical Characteristics</b>		
<b>Generation (PV panels)</b>		
InstalledMaxCapacity	kWh	Max production from resource
<b>Load (water heaters, space heating)</b>		
InstalledMaxCapacity	kWh	Max consumption in resource
<b>Charging (EV)</b>		
InstalledMaxCapacity	kWh	Max consumption in resource
<b>Battery technical parameter sheet</b>		
MaximumChargingPower	kW	Maximum charging power allowed for battery
MaximumDischargingPower	kW	Maximum discharging power allowed for battery
maxCapacity	kWh	Maximum state of charge allowed for battery
minCapacity	kWh	Minimum state of charge allowed for battery
EfficiencyOfCharging	p.u.	Efficiency parameter for charging storage unit
EfficiencyOfDischarging	p.u.	Efficiency parameter for discharging storage unit
ThresholdCharge	p.u.	Threshold in battery unit b charging process

ThresholdDischarge	p.u.	Threshold in battery unit b discharging process
ThresholdStoredEndPH	p.u.	Threshold in battery unit b of stored energy at the end of the planning horizon
<b>Battery economical parameter sheet</b>		
CostCharging	€/kWh	Technical parameter from the battery manual
CostDischarging	€/kWh	Technical parameter from the battery manual
<b>Flexibility contracts</b>		
<b>Generation (PV panels)</b>		
ActivationFeeNoneWorkdays /ActivationFeeWorkdays	€/NOK	Price of reducing generation output of the unit
MaxActivationTime	Minutes	Max activation time for resource
RestTime	Minutes	Minimum rest time between activations
MaxNumberOfActivations	Times per day	Max number of Activations per day
<b>Load (space heating, water heaters)</b>		
ActivationFeeNoneWorkdays /ActivationFeeWorkdays	€/NOK	Price for disconnecting the load unit
MaxActivationTime	Minutes	Max activation time for resource
RestTime	Minutes	Minimum rest time between activations
MaxNumberOfActivations	Times per day	Max number of Activations per day
<b>Charging (EV)</b>		
ActivationFeeNoneWorkdays /ActivationFeeWorkdays	€/NOK	Price for disconnecting the load unit
MaxActivationTime	Minutes	Max activation time for resource
RestTime	Minutes	Minimum rest time between activations
MaxNumberOfActivations	Times per day	Max number of Activations per day

<b>Battery technical parameter</b>		
ActivationFeeNoneWorkdays /ActivationFeeWorkdays	€/NOK	Price for disconnecting the load unit
MaxActivationTime	Minutes	Max activation time for resource
RestTime	Minutes	Minimum rest time between activations
MaxNumberOfActivations	Times per day	Max number of Activations per day
<b>Prices</b>		
Retail Buy (imported spot prices)	€/kWh	Price at energy part of retail contract for buying electricity in period
Retail Sell (imported spot prices)	€/kWh	Price at energy part of retail contract for selling electricity in period
Retail Buy (predicted spot prices)	€/kWh	Predicted price at energy part of retail contract for buying electricity in period
Retail Sell (predicted spot prices)	€/kWh	Predicted price at energy part of retail contract for selling electricity in period
Grid Sell	€/kWh	Price at energy part of grid contract for selling electricity in period t
Grid Buy	€/kWh	Price at energy part of grid contract for buying electricity in period t
Grid Buy Low	€/kWh	Price at energy part of grid contract for buying electricity in cases with subscribed power in period if bought electricity is below subscribed level
Grid Buy High	€/kWh	Price at energy part of grid contract for buying electricity in cases with subscribed power in period if bought electricity is above subscribed level
Capacity Constraint Consumption	kW	Maximum import capacity.  If given as parameter it will trigger the calculation of optimal capacity constraint as the sum of charge point schedules in the flexibility area.
Capacity Constraint Production	kW	Maximum export capacity

Tax	€/kWh	Sum price for all taxes that are related to buying electricity in period
VAT	p.u.	Parameter that adds Value Added Tax to the amount bought
Grid Buy Peak	€/kW/month	Price at grid contract for peak fee
Grid Buy Low - Volume Limit	kW	Limitation of basis for peak fee
Optimization Frequency	#	Periods per hour

**Table 5: Optimization Management input parameters**

### 3.6.2 Calculate optimal flexibility schedule

#### 3.6.2.1 Purpose

The purpose of this process is to generate an optimal flexibility schedule, optimal capacity regulation schedule for the Dutch pilot and relevant flexibility for the set of flexibility request messages.

#### 3.6.2.2 Involved roles

All processes are automated and hence involve only system roles.

#### 3.6.2.3 Preconditions

Historical resource energy values and historical events and external information exists.

Predicted energy values and charge point status exist.

Flexibility contract, contract product and contract asset parameters exist.

Configuration of messages involved in the flexibility request management.

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

The automated process of prosumer optimization will be like this:

1. A schedule triggers the optimization service with the correct parameters for each flexibility contract.
2. The optimization service calculates an optimal flexibility schedule and an optimal capacity regulation schedule.
3. The calculated schedules are returned from the optimization service and stored to the correct time series in the system.

The following user stories are covered:

1. Calculate optimal flexibility schedule.
2. Calculate optimal flexibility schedule and optimal capacity schedule.
3. Calculate basis for flexibility request management

#### 3.6.2.5 Calculate optimal flexibility schedule

##### **Purpose**

To calculate an optimal flexibility schedule for the resources at a prosumer site.

##### **Description**

As an FO I want the system to

- Trigger optimization service at a scheduled time.
- Store the optimization schedule to time series.

##### **Roles**

The IIP.

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

Required parameters to the optimization service exists.

##### **Acceptance Criteria**

Optimal flexibility schedule stored to the systems time series repository.

#### 3.6.2.6 Calculate optimal flexibility schedule and optimal capacity schedule

##### **Purpose**

To calculate an optimal flexibility schedule and an optimal capacity schedule for the resources at a prosumer site.

##### **Description**

As an FO I want the system to

- Trigger optimization service at a scheduled time.

- Store the optimal flexibility schedule and the optimal capacity schedule to time series.

**Roles**

The IIP.

**Preconditions and Assumptions**

Required parameters to the optimization service exists.

Capacity constraint exists as input parameter to the optimization service.

**Acceptance Criteria**

Optimal flexibility schedule and optimal capacity schedule are stored to the systems time series repository.

**3.6.2.7 Calculate basis for flexibility request management****Purpose**

To calculate and store the basis for flexibility request management.

**Description**

As an FO I want the system to

- Trigger optimization service at a scheduled time
- Store the basis for flexibility request management messages to time series

**Roles**

The IIP.

**Preconditions and Assumptions**

Required parameters to the optimization service exists.

Configuration of messages involved in the flexibility request management

**Acceptance Criteria**

Basis for flexibility request messages are stored to time series

### 3.7 Follow up and reporting

This section replaces the section 3.7 Follow up and reporting in D8.2.

#### 3.7.1 Overview

This section will describe in general the follow up and reporting of base time series and calculated time series in the IIP. Base time series include the received meter values for prosumer resources and charge point status, in addition to the requested weather and prices described in section 3.2 Meter values management. Calculated time series include results from IIP calculations, prediction services, flexibility optimization and flexibility request messages. Flexibility request message time series include the basis for sending flexibility request messages described in section 3.9 Flexibility Services Management.

A set of search criteria enables the user to combine energy, predictions, flexibility schedules to get an overview on the status of the optimization process, energy meter values, predictions and flexibility request messages.

Data may be viewed as a chart or listed as meter values in numbers for a single resource or all the resources and areas connected to a site.

The links between different data objects is described in Figure 8.

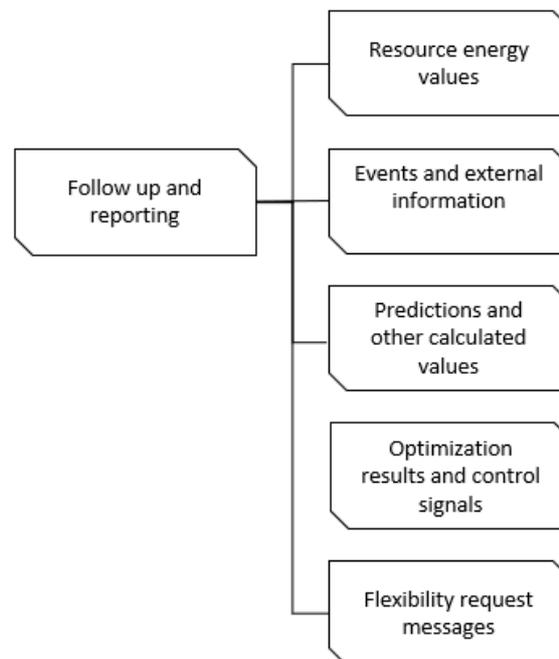


Figure 8: How data is linked in the Value Reporting screen

### 3.7.2 Follow up and reporting user stories

#### 3.7.2.1 Purpose

The purpose of this process is to view all the time series values involved in the process of flexibility in the IIP.

#### 3.7.2.2 Involved roles

FO follow up on time series values.

#### 3.7.2.3 Preconditions

Resources, sites and areas are defined in the system and relevant time series in each topology level have been imported or calculated.

Predictions are calculated.

Flexibility optimization have been executed.

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

The manual process of a follow up in value reporting will be like this:

1. The FO opens the Time Series Values UI from relevant data element
2. The FO chooses to view the values in a chart view or as meter values as numbers.

The following user stories are covered:

1. View time series as chart and values for a charge point.
2. View time series as chart and values for a site.
3. Edit time series values for estimated energy values for a resource.
4. View multiple time series as chart for the optimization process
5. View flexibility message as chart and values for a flexibility contract

#### 3.7.2.5 View time series as chart and values for a Charge point

##### **Purpose**

To examine all time series of a charge point as a manual validation of imported energy values, predictions and the optimization schedule.

**Description**

As an operator of the system I want the system to

- Enter search criteria for a specific charge point.
- View all time series values for a prosumer charge point.

**Roles**

FO examine the time series values of a charge point.

**Preconditions and Assumptions**

Charge point and its connection to flexibility area is defined in the system.

Charge point energy value and status time series exist in the system.

Charge point predictions and optimization schedule exist in the system.

**Acceptance Criteria**

Charge point time series are presented in the system.

**3.7.2.6 View time series as chart and values for a Site****Purpose**

To examine time series of a Site as a manual validation of aggregated values to the site.

**Description**

As an operator of the system I want the system to

- Enter search criteria for a specific site.
- View time series values for the site.

**Roles**

FO examine the time series values of a site.

**Preconditions and Assumptions**

Site and connected resources are defined in the system.

Site time aggregated time series exist in the system.

**Acceptance Criteria**

Site time series are presented in the system.

**3.7.2.7 Edit time series values for estimated energy values for a resource****Purpose**

To update estimated energy values when a time interval has been missing.

**Description**

As an operator of the system I want to

- Enter search criteria for a specific resource.
- Update energy values to replace estimated values.

**Roles**

FO updates estimated energy values.

**Preconditions and Assumptions**

Energy values have been estimated as a result of missing previous value when receiving new energy values.

**Acceptance Criteria**

Estimated energy values have been replaced manually and stored to the systems time series repository.

**3.7.2.8 View multiple time series as chart for the optimization process****Purpose**

To investigate the status of the optimization process.

**Description**

As an operator of the system I want to

- Enter search criteria for a site.

- Investigate the meter values, predictions and optimization schedules in a combined view.

**Roles**

FO investigates the optimization process.

**Preconditions and Assumptions**

Site and flexibility contract and all parameters are defined in the system.

Predictions have been calculated.

Optimization service have returned optimization schedules for the site.

**Acceptance Criteria**

FO has an overview of the optimization process at the site.

**3.7.2.9 View flexibility message values for a flexibility contract****Purpose**

To investigate the status of the flexibility request management

**Description**

As an operator of the system I want to

- Enter search criteria for a flexibility request contract
- Investigate the flexibility messages time series in a combined view

**Roles**

FO investigates the flexibility messages management.

**Preconditions and Assumptions**

Flexibility request contract and all parameters are defined in the system.

Flexibility request messages have been sent or received by the IIP.

**Acceptance Criteria**

FO has an overview of the flexibility request messages.

3.7.2.10 User interfaces

User interface	Short description
Value Reporting	Provides a single screen for presentation of time series in the system as chart or value numbers.  Search criteria must be entered as filter for required data.

**3.8 Control signals management**

This section is not updated in this document but copied from D8.2 for consistency.

**3.8.1 Overview**

The purpose with control signals management is to determine where and how to send the optimized schedule generated by the optimization management. The process is triggered at specified times by the control signal management schedule.

End points will be configured by each pilot company specifying the input to the pilot system.

Control signals management have three main types of messages. These are:

- Asset Control On/Off message – for resources having controllable type OnOff
- Control Regulation Message – for resources having controllable type Continuous
- Optimal Capacity Regulation Message – for flexibility areas providing capacity constraint as input to optimization

The management system at pilot sites has the responsibility to execute the schedules to the resources according to the received schedules.

The main process is split into sub-processes as illustrated in Figure 9.

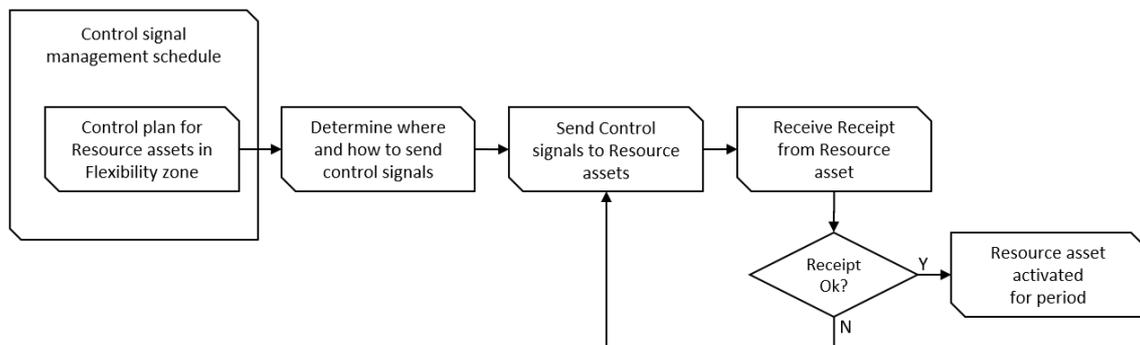


Figure 9. Main process and sub-processes for Control signals management

### 3.8.2 Control signals management user stories

#### 3.8.2.1 Purpose

The purpose of this process is to send control signals and schedules to end points defined in the system.

#### 3.8.2.2 Involved roles

All processes are automated and hence involve only system roles.

#### 3.8.2.3 Preconditions

Optimal flexibility schedule and optimal capacity schedule exists.

Flexibility resources and areas have end points defined in the system.

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

The automated process of control signals management will be like this:

1. A schedule triggers the control signal management.
2. The end point parameters are fetched from the tenant configuration.
4. Optimal schedules are sent at the defined format to the pilot system.

The following user stories are covered:

1. Send optimal flexibility schedule to the pilot system.
2. Send optimal capacity schedule to the pilot system.

#### 3.8.2.5 Send optimal flexibility schedule to the pilot system

##### **Purpose**

To send the optimal flexibility schedule to the pilot system.

##### **Description**

As a system integrator I want the system to

- Call the pilot system with an optimized flexibility schedule.

##### **Roles**

System to system integration.

### **Preconditions and Assumptions**

Optimal flexibility schedule exists.

Flexibility resources and areas have end points defined in the system.

### **Acceptance Criteria**

Optimal flexibility schedule is received and stored in the pilot system.

#### 3.8.2.6 Send optimal capacity schedule to the pilot system

### **Purpose**

To send the optimal capacity schedule to the pilot system.

### **Description**

As a system integrator I want the system to

- Call the pilot system with an optimized capacity schedule.

### **Roles**

System to system integration.

### **Preconditions and Assumptions**

Optimal capacity schedule exists.

Flexibility resources and areas have end points defined in the system.

### **Acceptance Criteria**

Optimal capacity schedule is received and stored in the pilot system.

### 3.9 Flexibility Services Management

This chapter is new in D8.4 thus no reference to D8.2.

#### 3.9.1 Overview

A distribution system operator (DSO) or a balance responsible party (BRP) may request flexibility from a flexibility operator (FO), get a flexibility offer with reference to their request and finally a delivery note of the supplied flexibility.

This section will describe the use cases required to handle this process and how the system should generate and send the related messages.

The messages are described in D7.4.

The usage of the messages for each pilot is described in D4.3.

**Note!** The pilot cases differ from the below described generic sequence diagram. Some of the communication is not relevant and thus, will not be implemented.

Figure 10 (Figure 12 in D4.3) describes the generic interactions for offering flexibility services to DSOs, BRPs and prosumers using flexibility devices (battery, controllable load, EV charging, etc.) coordinated by a FO.

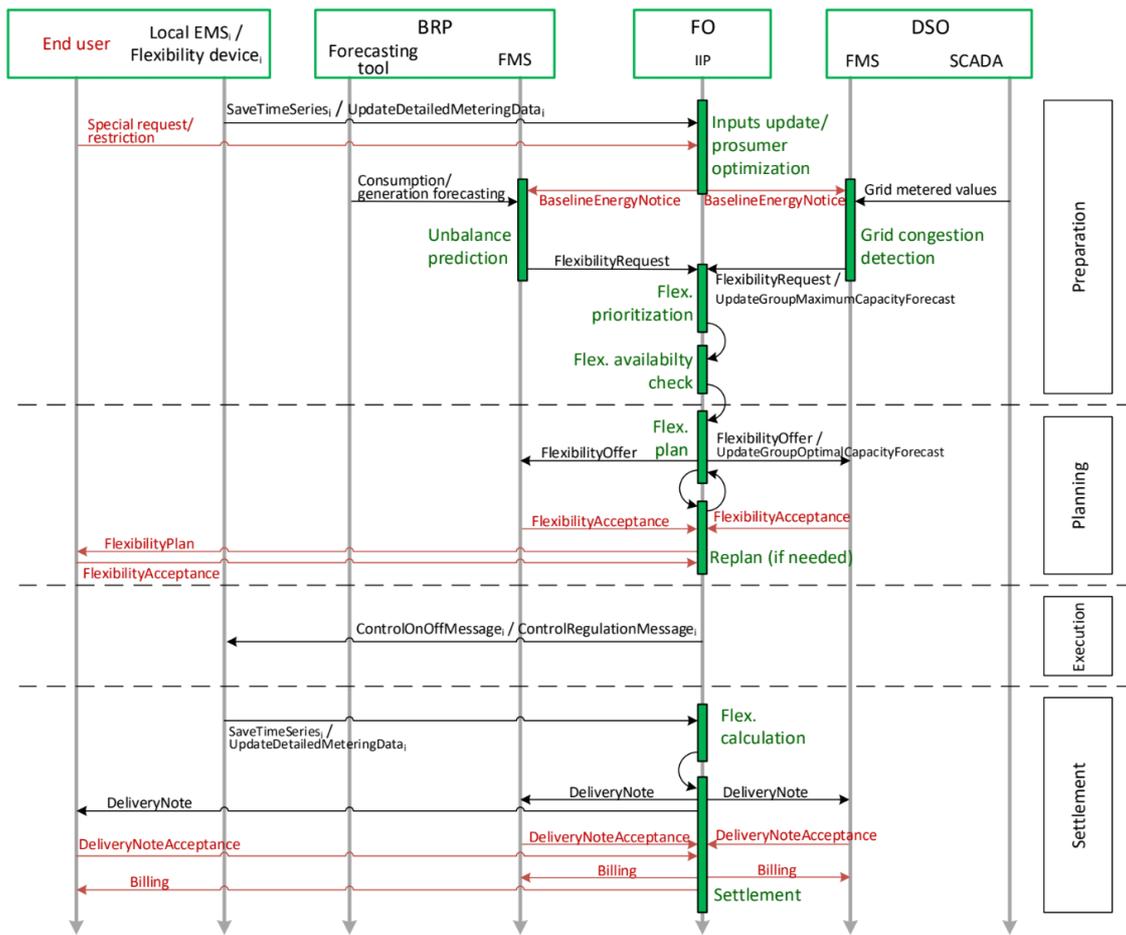


Figure 10. Generic sequence diagram of flexibility services

### 3.9.2 Flexibility Messages user stories

#### 3.9.2.1 Purpose

The purpose of this process is to handle messages between a FO and a DSO and/or BRP required for a request of flexibility.

#### 3.9.2.2 Involved roles

All processes are automated and hence involve only system roles.

#### 3.9.2.3 Preconditions

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

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

The automated process of requesting flexibility will be like this:

1. The IIP does prosumer optimization and calculates BaselineEnergyNotice and sends it to DSO and BRP systems.
2. DSO detects a grid congestion and sends a FlexibilityRequest message to the IIP.
3. BRP predicts an imbalance and sends a FlexibilityRequest message to the IIP.
4. The IIP performs a flexibility prioritization, calculates available flexibility and generates a flexibility plan. The IIP sends the flexibility plan to DSO and BRP as a FlexibilityOffer.
5. BRP or DSO sends a FlexibilityAcceptance to the IIP.
6. The IIP sends ControlOnOffMessage/ControlRegulationMessage to the pilot system for each flexibility device to be regulated. This user story is described in section 3.8.2 Control signals management user stories
7. For a site having regulated devices the difference between baseline and actual energy will be calculated as a DeliveryNote after the flexibility period is passed.
8. The IIP sends a DeliveryNote to DSO or BRP
9. The DSO or BRP sends a DeliveryNoteAcceptance to the IIP. This message type is optional and will not be implemented in the IIP
10. The IIP calculates billing information and sends billing to BRP / DSO and End User. This message type is optional and will not be implemented in the IIP

The following user stories are covered:

1. Calculate and send BaselineEnergyNotice to BRP and DSO
2. Receive FlexibilityRequestMessage from BRP for covering predicted imbalance
3. Receive FlexibilityRequestMessage from DSO for helping in case of grid congestion
4. Calculate flexibility plan and send as FlexibilityOffer to BRP and DSO
5. Receive FlexibilityOfferAcceptance from BRP and DSO
6. Calculate and send DeliveryNote to BRP and DSO

### 3.9.2.5 Calculate and send BaselineEnergyNotice to BRP and DSO

#### **Purpose**

To calculate and send a BaselineEnergyNotice to BRP and DSO

### **Description**

As an FO I want the system to

- Calculate a BaselineEnergyNotice on a schedule
- Send the BaselineEnergyNotice to the relevant BRP or DSO

### **Roles**

System to system integration.

### **Preconditions and Assumptions**

FO has a portfolio of prosumers with available flexibility and has performed prosumer optimization as well as calculated a baseline

A contract between FO and BRP with a definition of flexibility area and BaselineEnergyNotice has already been defined in the system.

### **Acceptance Criteria**

BaselineEnergyNotice is calculated and sent to the relevant BRP or DSO.

#### **3.9.2.6 Receive FlexibilityRequestMessage from BRP for covering predicted imbalance**

### **Purpose**

To capture FlexibilityRequestMessage from BRP for covering the BRPs predicted imbalance.

### **Description**

As a BRP I want to be able to

- Request flexibility from an FO to cover predicted imbalance.

### **Roles**

System to system integration

### **Preconditions and Assumptions**

FO has a portfolio of prosumers with available flexibility and has performed prosumer optimization as well as calculated a baseline

A contract between FO and BRP with definition of flexibility area and FlexibilityRequestMessage has already been defined in the system.

### **Acceptance Criteria**

On supply of valid data, the flexibility request data are stored to the correct time series in the systems time series repository and used as parameter for optimization.

#### **3.9.2.7 Receive FlexibilityRequestMessage from DSO for helping to mitigate grid congestion**

### **Purpose**

To capture FlexibilityRequestMessage from DSO for helping to mitigate grid congestion.

### **Description**

As a DSOI want to be able to

- Request flexibility from an FO for helping to mitigate grid congestion

### **Roles**

System to system integration

### **Preconditions and Assumptions**

FO has a portfolio of prosumers with available flexibility and has performed prosumer optimization as well as calculated a baseline

A contract between FO and DSO with a definition of flexibility area and FlexibilityRequestMessage has already been defined in the system.

### **Acceptance Criteria**

On supply of flexibility request, data are stored to the correct time series in the systems time series repository.

#### **3.9.2.8 Calculate flexibility plan and send as FlexibilityOffer to BRP or DSO**

### **Purpose**

To perform flexibility prioritization, check flexibility availability and calculate a flexibility plan to be sent to relevant BRP or DSO as FlexibilityOffer.

### **Description**

As an FO I want the system to

- Trigger optimization from arrival of FlexibilityRequestMessage to perform flexibility prioritization, check available flexibility and calculate a flexibility plan
- Send the flexibility plan as a FlexibilityOffer to the relevant DSO or BRP

### **Roles**

System to system integration

### **Preconditions and Assumptions**

FlexibilityRequestMessage is received from BRP or DSO.

### **Acceptance Criteria**

On supply of valid flexibility request the system is triggered to perform flexibility prioritization, check flexibility availability and calculate a flexibility plan to be sent to relevant BRP or DSO as FlexibilityOffer.

A flexibility offer is sent to the relevant BRP or DSO.

#### **3.9.2.9 Receive FlexibilityOfferAcceptance from BRP or DSO**

### **Purpose**

To capture FlexibilityOfferAcceptance from BRP or DSO based on a FlexibilityOffer

### **Description**

As a BRP or DSO I want to be able to

- Accept the offered flexibility from an FO

### **Roles**

System to system integration

**Preconditions and Assumptions**

FO has sent a FlexibilityOffer message

**Acceptance Criteria**

On supply of valid data, the flexibility offer acceptance data are stored to the correct time series in the systems time series repository

**3.9.2.10 Calculate and send DeliveryNote to BRP and DSO****Purpose**

To calculate the difference between baseline and actual energy after the flexibility period is passed and actual energy values are captured. The difference between baseline and actual energy is sent as DeliveryNote to relevant BRP or DSO

**Description**

As an FO I want the system to

- Calculate a DeliveryNote on a schedule after the flexibility period is passed
- Send the DeliveryNote to the relevant BRP or DSO

**Roles**

System to system integration.

**Preconditions and Assumptions**

A flexibility offer is sent to BRP or DSO.

The flexibility period is passed and actual energy values are captured.

**Acceptance Criteria**

Delivery note is calculated and sent to the relevant BRP or DSO.

## **4 Integrated INVADE Platform - Web UI**

This section replaces the section 4 Integrated INVADE Platform - Web UI in D8.2.

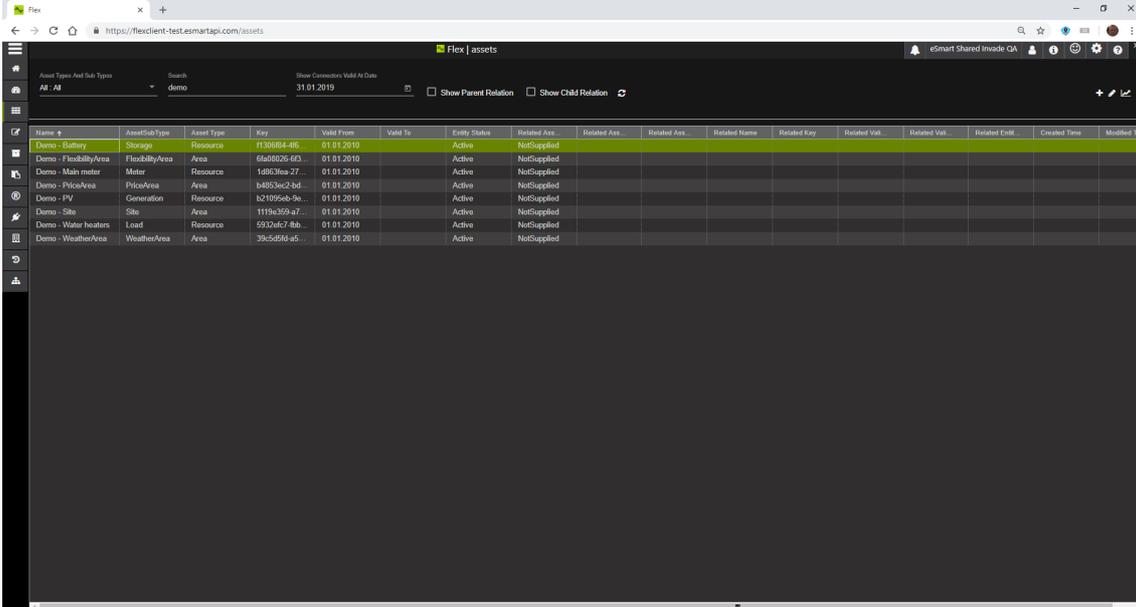
## 4.1 Overview

All user interfaces in the IIP are released as one common installation for all pilots and are accessed from a web browser with a URL provided by eSmart. Google Chrome has been selected as the preferred browser. Thus, all users should have this browser available and strive to update regularly to latest version.

New version will be installed by eSmart to maintain compatibility between client and data repositories.

### 4.1.1 List view

All registration screens are opened with a list view to filter relevant data.



Name	AssetSubType	Asset Type	Key	Valid From	Valid To	Entity Status	Related Ass	Related Ass	Related Ass	Related Name	Related Key	Related Val	Related Val	Related Entit	Created Time	Modified Time
Demo - Battery	Storage	Resource	f136684-46...	01.01.2010		Active	NotSupplied									
Demo - FlexibilityArea	FlexibilityArea	Area	66d8025-6D...	01.01.2010		Active	NotSupplied									
Demo - Math meter	Meter	Resource	f48038ea-2f...	01.01.2010		Active	NotSupplied									
Demo - PilotArea	PilotArea	Area	b4833e02-bd...	01.01.2010		Active	NotSupplied									
Demo - PV	Generation	Resource	b21075eb-9e...	01.01.2010		Active	NotSupplied									
Demo - Site	Site	Area	1119e359-a7...	01.01.2010		Active	NotSupplied									
Demo - Water heaters	Load	Resource	5932e6c7-fb...	01.01.2010		Active	NotSupplied									
Demo - WeatherArea	WeatherArea	Area	39c5e6f0-a5...	01.01.2010		Active	NotSupplied									

Figure 11 – List screen

### 4.1.2 Detailed screen

Double-clicking on a row, or pressing the + sign for new item opens a comprehensive view to add details to the item.

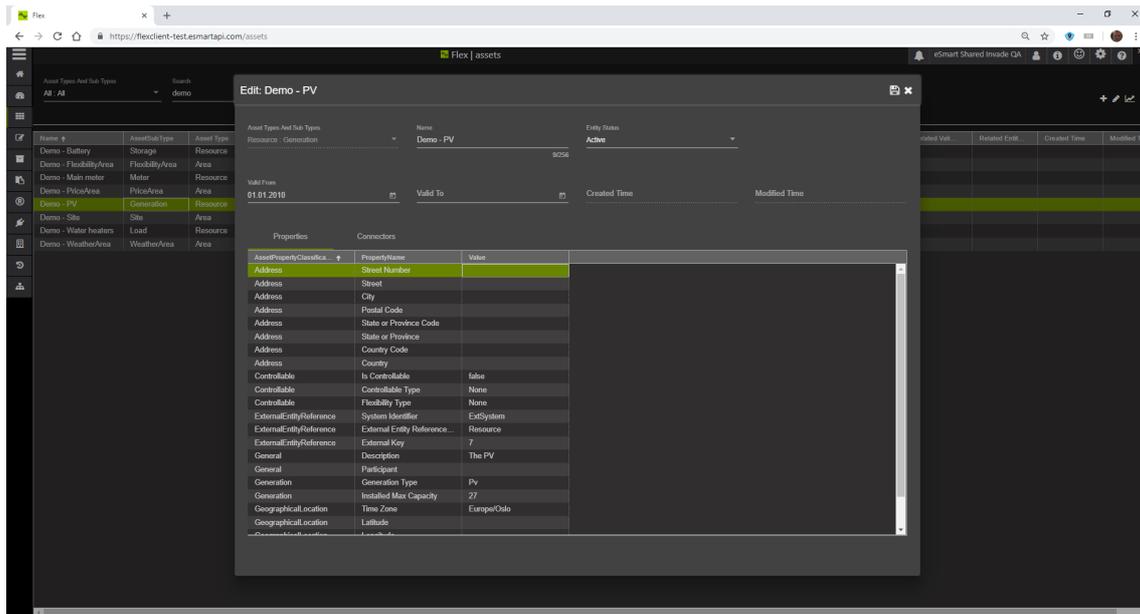


Figure 12 - Detailed screen

### 4.1.3 Time series values

When selecting a row in a list a chart symbol will display in the upper left corner. Pressing this symbol will open the Time Series Values screen. Values may be displayed as chart or matrix as described in section 3.7 Follow up and reporting.

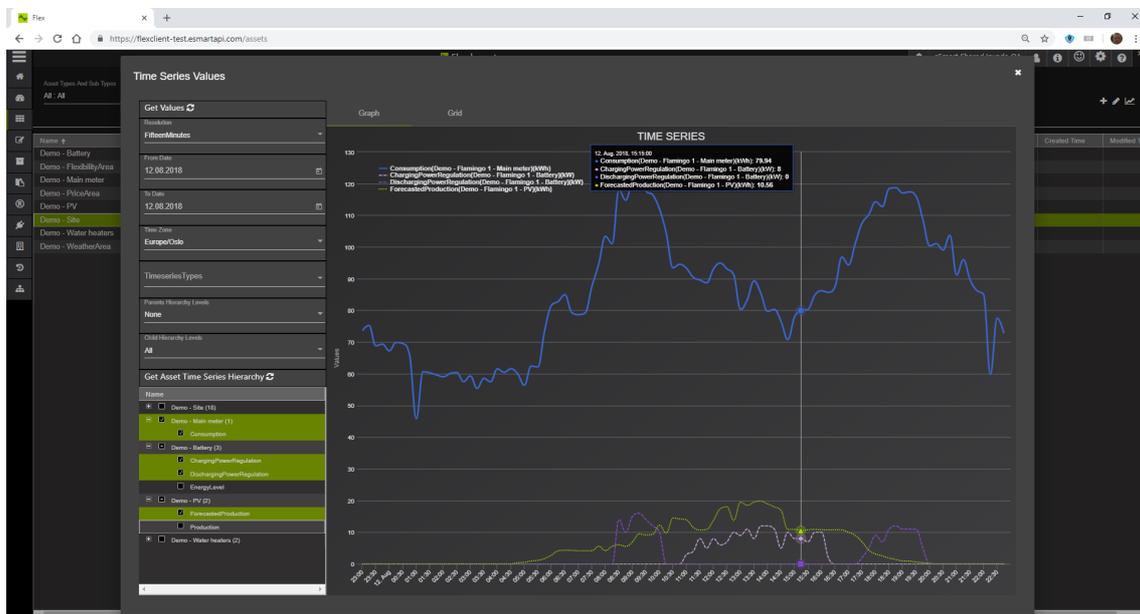


Figure 13 - Time Series Values

## 4.2 Security

### 4.2.1 Tenants

Each pilot will be setup as a separate tenant in Microsoft Azure to ensure no data is visible across pilot companies. GreenFlux and ElaadNL will be setup with separate tenants since these are separate companies within the Netherland pilot.

### 4.2.2 User configuration

Login to the IIP is handled with Auth0 security.

Pilots will be asked for a list of users containing name and a valid e-mail address, per company. User name could be an anonymous user code to maintain user anonymity which could be completed with a matching anonymous e-mail address.

Users will be configured by eSmart in IIP version 1.

The first time a user logs in with the provided password, he or she will be asked to change password.

Login screen will look like this:

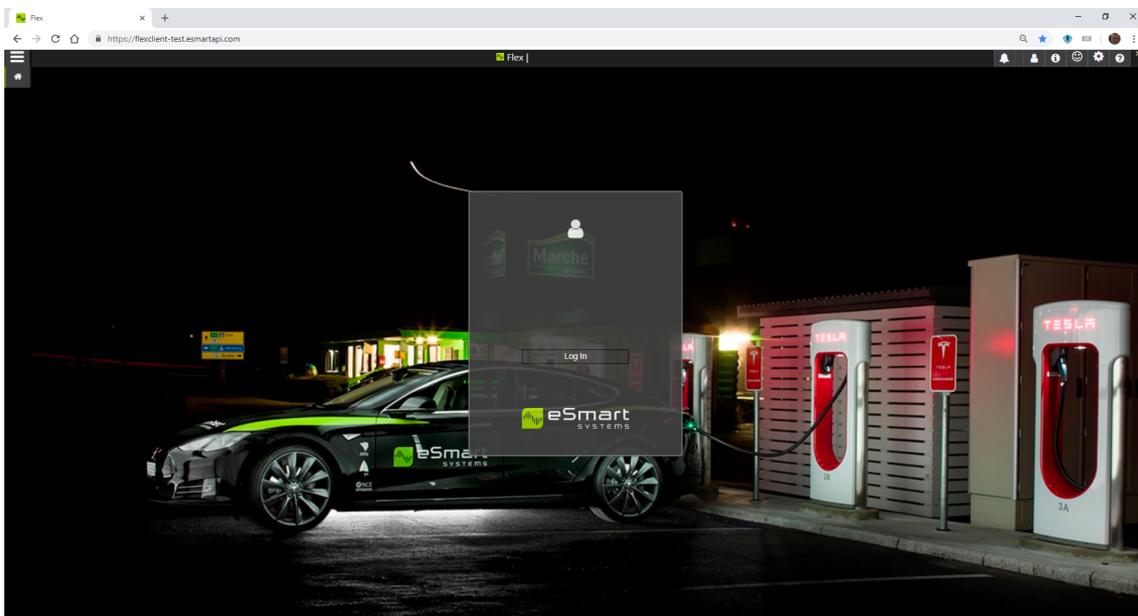


Figure 14 – eSmart login screen