

# GXF – Microgrids integration

alliander

# Application architecture MGP

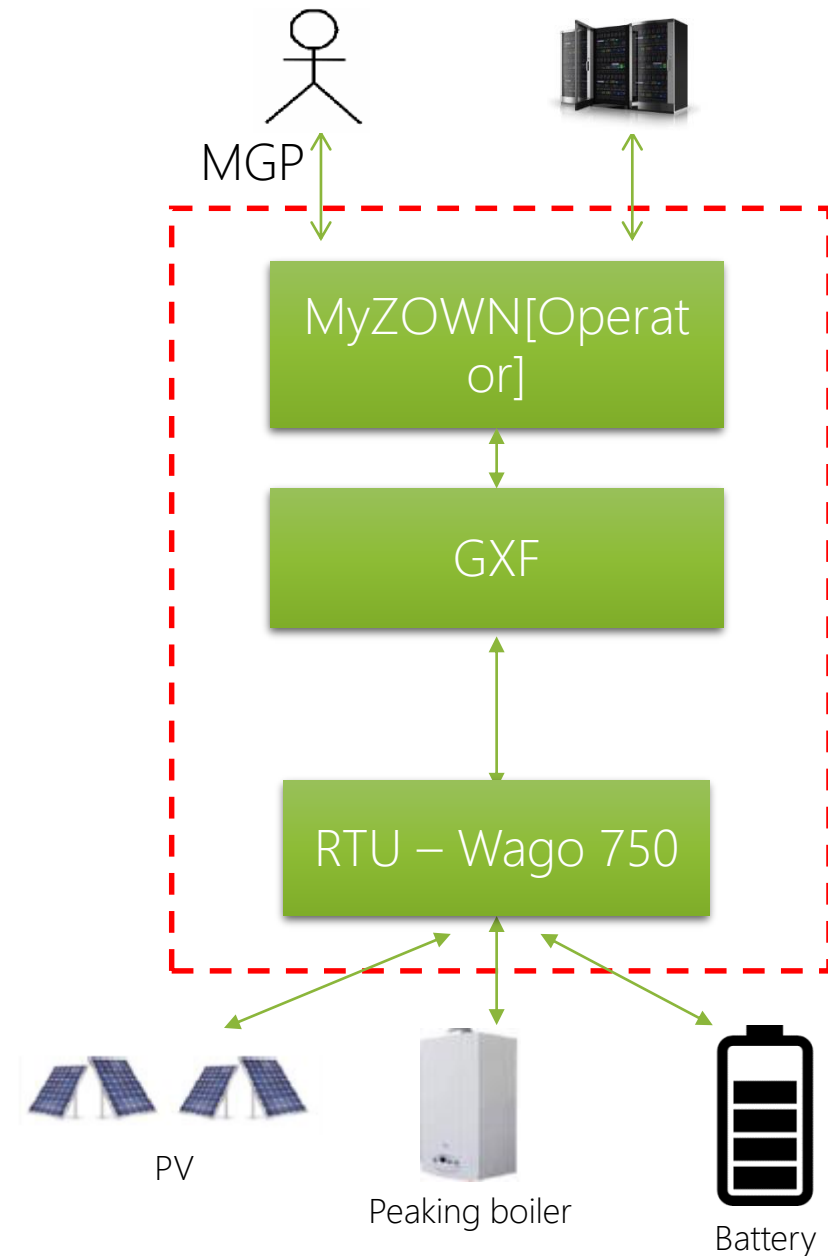
The Microgrid Platform (MGP) is a system to monitor, control and manage a microgrid.

A microgrid can contain multiple devices, as for example:

- PV installation
- Wind turbine
- Battery
- Peaking Boiler
- Diesel generator
- Grid connection
- Load
- Etc.

The diagram on the right provides an overview of the high-level architecture of Mircogrid platform.

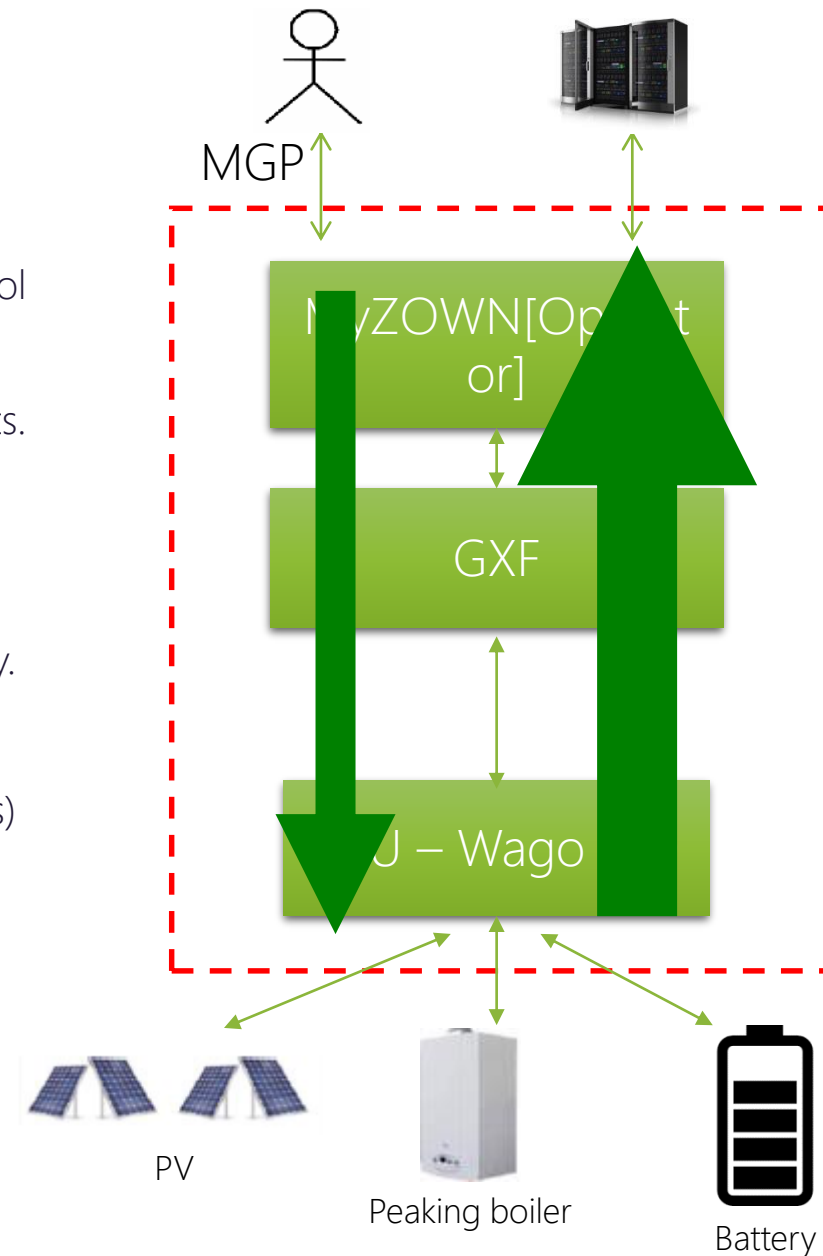
In the following slides describes this architecture in more detail.



# Control- and monitoring signals

The MGP communicates with devices to obtain reads, events and can control the devices remotely.

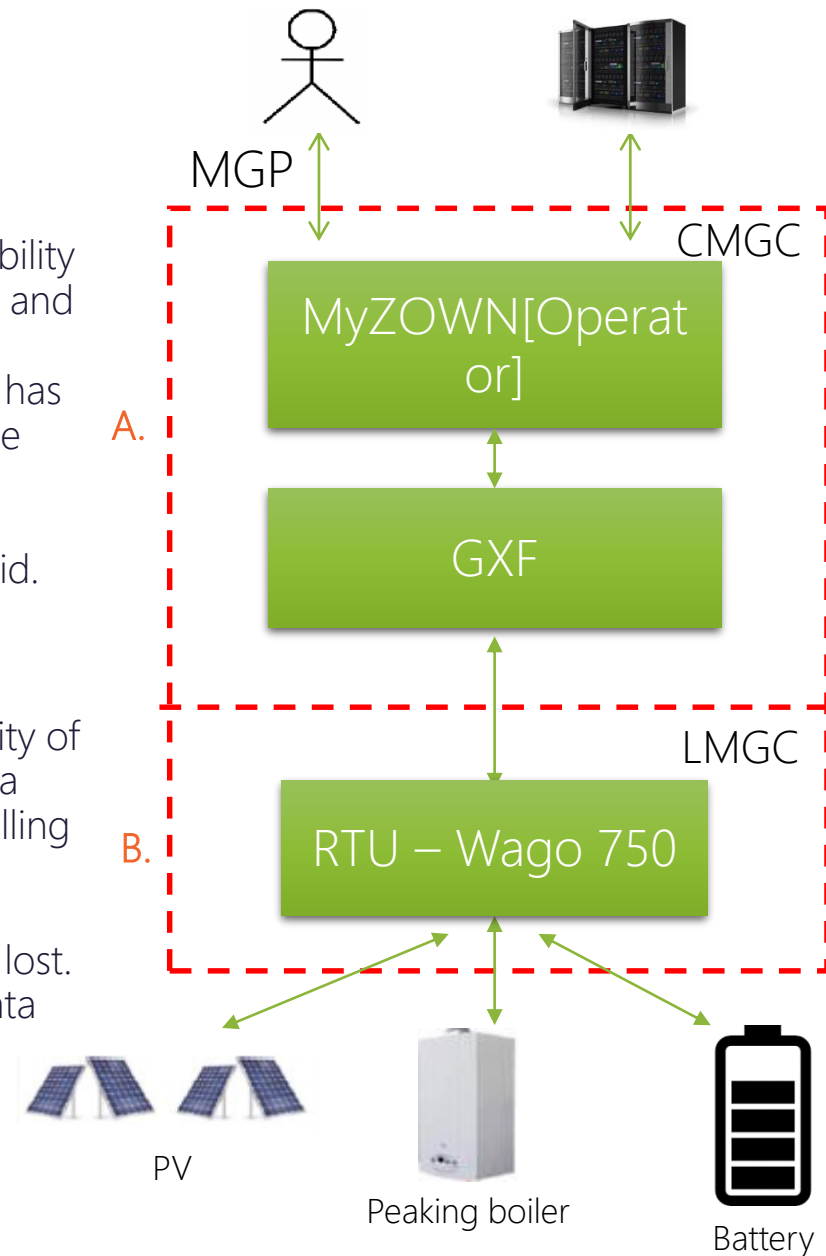
- A. **Control signals:** Control-signals are signals to control a device. This control-signal is initiated by Stream. For example: to disable loading of the battery. A control-signal can contain parameters, like setpoints. A setpoint is used by a rule (more details about rules in a few pages).
- B. **Monitoring-signals:** Monitoring-signals are signals which are subscribed from Stream and return data about the device, for example: the load of a battery. The monitoring signals are pushed by the RTU to GXF to MyZOWN[Operator] on:
  - Time-based event (for example every 5 minutes)
  - Change event (for example a setpoint is changed)



# Central versus local control

The MGP consists of two parts:

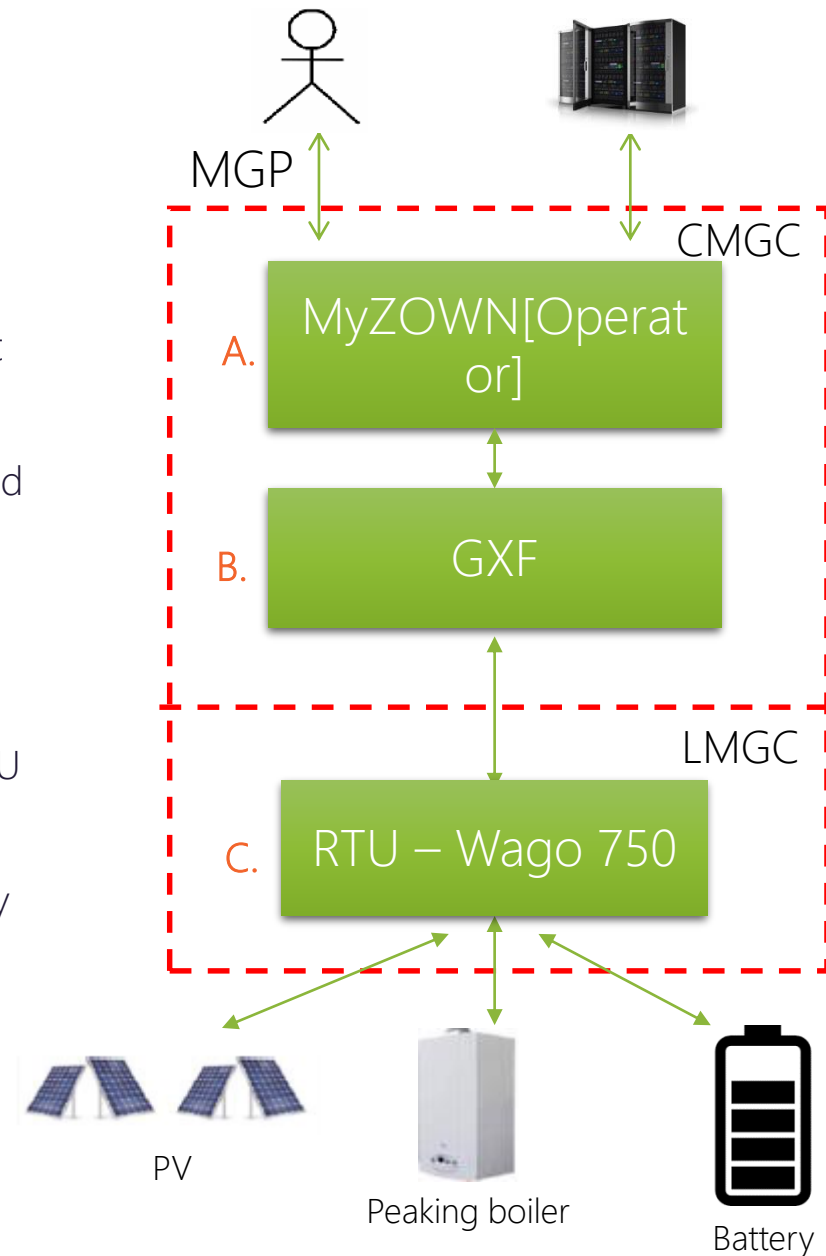
- A. **Central Micro Grid Controller (CMGC)** The responsibility of the CMGC is to expose services for the end-user and convert the needs of the end-user to control, monitoring and manage the microgrid. The CMGC has an response time of several seconds to minutes. The CMGC contains services which can for instance optimize the usage of the micro grid, or services to predict maintenance on the devices in the micro grid. But, also services for billing are provided.
- B. **Local Micro Grid Controller (LMGC)** The responsibility of the LMGC is to balance the supply and demand of a commodity, like energy, in the micro grid by controlling and monitoring devices. The LMGC has an almost instantaneously response time and will be self sustainable when the connection with the CMGC is lost. There is a connection with the CMGC to transfer data gathered in the LMGC.



# Component description

The MGP consists of three components:

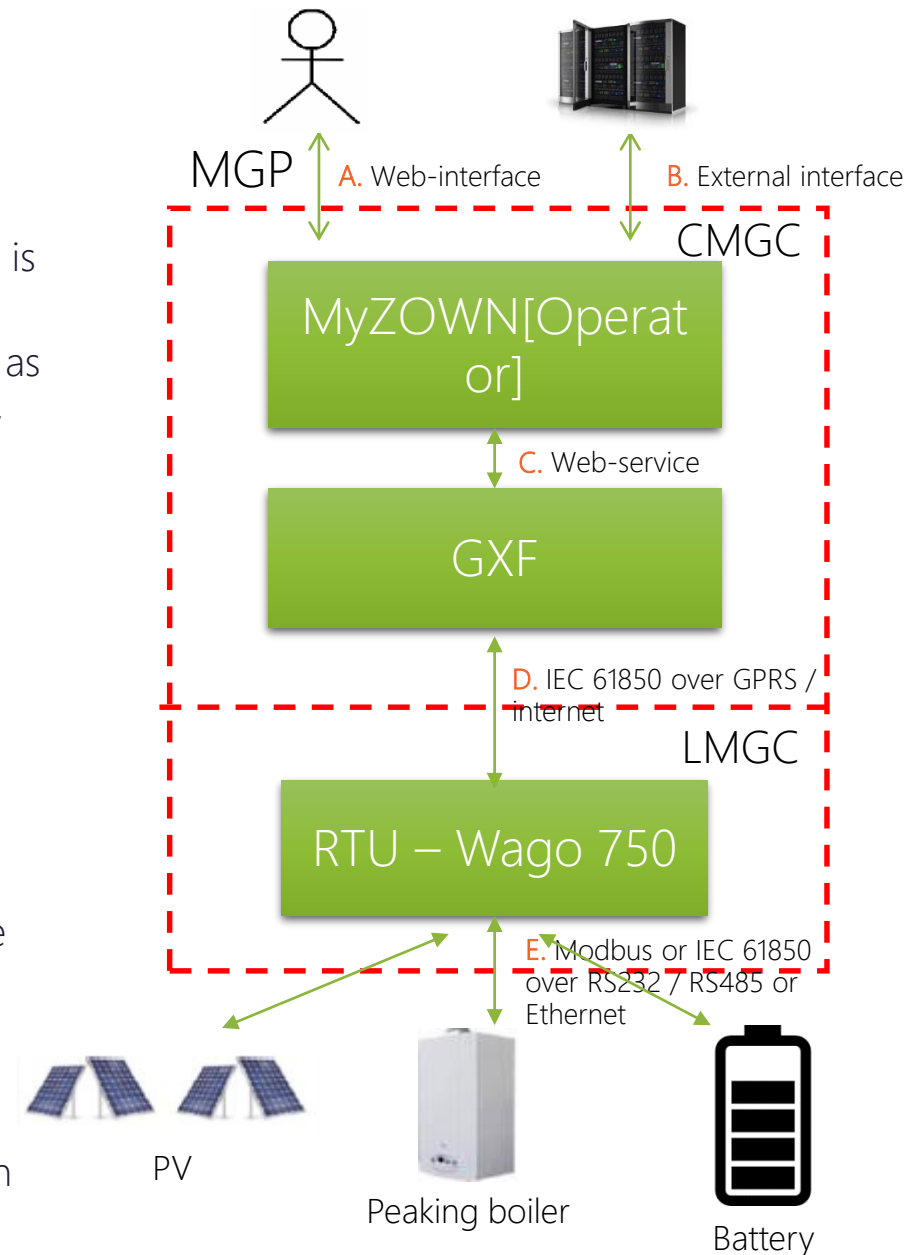
- A. **MyZOWN[Operator]**: top component which serves end-user services like insights of the microgrid and executes also high level tasks, like handling forecast and optimizing the micro grid. Billing / contracts and settlement is also part of this component. This component also gathers all data from the micro grid which is used for analytics, forecasts and optimization of the micro grid.
- B. **GXF**: middleware component which is used to have an abstraction of the devices and the separation between the LMGC and the CMGC. GXF is responsible to maintain the connection with the RTU in the micro grid. Abstraction of topology and device configuration at north-bound of GXF makes it possible to connect external services and not only Stream.
- C. **RTU**: Gateway or tele-controller (OT) which is responsible to maintain the balance in the micro grid by executing rules and controlling devices.



# Interfaces

Five interfaces are defined in the MGP:

- A. **Web-interface with the end-user:** The web-interface is provided for different users with different roles.
- B. **External interface:** The external interface is provided as external service to external parties. In the beginning, this will not be used (or other parties can connect directly to GXF).
- C. **Web-services:** The GXF interface is based on asynchronous web-services. These web-services provide a connection to MyZOWN[Operator] and makes it possible for other applications to connect. See [Appendix B](#) for a detailed description of the available webservices.
- D. **IEC 61850:** service which communicates with (RTU) devices in the local micro grid. This service needs to be as independent as possible to 'loosely' couple the RTU to GXF. See [Appendix C](#) for a detailed description of IEC61850.
- E. **IEC61850 or Modbus:** Most of the time Modbus on Ethernet is used to communicate with the devices in the micro grid. In the future this can be IEC 61850 on Ethernet.

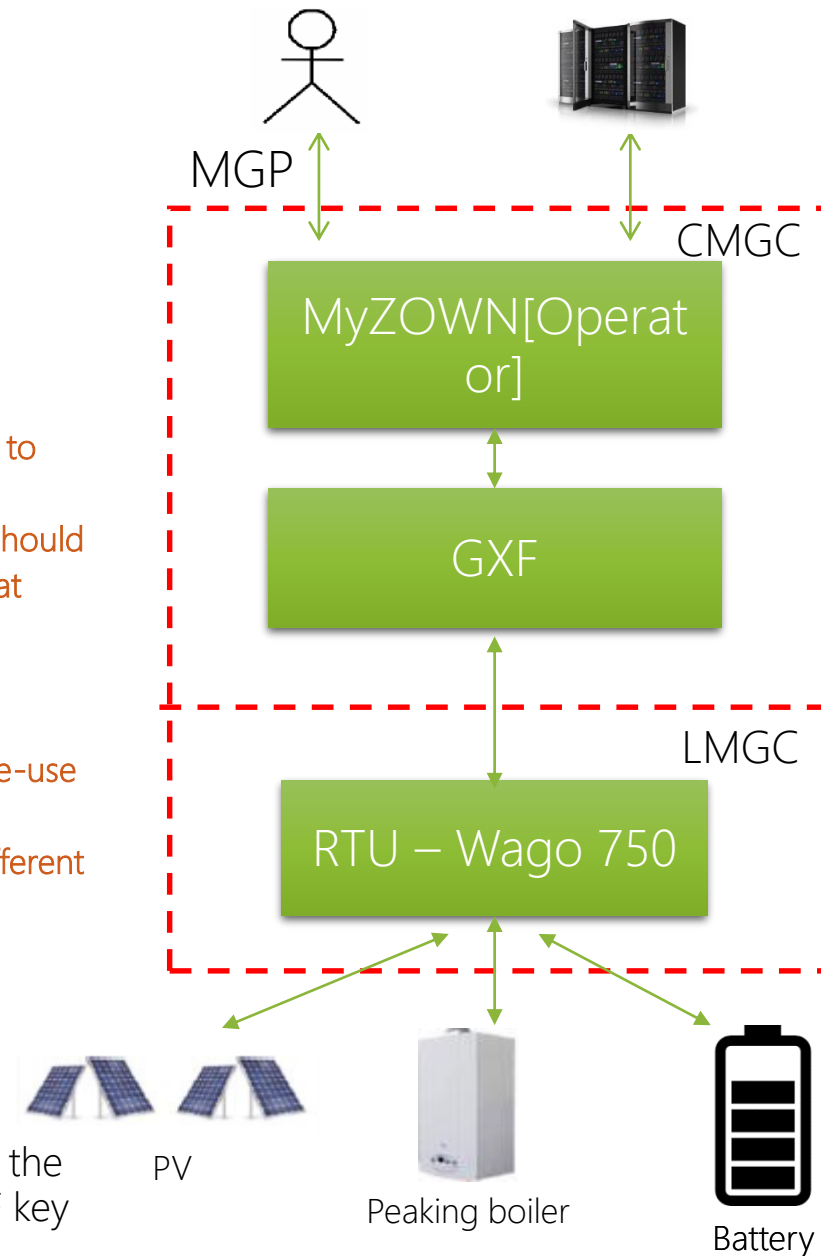


# Key Architectural design principles\*

The eight key architecture design principles are:

- A. Safe & reliable. The security and reliability monitored and controlled microgrid should be guaranteed at all times.
- B. Interoperability. The MGP is hardware independent and communicates with devices using open market standards.
- C. Multi-commodity. The MGP must be able to accommodate multiple commodities (power, gas, heat) and it must be able to optimize them.
- D. Data ownership. Participants in microgrids (= data owners) should at all times be able to control who can use their data for what purpose.
- E. Fit-for-purpose. Customers of the MGP only pay for the functionality they need.
- F. Re-usability (& multi-tenant). ZOWN aims to maximize the re-use of functionality to keep cost down (70-30 ambition).
- G. Best-of-Breed. The MGP must be able to easily integrate different applications from different vendors.
- H. Future proof. The MGP is able to grow with the number of customers (in terms of functionality, cost and performance).

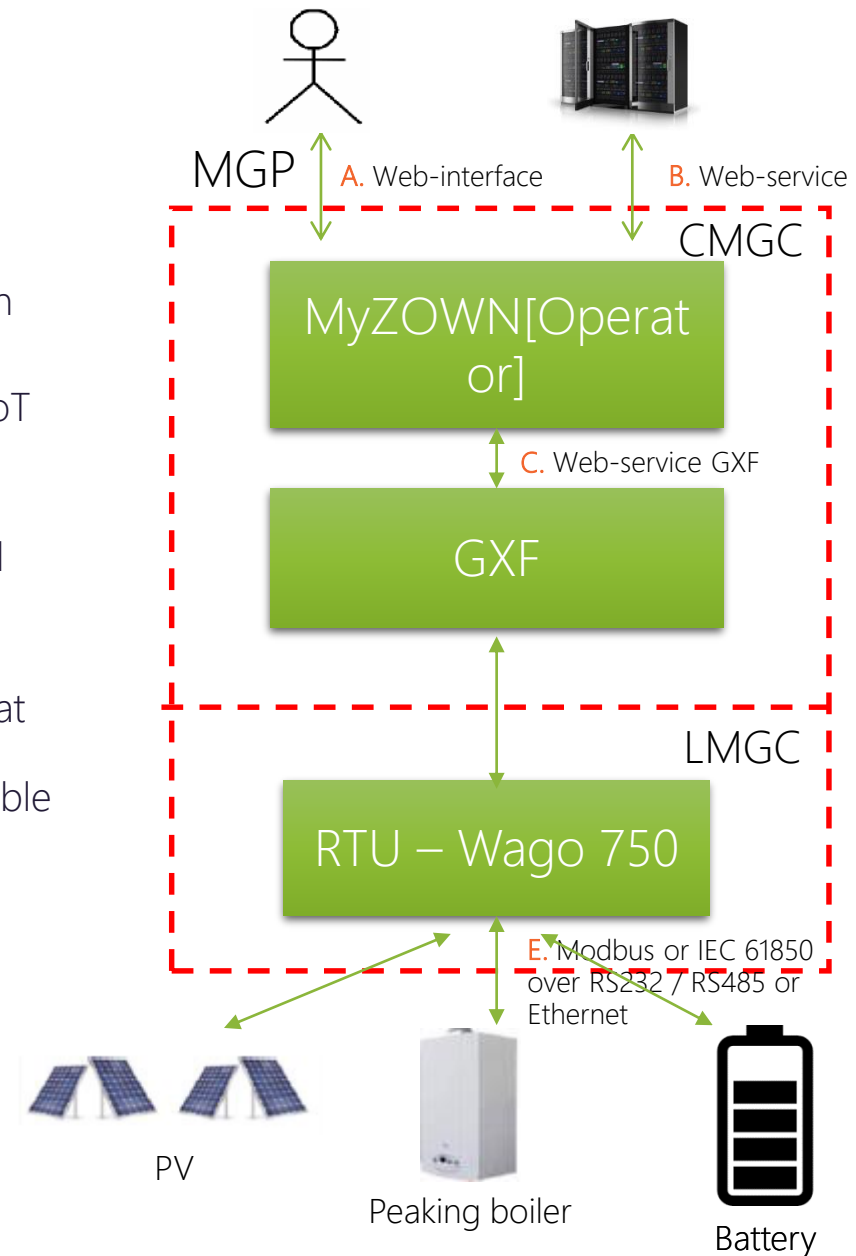
\* These are the key architectural design principles for the ZOWN MGP and do not map one-to-one to the GXF key architectural design principles.



# Key Architectural decisions

The four key architecture decisions:

- A. **Why MyZOWN[Operator]?** MyZOWN[Operator] provides a good starting point for the services which are need for a MVP.
- B. **Why GXF as middle-tier?** GXF is open and generic IoT platform, which provides security, scalability, availability, vendor independency and abstraction.
- C. **Why IEC 61850?** IEC 61850 has context and is (or will be) implemented by many manufacturers. Also has GXF already IEC 61850 adapters which can be used.
- D. **Why choose for Wago 750?** Wago 750 is also used at other places in Alliander. Also ICT Automatisering confirms the decision. Wago specialist say, it is possible to implemented the wanted functionality.





# Appendix B: Microgrid webservices

For the interface between Stream and GXF the following three webservices are defined:

- A. GetData. The GetData is a generic webservice to get device data of a RTU, PV, Battery or any other Microgrid device triggered by the RTU or triggered by MyZOWN[Operator].
- B. SetData. The SetData is a generic webservice to set device variable of a RTU, PV, Battery or any other Microgrid device triggered by MyZOWN[Operator].
- C. GetTopology (proposed). The SetTopology is a generic webservice to get the topology of a microgrid.

For the latest version of the WSDL of these webservices see [github.com](https://github.com)

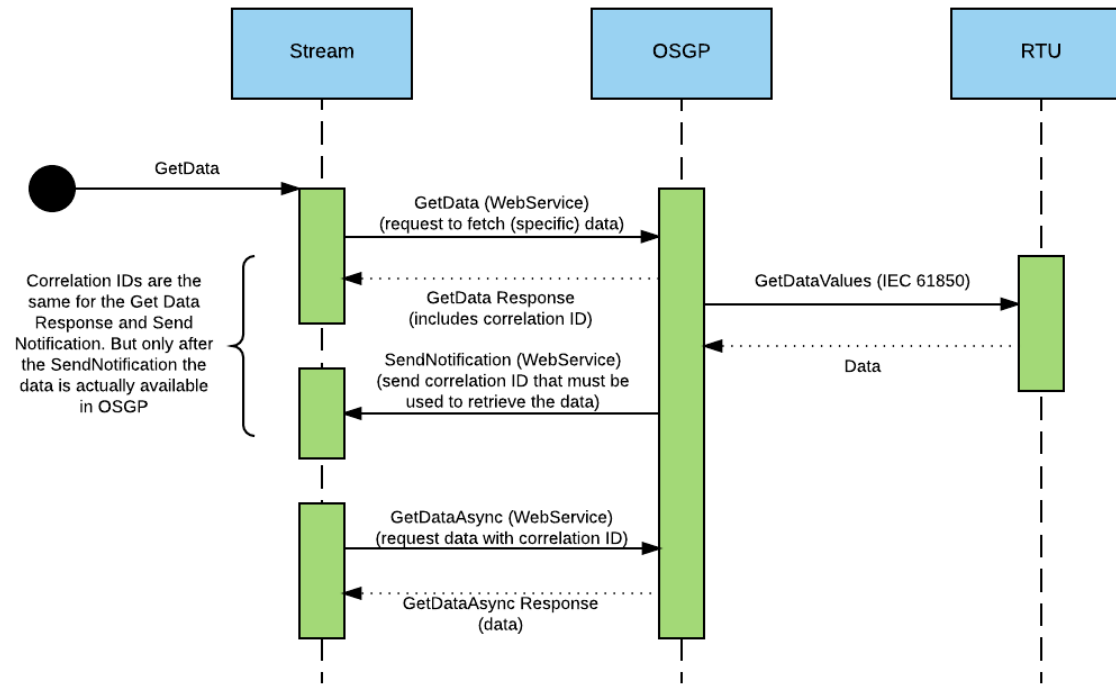
# Appendix B: Microgrid webservices

GetData webservice (when triggered by Stream):

The GetData is a generic webservice to get device data of a RTU, PV, Battery or any other Microgrid device triggered by Stream. The device data is requested by Stream.

The following sequence diagram describes the asynchronous communication between Stream, GXF en RTU when the Stream is request data.

For the latest and update version of the WSDL of this webservices see [github.com](https://github.com)



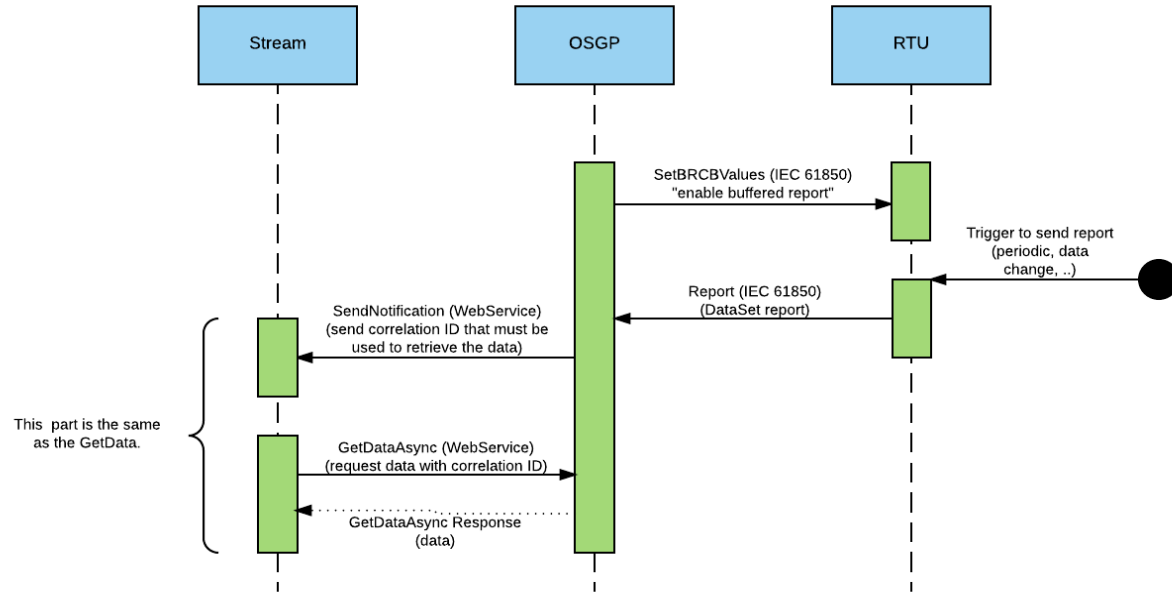
# Appendix B: Microgrid webservices

GetData webservice (when triggered by RTU)

The GetData is a generic webservice to get device data of a RTU, PV, Battery or any other Microgrid device triggered by the RTU. The device data is pushed from the RTU to GXF through a report. A report contains one or more data values.

The following sequence diagram describes the asynchronous communication between Stream, GXF en RTU when the RTU is pushing an report.

For the latest and update version of the WSDL of this webservices see [github.com](https://github.com)



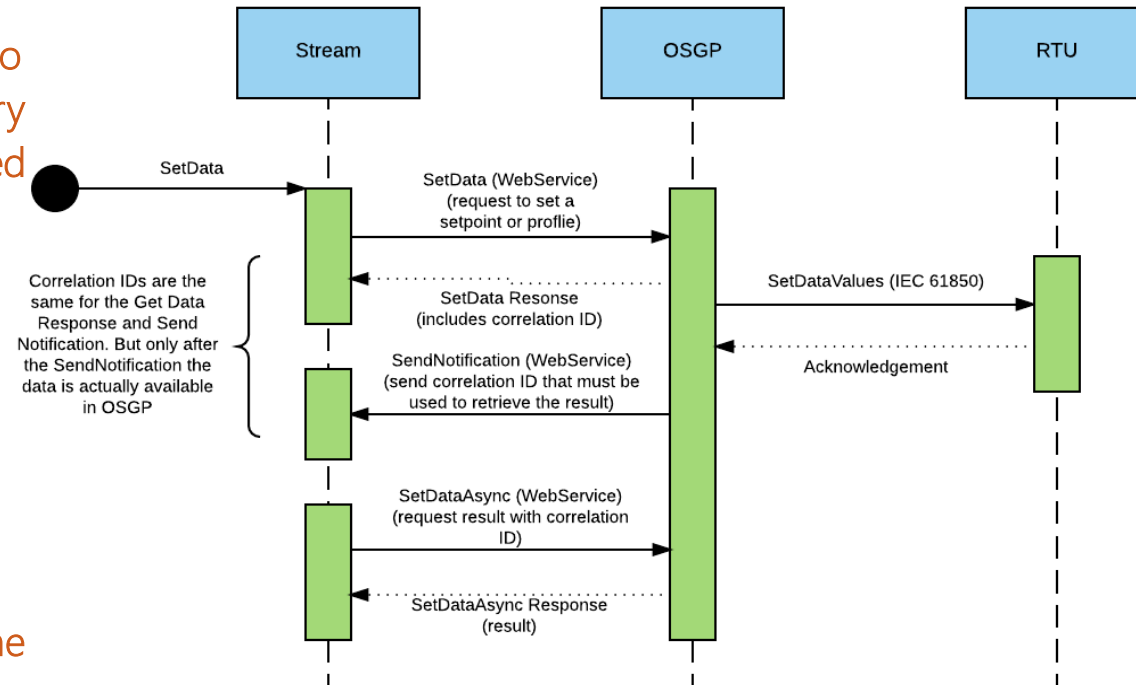
# Appendix B: Microgrid webservices

SetData webservice:

The SetData is a generic webservice to set device variable of a RTU, PV, Battery or any other Microgrid device triggered by Stream.

The following sequence diagram describes the asynchronous communication between Stream, GXF en RTU when the Stream is setting a setpoint or profile.

For the latest and update version of the WSDL of this webservices see [github.com](https://github.com)



# Appendix B: Microgrid webservices

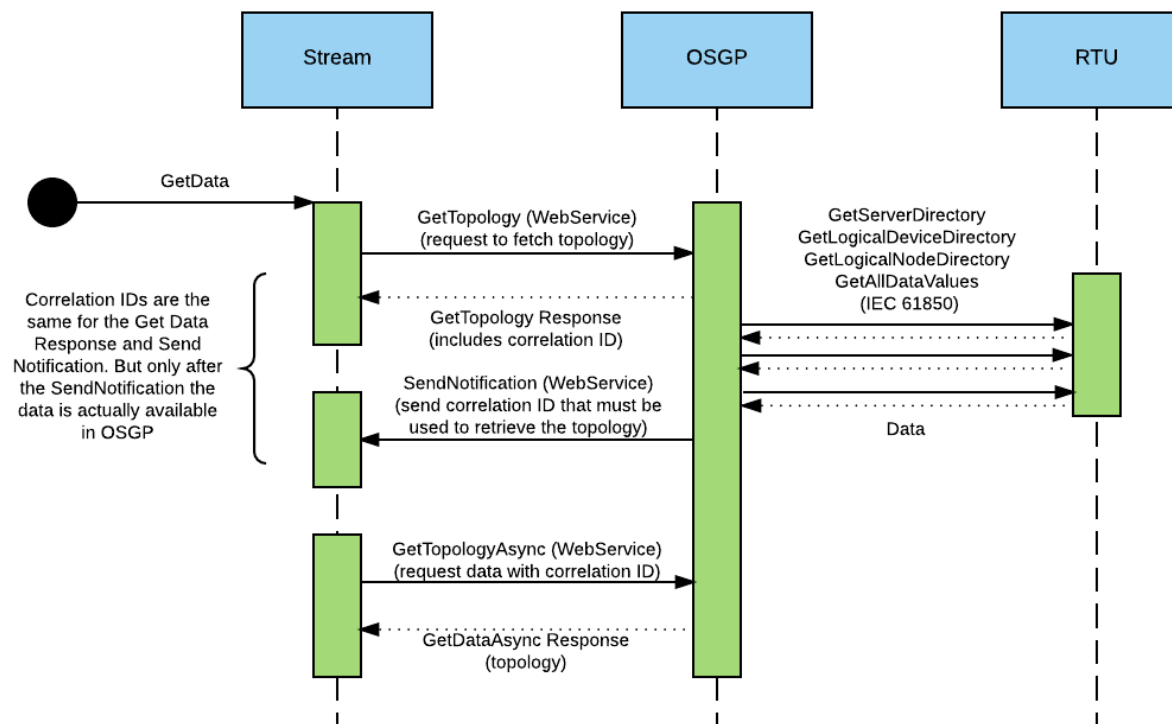
GetTopology webservice

(proposed):

The SetTopology is a generic webservice to get the topology of a microgrid. The topology describes what devices are in a microgrid. For example: 1 RTU, 2 PV' s and 2 Batteries.

The following sequence diagram describes the asynchronous communication between Stream, GXF en RTU when the Stream is getting the topology.

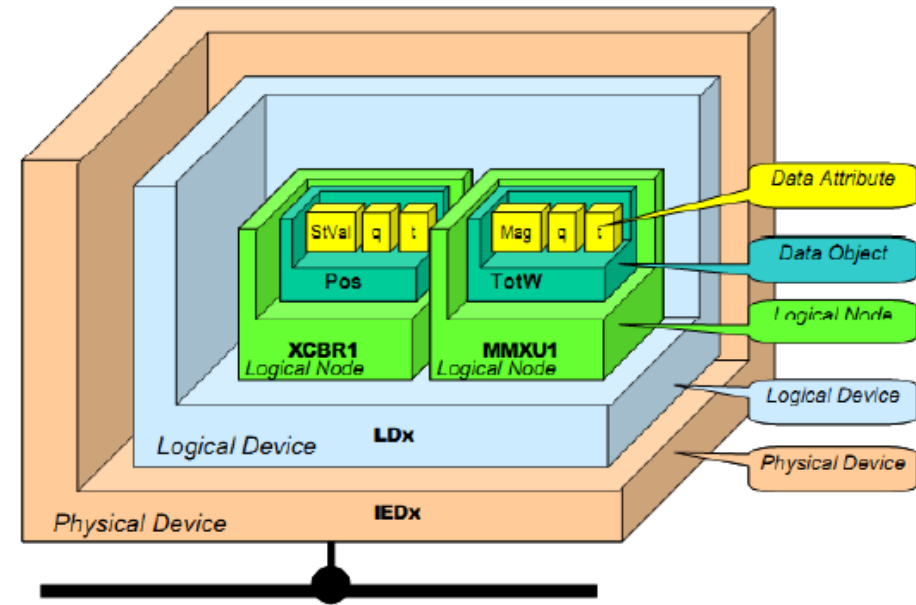
For the latest and update version of the WSDL of this webservices see [github.com](https://github.com)



# Appendix C: IEC 61850

For the interface between GXF and RTU is based on the protocol IEC 61850. IEC 61850 is communication standard for electrical substation automation systems.

The IEC 61850 model is that a physical device is represented by a logical device and the logical device consists of one or more logical nodes. Each node has several data objects (descriptions, status info, controls, measures, settings) which have various data attributes.



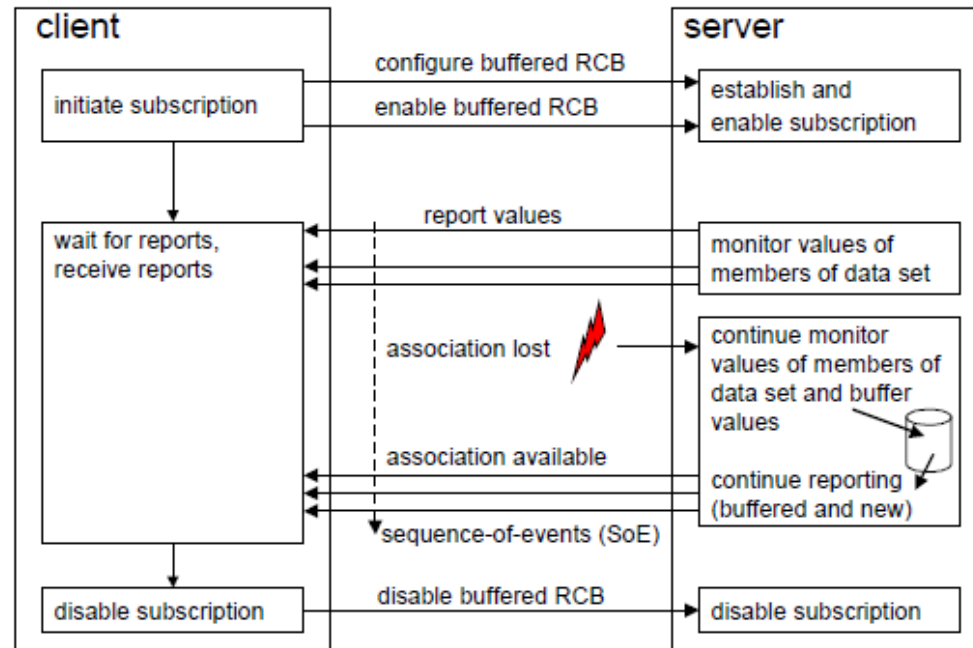
# Appendix C: IEC 61850

## Reports

In order for the RTU to push device data to GXF, reports are used. The sending of reports is triggered by preconfigured events. Triggers can e.g. be if the value or quality of a variable in the monitored data set changes.

IEC 61850 distinguishes between buffered and unbuffered reporting. In unbuffered reporting events will not be logged and reported if the associated client for the unbuffered report control block is not connected. In the case of buffered reporting the events will be logged for a specific amount of time and sent later when the client is connected again.

In the MGP is chosen for buffered reporting.



IEC 1424/11