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

Deliverable nº:	**D8.3**
Deliverable name:	**End-user mobile apps and management dashboards, phase 1**
Version:	**1.0**
Release date:	**25/06/2018**
Dissemination level:	**Public**
Status:	**Approved**
Author:	**eSmart Systems – Susann K. Eriksen, Jon Egil Spetaas and Terje Lundby**

Document history:

Version	Date of issue	Content and changes	Edited by
0.1	16/05/18	First draft version	Susann Kjellin Eriksen
0.2	25/06/18	Draft sent for peer-review	Susann Kjellin Eriksen
0.3	27/06/18	Draft updated with FAT results and executive summary	Jon Egil Spetaas, Susann Kjellin Eriksen, Terje Lundby

Peer reviewed by:

Partner	Reviewer
SmartIO	Joseph Negreira

Deliverable beneficiaries:

WP / Task
WP4 / T4.6
WP9 / T9.4
WP10 / T10.2 – T10.7

Table of contents

Executive summary

The Integrated INVADE platform is a central delivery in the INVADE project. The platform will be used by the Flexibility Operator to manage flexibility from flexibility providers and offer this to flexibility customers.

This document describes how the Factory Acceptance Test (**FAT**) of the Integrated INVADE platform version 1 is conducted and the results from the test. The test is run to demonstrate and verify the functionality of the platform and is run from eSmart's office in Halden, Norway.

The results from the test show that all test steps are passed, except optimization of EV and the belonging SiteHousehold2, as the EV model in the optimization algorithm was not triggered. This will be investigated and fixed before the platform is taken into use by pilots. Four steps are marked with comments, which also will be followed up. In total, the FAT of the Integrated INVADE platform version 1 is approved. The pilots will start using the platform during autumn 2018.

1 Introduction

The delivery D8.3 End-user mobile apps and management dashboards, phase 1, is the Integrated INVADE platform (**IIP**) version 1 software. The IIP will be implemented at pilot sites during autumn 2018 and used by the Flexibility Operator during the pilot phase of the INVADE project.

This document describes how the Factory Acceptance Test (FAT) for the IIP version 1 is conducted. The test is run to demonstrate and verify the platform functionality developed in phase 1. The functionality of the platform is described in other deliveries in both WP5, WP7 and WP8. More specifically, the platform's flexibility algorithms are presented in D5.3, the communication platform API is presented in D7.2, the platform's asset management, asset control and data repository are presented in D8.1 and the business intelligence part of the IIP is presented in D8.2.

The test cases presented in this delivery are based on the use cases defined in document D.8.1 and D.8.2, and describe the required test steps, including expected results when appropriate. The test cases also define purpose, assumptions and pre-conditions that must be met before the test cases can be run. The FAT is being run from eSmart's office in Halden, Norway.

The structure of the document follows the IIP main processes, which can be seen in Appendix 1. All processes expect control request management, control signals management and follow-up and reporting are covered in the test. The process overview has previously been presented and explained in D8.1 and D8.2.

2 Test sites

The FAT is focused around two fictive test sites. The sites have been constructed to demonstrate various functionality in the IIP. Both sites are part of the same weather, price and flexibility area, but are set up with different resources. Site 1 has a battery, PV and non-controllable load, while site 2 has an EV charger, PV and non-controllable load. The asset hierarchy can be seen in Figure 1.

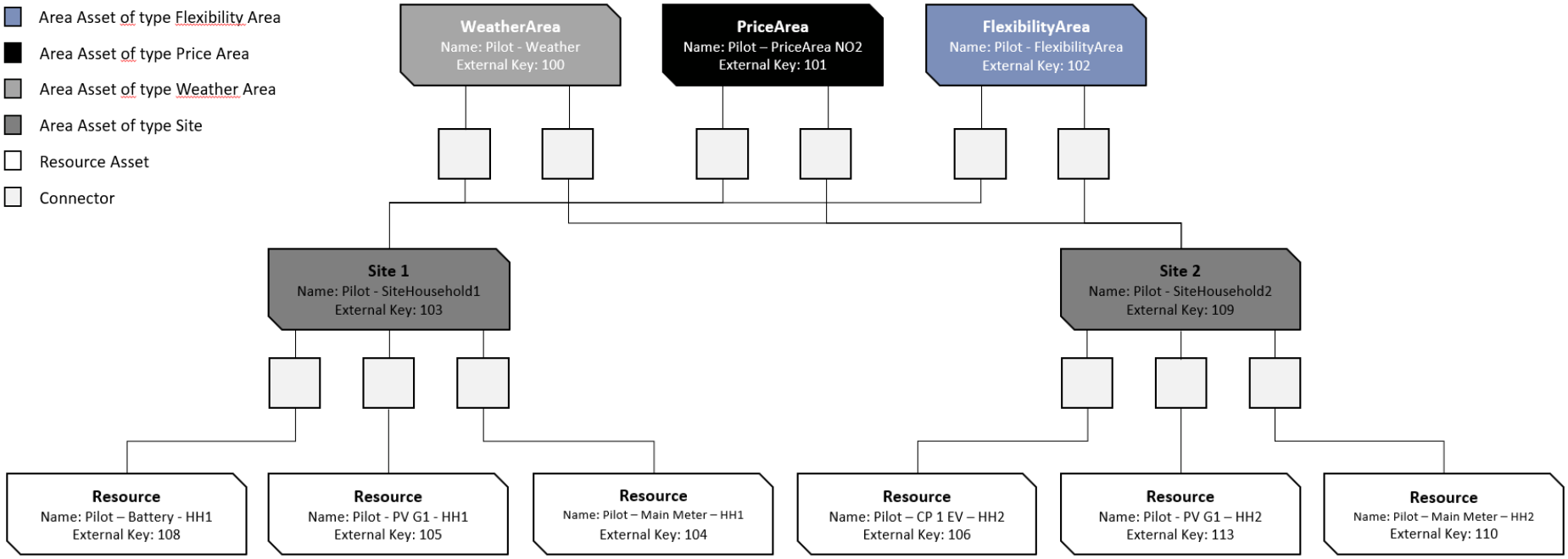

Figure 1: FAT test sites

3 Master data and Configurations

This section describes the test cases related to the master data and configurations process.

3.1 Asset, area and zone management – Automatic

3.1.1 Purpose

The purpose of this prosses is to import asset data through the API.

In this FAT, asset data have already been imported through an excel API template.

In this testcase we will import a dummy asset.

3.1.2 Assumptions and Pre-Conditions

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

3.1.3 Test Steps: New Asset, area and zone

	Action	Expected result	Result
1.	Open the document: Asset loader – FAT.txt	Excel document opens with pre-configured dummy asset	OK
2.	Open Swagger: https://invadectplatformapi.azurewebsites.net/swagger/ui/index	Swagger is opened	OK
3.	Click on AssetAPi		OK
4.	Click on /api/External/AssetAPi/UploadAssetCsvData	UploadAssetsCSVData is open	OK
5.	Go to Asset loader – FAT.txt and copy all the text		OK
6.	Go to UploadAssetsCSVData and paste the parameter string into assetCsvdata field.	Parameter string is pasted correctly	OK

7.	Click on button “Try it out!”	The dummy assets will be imported into the system	OK
8.	Open the Asset Screen	Asset Screen is opened	OK
9.	Click on Search and enter eSmart Click on Refresh	All eSmart Assets are opened in the grid and imported asset is shown in the list.	OK

3.2 Asset, area and zone management – Manual entering

3.2.1 Purpose

The purpose of this process is to enter and update information about assets, areas and sites in the asset screen.

3.2.2 Assumptions and Pre-Conditions

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

3.2.3 Test Steps: New Asset, area and zone

	Action	Expected result	Result
1.	Open the Asset Screen	Asset Screen is opened	OK
1.	Click on “+” icon to add new Asset	“Create Asset” window is opened.	OK
2.	Click on “Asset Types and Sub Types” dropdown menu. Choose Resource Meter	Selected “Asset Types and Sub Types” is displayed. Once the Asset type is selected, list of	OK

		property is displayed below under the “Properties” tab.	
3.	Select required “Entity Status”	Selected “Entity Status” is displayed	OK
4.	Enter the Name “eSmart - Main Meter” and Valid from and Valid To date		OK
5.	Fill in the required Property: **Description:** eSmart - Main Meter **TimeZone:** CET **Latitude:** **Longitude:** **StreetNumber:** **Street:** **City:** **PostalCode:** **StateOrProvinceCode:** **StateOrProvince:** **CountryCode:** **Country**: **MeterType**: EnergyProductionAndConsumption **IsMainMeter**: TRUE		OK
6.	Select “Connectors” tab	Connectors tab is opened	OK
7.	Hover over the first “+” button and second “+” button.	The tooltips for both buttons are displayed. “Add Child” for first button and “Add Parent” for second button.	OK

8.	Click on “Add Parent” button	“Add Parent” window pops up.	OK
9.	Click on the “Asset Types and Sub Types” and select Area: Site	Select Asset is displayed	OK
10.	Click directly on refresh or enter text to search for a site that already exists (e.g eSmart - Site) and click on refresh button	Display with sites matching the search criteria is displayed.	OK
11.	Select the Asset “eSmart – Site” from dropdown menu	Select Asset is displayed	OK
12.	Click the OK button	“Add Parent” window is closed, and the new parent will be listed in Connector tab.	OK
13.	Click the Save button	The Asset, properties and connectors will be saved.	OK

3.2.4 Test Steps: Edit Asset, area and zone

	Action	Expected result	Result
1.	Double click on one of the assets to edit.	Detail window opens in Edit mode.	OK
2.	Edit the desired field and tab out of the field.		OK
3.	Click on the Save button	Data is saved and detail window is closed.	OK
4.	Click on the asset that was edited and check if it's correct.	Edit window is opened to verify that recent changes are saved.	OK

3.3 Asset, area and zone management – Pre-configuration

3.3.1 Purpose

The purpose of this process is to go through the pre-configuration for an asset done by eSmart before the FAT.

3.3.2 Assumptions and Pre-Conditions

The assets, areas and sites used in the FAT have been pre-configured by eSmart.

3.3.3 Test Steps: Open FAT Configuration for Asset

	Action	Expected result	Result
1.	Open Asset screen	Asset Screen is opened	OK
2.	Click on Search field and enter Pilot Click on the refresh button	Assets types matching the search criteria are displayed in the grid.	OK
3.	Go through the configuration and compare it with the hierarchy drawing.	The configuration matches the hierarchy drawing.	OK with comment: EV should have a property for Minimum Charging Power

4 Contract Management

This section describes the test cases related to the contract management process.

4.1 Legal Entity

4.1.1 Purpose

The purpose of this process is to enter and update Legal Entity, in addition to go through the FAT configuration.

4.1.2 Assumptions and Pre-Conditions

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

4.1.3 Test Steps: New Legal Entity

	Action	Expected result	Result
1.	Open Entities screen	Entities screen is opened	OK
2.	Click on “+” icon to add new legal entity	List of fields appear on the right-hand side of the window.	OK
3.	Fill in all the required fields: **Name**: eSmart **Roles:** Flexibility Operator, owner and counterparty Entity Status: Active Company (org. no): 987654321	The Save button is enabled.	OK with comment: Role type Counterparty will include consumer and prosumers.
4.	Click on Save button	The recently created entity is saved and displayed in the list.	OK
5.	Verify that recently entered entity is saved by entering eSmart in the Name field and click “Refresh” button.	Recently entered entity is displayed in the grid.	OK

4.1.4 Test Steps: Edit Legal Entity

	Action	Expected result	Result
1.	Search for a legal entity to edit by entering eSmart in the Name field or choose a role and click “Refresh” button.	Entities matching the search criteria are displayed in the list.	OK
2.	Double click on “eSmart” to edit.	List of fields with previously entered values appear on right-hand side of the window.	OK
3.	Edit the desired field	Edited field is changed and save button is enabled.	OK
4.	Click on “Save” button	Recently made changes are saved.	OK
5.	Double click again on same entity and verify that recently made changes are saved.	Recently made changes are saved.	OK

4.2 Legal Entity – Pre-configuration

4.2.1 Purpose

The purpose of this process is go through the Pre-configuration for a Legal Entity done by eSmart before the FAT.

4.2.2 Assumptions and Pre-Conditions

The legal entities used in the FAT have been pre-configured by eSmart.

4.2.3 Test Steps: Open FAT Configuration for Entity

	Action	Expected result	Result
1.	Open Entities screen	Entities screen is opened	OK
2.	Click on name field and enter Pilot. Click on the refresh button.	Search criteria matching “pilot” are displayed in the grid.	OK
3.	Double click on the entity “PilotSystem”.	“PilotSystem” entity is opened.	OK
4.	Go through the pre-configured legal entities: PilotSystem – FlexOp PilotSystem - Prosumer	Configuration is ok.	OK

4.3 Contract Product

4.3.1 Purpose

The purpose of this process is to enter and update Contract Product for prosumer optimization contracts.

4.3.2 Assumptions and Pre-Conditions

Relevant information about Contract Product used in the FAT is available.

The Legal Entities used in the FAT have already been created in the system.

4.3.3 Test Steps

	Action	Expected result	Result
1.	Open the Contract Product Screen	Contract Product Screen is opened	OK
2.	Click on “+” icon to add new Contract Product	“Create Contract Product” window is opened.	OK
3.	Fill up all the required field for Contract Product details: **Name:** eSmart **Contract type**: ProsumerOptimization **Entity Status:** Active **Price Unit**: NOK/kWh **Currency Unit**: NOK **Quantity Unit**: kWh **VAT Percentage**: 25 **Available From Time**: 01/01/2018	Entered details are shown in required fields, and “Save” button is enabled.	OK
4.	Click on “Save” button	Once the “Save” button is clicked, the screen closes.	OK
5.	Search for and open the saved Contract product and verify that Flexibility Rules are filled in with default values. Change the CapacityConstraintAppliesToResourceTypes to blank.	Verify that recently created entity still exist in the list, and that CapacityConstraintAppliesToResourceTypes is blank.	OK
6.	Click on “Save” button	Entered items are saved and the screen closes.	OK
7.	Open the saved Contract product and enter Contract Products Elements by pressing the +:		OK

	FixedFee: **Amount Type:** FixedFee **Related ContractType:** Energy **Calculation Frequency:** Month **Amount Period**: Month **Quantity Limit Tier:** No **Quantity Lower Limit:** **Quantity Lower Limit Operator:** **Quantity Upper Limit :** **Quantity Upper Limit Operator:** **Price Value:** 47 **Price Multiplier:** **Price MarkUp:** **Vat Percentage:** 25 **Entity Status:** Active VariableConsumptionAndSpotPriceWithMarkUp: **Amount Type:** VariableConsumptionAndSpotPriceWith MarkUp **Related ContractType**:Energy **Calculation Frequency**: Hour **Amount Period**: Month **Quantity Limit Tier No:** **Quantity Lower Limit Operator:** **Quantity Lower Limit:** **Price Value:** **Price Multiplier** **Price MarkUp:** 0.045		

	Vat Percentage: 25 **Entity Status**: Active		
8.	Click the “Save” button in Contract Product Element popup	Entered items are saved and the Contract Product Element popup screen closes.	OK
9.	Click the “Save” button in Contract Product screen	Contract Product is saved and the screen is closed	OK

4.3.4 Test Steps: Edit Contract Product

	Action	Expected result	Result
1.	Double click on eSmart to edit.	Popup displays previously entered values.	OK
2.	Edit the desired field.	Desired fields are edited and save button is enabled.	OK
3.	Click on “Save” button.	Recently made changes are saved.	OK
4.	Double click again on the same contract product and verify that recently made changes are saved.	Recently made changes are saved.	OK

4.4 Contract Product – Pre-configuration

4.4.1 Purpose

The purpose of this process is go through the Pre-configuration of contract product done by eSmart before the FAT.

4.4.2 Assumptions and Pre-Conditions

The Contract Product used in the FAT have been pre-configured by eSmart.

4.4.3 Test Steps: Open FAT Configuration for Contract Product

	Action	Expected result	Result
1.	Open Contract Product screen.	Contract Product Screen is opened.	OK
2.	Click on search field and enter Pilot. Click on the refresh button	Search criteria matching pilot are displayed in the grid.	OK
3.	Double click on the contract product.	PilotSystem CP is opened.	OK
4.	Go through the configuration and verify. eSmart to explain each element in the Contract Product	Configuration is ok.	OK

4.5 Contract

4.5.1 Purpose

The purpose of this process is to enter and update Contract and all parameters for prosumer optimization.

4.5.2 Assumptions and Pre-Conditions

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

The Legal Entity have already been created in the system.

Contract Product have already been created in the system.

4.5.3 Test Steps: New Contract

	Action	Expected result	Result
1.	Open the Contract Screen	Contract Screen is opened	OK

2.	Click on “+” icon to add new Contract	“Create Contract” window is opened.	OK
3.	Fill up all the required field including for **Contract details:** **Name:** eSmart **Contract Type:** ProsumerOptimization **Buy Sell type**: Sell **Owning Legal Entity**: eSmart **Counterparty:** eSmart **Contract Product**: eSmart **Delivery From Time**: 01/01/2018 **Entity Status**: Active **Area type**: FlexibilityArea **Area Name**: eSmart FlexArea **Optimization Level:** Site **Master Agreement**: N/A **Master Contract**: N/A		OK
4.	Contract Parameters are filed out with default values. Go through the parameters and verify if it’s ok.	Parameters are found to be ok.	OK
5.	Click on “Save” button	New contract is saved and the screen closes.	OK

4.5.4 Test Steps: Edit Contract

	Action	Expected result	Result
1.	Double click on one of the displayed contracts.	Popup displays previously entered values.	OK
2.	Edit the desired field	Save button is enabled once a field is edited.	OK
3.	Click on “Save” button	Verify that recently made changes are saved.	OK
4.	Double click again on the same contract and verify that recently made changes are saved.	Recently made changes are saved.	OK

4.6 Contract Product – Pre-configuration

4.6.1 Purpose

The purpose of this process is go through the Pre-configuration of contract done by eSmart before the FAT.

4.6.2 Assumptions and Pre-Conditions

The Contract used in the FAT have been pre-configured by eSmart.

4.6.3 Test Steps: Open FAT Configuration for Entity

	Action	Expected result	Result
5.	Open Contract screen	Contract Screen is opened	OK
6.	Click on search field and enter Pilot. Click on the refresh button	Search criteria matching “Pilot” are displayed in the grid.	OK
7.	Double click on the contract.	PilotSystem Opt Contract is opened.	OK
8.	Go through the configuration and verify.	Configuration is ok	OK

5 Events and external information

This section describes the test cases related to the events and external information process.

5.1 Import weather observations and forecasts

5.1.1 Purpose

The purpose of this process is to import weather observations to investigate the data.

In this test case weather observations have already been imported through an Excel API template. Automatic import will be implemented later.

5.1.2 Assumptions and Pre-Conditions

The weather area and its topology have already been defined in the system.

5.1.3 Test Steps

	Action	Expected result	Result
1.	Open the Asset Screen	Asset Screen is opened	OK
2.	Click on Search and enter Pilot. Click on Refresh	All Assets matching search criteria Pilot are displayed in the grid.	OK
3.	Select the Pilot - Weather and click on the Time Series Values ikon 	The Time Series Values screen is opened.	OK
4.	Choose the period, press the plus on Pilot – Weather, check off the ActualTemperature and press refresh.	Values are shown in the graph and the grid.	Ok with comment: Display of Weather time series ActualCloudiness did not display in the Graph the first time.

5.	Verify that the data looks logical.	Data is logical	OK
6.	Repeat the step 4 and 5 for: - ActualWindSpeed - Actualcloudiness - ActualWindDirection		OK

5.2 Scheduled request for day ahead prices

5.2.1 Purpose

The purpose of this process is to import day ahead prices and investigate the data.

In this testcase, day ahead prices have already been imported through an Excel API template. Automatic import will be implemented later.

5.2.2 Assumptions and Pre-Conditions

The Price Area and its topology have already been defined in the system.

5.2.3 Test Steps

	Action	Expected result	Result
1.	Open the Asset Screen	Asset Screen is opened	OK
2.	Click on Search and enter Pilot. Click on Refresh	All Assets matching search criteria Pilot are displayed in the grid.	OK
3.	Select the Pilot - PriceArea and click on the Time Series Values ikon 	The Time Series Values screen is opened	OK
4.	Choose the period, press the plus on Pilot –	Values will be shown in the graph and the grid.	OK

	PriceArea, check off the SpotPrice and press refresh.		
5.	Verify that the data looks logical.	Data is logical.	OK

6 Meter values management and prediction

This section describes the test cases related to the meter values management and prediction processes.

6.1 Receive PV Meter Readings

6.1.1 Purpose

The purpose of this process is to import PV meter readings, calculate the delta values, estimate (interpolate) any missing values and calculate predictions.

In this test case time series will be imported through an Excel API template.

6.1.2 Assumptions and Pre-Conditions

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

6.1.3 Test Steps

	Action	Expected result	Result
1.	Open the text files for PV in Site 1 and Site 2 and copy all the text: - 6.1.3-Pilot - PV G1 - HH1.txt - 6.1.3-Pilot - PV G1 – HH2.txt	Text files with pre-configured time series values	OK
2.	Open Swagger: https://invadectplatformapi.azurewebsites.net/swagger/ui/index	Swagger is opened	OK
3.	Enter login information		OK
4.	Click on TimeSeriesApi		OK
5.	Click on /api/External/TimeSeriesApi/ SaveTimeSeriesListForEntitiesV1	SaveTimeSeriesList-ForEntitiesV1 is opened	OK
6.	Go to SaveTimeSeriesListForEntitiesV1 and	Parameter string is pasted correctly.	OK

	paste the parameter string into timeSeriesV1List field.		
7.	Click on button Try it out!	Time series values are imported into the system	OK
8.	Repeat step 6. to 8. for text copied from: - 6.1.3-Pilot - PV G1 – HH2.txt		OK
9.	When the values are imported these calculations/Workers will automatically be run: **Delta Calculation** – Converting Meter Reading to energy **Estimation Calculation** - If Meter Reading is missing the calculation will interpolate. **Forecast Calculation:** Production time series are pushed to Prediction service. Prediction service push back the forecasted time series for production.	Energy and prediction Time Series are created: - Production time series - ForecastedProduction time series	OK
10.	Open the Asset Screen	Asset Screen is opened	OK
11.	Click on Search and enter Pilot. Click on Refresh	All Assets matching search criteria Pilot are displayed in the grid.	OK
12.	Select the Pilot – PV G1 - HH1 and click on the Time Series Values ikon 	The Time Series Values screen is opened.	OK
13.	Choose the period, press the plus on Pilot – PV G1 - HH1, check off the Production meter reading and Production and press refresh button.	Values are shown in the graph and the grid.	OK

14.	Verify that the production data is calculated correctly. Formula: MeterReading t+1 – MeterReading t = Energy per 15 min	Data is correct.	OK
15.	Verify that the estimation (interpolation) is correct where the Meter Reading was missing. Formula: Interpolate between previous value and last value.	Data is correct	OK
16.	Go to the Time Series Values - Search filter, check off the time series Production and ForecastedProduction and press refresh button. Verify that the prediction is logical.	Values are shown in the graph and the grid. Predictions are logical.	OK
17.	Repeat from step 1-16 for PV-Pilot site 2		OK

6.2 Receive Battery Meter Readings

6.2.1 Purpose

The purpose of this process is to import Battery state of charge and meter readings for charge and discharge, calculate the delta values and estimate (interpolate) any missing values.

In this testcase time series will be imported through an Excel API template.

6.2.2 Assumptions and Pre-Conditions

The Battery resources and its topology have already been defined in the system.

6.2.3 Test Steps

	Action	Expected result	Result
1.	Open the text files for Battery and copy the text: - 6.2.3-Pilot - Battery ChargingMeterReading.txt - 6.2.3-Pilot - Battery DischargingMeterReading.txt - 6.2.3-Pilot – Battery EnergyLevel.txt	Text file opened with pre-configured time series values.	OK
2.	Open Swagger: https://invadectplatformapi.azurewebsites.net/swagger/ui/index	Swagger is opened	OK
3.	Enter login information		OK
4.	Click on TimeSeriesApi		OK
5.	Click on /api/External/TimeSeriesApi/ SaveTimeSeriesListForEntitiesV1	SaveTimeSeriesList-ForEntitiesV1 is opened	OK
6.	Go to SaveTimeSeriesListForEntitiesV1 and paste the parameter string into timeSeriesV1List field.	Parameter string is pasted correctly	OK
7.	Click on button Try it out!	Time series values are imported into the system.	OK
8.	Repeat step 6. to 8. with text from : - 6.2.3-Pilot - Battery DischargingMeterReading.txt - 6.2.3-Pilot – Battery EnergyLevel.txt		OK

9.	When the values are imported these calculations/Workers will automatically be run: **Delta Calculation** – Converting Meter Reading to energy **Estimation Calculation** - If Meter Reading is missing the calculation will interpolate	Energy and prediction Time Series are created: - ChargingEnergy time series - DischargingEnergy time series - EnergyLevel time series	OK
10.	Open the Asset Screen	Asset Screen is opened	OK
11.	Click on Search and enter Pilot. Click on Refresh	All Assets matching search criteria Pilot are displayed in the grid.	OK
12.	Select the PV – Pilot and click on the Time Series Values ikon 	The Time Series Values screen is opened.	OK
13.	Choose the period, press the plus on Pilot – Battery, check off the ChargingEnergy, DischargingEnergy and EnergyLevel and press refresh button.	Values are shown in the graph and the grid.	OK
14.	Verify that the estimation (interpolation) is correct where the Meter Readings was missing. Formula: Interpolate between previous values and last values	Data is correct	OK

6.3 Receive Main Meter values

6.3.1 Purpose

The purpose of this process is to import Main Meter meter readings, calculate the delta values, estimate (interpolate) any missing values, calculate NonControllableConsumtion and predictions.

In this testcase time series will be imported through an Excel API template.

6.3.2 Assumptions and Pre-Conditions

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

6.3.3 Test Steps

	Action	Expected result	Result
1.	Open the text files and copy all the text: - 6.3.3-Pilot – Main Meter Consumption HH1.txt - 6.3.3-Pilot – Main Meter Consumption HH2.txt - 6.3.3-Pilot – Main Meter Production HH1.txt - 6.3.3-Pilot – Main Meter Production HH2.txt	Text file opens with pre-configured time series values.	OK
2.	Open Swagger: https://invadectplatformapi.azurewebsites.net/swagger/ui/index	Swagger is opened.	OK
3.	Enter login information		OK
4.	Click on TimeSeriesApi		OK
5.	Click on /api/External/TimeSeriesApi/SaveTimeSeriesListForEntitiesV1	SaveTimeSeriesList-ForEntitiesV1 is opened	OK

6.	Go to SaveTimeSeriesListForEntitiesV1 and paste the parameter string into timeSeriesV1List field.	Parameter string is pasted correctly	OK
7.	Click on button Try it out!	Time series values will be imported into the system.	OK
8.	Repeat 6. to 8. with the text from: - 6.3.3-Pilot – Main Meter Consumption HH2.txt - 6.3.3-Pilot – Main Meter Production HH1.txt - 6.3.3-Pilot – Main Meter Production HH2.txt		OK
9.	When the values are imported these calculations/Workers will automatically be run: **Delta Calculation** – Converting Meter Reading to energy **Estimation Calculation** - If Meter Reading is missing, the calculation will interpolate **Forecast Calculation:** Production time series are pushed to Prediction service. Prediction service push back the forecasted time series for production.	Energy, Noncontrollable and predictions time series are created: - Production time series - Consumption time series - NonControllableConsumption time series - ForecasedNoneControllableConsumption time series	OK
10.	Open the Asset Screen	Asset Screen is opened.	OK
11.	Click on Search and enter Pilot. Click on Refresh.	All Assets matching search criteria Pilot are displayed in the grid.	OK

12.	Select the Pilot – SiteHousehold1 and click on the Time Series Values ikon 	The Time Series Values screen is opened.	OK
13.	Choose the period, press the plus on Pilot – Main Meter – HH1, check off the Production meter reading and Production and press refresh button.	Values are shown in the graph and the grid.	OK
14.	Verify that the production data is calculated correctly. Formula: MeterReading t+1 – MeterReading t = Energy per 15 min	Data is correct.	OK
15.	Go to the Time Series Values - Search filter, uncheck the time series and then check off the time series Consumption Meter Reading and Consumption	Values will be displayed in the graph and the grid.	OK
16.	Verify that the production data is calculated correctly. Formula: MeterReading t+1 – MeterReading t = Energy per 15 min	Data is correct.	OK
17.	Verify that the estimation (interpolation) is correct where the Meter Reading was missing. Formula: Interpolate between previous value and last value	Data is correct	OK
18.	Press plus on Pilot – SiteHousehold1 and check off the time series NonControllableConsumption and ForecastedNoneControllableConsumption. Verify that the prediction is logical.	Values will be displayed in the graph and the grid. The prediction is logical.	OK

19.	Repeat step 1-18 for site Pilot – Household2		OK

6.4 Receive EV values

6.4.1 Purpose

The purpose of this process is to import EV meter readings, Charging Status, calculate the delta values and predictions.

In this testcase time series will be imported through excel API template.

6.4.2 Assumptions and Pre-Conditions

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

6.4.3 Test Steps

	Action	Expected result	Result
1.	Open the text files and copy all the text: - 6.4.3-Pilot - EV1 ChargingMeterreading.txt - 6.4.3-Pilot - EV1 ChargePointState.txt	Text files with pre-configured time series values	OK
2.	Open Swagger: https://invadectplatformapi.azurewebsites.net/swagger/ui/index	Swagger is opened	OK
3.	Enter login information		OK
4.	Click on TimeSeriesApi		OK
5.	Click on /api/External/TimeSeriesApi/ SaveTimeSeriesListForEntitiesV1	SaveTimeSeriesList-ForEntitiesV1 is opened	OK
6.	Go to SaveTimeSeriesListForEntitiesV1 and	Parameter string is pasted correctly	OK

	paste the parameter string into timeSeriesV1List field.		
7.	Click on button Try it out!	Time series values are imported into the system	OK
8.	Repeat 6. to 8. with the text from: - 6.4.3-Pilot - EV1 ChargePointState.txt		OK
9.	When the values are imported these calculations/Workers will automatically be run: **Delta Calculation** – Converting Meter Reading to energy **Estimation Calculation** - If Meter Reading is missing the calculation will interpolate **Forecast Calculation:** Production time series is pushed to Prediction service. Prediction service push back the forecasted time series for production	Energy and prediction Time Series will be created: - ChargingEnergy - ChargingPointState - ForecastedChargingEnergy - ForecastedChargingPointState	OK
10.	Open the Asset Screen	Asset Screen is opened	OK
11.	Click on Search and enter Pilot. Click on Refresh	All Assets matching search criteria Pilot are displayed in the grid.	OK
12.	Select the Pilot – EV and click on the Time Series Values ikon 	The Time Series Values screen is opened.	OK
13.	Choose the period, press the plus on Pilot – EV, check off the ChargingMeterReading and	Values are shown in the graph and the grid.	OK

	ChargingEnergy and press refresh button.		
14.	Verify that the ChargingEnergy data is calculated correctly. Formula: MeterReading t+1 – MeterReading t = Energy per 15 min	Data is correct.	OK
15.	Go to the Time Series Values - Search filter and check off the time series ChargingEnergy and ForecastedChargingEnergy. Verify that the predictions are logical.	The predictions are logical.	OK
16.	Go to the Time Series Values - Search filter and check off the time series ChargingPointState and ForecastedChargingPointState. Verify that the predictions are logical.	The predictions are logical.	OK

7 Flexibility Management Optimization

This section describes the test cases related to the flexibility management optimization process.

7.1 Calculate optimal flexibility schedule

7.1.1 Purpose

The purpose of this process is to generate an optimal flexibility schedule and an optimal capacity regulation.

The testcase will be repeated for these 15-minute intervals:

- 15:00
- 15:15
- 15:30
- 15:45

7.1.2 Assumptions and Pre-Conditions

Historical Resource energy values and historical Events and external information exists.

Predicted energy values and charge point status exists.

Flexibility contract, contract product and contract asset parameters exist.

7.1.3 Test Steps

	Action	Expected result	Result
1.	Open the text file and copy all the text: - 7.1.3-Pilot - All assets 1445.txt The file has data for all assets for meter reading at 14:45.	Text files with pre-configured time series values	OK
2.	Open Swagger: https://invadectplatformapi.azurewebsites.net/swagger/ui/index	Swagger is opened	OK

3.	Enter login information		OK
4.	Click on TimeSeriesApi		OK
5.	Click on /api/External/TimeSeriesApi/ SaveTimeSeriesListForEntitiesV1	SaveTimeSeriesList-ForEntitiesV1 is opened	OK
6.	Go to SaveTimeSeriesListForEntitiesV1 and paste the parameter string into timeSeriesV1List field.	Parameter string is pasted correctly	OK
7.	Click on button Try it out!	Time series values are imported into the system	OK
8.	Energy values, predictions and aggregations will be calculated and stored as data for the 15-minute interval starting at 14:30 (and ending at 14:45)		OK
9.	Optimization will trigger at 14:50	These time series will be saved as result from optimization: - ProductionPower Regulation - ProductionOnOff PowerRegulation - ChargingPower Regulation (CP) - ChargingPower Regulation (Battery) - Discharging PowerRegulation (Battery)	OK with comment: Optimization of EV did not go through, consequently neither for SiteHousehold2. ChargingPower Regulation (CP) and OptimalCapacity regulation were therefore not saved.

		- OptimalCapacity Regulation	
10.	Open the Asset Screen	Asset Screen is opened	OK
11.	Click on Search and enter Pilot. Click on Refresh	All Assets matching search criteria Pilot are displayed in the grid.	OK
12.	Select the Pilot – SiteHousehold1 and click on the Time Series Values ikon 	The Time Series Values screen is opened.	OK with comment: Different types of data should have the same colour. E.g. consumption should be blue, production should be green.
13.	Investigate and discuss Optimization result time series	Values are shown in the graph and the grid.	OK
14.	Repeat 12. and 13. for Pilot – SiteHousehold2		Optimization for SiteHousehold2 did not go through. Results could therefore not be investigated and discussed.

8 Appendix 1

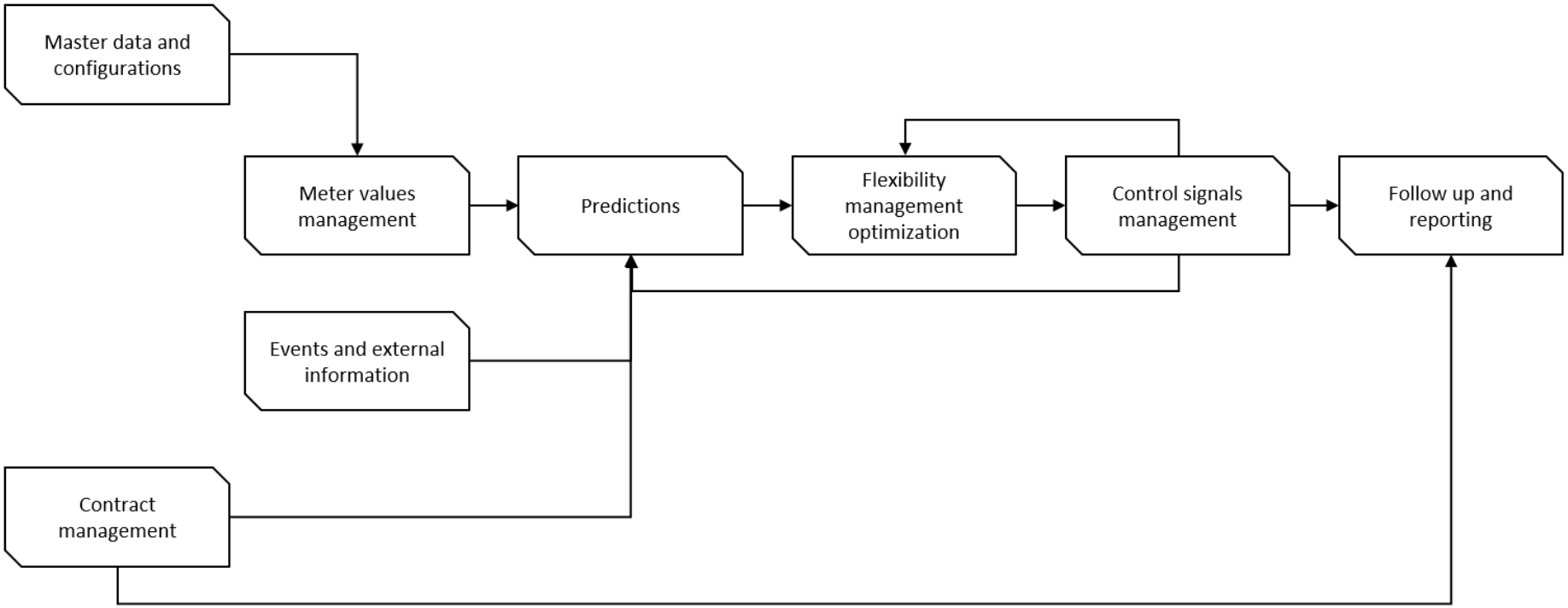

Figure 2: Overview of the Integrated INVADE platform main processes