Generalized Operational FLEXibility for Integrating Renewables in the Distribution Grid (GOFLEX)

D5.1 – Cloud-based Service Platform Requirement and Interface Specification

April 2017

Imprint

Contractual Date of Delivery to the EC:	**30 April 2017**
Actual Date of Delivery to the EC:	28 April 2017

Author(s):	**Mark Purcell, John Sheehan, Robert Gormally (IBM)**
Participant(s):	**Zoran Marinšek (INEA), Seshu Tirupathi (IBM)**
Reviewer(s):	**Sašo Brus (INEA)**

Project:	Generalized Operational FLEXibility for Integrating Renewables in the Distribution Grid (GOFLEX)
Work package:	**WP5**
Confidentiality:	Public
Version:	**1.1**
Contact:	**Mark Purcell – markpurcell@ie.ibm.com**
Website:	www.goflex-community.eu

Legal disclaimer

The project Generalized Operational FLEXibility for Integrating Renewables in the Distribution Grid (GOFLEX) has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731232. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the Innovation and Networks Executive Agency (INEA) or the European Commission (EC). INEA or the EC are not responsible for any use that may be made of the information contained therein.

Copyright

Executive Summary

This document (D5.1) presents the first activities of Task 5.1, and describes the requirements for a cloud-based service platform to support the GOFLEX project. Detailed in this document are the requirements for interoperability between the cloud-based service platform and the various other GOFLEX components provided by other work packages.

Document History

Version	Date	Status	Author	Comment
0.1	24th Jan 2017	Draft	Mark Purcell	First version
0.2	17th February 2017	Draft	Mark Purcell	Ljubljana Comments
0.3	24th March	Draft	Mark Purcell	Internal Review
1.0	5th April 2017	For review	Mark Purcell	For consortium Review
1.1	21st April 2017	Reviewed	Mark Purcell	Addressed comments

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

Abbreviation	Definition
API	Application Programming Interface
ATP	Automatic Trading Platform
DOMS	Distribution Observability and Management System
DSO	Distribution System Operators
ECAST	Energy Forecasting System
EMS	Energy Management System
FMAN	Flex-offer Manager
FMAR	Flex-offer Market
FOA	Flex-offer Agent
GIS	Geographic Information System
IoT	Internet of Things
JSON	JavaScript Object Notation
MQTT	Message Queue Telemetry Transport
SCADA	Supervisory Control and Data Acquisition
WP	Work Package

1 Introduction

1.1 Purpose

This document provides a description of the requirements for the core components of the *Cloud-based Service Platform*, the application programming interfaces (APIs) that need to be exposed to external components, the required functional & data interactions between internal and external components, architectural assumptions, and a proposed delivery sequence.

The GOFLEX project contains three distinct demonstration sites, represented by WP7, 8, and 9. As such, it is assumed that it is necessary to have three distinct instances of the *Cloud-based Service Platform*.

The purpose is to aid in the alignment of the relevant partners' deliverables to ensure a coherent, functional, and performant system is delivered when the project concludes.

1.2 Related Documents

This document should be read in conjunction with the following *Requirement & Interface Specification* documents – D2.1, D3.1, D4.1, and D6.1.

1.3 Document Structure

Section two describes the core components required for the *Cloud-based Service Platform*, and the APIs it is required to expose to external components, which are provided by other work packages. Section three describes external components' functional & data dependencies on the service platform. Section four describes the cloud platform's functional & data dependencies on external components. Sections five and six list the functional and non-functional requirements to be delivered by the service platform. Section seven details architectural assumptions relevant to the requirements described. Section eight describes the sequencing of the delivery of requirements.

2 Work Package Description

The *Cloud-based Service Platform* (WP5) provides a cloud-based energy network data and forecast service platform. This provides data and forecasting services to the Automatic Trading Platform (ATP) delivered by *Automatic Demand Response Trading Work Package* (WP2), the Energy Management Systems (EMS) described in the *Augmented Demand Response Ready Prosumer Work Package* (WP3), and the *Distribution Observability and Management System* (WP4).

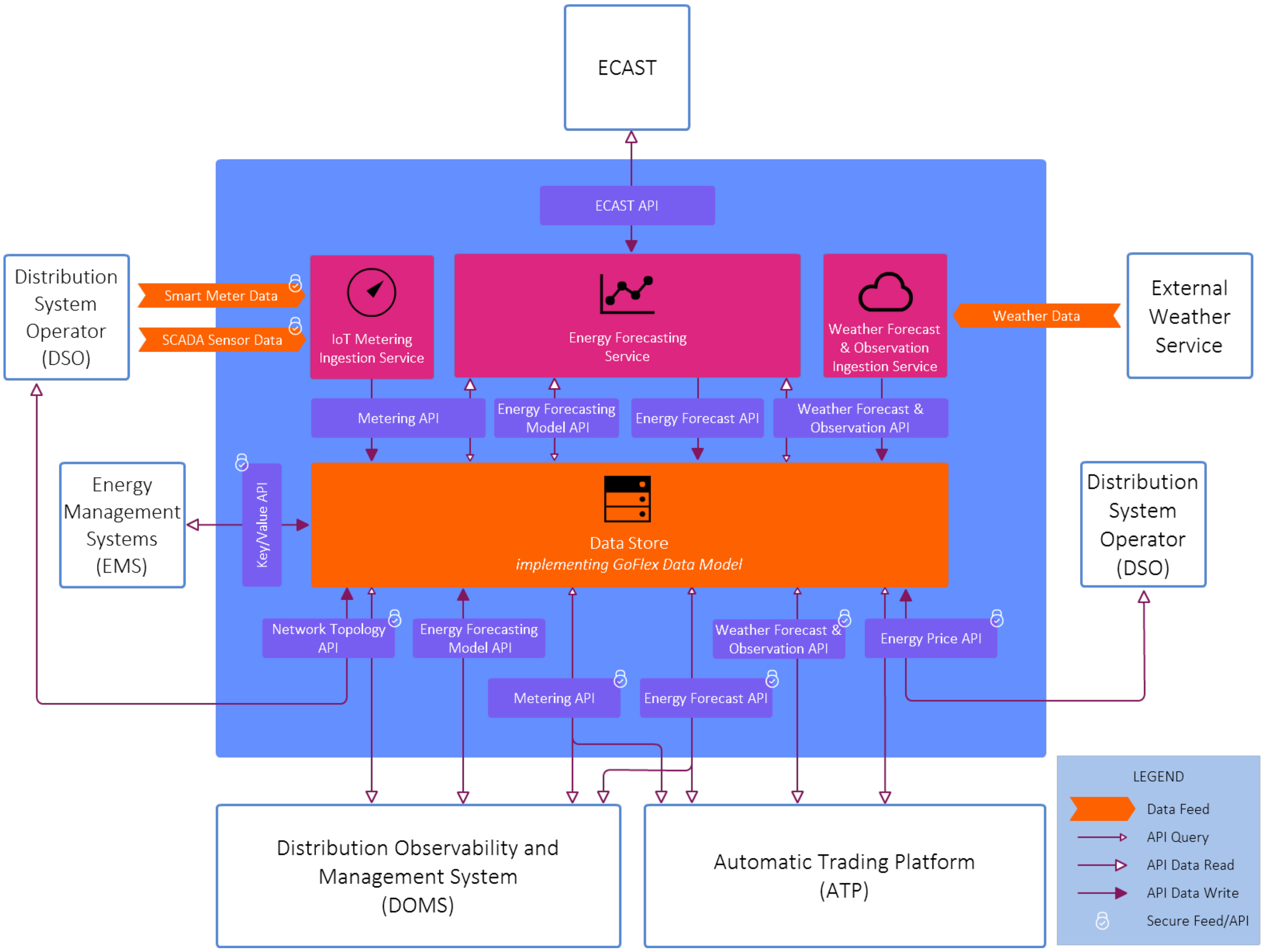

Figure 1 - High-level overview of the Cloud-based Service Platform

A functional implementation of WP5 and its interactions with WP2, WP3, and WP4 will be demonstrated by the *Cypriot System Deployment & Evaluation* (WP7), *Swiss System Deployment & Evaluation* (WP8), and *German System Deployment & Evaluation* (WP9).

The requirements gathering phase has identified many individual requirements, which can broadly be categorized into four functional areas. Therefore, the service platform provided by WP5 is discussed in terms of these four areas – an *IoT Data Ingestion Service*, a *Weather Forecast & Observation Ingestion Service*, an *Energy Forecasting Service*, and a *Data Store* that implements a *GOFLEX Data Model* (see Figure 1). These functional areas are supported by several APIs and are protected via a security policy.

2.1 IoT Data Ingestion Service

The *IoT Data Ingestion Service* must accept both energy consumption and generation data from a diverse range of sources, such as prosumer smart meters and energy grid SCADA sensors. This service could potentially be implemented using Message Queue Telemetry Transport (MQTT). The service must be secure, providing secure data transport, authentication, and authorization.

Each demonstration site must design, implement and deploy their own ingestion clients to extract and anonymize data from their local data stores, and to publish the data to the service for ingestion - see Sections 4.1 and 4.2. Sample client software highlighting how to implement the secure data transport must be provided by WP5.

The ingested metering data must be persisted to the Data Store.

2.2 Weather Forecast & Observation Ingestion Service

The *Weather Forecast & Observation Ingestion* Service consumes timely weather forecast and historical observation data from external weather data providers for each energy network entity location.

Forecasts of weather variables including surface temperature, solar irradiance, wind speed, and cloud cover, with an interval between fifteen and sixty minutes over a 48-hour horizon are persisted to the Data Store and made available to other components via the *Weather Forecast & Observation API.*

Retrospective weather observations, with an interval between fifteen and sixty minutes over a two-year period, are also consumed enabling the *Energy Forecasting* Service to perform energy forecast model training.

2.3 Energy Forecasting Service

The *Energy Forecasting Service* produces energy consumption and energy generation forecasts with a temporal resolution between fifteen and sixty minutes over a 48-hour horizon for all energy network entities with energy forecast model instances attached.

Energy forecast model instances are created, managed, and attached to network entities of interest via the *Energy Forecast Model API*.

During a forecast run, all forecast model instances are retrieved via the *Energy Forecast Model* API, and forecast model predictor signals identified. Entity-specific predictor signal data, such as meter observations and local weather forecasts, are retrieved via the *Metering API* and the *Weather Forecast & Observation API.* Generic predictor signal data, such as calendar day categorical data, are also retrieved from the *Data Store*. These predictor signal data are consumed by the energy forecast model instances, and energy consumption and generation forecasts generated for each network entity. The resulting forecasts are persisted to the *Data Store,* and made available to other components, via the *Energy Forecast API.*

In addition, the service performs scheduled automated energy forecast model training to ensure the continued accuracy of energy forecasts generated by the service. Historical metering and weather observation data over a two-year period are retrieved via the *Metering API and Weather Forecast & Observation API*, and all energy forecast model instances retrained. The re-weighted model instances are updated via the *Energy Forecast Model API*.

2.4 Data Store

The *Data Store*, which implements a *GOFLEX Data Model*, represents and persists energy network entities and their associated metadata, meter data from prosumer smart meters & SCADA sensors, weather forecasts and historical observations, reference electricity market prices, and calendar data. It also represents and persists the energy forecast models and their instances, and the energy forecasts they produce. Additionally, a generic key/value store caters for bespoke data persistence requirements not yet anticipated. The data in the *Data Store* is made available to other components/users, such as DSOs, ATPs, and EMSs, via the APIs already described.

2.5 Additional API's

In addition to the APIs described above, the following three APIs are also exposed – the *Network Topology API*, the *Energy Price API*, and the *Key/Value API*.

The *Network Topology API* allows DSOs to register and manage energy network entities on the *Cloud-based Service Platform*, such as prosumer smart meters, SCADA sensors, energy network feeders & substations. Entity metadata such as location, layout, and connectivity are also managed via this API. Other components, such as the *Weather Forecast Ingestion Service* and DOMS, can query this API for relevant entity metadata.

The *Energy Price API* allows the management of reference market energy prices on the service platform permitting other components, such as the ATP, to query for up-to-date spot energy prices.

Finally, the *Key/Value API* exposes a generic interface through which unstructured data can be persisted and retrieved to and from the *Data Store* by component such as EMSs. This allows the *Data Store* to be used for unanticipated data requirements as the project progresses. Client components are responsible for the coherence of data crossing this API and must implement the appropriate client side business logic to ensure data integrity. Also, where bespoke aggregated data is required, such data transformations will be performed by the API clients.

2.6 Security, Authentication & Authorization

Security, in the context of the WP5, refers to the secure transfer of anonymized data between partners' sites and the remote cloud based service platform. As the data will travel over the public internet, it is subject to interception, inspections, and possibly manipulation. To combat this, public facing APIs must provide secure, encrypted endpoints. This allows clients to negotiate a connection enabling the secure transfer of data.

Authentication is a method to identify users of the system. This allows legitimate users to access the public APIs and could be managed using for example, authentication tokens.

Authorization is the mechanism to determine what actions authenticated users can perform. Distinct user roles and capacities as well as enforcement policies are required. User roles could potentially vary from open access to public data to authenticated and encrypted access to more sensitive data.

3 Provided to other work packages / components

3.1 Functionality

Functionality provided by the service platform is now discussed in terms of the relationships to other work packages.

3.1.1 WP 2 – Automatic Trading Platform (ATP)

The *Cloud-based Service Platform* exposes several APIs to the three subsystems of the ATP – the Flex-offer Agent (FOA), the Flex-offer Manager (FMAN), and Flex-offer Market (FMAR). The data provided over these APIs enables the ATP to perform its function in energy market flex-offer generation and validation.

The APIs to be utilised by the ATP are: The *Metering API*, the *Weather Forecast & Observation API*, the *Energy Forecast API*, the *Energy Price API* and the *Transmission API*. The data traversing these APIs to and from the ATP are described in Section 3.2.1.

3.1.2 WP 2 – Distribution System Operator (DSO)

The *Cloud-based Service Platform* exposes to the DSO an *IoT Data Ingestion Service* for the ingestion of metering data, both prosumer meters and energy grid sensor data. DSOs can publish real-time and historical data to this service and must design and implement service clients based on their own unique data storage and acquisition features.

The platform also exposes the following APIs, through which DSOs can read and write data to the platform *Data Store* - the *Network Topology API* and the *Energy Price API*. The data traversing these APIs to and from the DSO are described in Section 3.2.2.

3.1.3 WP 3 – Energy Management System (EMS)

The *Cloud-based Service Platform* makes available to the EMSs a *Key/Value API* for the storage and retrieval of bespoke data objects and time-series observations. Due to the unstructured nature of the data traversing this API, data coherence and integrity must be implemented and managed by the client-side business logic embedded in the EMSs.

3.1.4 WP 4 – Distribution Observability and Management System (DOMS)

The platform makes available to DOMS a variety of APIs to enable it to perform its core functions of energy network observability and predictions of near-term power flow and state.

The APIs to be made available to DOMS are – the *Metering API*, the *Energy Forecast Model API*, the *Energy Forecast API*, and the *Network Topology API*. The data traversing these APIs to and from the DOMS are described in Section 3.2.4.

3.1.5 WP 5 – ECAST

The *Energy Forecasting Service* utilises multiple energy forecasting models per network entity of interest. To select the best model, the service platform utilises the external ECAST service to rank the energy forecasting model performance in terms of forecasting error and/or computational cost.

The *Energy Forecasting Service* retrieves energy forecast models and predictor data via the *Energy Forecast Model*, *Metering*, and *Weather Forecast & Observation APIs*, and invokes the ECAST service via the *ECAST API*. The data traversing these APIs to and from ECAST are described in Section 3.2.5.

3.2 Data

Service platform data requirements are now discussed, categorised by the data provided to other work packages and linked to defined functional requirements.

3.2.1 WP 2 – Automatic Trading platform (ATP)

Table 1 describes the data flows between the ATP and the *Cloud-based Service Platform*, and the corresponding WP5 functional requirement for data storage, querying, and retrieval.

Table 1 - ATP and Cloud Service Platform Data Flows

Data Flow	Functional Requirement
Querying & retrieval of prosumer meter data for flex offer generation.	F5.4 – *Metering API*
Querying & retrieval of surface temperature & solar irradiance forecasts at prosumer locations, with a temporal resolution between fifteen and sixty minutes over a 48-hour forecasting horizon, for flex offer generation.	F5.6 – *Weather Forecast & Observation API*
Querying & retrieval of prosumer energy consumption & generation forecasts with a temporal resolution between fifteen and sixty minutes over a 48-hour horizon, for flex offer validation.	F5.9 – *Energy Forecast API*
Querying & retrieval of energy market spot prices for flex offer generation.	F5.11 – *Energy Price API*
Querying & retrieval of transmission loss between two points on the energy grid.	F5.14 – *Transmission API*

3.2.2 WP 2 – Distribution System Operator (DSO)

Table 2 describes the data flows between DSOs and the *Cloud-based Service Platform*, and the corresponding WP5 functional requirement for data storage, querying, and retrieval.

Table 2 - DSO and Cloud Service Platform Data Flows

Data Flow	Functional Requirement
Storage, to an underlying *Data Store* of anonymized real-time & historical prosumer and energy network sensor data.	F5.3 – *IoT Data Ingestion Service*
Storage & maintenance of the energy network topology representation, including prosumers and other physical & logical network entities. Metadata relating to entity GIS location, interconnectivity, & capacity are also stored and maintained by the DSO via this API.	F5.10 – *Network Topology API*
Storage & maintenance of energy market spot prices.	F5.11 – *Energy Price API*

3.2.3 WP 3 – Energy Management System (EMS)

Table 3 describes the data flows between EMSs and the *Cloud-based Service Platform*, and the corresponding WP5 functional requirement for data storage, querying, and retrieval.

Table 3 - EMS and Cloud Service Platform Data Flows

Data Flow	Functional Requirement
Storage, querying, and retrieval of bespoke unstructured data & time-series to and from the *Data Store*.	F5.12 – *Key/Value API*

3.2.4 WP 4 – Distribution Observability and Management System (DOMS)

Table 4 describes the data flows between DOMS and the *Cloud-based Service Platform*, and the corresponding WP5 functional requirement for data storage, querying, and retrieval.

Table 4 - DOMS and Cloud Service Platform Data Flows

Data Flow	Functional Requirement
Querying & retrieval of real-time & historical prosumer & SCADA sensor data to enable the estimation of the current state of energy network entities, such as substations and feeders.	F5.4 – *Metering API*
Registration of energy forecasting model instances with network entities of interest to predict near-term future energy consumption & generation by those entities.	F5.7 – *Energy Forecasting Model API*
Querying & retrieval of energy forecasts for network entities to predict the short-term future states of energy network assets.	F5.9 – *Energy Forecast API*
Querying & retrieval of prosumers GIS location & energy grid connectivity.	F5.10 – *Network Topology API*

3.2.5 WP 5 – ECAST

Table 5 describes the data flows between ECAST and the *Cloud-based Service Platform*, and the corresponding WP5 functional requirement for data storage, querying, and retrieval.

Table 5 - ECAST and Cloud Service Platform Data Flows

Data Flow	Functional Requirement
Querying & retrieval of energy forecasting model predictor data (meter readings) by the *Energy Forecasting Service*.	F5.4 – *Metering API*
Querying & retrieval of energy forecasting model predictor data (weather forecasts & observations) by the *Energy Forecasting Service*.	F5.6 – *Weather Forecast & Observation API*
Querying & retrieval by the *Energy Forecasting Service* of energy forecasting models to be ranked by ECAST. Storage of forecast model rankings to the *Data Store*.	F5.7 – *Energy Forecasting Model API*
Submission of forecast models to be ranked and related predictor data. Retrieval of forecast model ranking based on forecast accuracy and/or computational cost.	F5.13 – *ECAST API*

4 Depends on other work packages / components

4.1 Functionality

4.1.1 WP 7/8/9 - Data Ingestion Client

The service platform provides an IoT Data Ingestion Service (Section 2.1) for the ingestion of anonymized smart meter data. Consequently, work packages which submit such data must provide a software client for this purpose. The client must:

- extract meter readings from the energy providers' data store
- anonymize this data
- convert any local timestamps to UTC
- package measurement samples into a payload

Once the client has prepared data for submission, it connects over a secure channel to the Ingestion Service, authenticates itself to this service, and then transmits its payload. The client can then disconnect, or maintain an active connection for future data transfers. Preferably the client dispatches new data to the Ingestion Service once it is generated, but bursts of messages can also be accepted. As noted in section 2.1, work package 5 provides example client software for this ingestion client.

4.1.2 WP 2/3 - Management of ad hoc storage

GOFLEX components may also wish to store anonymized metadata. The service platform provides a key/value store for data such as this. Since this data is assumed to be relatively static and of small quantity, it can be manually entered to the key/value store on an ad hoc basis. Should it transpire, that this process is semi-regular or contains large data quantities, automation can be considered, leveraging the key/value API provided by the service platform. If this route is taken, components wishing to use the key/value API must provide a client along with key/value appropriate message formats.

4.1.3 WP 7/8/9 - Energy Grid Topology Ingestion Client

The cloud service platform also expects data detailing the topology of the energy grid for each demonstration site. This data could be transferred to the service platform using for example, a common infrastructure model (CIM) with all data represented in a structured

JSON format. However, as this data transfer is likely to be a rare occurrence, a manual transfer process in partnership with each of the work packages may be preferable.

4.1.4 WP 4 - Distribution Observability and Management System (DOMS)

The Transmission API provides a means of determining the transmission loss between two points on the energy grid. The module performing the actual calculation of the loss is a feature of DOMS, and must be available to the service platform.

4.2 Data

4.2.1 Time and date formats

Time and date formats (timestamps) on the service platform are always represented as UTC. It is a requirement of client components to provide all timestamps in UTC format. Retrieved timestamps from the service platform are also in UTC format. As such, if a local timestamp is required by a dependent component, the component itself must provide the appropriate conversion.

4.2.2 Meter data

Smart meter data consists of time series data sampled by the meters at regular intervals, as well as meta data about the smart meter devices themselves. Smart meters can potentially provide the following data:

- Current magnitude (Ampere)
- Voltage magnitude (Volt)
- Active power (Watt)
- Reactive power (Volt-Ampere reactive)
- Interval energy consumption (Watt-Hour)
- Interval energy generation (Watt-Hour)
- Timestamp (converted by client to UTC)

Metadata relating to smart meter data can potentially take a more unstructured form, consisting of device id, device type, and possibly free form text (Unicode).

SCADA data consists of time series data, as well as meta data from SCADA itself and can be sampled at irregular intervals. Often, the following data can be provided:

- Active power (Watt)
- Reactive power (Volt-Ampere reactive)
- Voltage magnitude (Volt)
- Current magnitude (Ampere)
- Timestamp (converted by client to UTC)

4.2.3 Grid Topology

The cloud service platform also requires data detailing grid network topology. Network topology includes the list of assets such as transformers, substations, feeders, service points etc. This list may also have associated metadata such as GIS location and electrical parameters, e.g. impedance of lines or transformers, type of transformer, class of service points, voltage levels etc. Network topology also covers connectivity and connectivity descriptions e.g. which network entities are connected to which other network entities etc.

4.2.4 Demonstration sites

Correct implementation of the client (Section 4.1) allows for the submission of meter data. Upon installation of this ingestion client, further information from WPs 7/8/9 may be required to determine additional smart meter metadata that should be stored. Streamlined or automated transfer of grid network topology information can also be considered.

5 Functional Requirements

Table 6 - WP5 Functional Requirements

Requirement Number	Requirement Description
F5.1	A *GOFLEX Data Model* to represent the energy network entity, smart meter, SCADA, weather forecast, reference electricity market price, and calendar data. The data model also represents the energy demand & generation forecasting models, and the forecasts generated by those models.
F5.2	Data conforming to the model needs to be persisted to an underlying *GOFLEX Data Store*, and the data store initially populated with GOFLEX metadata and base energy forecasting models.
F5.3	A cloud-based *IoT Data Ingestion Service* exposing an internet facing service for the consumption of real-time & historical smart meter & SCADA sensor data. The service provides secure data transport, authentication, and authorization. Smart meter and SCADA sensor data is persisted to the *GOFLEX Data Store* via the *Metering API* in F5.4.
F5.4	A *Metering API* through which smart meter and SCADA sensor data can be managed and queried in the *GOFLEX Data Store*.
F5.5	A *Weather Forecast & Observation Ingestion Service* to retrieve weather forecast data from one or more external forecast providers. Surface temperature and solar irradiance predictions and observations are retrieved with an interval between fifteen and sixty minutes over a forecast horizon of 48 hours. Forecast data is persisted to the *GOFLEX Data Store* via the *Weather Forecast & Observation API* in F5.6.
F5.6	A *Weather Forecast & Observation API* to manage and query weather forecast and historical observation data in

	the *GOFLEX Data Store,* persisted there by the *Weather Forecast & Observation Ingestion Engine* in F5.5.
F5.7	An *Energy Forecasting Model API* through which energy demand & generation forecasting models and instances can be managed in the *GOFLEX Data Store*. Model instances can be attached to one or more energy network entities that are registered via the *Network Topology API* described in F5.10.
F5.8	An *Energy Forecasting Service* to run the energy demand & generation forecasting models registered via the *Energy Forecasting Model API*, resulting in energy forecasts with a temporal resolution between fifteen and sixty minutes over a forecast horizon of 48 hours. Energy forecasts are persisted to the *GOFLEX Data Store* via the *Energy Forecast API* described in F5.9. The service also periodically performs energy forecast model retraining to ensure continued forecast accuracy.
F5.9	An *Energy Forecast API* through which energy demand and generation forecasts can be managed and queried in the *GOFLEX Data Store*.
F5.10	A *Network Topology API* through which energy network entity data, such as energy grid network layout, and prosumer GIS location & energy grid connectivity, can be managed and queried in the *GOFLEX Data Store*.
F5.11	An *Energy Price API* via which reference market energy prices can be registered, updated, and queried in the *GOFLEX Data Store*.
F5.12	A *Key/Value API* for the storage & retrieval of bespoke unstructured data objects & time-series to/from the *Data Store*.
F5.13	An *ECAST API* to interface with the external ECAST energy forecast model ranking service. Metering and weather data, as well as forecast model and predictor data are passed to the service, and a ranking of forecasting model performance is retrieved.

F5.14	A *Transmission API* to interface to the DOMS module. Two points on the energy grid are passed to the service, which delegates the calculation to DOMS. The fractional transmission losses are returned.

6 Non-functional requirements

Now follows a list and brief description of the non-functional requirements for Work Package 5.

Table 7 - WP5 Non-functional Requirements

Requirement Number	Requirement Description
NF5.1	Incremental backup and recoverability of data store described in F5.2.
NF5.2	The data store described in F5.2 must scale to accommodate large amounts of data, and large volumes of granular or complex queries.
NF5.3	F5.3 supports JSON encoded text (Unicode) messages – not binary. Location coordinates are in wgs1984 (lat/long), measurements are in metric.
NF5.4	The *IoT Data Ingestion Service* described in F5.3 requires load balancing to ensure that all inbound data is consumed and persisted.

7 Architectural considerations / assumptions

It is assumed that:

- Demonstration sites must design and implement their own data ingestion clients to anonymize and publish real-time and historical metering data to the *IoT Data Ingestion Service.* All time-stamps emanating from these clients must be UTC.
- These clients must be hosted on a platform capable of reaching the *Cloud-based Service Platform* (Internet). For example, appropriate firewall rules must be in place to allow this data communication.
- All data anonymizations performed by these clients are assumed to be appropriate for their jurisdiction and in accordance with general EU guidelines.
- It is assumed that grid network topology information is available and in a format that is reasonably easy to consume.

8 Implementation Plan

Requirements outlined above will be implemented in a phased project. We note here the prioritization of requirements through the project phases

8.1 Prototype

In this first phase, prototypes of the core components of the *Cloud-based Service Platform* will be implemented. The *GOFLEX Data Store* (F5.2) implementing the *GOFLEX Data Model* (F5.1), *IoT Data Ingestion Service* (F5.2), and *Weather Forecast & Observation Ingestion Service* (F5.5) will be developed and implemented.

The *Energy Forecasting Service* (F5.8) will be implemented and validated with historical data from the demonstration sites in Cyprus, Switzerland, and Germany.

A basic implementation of the APIs connecting these core components will be implemented and tested – they are the *Metering API* (F5.4), the *Weather Forecast & Observation API* (F5.6), the *Energy Forecasting Model API* (F5.7), and the *Energy Forecast API* (F5.9).

Security will be considered from the beginning, and appropriate security components will be amongst the earliest features available.

8.2 Full Version

In this second phase, the *IoT Data Ingestion Service* (F5.2) and *Weather Forecast & Observation Ingestion Service* (F5.5) will be validated with real-time data feeds. These components will be enhanced and their performance tuned based on observation of their real-world behaviour.

The performance of the *Energy Forecasting Service* (F5.8) will be ascertained with real-time data and its performance benchmarked against that observed during the prototyping phase to ensure equal or superior forecasting results.

The APIs implemented in the prototyping phase will be further functionally enhanced for use by components external to the platform.

In addition, the following API interfaces will be implemented in this phase – the *Weather Forecasting Model API* (F5.7), the *Network Topology API* (F5.10), the *Energy Price API* (F5.11), and the *Key/Value API* (F5.12).

All API interface functionality will be validated, and any unforeseen requirements implemented.

8.3 Final Version

Due to the heterogeneous nature of the number & type of data sources in place at the demonstration sites in Cyprus, Switzerland, & Germany, demonstration site specific functionality will be identified and implemented in this final phase.

Also, due to potential variations in the quality and availability of metering sensor data, additional site-specific performance tuning of the energy forecasting models implemented in the *Energy Forecasting Service* (F5.8) will be undertaken.

Finally, work package key performance indicators will be measured and performance enhancements applied to the *Cloud-based Service Platform* to ensure that they are achieved.

8.4 Performance Indicators

A subset of KPIs defined in the GOFLEX proposal will be considered in the context of the cloud service platform. These KPIs are summarized as follows:

- Indicator 2.1.1.1c: "**Demonstrated solutions have the potential to be scaled (if needed) and replicated**"

The cloud service platform provides a scalable service, ensuring that the system will scale to support an increasing number of potential prosumers. As each demonstration site is required to have a dedicated instance of the cloud service platform, replicability will be a core goal in the implementation phase.

In addition, the underlying inter-component communication technologies rely on the use of standard, ubiquitous technologies (e.g. MQTT, AMQP, REST etc.), all backed by open-source or licence free implementations.

- Indicator 2.1.1.2b: "**Validated contributions for improved stability and flexibility in the distribution grid, avoid congestion; enabling near real-time pan European energy balancing market**"

The cloud service platform provides a scalable service, ensuring that the system will scale to support an increasing number of queries from dependent GOFLEX sub-systems (e.g. ATP), staying within our latency targets.

The cloud service platform also provides an interface to one or more external weather forecasting systems. These weather forecasting systems will be queried at a fine-grained resolution, leading to enhanced accuracy of the energy forecasting service.

9 Conclusion

This document presents the D5.1 deliverable of WP5. A description of the required components that compose the cloud-based service platform are discussed. In addition, all external component interactions are described from a functional and data perspective. APIs to meet these component interactions are proposed. Dependencies on other work packages are highlighted and areas where contributions from WP5 may assist other work packages are described (e.g. sample data ingestion client software). Therefore, this document and the requirements gathering process underpinning it provide a strong basis for progression.