TrustwOrthy model-awaRE Analytics Data platfORm

RIA - Grant No. 688797

Toreador Platform Implementation for the Energy Production Data Analysis Pilot

Deliverable 10.3
Rev. 1.0

.

Project Title	TOREADOR – TrustwOrthy model-awaRE Analytics Data platfORm
Grant Number	688797
Project Type	RIA

Title of Deliverable	Toreador platform implementation for the Energy Production Data Analysis pilot
Subtitle of Deliverable	
Deliverable Number	D10.3
Dissemination Level	Public
Internal Rev. No.	1.0
Contractual Delivery Date	31 December 2017
Actual Delivery Date	31 December 2017
Contributing partners	LIGHT, CINI, CITY
Editor(s)	LIGHT
Author(s)	Christos Polydoros, Michal Poszumski, Menelaos Ioannidis (LIGHT)
Reviewer(s)	TAIGER, JOT

Executive Summary

This document presents the current architecture of LIGHT and TOREADOR's platform, a description of specific use cases, a detailed analysis of the declarative, procedural and deployment models and how the TOREADOR factor helped for this deliverable. Finally, some conclusions and perspectives for the next period (Y3) of the TOREADOR project have been provided.

Contents

Chapter 1

INTRODUCTION

1.1 Use case description and architecture

This Chapter compares the current architecture of LIGHT and TOREADOR's platform, and provides a description of specific use cases, with a detailed analysis of the declarative, procedural and deployment models

1.1.1 Architecture

In Figure 1 below, the LIGHT's architecture is displayed together with its interface with the TOREADOR one.

In more detail, in Figure 1 Kw/h meters and inverters are sensor devices generating data about power emission and consumption. Currently this data is being collected by a Data logger, via the MODBUS protocol. MODBUS is a serial communications protocol originally published in 1979 by the Modicon company (now Schneider Electric) for use with its programmable logic controllers (PLCs). Modbus has become a de facto standard communication protocol for connecting industrial electronic devices. Data loggers transmit data to the LIGHT cloud infrastructure over HTTPS.

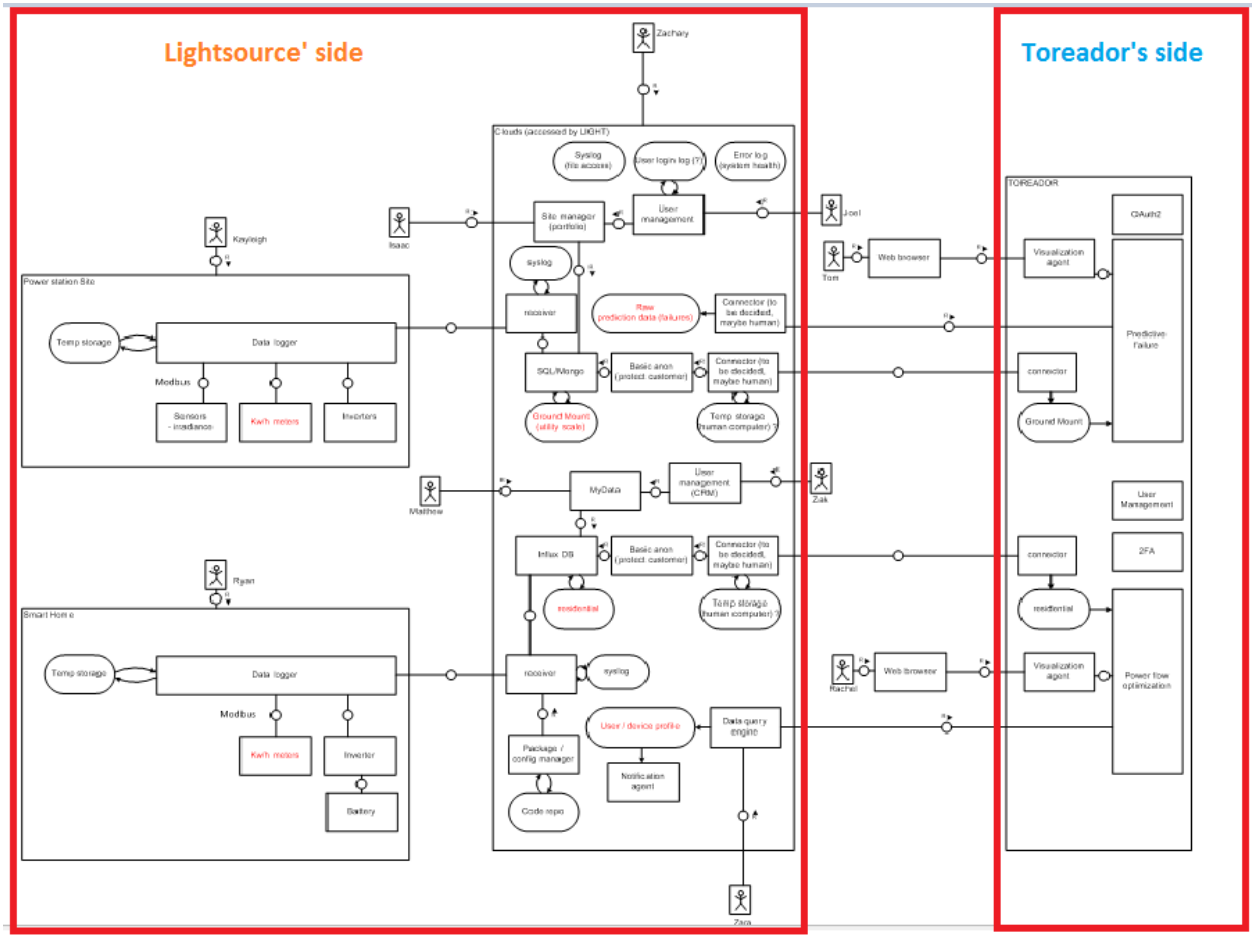


Figure 1: LIGHT – Toreador Architecture

Again, with reference to Figure 1, Power Station sites are physical sites where solar panels are being deployed. These panels are monitored by sensors connected to a data logger hosted in a dedicated building where only authorized personnel can enter (e.g. Kayleigh).

Smart Homes are homes belonging to private individuals, into which physical boxes owned by LIGHT are being installed. These boxes contain the data logger and are plugged to the house's power sensors. The boxes shall not be physically opened by unauthorized personnel.

The main servers of LIGHT are hosted on a Cloud infrastructure. There, receivers are collecting data from Power stations and from Smart homes,

and storing it into separate databases. Dedicated applications run on top of this data, respectively for site managers (such as Isaac) to monitor its power plants and for individuals (such as Matthew) to review data about his smart house. These applications are being administered by role managers, ensuring that each external user has access to his data and to his data only.

Diverse logs are being kept, such as user access log by the user management agent and syslog and error logs by the database platforms.

Both main data storages (Ground Mount and Residential) follow the same (separate) architecture model. They are being polled by a Basic Anon agent, performing removal of data un-necessary for the analysis (a filtering service to be refined in later WP10 deliverables) and pseudonymizing of identifiers (such as device vendors for Ground Mount and end-users for Residential) for having the data as minimal as possible for analysis by the TOREADOR platform. Analysis will support decision on which device to replace / buy (in use-case #1) and on which user to notify (in use-case #2).

The data will be sent to the TOREADOR platform, possibly hosted on a separate cloud platform. It shall be decided how the data should be uploaded to the TOREADOR platform from the LIGHT cloud platform, either automatically or via a manual process. Both approaches have advantages and disadvantages, and the risk analysis provided in the later section expands on these considerations.

Once ingested by the TOREADOR platform, the data will be processed by specific analysis algorithms and made available via a dedicated service for visualization through a web browser, as well as for download (as text data) for further analysis locally on the LIGHT cloud platform. Once downloaded, the data is made available to authorized persons; the authorizations will be managed by the LIGHT user management tools (and are beyond the scope of this deliverable).

The TOREADOR platform has a User Management module, supports 2-factor authentication via its 2FA module and uses the OAuth2 protocol for session and authorization management.

1.1.2. Energy pilot use cases

As mentioned in the previous Section and discussed in past deliverables, two main use-cases will be supported by the WP10 pilot through the TOREADOR platform, for both the large scale and residential scale assets.

The energy pilot work package WP10 is aimed at maximizing the efficiency of solar energy. It consists in a set of components retrieving data from solar plants as well as from smart homes running their own local solar appliance in an effort to respectively predict hardware failure (avoiding power cuts) and optimize energy consumption (avoiding waste and minimizing costs).

The data comes from locally deployed sensors and is being sent to a cloud platform. It is there used directly for diverse metrics and reporting. It is also prepared before being sent to another cloud platform, where it is processed, generating respectively reports predicting failures and power consumption analysis graphs for optimizing energy creation and consumption.

1.1.2.1 Large scale asset case

This use case deals with a predictive maintenance scenario in a large-scale asset example, which is described by the steps below:

1- Sensors placed at power station sites send data about power to their data logger
2- The data logger sends the data to a receiver, which stores the data into the Ground Mount data storage
3- The Basic Anon agent retrieves the data and applies some preparation including removal or pseudonymizing of metadata related to customer names and details
4- The prepared data is being sent to a connector on the Toreador platform (either automatically or through a manual process, to be decided upon implementation)
5- The Ground Mount data is locally stored, then processed for predicting failures
6- Tom logs into the Visualization agent and can drill into the prediction details
7- Upon a LIGHT's employee's action or via a batch job, the prediction's raw data is being retrieved to the Raw Prediction Data storage for further action which is beyond the scope of the pilot (including making buying decisions and pro-actively changing components at the power station)

Please see below some screenshots from this use case (LIGHT monitoring system – Large scale assets)

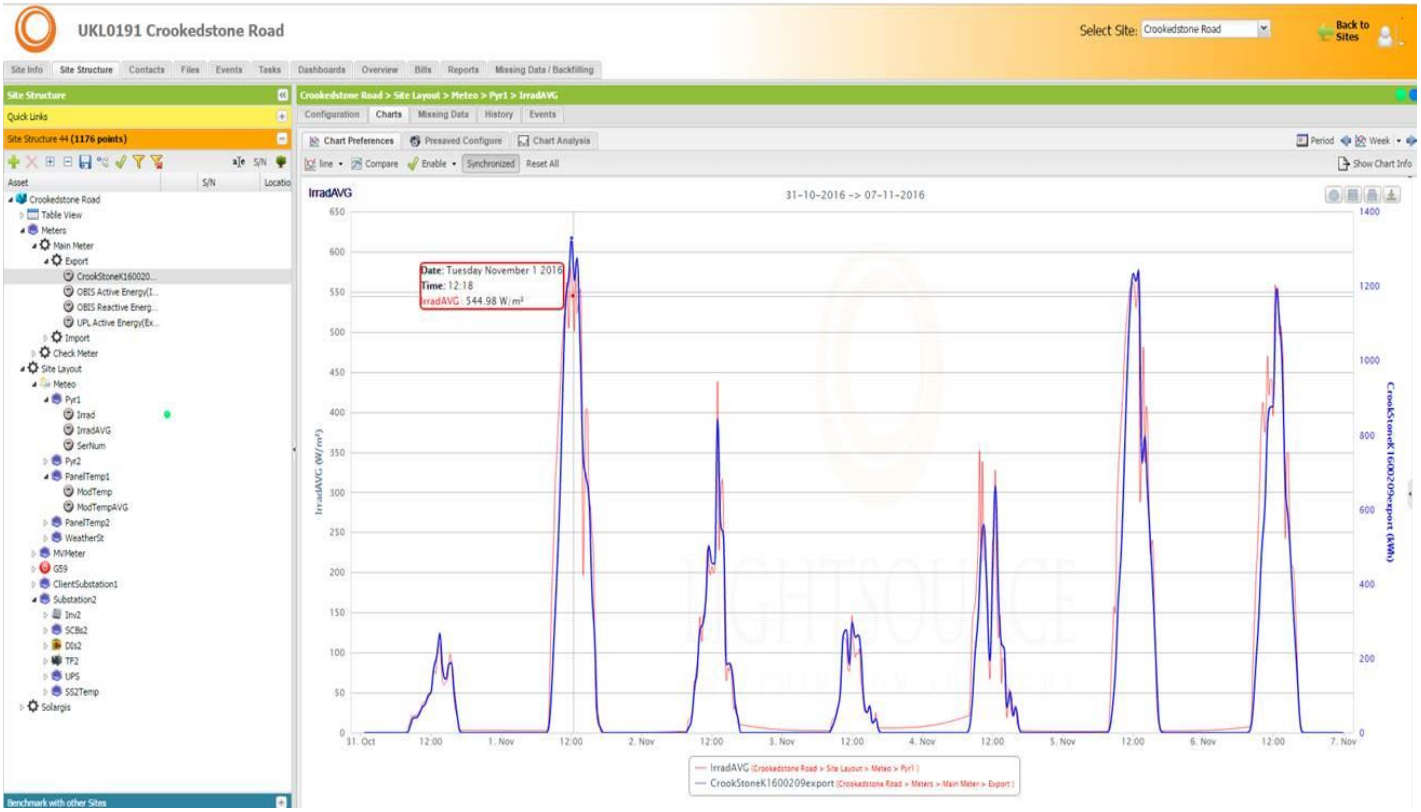


Figure 2: Red line presents the irradiance date and the blue line presents the export data to the grid

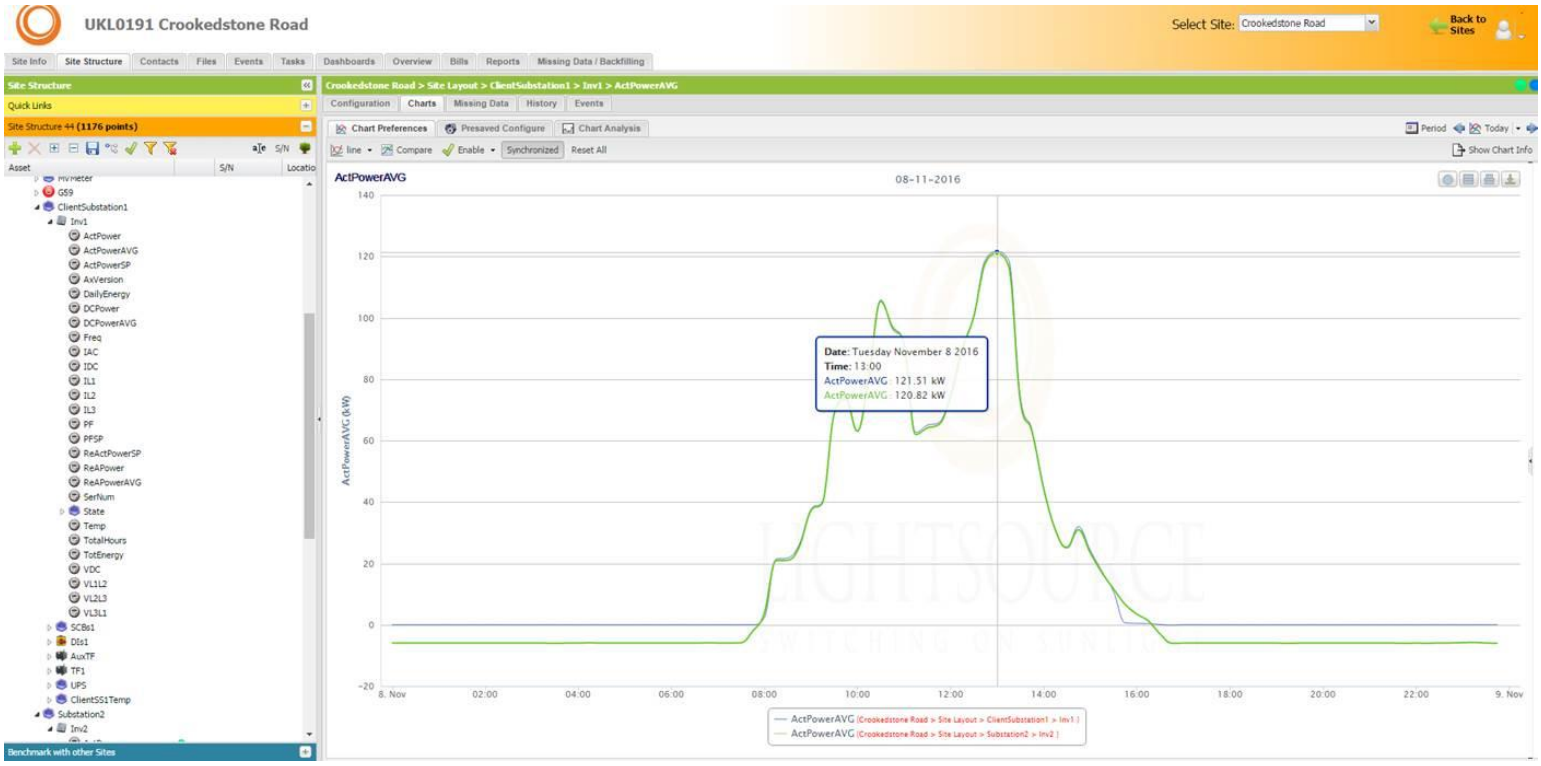


Figure 3: The green line displays the quarterly power production of two inverters within a day.

1.1.2.2 Residential scale asset case

The power optimization in a smart home is described into the below steps:

1- Sensors located in smart houses send data about power to their data logger
2- The data logger sends the data to a receiver, which stores the data into the Residential data storage
3- The Basic Anon agent retrieves the data and applies some preparation including removal or pseudonymizing of metadata related to customer names and details
4- The prepared data is being sent to a connector on the Toreador platform (either automatically or through a manual process, to be decided upon implementation)
5- The Residential data is locally stored, then processed for power optimization
6- A LIGHT's employee logs into the Visualization agent and can drill into the analyzed data, such as inference about the types and usage of electric devices at the customers' smart home
7- The data is further being pulled into the User/device profile storage area, where from LIGHT's side can perform several queries, including the possibility to re-identify which house is being concerned by certain data elements, offering the possibility to send notifications to the customer with information about abnormal power usage or suggestions for better

A detailed overview of LIGHT's Home Energy Management System, which will be the main source of residential data, is presented in the next pages.

LIGHT has already developed a residential scale scheme called GoSunplug[1], which provides energy services to residential customers. This system will be the main source of data towards the TOREADOR platform, regarding the power optimization in a smart home, as described into the previous section.

[1] [4] http://gosunplug.com/

Below we report some screenshots from the HEMS:

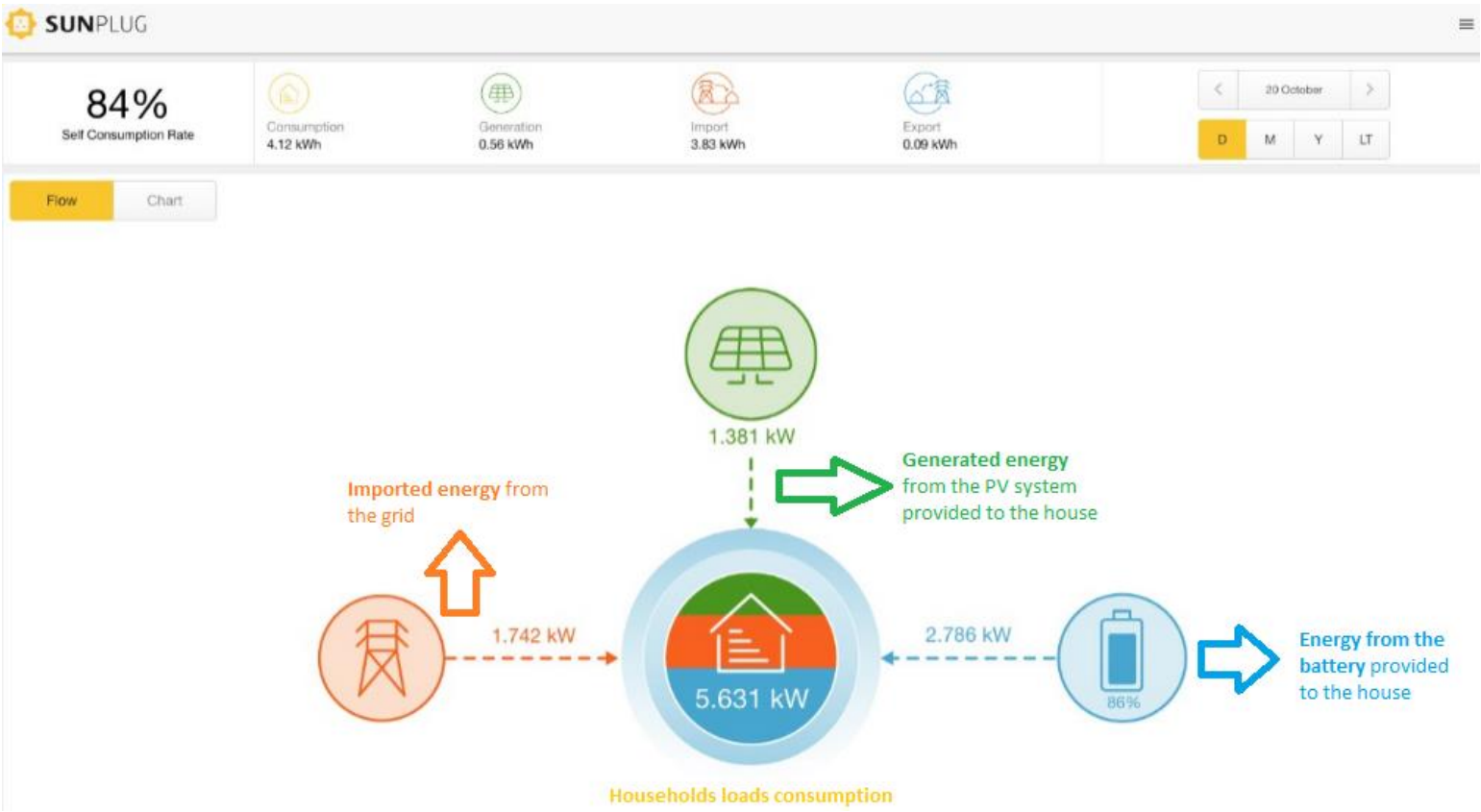


Figure 4: Overview of the HEMS platform with three main categories of data sources (*Generated energy, Imported energy, Energy from the battery*) – Real time graph

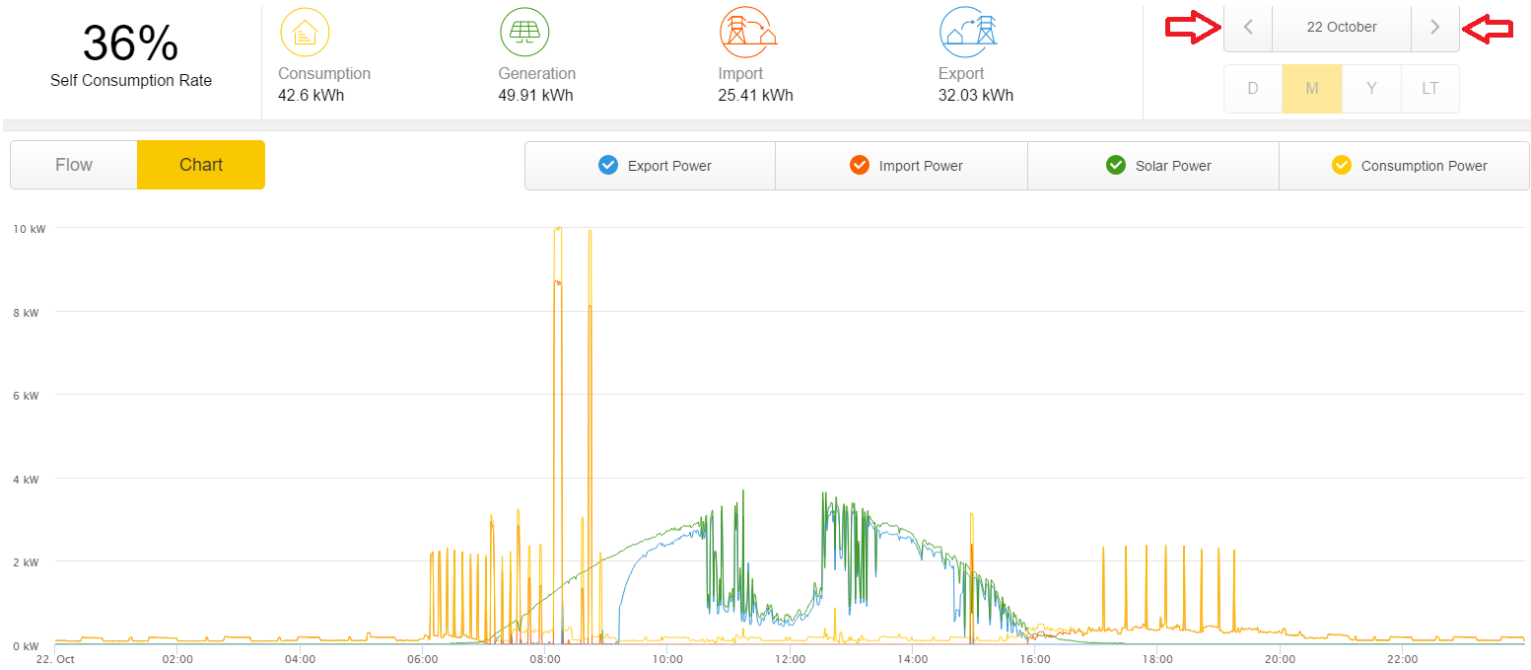


Figure 5: Electricity chart in hourly resolution, in daily basis

The power chart displays a more detailed reading of the electricity that is consumed, generated, imported and exported on a daily basis with hourly resolution. The chart only shows readings of the SunPlug system on a daily basis, but also stores historical information on the system functions, which can be accessed by scrolling through different days. All this data can be used and implemented into the TOREADOR platform for producing different outcomes (presented in detail into the next section), which will be beneficial to LIGHT.

1.2 Recall on pilot's goals

The Big Data integration that can be achieved via the TOREADOR project, based on different circumstances that influence a PV asset (weather forecasting, irradiance forecasting, temperature data etc. from our monitoring system) or the electrical network (voltage and frequency instability etc.) can give important outcomes that LIGHT could take advantage of.

More specifically the overall targets, that LIGHT expects and that could be achieved by the TOREADOR results are listed below:

- Real time monitoring analysis by gathering a big amount of data from several assets.
- Onsite & Cloud analytics.
- Real time aggregation in a Virtual Power Plant (VPP mode).
- Real time Decision Support System that could be achieved by the bi-directional circulation of information via the PPCs and the HEMS.
- Real time controls of key equipment into both the large scale and residential assets.
- Data logging of balancing events performance.
- Providing services to the grid when it is necessary (frequency response/voltage control).
- Predicting failures of key components of the assets.
- Increasing the assets' lifetime by using Mean Time Before Failures models (MTBF) with the predictive analytics and the Toreador platform.

Components from the TOREADOR platform, which can help make decisions on the functionality of different key equipment components will be to a considerable extent the visual ones we have just examined. Aside from these, TOREADOR offers another type of useful resource in the form of reports –the format of which can vary between CSV, Excel and so forth. These reports will contain the results obtained after the analysis phase and after applying machine learning algorithms and data mining on the TOREADOR platform, in order to deliver the above outcomes.

1.3 Coverage

In this section we provide a recap of the features, that LIGHT would like to see from the TOREADOR platform. The recap develops on what has been discussed in the previous deliverables reports.

Combining a smart network with the usage of the TOREADOR platform services, can give a huge boost to LIGHT with significant benefits in different categories of our activities.

Firstly, predictive analytics models, which are based on smart learning machine algorithms, can increase the efficiency of the assets (a solar park or a smart home), by maintaining and operating these assets according to real time data, historical data and data from similar assets and past maintenance records.

In the future, LIGHT wants to achieve 1 sec granularity in its database, as mentioned before. Predictive analytics is defined as predicting at a more detailed level of granularity. LIGHT would like to follow the model (graphically represented below) with the combination of the Toreador platform and the predictive analytics, in order to grow its business value:

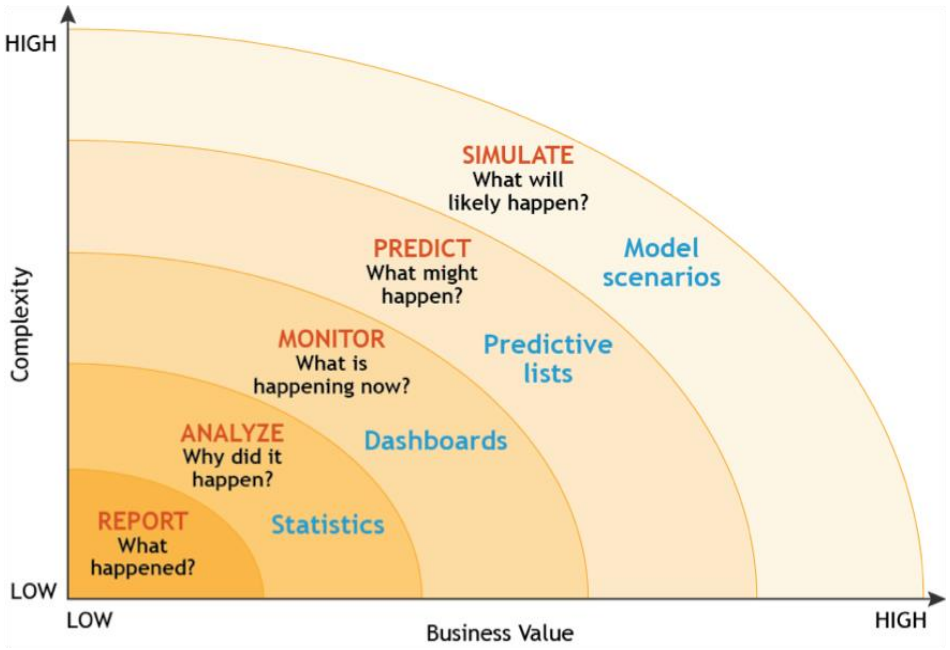


Figure 6: Predictive analytics contribution

More specifically, the first three categories-labels (REPORT, ANALYZE, MONITOR) have already been developed by LIGHT, via its monitoring system although the Toreador platform can be used to connect all different assets, manage the amount and variety of collected data and enable a predictive approach, so that the analysis on these data can be used to optimize in real-time the performance of the energy generation processes. The predictive analytics (PREDICT, SIMULATE labels), connected to the Toreador platform, will play a fundamental role in complex challenges (Y axis) by providing higher business value for LIGHT through simulation of several scenarios for future cases (X axis). Below, Figure 6 shows the addition of the Toreador platform and the predictive analytics:

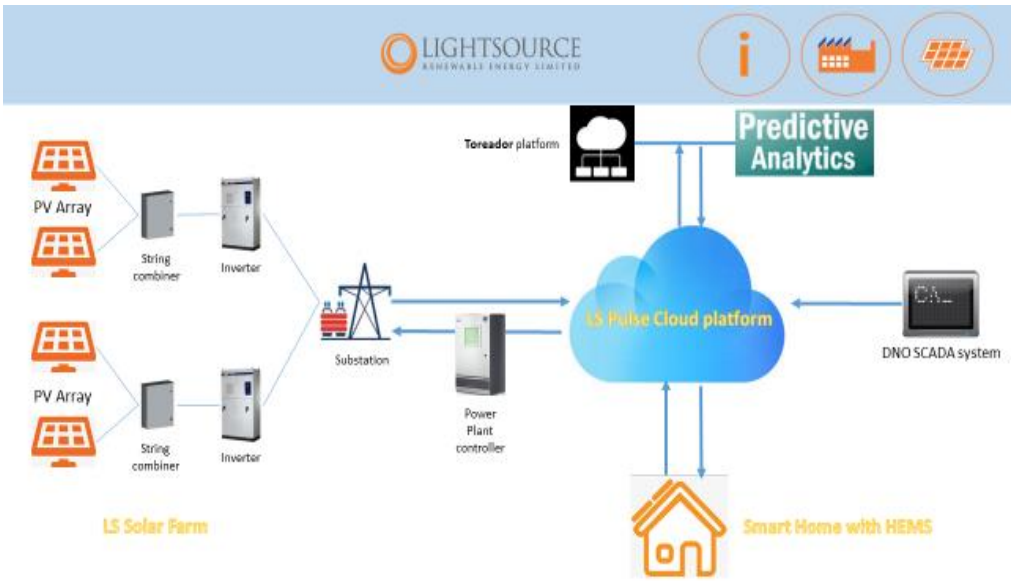


Figure 7: "Smart network" with the Toreador platform and predictive analytics

For the large assets/solar farms, based on the above schematic, LIGHT could reap a lot of benefits. Instead of using, the standard maintenance services which are quite expensive and time-consuming, the predictive models can store all the variables and conditions that provoke past failures to predict future failures (inverters, transformers, batteries etc. failures).

Predictive analytics models can provide to LIGHT various advantages such as:

- Cheaper component repair and replacement services
- Reduced revenue loss
- Focussed and more efficient O&M activity
- Better equipment management
- Better operational practices

For the residential assets, LIGHT will use the Toreador platform to deliver towards the grid several services such as frequency response and voltage stability during different circumstances that would be alerted from the DNO SCADA system. For instance, in frequency emergency situations (peak demand times), the community could export surplus energy from the storage systems towards the grid. Please see below the relevant schematic:

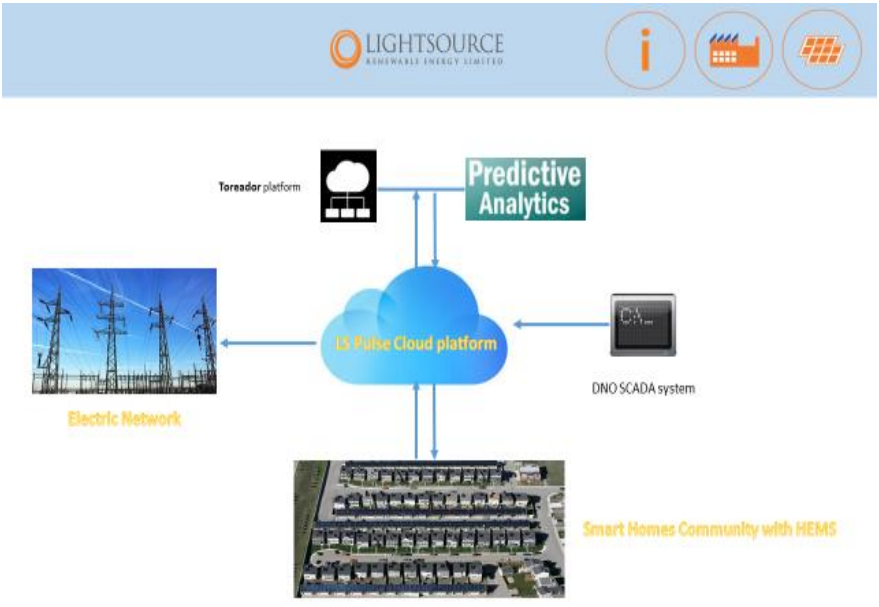


Figure 8: "Smart Homes" Community with Toreador and Predictive analytics

The combination of Figure 7 and Figure 8 produces the below Figure 9 which represents the energy pilot into the TOREADOR platform:

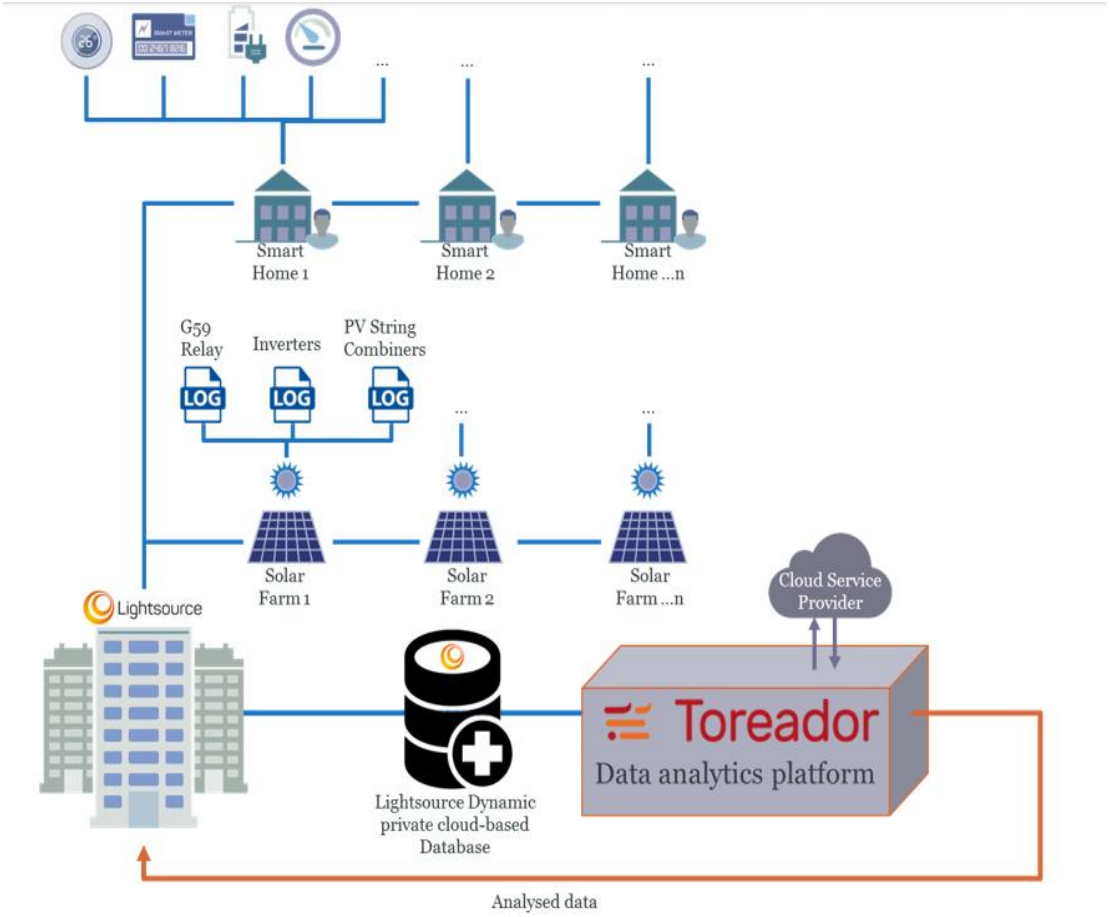


Figure 9: Overview of the energy pilot

All the above details show clearly that LIGHT's energy scenario could have different benefits, as the Toreador platform could develop the best predictive services for different scenarios. The Big Data integration that can be achieved via the Toreador project, based on different circumstances that influence a PV asset (weather forecasting, irradiance forecasting, temperature data etc.) or the electrical network (voltage and frequency instability etc.) can give important outcomes that LIGHT's could take advantage of them. The requirements from LIGHT's side are presented in more detail into the next chapter.

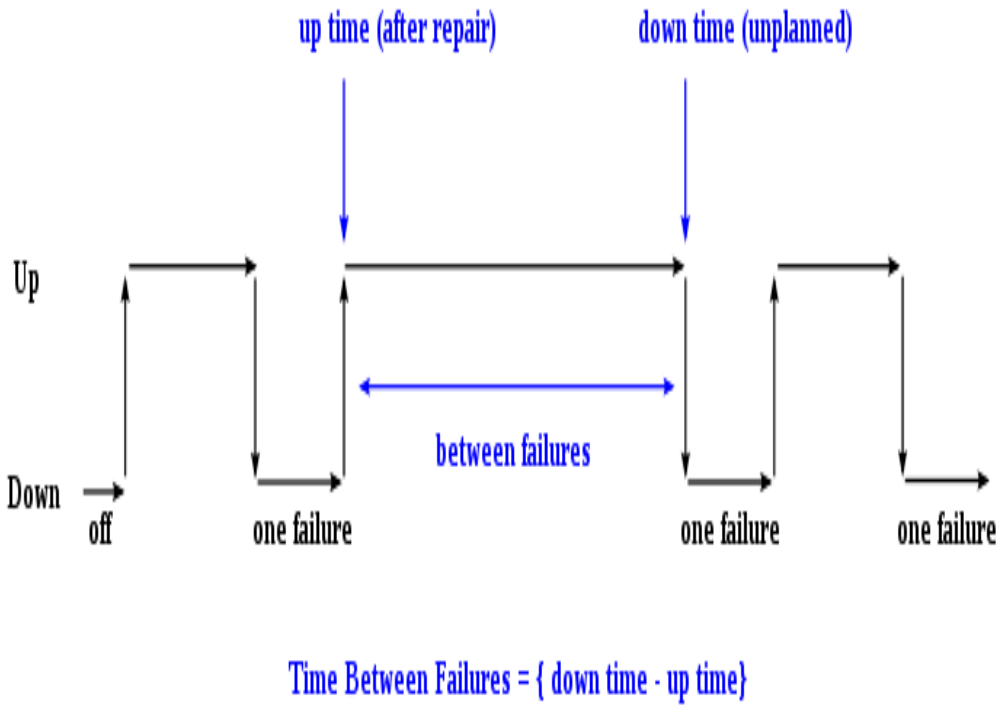
Moreover, the topics that LIGHT will cover are displayed below together with a description for each one of them.

- Topic 01: TOREADOR platform shall support predictive analysis. Prediction will be performed based on the prediction models that will be used on the platform in order to support knowledge extraction and prediction of our equipment's functionality.

- Topic 02: TOREADOR analytics must support real time data integration via APIs support in order to allow real time combination of data extracted from different sources (both large scale and residential scale assets).

- Topic 03: Prediction for equipment maintenance based on historical data and for equipment anomalies in work cycles of the devices (inverters, transformers, smart meters' failures). More specifically, data from the large scale solar of 3 years and from our residential assets will be provided by LIGHT, for preventing anomalies regarding spikes in voltage, giving frequency response of the grid quality, and receiving temperature of inverters and ampere information. We expect the platform to support predict maintenance of tools and machines involved in the monitoring process.

- Topic 04: Increasing the assets' lifetime by using Mean time between failures methodology, in order to protect our equipment for future failures. In more detail, LIGHT considers that the usage of the MTBF models could be very significant as they refer to the average amount of time that a device or product functions before failing. This unit of measurement includes only operational time between failures and does not include repair times, assuming the item is repaired and begins functioning again. MTBF figures are often used to project how likely a single unit is to fail within a certain period of time. The definition of MTBF relies on the category of a system failure and specifically for the solar assets there are several failures such as inverters failures, transformers failures, smart meter failures etc. which are quite expensive and time-consuming to be repaired and replaced if it is necessary (long / very long lead items for a solar asset). A brief summary of the MTBF models is presented via the below schematic:



A big amount of revenue is lost for LIGHT during the "between failures" and these scenarios could be avoided by using the results from the Toreador platform that could combine both Predictive analytics and MTBF models. This development can also be very significant for increasing the lifetime and the value of LIGHT's large-scale and residential-scale assets.

- Topic 05: Privacy support and anonymization mechanisms to be supported with the name and location (general info as well) of the assets to be protected and anonymized. The protection of sensitive data that will be provided to the TOREADOR platform, will be extremely important and one of the main requirements from LIGHT.

- Topic 06: TOREADOR shall support behavioural analysis algorithms based on behavioural algorithms that could be used from the platform.

- Topic 07: Inspection process monitoring of data which means that the data correlation and analysis and the predictive maintenance to specific activities for the inspection process during and after a work cycle.

- Topic 08 : SLA specification, with 8 different requirements, such as: a) TOREADOR support of SLAs specification including service level objectives regarding functional, quality, security and privacy properties for BDA services, b) TOREADOR shall support actions that should be undertaken under different circumstances during the life of an SLA (e.g., when the SLA is violated). c) TOREADOR support the specification of SLAs covering two-party and multi-party agreements. d) TOREADOR shall provide reference catalogues of properties that could be used in defining SLAs. e) TOREADOR shall provide reference templates of properties that could be used in defining SLAs. f) TOREADOR shall provide a generic high-level language for BDA platform agnostic SLA specification that can be usable by non-expert service providers and service consumers. g) TOREADOR shall provide a reference language for the specification of operational SLAs at runtime and the transformation of SLAs expressed in its high-level platform agnostic language into it. h) TOREADOR shall support the transformation of SLAs expressed in its high-level language into SLAs that can be monitored and managed on specific targeted BDA platforms selected by the project and the reference operational SLA specification language

- Topic 09: SLA negotiation, with 8 different requirements, such as: a) TOREADOR shall support SLA negotiation over a full continuum of service levels for the guarantee terms defined in SLAs. b) TOREADOR shall support SLA negotiation based on a selection of specific pre-defined SLAs (e.g., bronze, silver, gold). c) TOREADOR shall support static SLA negotiation, i.e., negotiation of SLAs only prior to the deployment of the BDA service regulated by it. d) TOREADOR shall support dynamic SLA negotiation, i.e., negotiation of SLAs after the deployment and during the provision of the BDA service regulated by it. e) TOREADOR platform shall support automatic SLA negotiations. f) TOREADOR platform shall support semi-automatic SLA negotiations. g) TOREADOR platform shall allow parties to be able to define criteria for SLA negotiation. h) TOREADOR platform shall provide criteria for SLA negotiation that can be changed dynamically

- Topic 10: SLA assessment, with 12 different requirements: a) TOREADOR shall support the definition of SLA guarantee terms that can be monitored continually, b) TOREADOR should support the regular inspection of guarantee terms that can only be assessed in this manner, c) TOREADOR shall support the assessment of SLA terms that need to be assessed only by one off inspections, d) TOREADOR shall support predictive monitoring of SLA guarantee terms, i.e., estimates of the likelihood of having a violation of an SLA guarantee term at some point in the future, e) TOREADOR shall support the diagnosis of violations of SLA guarantee terms, i.e., the identifications of the reasons underpinning the detected violations, f) TOREADOR shall support SLA monitoring for the provider and the consumer of BDA services, g) TOREADOR shall support scalable SLA monitoring, h) TOREADOR shall support efficient SLA monitoring, i.e., monitoring which is guaranteed to produce and notify its results within a specific time boundary following the detection of a violation or the confirmation of the satisfaction of an SLA guarantee term, i) TOREADOR shall support the definition of SLAs that determine the acceptable level of intrusiveness of their monitors, j) TOREADOR shall support automatically configurable SLA monitoring when SLAs change dynamically and when BDA provision platforms change, k) TOREADOR shall support secure SLA monitoring, l) TOREADOR shall support trustworthy SLA monitoring,

Chapter 2

TOREADOR Approach – End to end

2.1 Introduction – Use cases analysis

As mentioned in the previous Sections of this deliverable, LIGHT has gathered a very big portfolio of both solar large and residential scale assets and every activity from the equipment is tracked from our monitoring system and data is registered into the relevant database. Performance issues influence the efficiency of our portfolio; therefore, LIGHT's income receives a significant impact from time to time.

For the end to end approach under Deliverable 10.3 purposes, LIGHT would like to focus on two different cases, related to the residential/smart home portfolio. We have already built a dynamic and a fast pace growing portfolio of smart homes around the UK, and as we have the ultimate target to expand this network with many more installations, we would like to face different challenges by finding smart solutions, using specific methodologies. More specifically, we would like to focus on the Mean Time Before Failures model (mentioned in the overall goals that LIGHT expect to see from the TOREADOR platform) which will help us to detect failures in our key equipment, based on historical data coming from our database. Before providing a more detailed explanation about the MTBF model and how this will be linked with our smart home portfolio, it's essential to analyse in a very high-level detail how our smart home monitoring system

works. Description, graphs, performance analysis results, failure alerts etc are displayed into the next pages.

Moreover, two specific scenarios will be analysed which can be used under the MTBF methodology that we would like to see from the TOREADOR platform, in order to keep our equipment's efficiency in very high-quality levels, increasing their lifetime and of course avoiding loss of income caused by this failure. As it is displayed into the previous sections of the document, for the smart home portfolio has developed a very interactive monitoring system which receives and transfers to our database 1sec resolution data, across the portfolio. Our database includes two main categories of data points, electrical **power** and **energy**, tracked and registered for every installation in daily, weekly, monthly, annually and lifetime basis. More specifically, for the power points we developed a live flow chart which displays in live time the condition of the generated power from the panels, the import/export power from and to the grid respectively, the consumption power from house's loads and finally battery's power circulation into the whole system. This dynamic power flow is displayed below with the relevant explanation per icon:

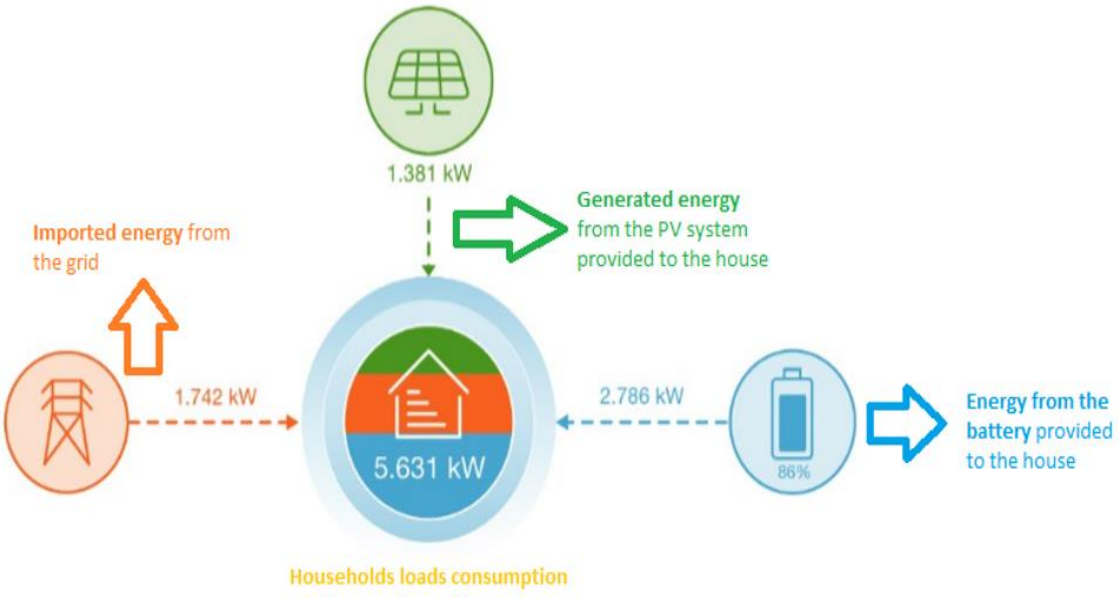


Figure 10: Dynamic power flow for the different key components from a smart home installation

LIGHT's main target by developing the smart home solution is to decrease the imported energy from the grid, which is much more expensive than the fixed priced solar generated energy from the panels, installed on every smart home's roof. So, both the solar generated power and the battery power data points play the most significant role in our income and if we monitor failures from the inverter equipment, which is directly linked with the solar panels and the battery, it means that our revenue is significantly influenced.

Therefore, two scenarios will be exploited and will be included into the MTBF model, which are the below:

- ***Scenario 1: Battery charging failure***

The energy storage equipment, a residential battery, plays an important role in LIGHT's smart home project. This automatically means that the battery's functionality is fundamental and a series of various activities must be completed, to keep the battery in the highest efficiency rates, for both our customers and LIGHT's benefits. In more detail, for the battery to start charging from the excess solar generated power, there are two main principles that must be followed:

The generated power from the solar panels is higher than the consumed power from the house loads (Pgen > Pcons)

The battery's state of charge (SOC) is below 100%, which means that the battery is not fully charged

Both principles mentioned above are monitored in our live system and are supervised from us via the inverter which gives the relevant commands for this power flow circulation to be completed successfully. In case that everything mentioned previously is followed accordingly, under these two main rules, and we don't see the battery charging (Pbat<0) then automatically an alert is flagged which means that we have a failure in

our system (for various reasons), so this is where the MTBF model can assist us to reduce this kind of situations. Please see below a simulation of the MTBF for this scenario:

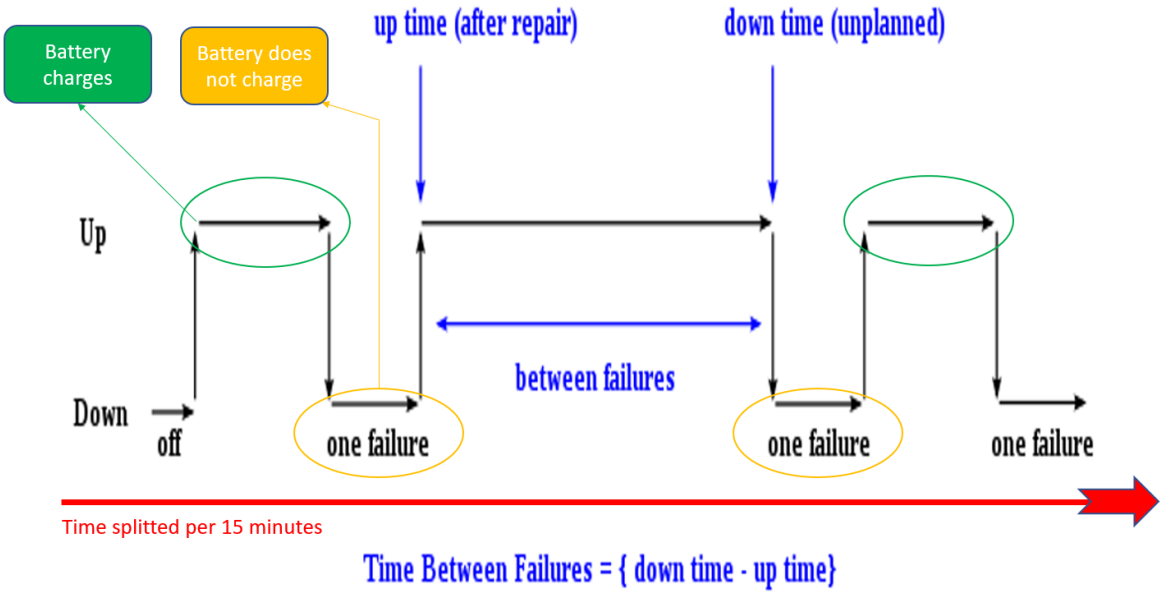


Figure 11: MTBF for the ***battery charging failure*** scenario

- ***Scenario 2: No generated power from the solar panels***

The key component of the smart home project is the solar panels on every smart home's roof. The ultimate target of the solar power, generated from the panels, is to cover most of the household needs and this power flow is controlled and monitored by the inverter, as in the scenario 1. LIGHT's income is totally based on the generated power from the panels towards the household, under a fixed price per kW, so any failures coming from this system influence our revenue. Our monitoring system has the ability to track the activity of the solar power figure, during daytime when it will available, so if we see this figure to be zero (Pgen=0) at 12.00 o'clock every day for one installation, this automatically means that there is fault that

has to be resolved as soon as possible. Please see below the relevant MTBF model for this scenario:

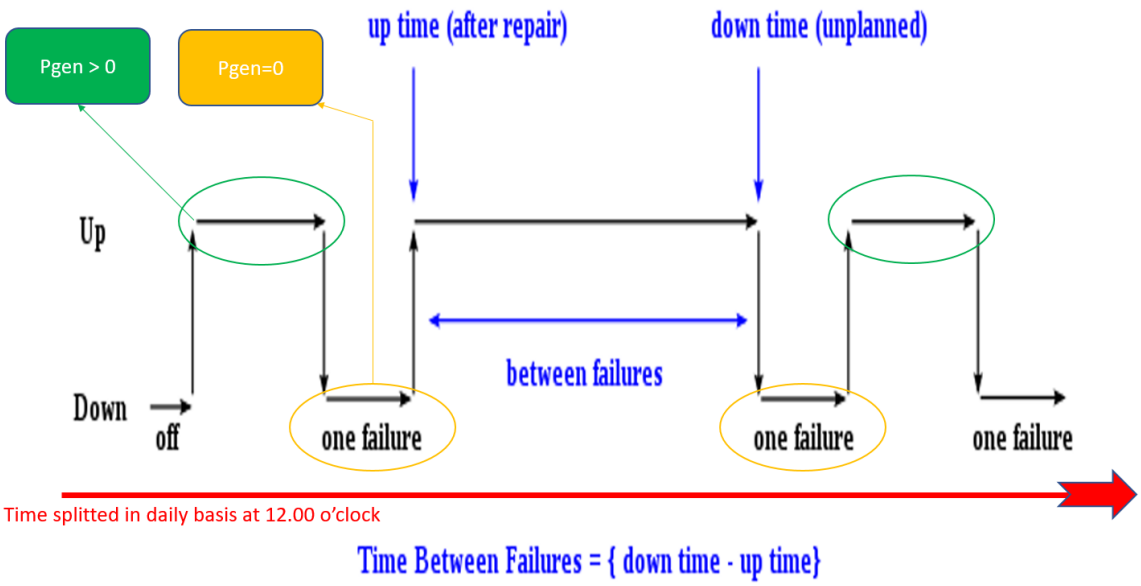


Figure 12: MTBF for the ***no generated power from solar panels*** scenario

Scenarios Results

By using the MTBF methodology for the two scenarios for several installations, we could produce a series of results using the MTFB formula, as it's displayed below:

$$\mathrm{MTBF} = \frac{\sum(\text{start of downtime} - \text{start of uptime})}{\text{number of failures}}.$$

Therefore, the results are presented on the below tables and graphs, which highlight the time between the failure incidents, for the two scenarios explained above:

Customers	Scenario 1 (days)	Scenario 2 (days)
Cust1	303	8
Cust2	84	193
Cust3	170	156
Cust4	216	357
Cust5	229	180
Cust6	258	24
Cust7	299	294
Cust8	116	297
Cust9	60	78
Cust10	294	208
Cust11	95	263
Cust12	297	327
Cust13	100	191
Cust14	80	206
Cust15	308	70
Cust16	294	151
Cust17	20	64
Cust18	343	14
Cust19	116	66
Cust20	89	133
Cust21	246	145
Cust22	281	295
Cust23	253	257
Cust24	326	314
Cust25	158	245
Cust26	18	231
Cust27	280	336
Cust28	241	8
Cust29	163	356
Cust30	264	5
Cust31	301	312
Cust32	288	143
Cust33	50	23
Cust34	82	214
Cust35	181	171
Cust36	310	320
Cust37	85	244
Cust38	351	9
Cust39	40	173
Cust40	197	187
Cust41	143	212
Cust42	87	88
Cust43	108	299
Cust44	82	114
Cust45	186	88
Cust46	144	341
Cust47	112	329
Cust48	105	183
Cust49	69	201
Cust50	199	2
Cust51	143	295
Cust52	252	363
Cust53	137	66
Cust54	154	149
Cust55	191	309
Cust56	167	298
Cust57	5	77
Cust58	208	78
Cust59	187	42
Cust60	229	178
Cust61	356	44
Cust62	86	260
Cust63	48	144
Cust64	178	73
Cust65	7	130
Average MTBF	**176.4461538**	**178.9384615**

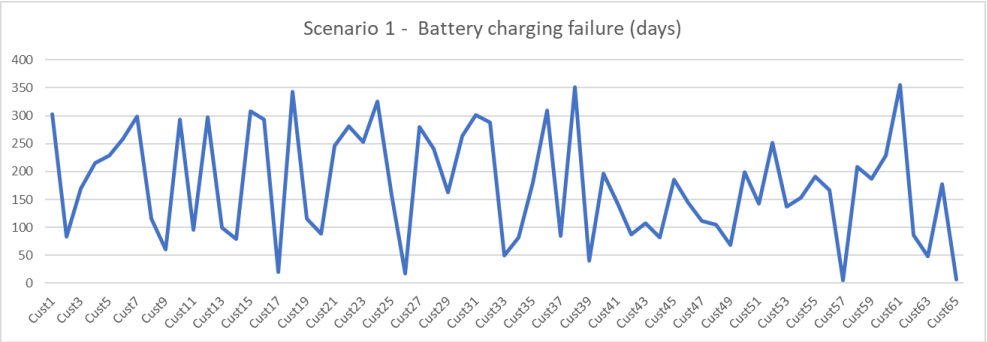


Figure 13: MTBF methodology results for Scenario 1 – Battery Charging failure

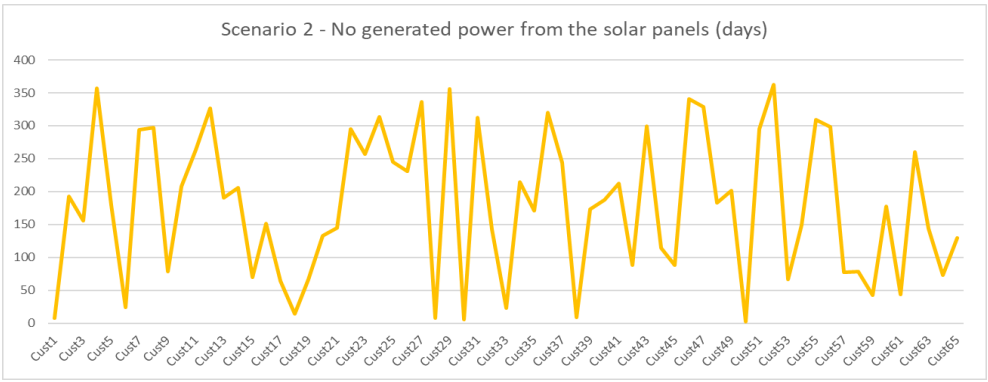


Figure 14: MTBF methodology results for Scenario 2 – No generated power from the solar panels

Based on the above, please see in the next pages the models (declarative, procedural, deployment) developed from LIGHT

2.2 Declarative Model

TOREADOR workflow for LIGHT pilot utilises MBDAaaS process (Figure 15), where objectives are stated via the Declarative Model Definition. Model is stored in the JSON format (Figure 16) reflecting key areas of the final outcome:

- Data Representation and Preparation – Data for LIGHT pilot is represented as an noSQL time series oriented database. It consists of a single table per *measurement* with relevant timestamp, value, and *GatewayID* accessible via API (application/json) calls. Described in detail in the relevant section of the JSON file (Figure 17). Required step for data preparation is anonymization of the *GatewayID* by hashing algorithm (SHA-256) declared in the section of the JSON (Figure 18).

- Data Analysis and Processing – Understanding impact of the equipment defects on the long-term modelling is one of the key objectives of the LIGHT pilot data analysis. *Generation* of the clean energy and correct functioning of the battery *charging* can be determined by applying Mean time between failures method, defined in the JSON file as "Anomaly/Fault detection" with the constrains of "Method":"Mean Time Between Failure", and TYPES of failures: "ChargerFailure" and "GeneratorFailure". Defined in JSON Figure 19.

- Data Display – Suitable data display for the LIGHT pilot is the 2D Graph and the table where MTBF figures can be presented for a given period of time (Figure 20)

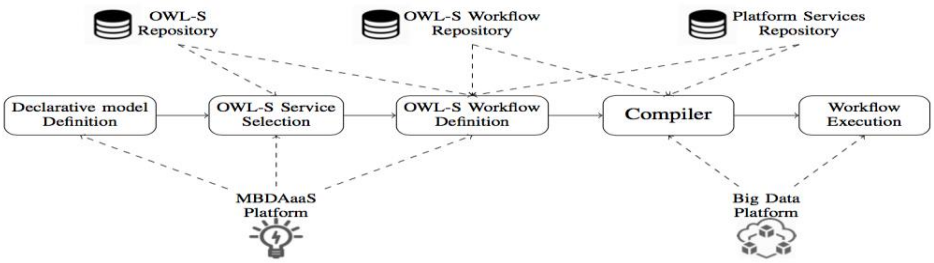
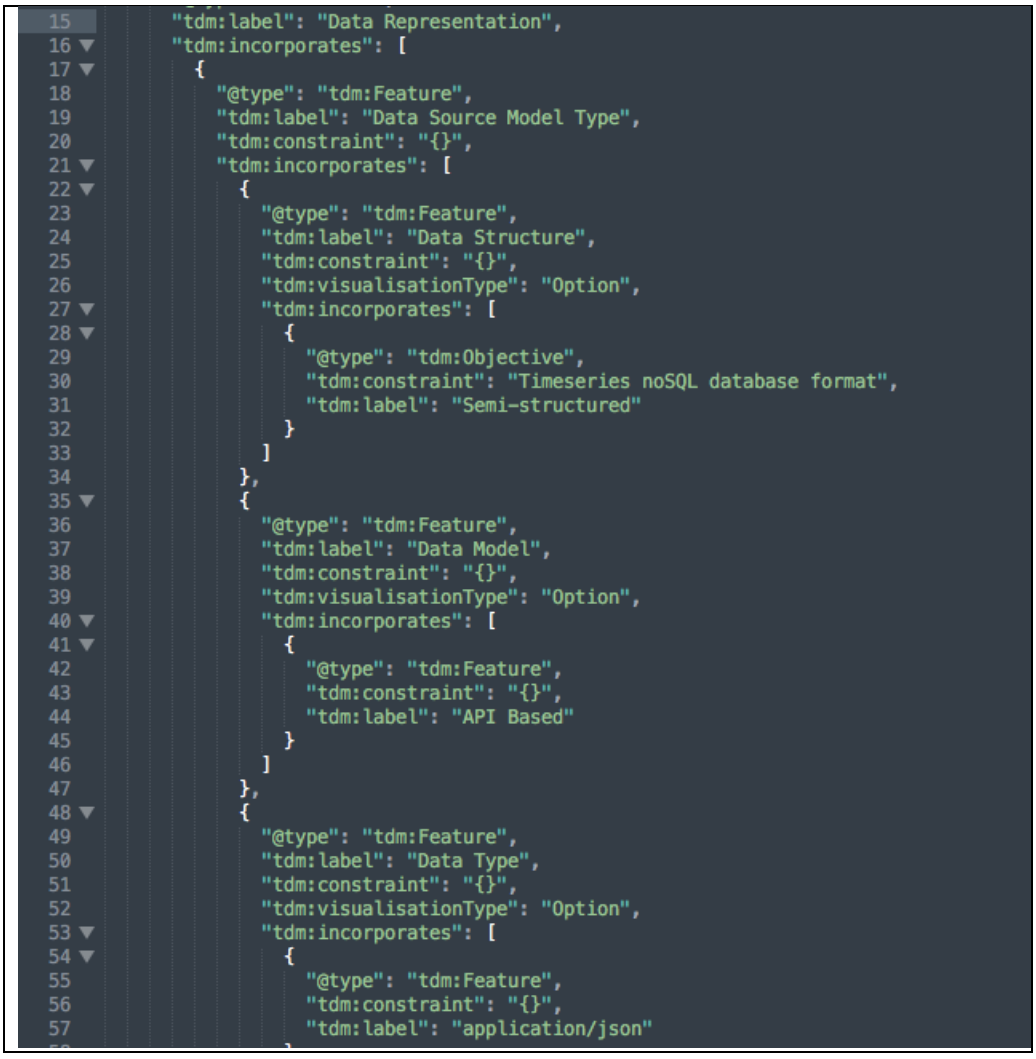


Figure 15: MBDAaaS process

Figure 16: Key sections of LIGHT's Declarative Model Definition in JSON format.



```
"tdm:label": "Data Representation",
"tdm:incorporates": [
  {
    "@type": "tdm:Feature",
    "tdm:label": "Data Source Model Type",
    "tdm:constraint": "{}",
    "tdm:incorporates": [
      {
        "@type": "tdm:Feature",
        "tdm:label": "Data Structure",
        "tdm:constraint": "{}",
        "tdm:visualisationType": "Option",
        "tdm:incorporates": [
          {
            "@type": "tdm:Objective",
            "tdm:constraint": "Timeseries noSQL database format",
            "tdm:label": "Semi-structured"
          }
        ]
      },
      {
        "@type": "tdm:Feature",
        "tdm:label": "Data Model",
        "tdm:constraint": "{}",
        "tdm:visualisationType": "Option",
        "tdm:incorporates": [
          {
            "@type": "tdm:Feature",
            "tdm:constraint": "{}",
            "tdm:label": "API Based"
          }
        ]
      },
      {
        "@type": "tdm:Feature",
        "tdm:label": "Data Type",
        "tdm:constraint": "{}",
        "tdm:visualisationType": "Option",
        "tdm:incorporates": [
          {
            "@type": "tdm:Feature",
            "tdm:constraint": "{}",
            "tdm:label": "application/json"
```

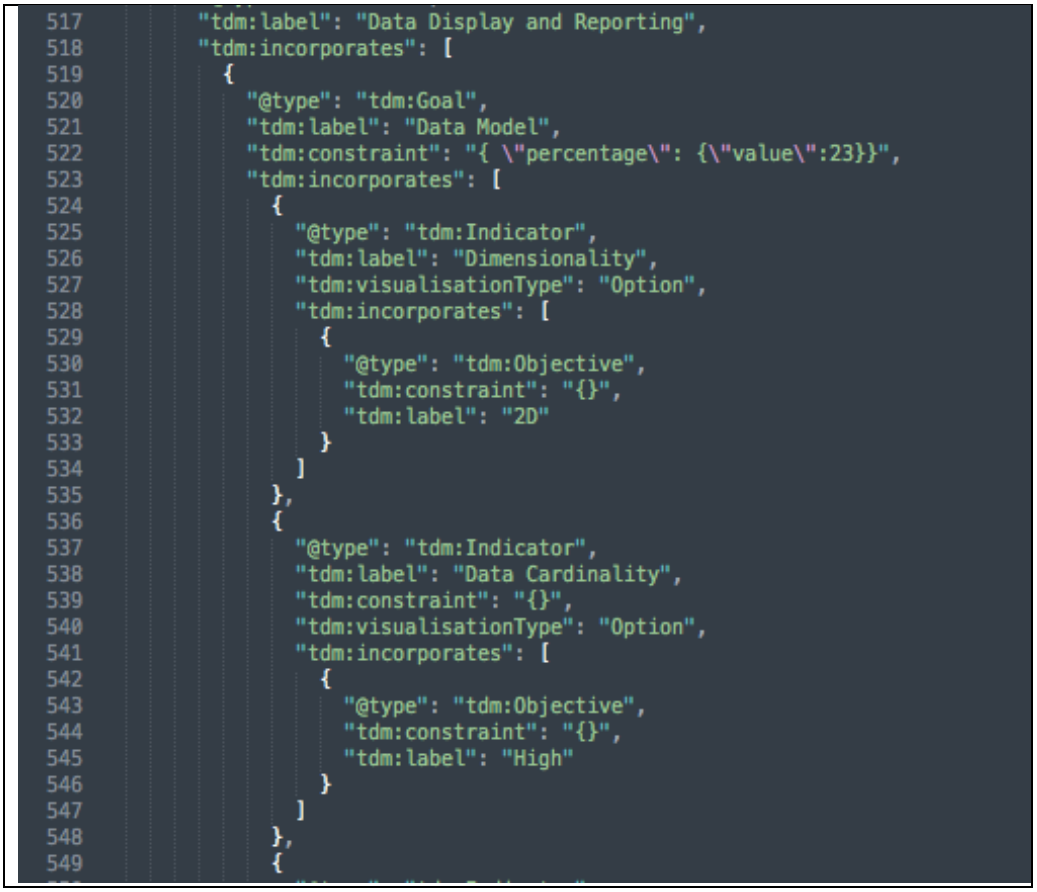
Figure 17: LIGHT Data Source structure in the Declarative Model (JSON).

```
{
  "@type": "tdm:Goal",
  "tdm:label": "Anonymization",
  "tdm:constraint": "{}",
  "tdm:incorporates": [
    {
      "@type": "tdm:Indicator",
      "tdm:label": "Anonymization Model",
      "tdm:constraint": "{}",
      "tdm:visualisationType": "Option",
      "tdm:incorporates": []
    },
    {
      "@type": "tdm:Indicator",
      "tdm:label": "Anonymization Techniques",
      "tdm:constraint": "{}",
      "tdm:visualisationType": "Option",
      "tdm:incorporates": [
        {
          "@type": "tdm:Objective",
          "tdm:constraint": "{ \"Algorithm\": \"SHA-256\",\n\"Target\": \"GatewayID\" }",
          "tdm:label": "Hashing"
        }
      ]
    }
  ]
}
```

Figure 18: Anonymisation specification in Declarative Model (JSON).

```
"tdm:analytics": {
  "@id": "http://www.toreador-project.eu/TDM/project-spec/area-analytics",
  "@type": "tdm:Area",
  "tdm:label": "Data Analytics",
  "tdm:incorporates": [
    {
      "@type": "tdm:Goal",
      "tdm:label": "Analytics Aim",
      "tdm:constraint": "{ \"percentage\": {\"value\":23}}",
      "tdm:incorporates": [
        {
          "@type": "tdm:Indicator",
          "tdm:label": "Task",
          "tdm:visualisationType": "Checkbox",
          "tdm:constraint": "{}",
          "tdm:incorporates": [
            {
              "@type": "tdm:Objective",
              "tdm:label": "Anomaly/Fault detection",
              "tdm:constraint": "{  \n   \"Method\":\"Mean Time Between Failure\",\n   \"ScenarioType\
            }
          ]
        },
        {
          "@type": "tdm:Indicator",
          "tdm:label": "Models",
          "tdm:constraint": "{}",
          "tdm:visualisationType": "Checkbox",
```

Figure 19: Declaration of the MTBF analytics in the JSON format.



```
517         "tdm:label": "Data Display and Reporting",
518         "tdm:incorporates": [
519           {
520             "@type": "tdm:Goal",
521             "tdm:label": "Data Model",
522             "tdm:constraint": "{ \"percentage\": {\"value\":23}}",
523             "tdm:incorporates": [
524               {
525                 "@type": "tdm:Indicator",
526                 "tdm:label": "Dimensionality",
527                 "tdm:visualisationType": "Option",
528                 "tdm:incorporates": [
529                   {
530                     "@type": "tdm:Objective",
531                     "tdm:constraint": "{}",
532                     "tdm:label": "2D"
533                   }
534                 ]
535               },
536               {
537                 "@type": "tdm:Indicator",
538                 "tdm:label": "Data Cardinality",
539                 "tdm:constraint": "{}",
540                 "tdm:visualisationType": "Option",
541                 "tdm:incorporates": [
542                   {
543                     "@type": "tdm:Objective",
544                     "tdm:constraint": "{}",
545                     "tdm:label": "High"
546                   }
547                 ]
548               },
549               {
```

Figure 20: Declaration of the data display preference in the JSON format.

2.3 Procedural Model

Following MBDAaaS process methodology (Figure 15) Declarative Model is transformed into the Procedural Model where selection of the services stored on the platform takes place. Example of anonymization for the hashing in Figure 21.

```
<owl:NamedIndividual rdf:about="tdm:spark:hashing">
  <rdf:type rdf:resource="tdm:MaterialService"/>
  <realizeSpecification rdf:resource="tdm:data_preparation:anonymization"/>
</owl:NamedIndividual>
  <owl:NamedIndividual rdf:about="tdm:data_preparation:anonymization">
   <rdf:type rdf:resource="tdm:Anonymization"/>
    <hasArea rdf:resource="tdm:dataPreparation"/>
    <hasConstraint rdf:resource="tdm:OWLNamedIndividual_00000"/>
    <hasIndicator rdf:resource="tdm:data_preparation:anonymization:technique"/>
    <hasObjective rdf:resource="tdm:data_preparation:anonymization:technique:hashing"/>
   </owl:NamedIndividual>
<owl:NamedIndividual rdf:about="tdm:OWLNamedIndividual_00000">
  <rdf:type rdf:resource="tdm:Constraint"/>
    <hasURIValue rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI"> http://toreador
  <onParameter rdf:resource="tdm:normalizedDataset"/>
</owl:NamedIndividual>
```

Figure 21: Example of OWL-S process selection for data anonymization.

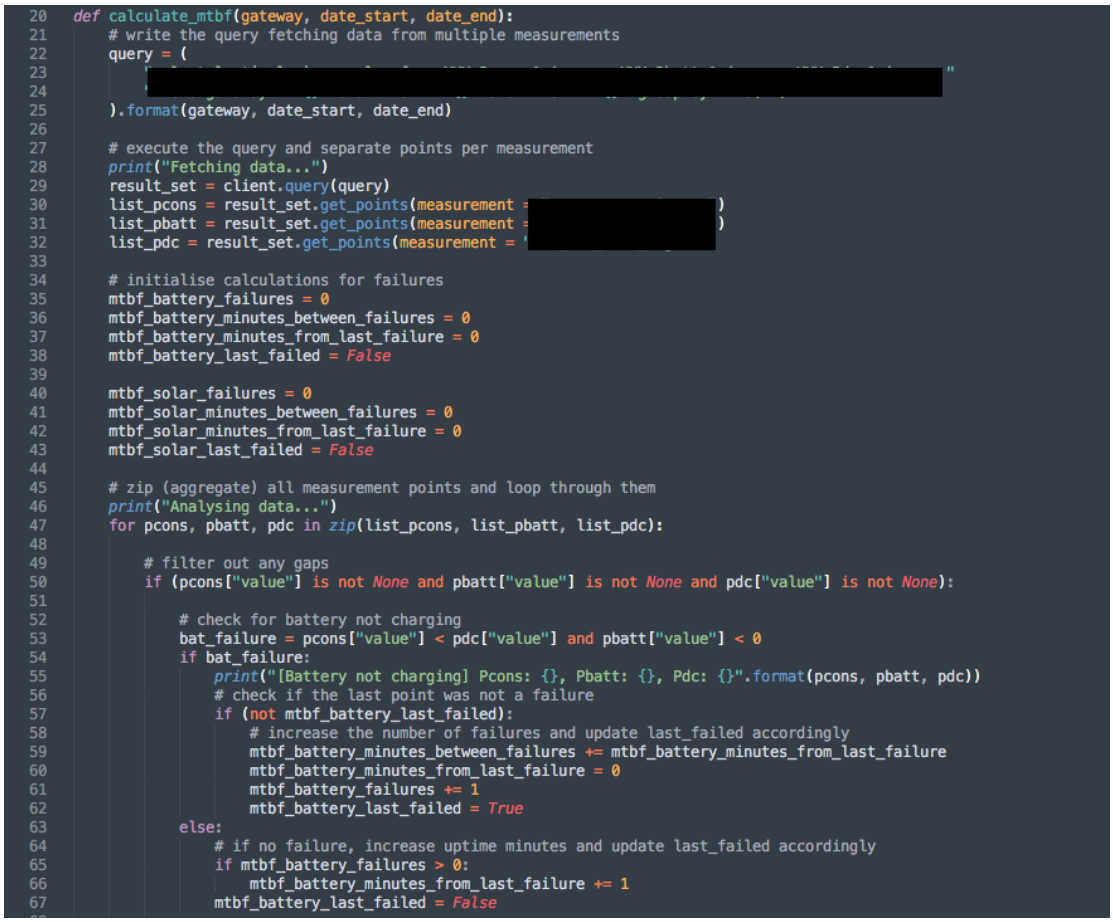
The last step in the MBDAaaS process is the definition the OWL-S workflow. Figure 22.

```
<process:CompositeProcess rdf:about="tdm:CompositeProcessXXXXX">
    <process:composedOf>
        <process:Sequence>
            <process:components>
                <process:ControlConstructList>
                    <list:first>
                        <process:Perform>
                            <process:process rdf:resource="tdm:AnonymizationService"/>
                        </process:Perform>
                    </list:first>
            [...]
                <process:ControlConstructList>
                    <list:first>
                        <process:Perform>
                            <process:process rdf:resource="tdm:MeanTimeBetweenFailures"/>
                        </process:Perform>
                    </list:first>
            [...]
                </process:ControlConstructList>
            </process:components>
        </process:Sequence>
    </process:composedOf>
</process:CompositeProcess>
```

Figure 22: Example of the OWL-S workflow for anonymization and MTBF analysis

2.4 Deployment Model

Process defined in Figure 15 transforms the procedural model into the deployment model which is to be executed directly on the platform using selected compiler. LIGHT pilot is using Python scripts to calculate Mean time between failures for *Generation* and *Charger*. Figure 23.



```python
def calculate_mtbf(gateway, date_start, date_end):
    # write the query fetching data from multiple measurements
    query = (
        [redacted]
    ).format(gateway, date_start, date_end)

    # execute the query and separate points per measurement
    print("Fetching data...")
    result_set = client.query(query)
    list_pcons = result_set.get_points(measurement = [redacted])
    list_pbatt = result_set.get_points(measurement = [redacted])
    list_pdc = result_set.get_points(measurement = [redacted]

    # initialise calculations for failures
    mtbf_battery_failures = 0
    mtbf_battery_minutes_between_failures = 0
    mtbf_battery_minutes_from_last_failure = 0
    mtbf_battery_last_failed = False

    mtbf_solar_failures = 0
    mtbf_solar_minutes_between_failures = 0
    mtbf_solar_minutes_from_last_failure = 0
    mtbf_solar_last_failed = False

    # zip (aggregate) all measurement points and loop through them
    print("Analysing data...")
    for pcons, pbatt, pdc in zip(list_pcons, list_pbatt, list_pdc):

        # filter out any gaps
        if (pcons["value"] is not None and pbatt["value"] is not None and pdc["value"] is not None):

            # check for battery not charging
            bat_failure = pcons["value"] < pdc["value"] and pbatt["value"] < 0
            if bat_failure:
                print("[Battery not charging] Pcons: {}, Pbatt: {}, Pdc: {}".format(pcons, pbatt, pdc))
                # check if the last point was not a failure
                if (not mtbf_battery_last_failed):
                    # increase the number of failures and update last_failed accordingly
                    mtbf_battery_minutes_between_failures += mtbf_battery_minutes_from_last_failure
                    mtbf_battery_minutes_from_last_failure = 0
                    mtbf_battery_failures += 1
                    mtbf_battery_last_failed = True
            else:
                # if no failure, increase uptime minutes and update last_failed accordingly
                if mtbf_battery_failures > 0:
                    mtbf_battery_minutes_from_last_failure += 1
                mtbf_battery_last_failed = False
```

```
            # check for system not producting solar
            solar_failure = pdc["time"].endswith("12:00:00Z") and pdc["value"] == 0
            if solar_failure:
                print("[System not producing Solar] Pdc: {}".format(pdc))
                # check if the last point was not a failure
                if (not mtbf_solar_last_failed):
                    # increase the number of failures and update last_failed accordingly
                    mtbf_solar_minutes_between_failures += mtbf_solar_minutes_from_last_failure
                    mtbf_solar_minutes_from_last_failure = 0
                    mtbf_solar_failures += 1
                    mtbf_solar_last_failed = True
            else:
                # if no failure, increase uptime minutes and update last_failed flag accordingly
                if mtbf_solar_failures > 0:
                    mtbf_solar_minutes_from_last_failure += 1
                mtbf_solar_last_failed = False

    mtbf_battery = mtbf_battery_minutes_between_failures / (mtbf_battery_failures if mtbf_battery_failures != 0 else 1)
    mtbf_solar = mtbf_solar_minutes_between_failures / (mtbf_solar_failures if mtbf_solar_failures != 0 else 1)

    return (mtbf_battery, mtbf_solar)

# entry point
gateway = [redacted]
date_start = datetime(2017, 6, 4, 0, 0, 0)
date_end = datetime(2017, 6, 4, 23, 59, 59)
mtbf_battery, mtbf_solar = calculate_mtbf(gateway, date_start, date_end)
msg = (
    "For Gateway {} between {} and {}\n"
    "- MTBF for Battery not charging: {} minutes\n"
    "- MTBF for System not producing Solar: {} minutes"
).format(gateway, date_start, date_end, mtbf_battery, mtbf_solar)
print(msg)
```

Figure 23: Deployment Model for the MTBF analysis in the Python script.

The expected output from the Python script for MTBF analysis for Charger and Generator , is displayed in Figure 24:

```
Fetching data...
Analysing data...
[Battery not charging] Pcons: {'time': '2017-06-04T09:29:00Z', 'value': 0.115}, Pbatt:
{'time': '2017-06-04T09:29:00Z', 'value': -0.011}, Pdc: {'time': '2017-06-04T09:29:00Z',
'value': 0.732}
[Battery not charging] Pcons: {'time': '2017-06-04T17:15:00Z', 'value': 0.302}, Pbatt:
{'time': '2017-06-04T17:15:00Z', 'value': -0.099}, Pdc: {'time': '2017-06-04T17:15:00Z',
'value': 0.721}
[Battery not charging] Pcons: {'time': '2017-06-04T17:22:00Z', 'value': 0.357}, Pbatt:
{'time': '2017-06-04T17:22:00Z', 'value': -1.579}, Pdc: {'time': '2017-06-04T17:22:00Z',
'value': 0.495}
[Battery not charging] Pcons: {'time': '2017-06-04T17:31:00Z', 'value': 0.329}, Pbatt:
{'time': '2017-06-04T17:31:00Z', 'value': -0.171}, Pdc: {'time': '2017-06-04T17:31:00Z',
'value': 0.364}
For Gateway UTXb98803 between 2017-06-04 00:00:00 and 2017-06-04 23:59:59
- MTBF for Battery not charging: 119.75 minutes
- MTBF for System not producing Solar: 0.0 minutes
```

Battery not charging:
- total minutes between failures: `479`
- total number of failures: `4`
- MTBF: `479/4` = `119.75`

Figure 24: MTBF Python script outcome

Code base compiler for LIGHT MTBF algorithm is to be applied on the data ingestion by Lightsource API into the Toreador Platform through its API. Detailed architecture is defined in the FIGURE 25.

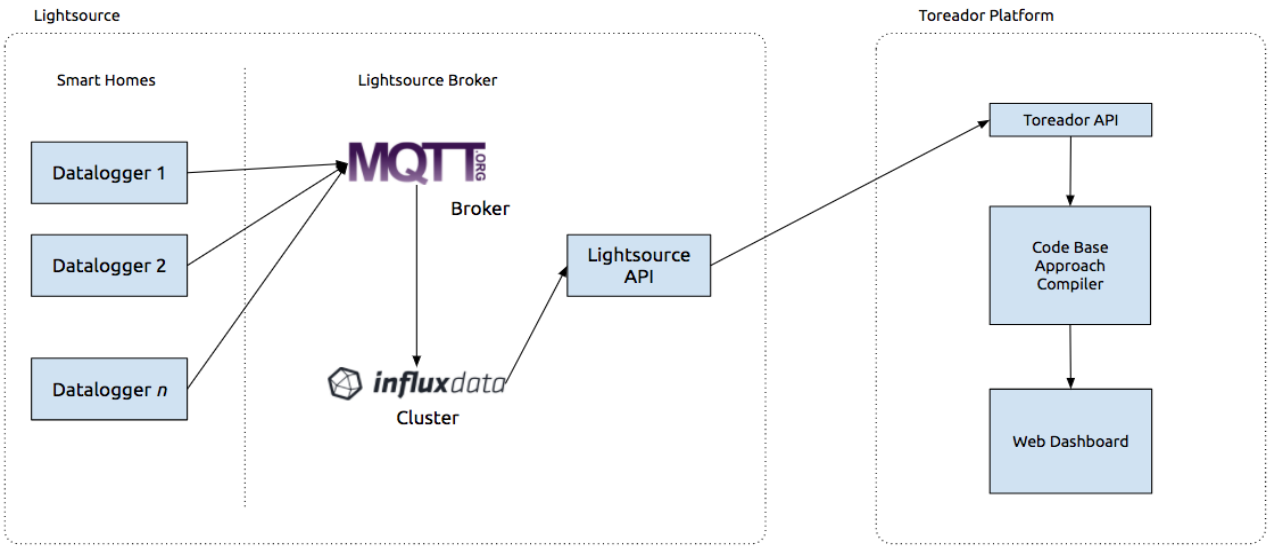


Figure 25: Deployment model architecture for the LIGHT pilot

Chapter 3

The TOREADOR factor

3.1 How TOREADOR helped with the design process

By using the MTBF model, under the TOREADOR project's umbrella, LIGHT wants to protect its equipment for future failures. As it was mentioned before, the unit of measurement includes only operational time between failures and does not include repair times, assuming the item is repaired and begins functioning again which can create a significant loss of income, as solar panels and battery are the key components for our smart home project and MTBF figures are often used to project how likely a single unit is to fail within a certain period, which can be proved extremely useful for LIGHT. The definition of MTBF relies on the category of a system failure and specifically for the solar assets there are several failures which are quite expensive and time-consuming to be repaired and replaced if it is necessary (long / very long lead items for a solar asset).

So, the process designed in the declarative, procedural and deployment models will definitely assist us to avoid critical and challenging situations which can influence the efficiency of our equipment and definitely the services quality.

Chapter 4

Conclusions and perspectives

The content of Deliverable 10.3 Toreador platform implementation for the Energy Production Data Analysis pilot will very helpful for LIGHT and the usage of the MTBF methodology with the combination of the correct models can bring a lot of benefits in the future. LIGHT is looking forward to seeing the results of the TOREADOR platform, within the next months, on the use cases described before.

Chapter 5

References/Bibliography

[1] TrustwOrthy model-awaRE Analytics Data platfORm (TOREADOR) Project. Available online: http://cordis.europa.eu/project/rcn/200253_en.html

[2] TOREADOR Project. Deliverable D10.1 Legal Aspects Specification for the Energy Pilot (2016)

[3] TOREADOR Project. Deliverable D10.2 Requirement specifications for big data analytics related to Energy Production Data Analysis processes (2016)