Modelling 2012 – Project Assignment

Instituto Superior Técnico,
Universidade Técnica de Lisboa, Portugal

Group A-18

Project Phase 4

66958, Cláudia Henriques, claudiafilipahenriques@ist.utl.pt
66964, Daniel Cardoso, daniel.cardoso@ist.utl.pt
66986, Francisco Raposo, francisco.afonso.raposo@ist.utl.pt
67043, Miguel Aragão, miguel.aragao@ist.utl.pt

Submission date: 20/05/2012

Table of Contents

Figure

Tables

Introduction

The artefacts were modelled having some assumptions under consideration. Otherwise there would be some obstacles that would be hard (or even impossible) to overcome. Here are just some of them. The other ones are specific to each artifact, so they are written alongside them.

Regarding the class and sequence diagrams for the MeteoNet software:

- Every data type that ends in "_type" are abstract data types to be implemented/chosen when implementing the MeteoNet software (using a specific programming language)
- The AWS class has some methods that behave the following way:
  - getAttributes(): attribute_type[]
    - Returns an array with the AWS attributes
  - getSensors(sensor_id_type): Sensor[]
    - Returns an array with the AWS' sensors that are of the same type as the sensor that has the same id passed as argument.
  - isWithinCircle(latitude_type, longitude_type, radius_type): boolean_type
    - Returns a true value if the AWS is within the circle with the radius passed as argument and center in the coordinates passed as arguments.
  - isSilentStation(timestamp_type, timestamp_type): boolean_type
    - Returns a true value if the AWS was silent in the time interval defined by the timestamps passed as arguments.
  - newSensor(sensor_id_type): void
    - Adds a new sensor with the sensor_id passed as argument.
- All arrays support the operations add(data_type), get(position_type) and concat(data_type[]) that add an element, get an element in the specified position and concatenate an array of elements.

Artefact #1 – Effort and Work-Breakdown

Number	Name	Phase 1	Phase 2	Phase 3	Phase 4	Total per student
66958	Cláudia Henriques	30	18	22	18	**78**
66964	Daniel Cardoso	30	18	22	18	**78**
66986	Francisco Raposo	30	18	22	18	**78**
67043	Miguel Aragão	30	18	22	18	**78**
	Total per phase	120	72	88	72	

Table 1. Effort per student.

Artefact	Page	66958	66964	66986	67043
1					**O,D**
2		**O,D**	**M**	**D**	
3		**M**	**O,D**	**d**	**D**
4		O,D	D		M
5			O,D	M	
6		M	D		O,D
7				O,D	D,M
8				M	O,D
9			M	O,D	
10				M,D	O,D
11			O,D,M		
12				O,D,M	
13		O,D,M			
14					O,D,M
15			O,D,M		
16			m		O,D,M
17		O,D,M		m	
18		m		O,D,M	
19			O,D,M		
20		O,D,M			
21		O,D,M			
22					O,D,M
23					O,D,M
24				O,D,M	
25			O,D,M	d	
26			O,D,M		
27				O,D,M	
28		O,D,M			
29		O, D			d

Table 2. Work-breakdown per student.

Artefact #2 – AWS context model

As we can see, the AWS has several stake holders involved.

The GPRS was considered as an enabling system because, generally thinking in the entire system, the main objective of the AWS is to send the data by GPRS (not concerning who will, or will not, receive the data).

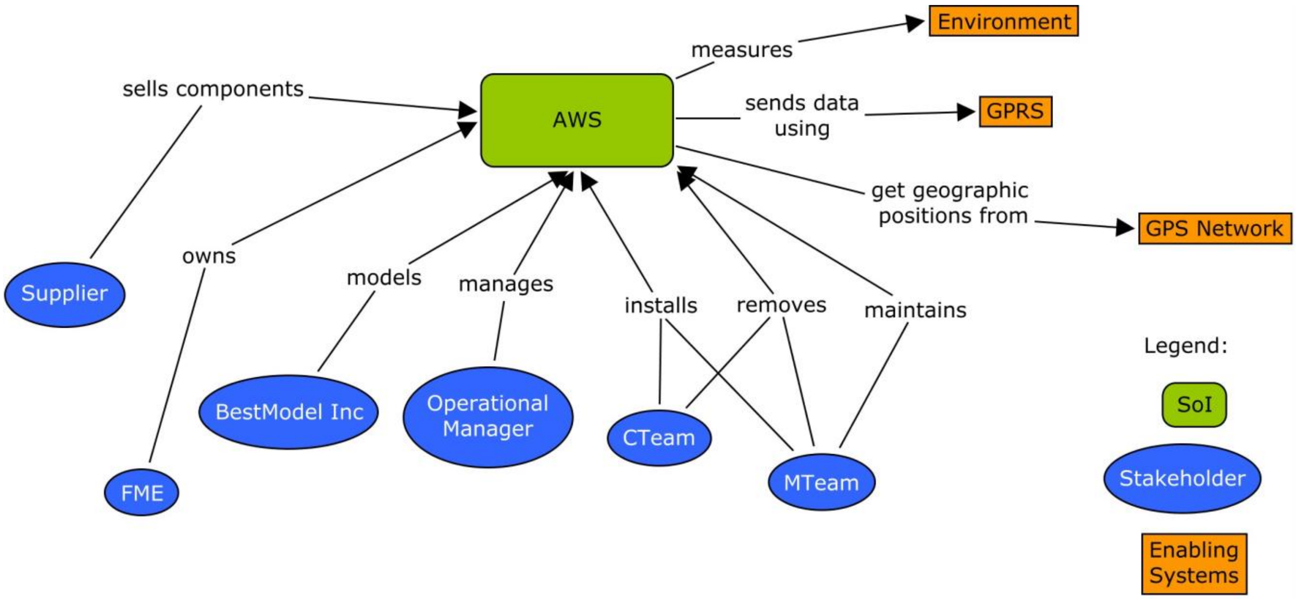

Figure 1 - AWS context model.

Artefact #3 – AWS requirements list

ID	Source	Type	Description
1	Line 39	NF	AWS must have a mast, which has a platform on top.
1.1	Line 51	NF	Mast can be 3, 10 or 30 meter tall.
1.1.1	Line 59	NF	The 10 meter masts shall have one group of tension cables.
1.1.2	Line 60	NF	The 30 meter masts shall have a minimum of two groups of tension cables.
1.2	Line 47	NF	Platform must support all the components.
1.3	Line 45	NF	The platform's socket is secured to the mast using six bolts.
2	Line 35	NF	AWS must have an enclosure.
2.1	Line 36	NF	Enclosure must have a power supply.
2.1.1	Line 143	NF	Power Supply must have a battery producing stable 12V DC to all the components.
2.1.2	Line 143	NF	Power Supply must have a regulator.
2.1.3	Line 144	NF	Power Supply must have an external power source.
2.2	Line 37	NF	Enclosure can have a data logger.
2.3	Line 61	NF	Enclosure shall have a sensor hub.
2.4	Line 64	NF	Enclosure shall have two circular cable routing openings.
2.4.1	Line 64	NF	Routing openings shall be covered with a weather-proof insulator.
2.4.2	Line 65	NF	The first opening shall be used to route all connector cables to and from the sensors.
2.4.3	Line 66	NF	The second opening shall be used to route all power supply.
2.4.4	Line 68	NF	Power cables and sensor cables can never be routed through the same routing opening.
2.5	Line 69	NF	The enclosure must be made of weather-proof fiberglass or stainless steel.
2.6	Line 98	NF	The enclosure shall have an electronic timer inside it, connected to each ADC
2.7	Line 110	NF	The enclosure shall have a Sensor Hub with exactly 16 input ports and 1 output port.
2.7.1	Line 111	NF	Each input port must accept the digital output of the ADC and have 4 switches..

ID	Source	Type	Description
2.7.2	Line 118	NF	Hub output port can be connected to data loggers.
2.8	Line 122	NF	An enclosure can have a maximum of 3 data loggers.
2.8.1	Line 123	NF	Data Logger shall have one input port and one USB port.
2.8.2	Line 127	NF	180 days of measurements history must be the minimum capacity of a Data Logger.
2.9	Line 152	NF	Enclosure holds the battery, the regulator, the rectifier and the transformer.
2.9.1	Line 145	NF	The battery must give energy to all the AWS components.
2.9.2	Line 146	F	Regulator recharges the battery.
2.9.3	Line 148	NF	Regulator can get its power from photovoltaic cell array.
2.9.4	Line 149	NF	Regulator can get its power from a wind turbine.
2.9.5	Line 150	NF	Regulator can get its power from a AC power grid.
2.9.6	Line 150	NF	12V transformer shall be connected to an AC power grid.
2.9.7	Line 151	NF	The enclosure can hold a photovoltaic Cell and a Wind Turbine, outside of the enclosure.
2.9.8	Line 153	NF	The battery has built-in a charge analyser that measures the remaining battery level.
2.9.8.1	Line 154	NF	When the charge analyser detects that the battery reached 5% of its capacity, it cuts power to all the AWS components.
2.9.8.2	Line 154	NF	When the charge analyser detects that the battery reached 10% of its capacity, it resumes power to all the AWS components.
2.10	Line 151	NF	The enclosure can hold a photovoltaic Cell outside of the enclosure.
2.11	Line 151	NF	The enclosure can hold a Wind Turbine outside of the enclosure.
2.12	Line 64	NF	The enclosure is attached to the platform with four bolts.
3	Lines 78, 35	NF	An AWS shall have a combination of meteorological sensors.
3.1	Line 84	F	A sensor can measure temperature, humidity, wind speed, wind direction, barometric pressure, precipitation, Ultraviolet radiation or location.
3.1.1	Line 105	NF	An anemometer must be installed with a wind vane and vice versa.
3.1.2	Line 107	NF	Rain Gauge must be installed with a Barometer.
3.1.3	Line 85	NF	The GPS sensor is the only sensor pack that is all provided as a unique component.
3.2	Line 79	NF	Each sensor shall be connected to one ADC (analogue-to-digital converter) located outside of the enclosure.
3.2.1	Line 87	NF	The output of each ADC is a 5 byte digital value.
3.2.2	Line 92	NF	The output of each ADC is connected to the sensor hub.
3.2.3	Line 93	F	Each ADC shall sample its sensor periodically.
3.2.4	Line 97	F	When the ADC receives the signal it shall sample the corresponding sensor and convert the electrical signal to a digital value.
3.2.5	Line 196	NF	Sensor packs must come from the same supplier.
3.3	Line 100	NF	Sensors for temperature and humidity must be installed.
3.4	Line 102	NF	AWS can have between 0 and 3 redundant sensors for measuring temperature, humidity, barometric pressure, precipitation, wind direction and wind speed.
3.5	Line 103	NF	Sensors for measuring Ultraviolet radiation and Location (latitude, longitude, altitude) can't be made redundant.
3.6	Line 108	NF	All sensors shall be housed outside the enclosure.
3.7	Line 234	NF	Sensors can be added and removed to the AWS.
4	Line 38	NF	AWS has a communication device.
4.1	Line 134	F	Communication Device shall send information.
4.2	Line 137	NF	Communication Device must have one input port connected to the sensor hub.
4.3	Line 138	NF	Communication Device shall be configured.
4.3.1	Line 138	NF	Communication Device shall be configured during installation.
4.4	Line 141	NF	Communication Device only communicates with one BWS.
5	Lines 94, 95	NF	Frequency of communication can be changed in the AWS.
6	Line 163	NF	One AWS can only be associated with one BWS (hence one WN).

Table 3 - AWS requirements list.

ID	Source	Type	Description
7	Line 4, 5	NF	An AWS has a power switch, which must be accessible from the outside of the enclosure, to control the power provided by the battery.
7.1	Line 6, 7	F	The switch must be turned off before any human invasive action on the AWS.
7.2	Line	F	The switch can turn off or on the AWS.

Table 4 - Additional AWS requirements list.

Artefact #4 – AWS Use Cases

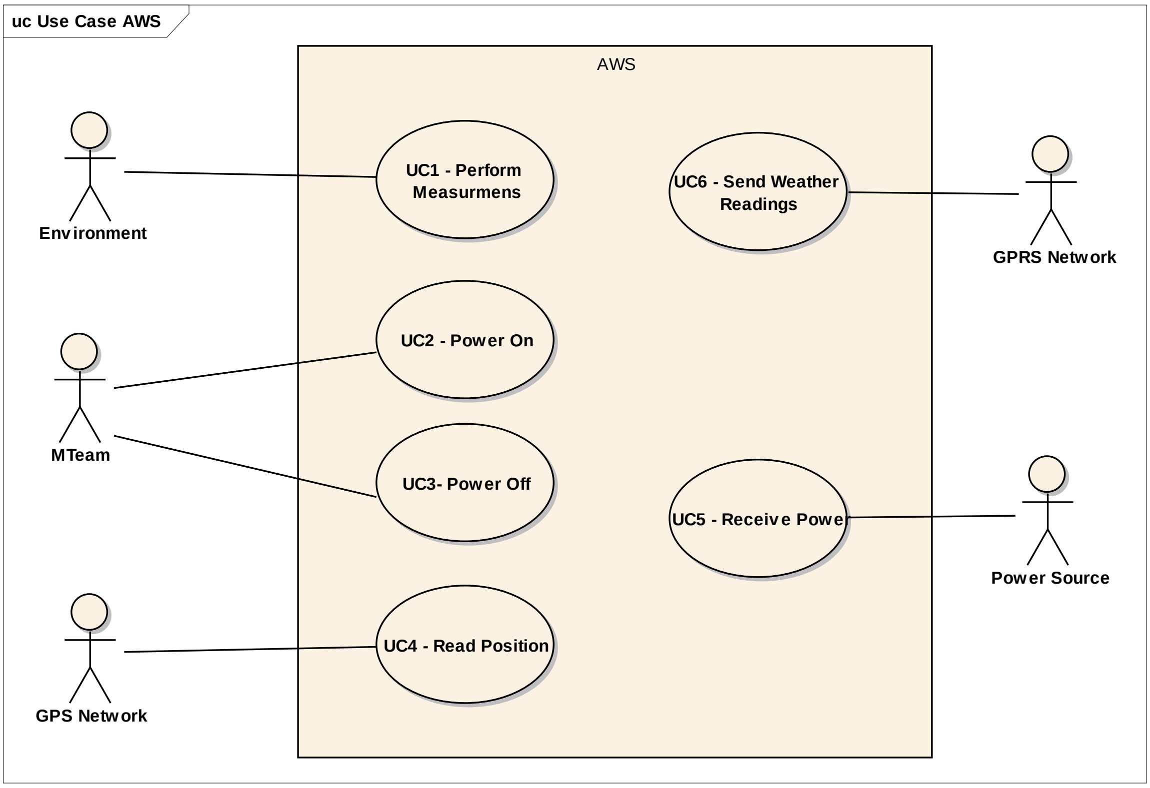

Figure 2 - Use Case diagram for the AWS system

Artefact #5 – AWS Use Case Scenarios

ID and Name	UC1 – Perform Measurements
Summary	A measurement of the weather conditions is perform and converted until it's ready to send and store.
Rationale	Sensor can be a thermistor, Resistive humidity sensor, Anemometer, Wind Vane, Barometer, Rain Gauge, Photometric UV sensor and a GPS.
Actors	Environment.
Pre-conditions	- AWS must be operational.
Primary Scenario	1. Sensor makes measurements from the current weather. 2. The measurements are sent for the communication device.
Post-conditions	- Readings plus Sensor ID in the communication device.
Alternative Scenario "Temperature"	1. The sensor which is performing the measurements is a thermistor.
Alternative Scenario "Humidity"	1. The sensor which is performing the measurements is a resistive humidity sensor.
Alternative Scenario "Wind Speed"	1. The sensor which is performing the measurements is a anemometer.

Alternative Scenario "Wind Direction"	1. The sensor which is performing the measurements is a wind vane.
Alternative Scenario "Barometric Pressure"	1. The sensor which is performing the measurements is a barometer.
Alternative Scenario "Precipitation"	1. The sensor which is performing the measurements is a rain gauge.
Alternative Scenario "Ultraviolet Radiation"	1. The sensor which is performing the measurements is a photometric UV sensor.
Alternative Scenario "Location"	1. The sensor which is performing the measurements is a GPS.
Alternative Scenario "Store Data"	2. The measurements are both sent to the communication device and stored on memory, case exists a data logger on the AWS which makes the store possible.

Table 5 - Scenarios of the AWS UC1- Perform Measurements

ID and Name	UC2 – Power ON
Summary	The ON/OFF button is pressed to turn on the AWS.
Rationale	The AWS is off, so a member of the Mteam press the button to the turn the station on.
Actors	Mteam
Pre-conditions	- AWS has to be off.
Primary Scenario	1. One member from the Mteam press the ON/OFF button. 2. The AWS turns on.
Post-conditions	- The AWS is turned on.

Table 6- Scenarios of the AWS UC2 –Power On.

ID and Name	UC3 – Power OFF
Summary	The ON/OFF button is pressed to turn off the AWS.
Rationale	The AWS is on, so a member of the Mteam press the button to the turn the station off.
Actors	Mteam
Pre-conditions	- AWS has to be on.
Primary Scenario	1. One member from the Mteam press the ON/OFF button. 2. The AWS turns off.
Post-conditions	- The AWS is turned off.

Table 7 - Scenarios of the AWS UC3 – Power Off.

ID and Name	UC4 – Read Position
Summary	The AWS gets its location form the GPS Network.
Rationale	Each AWS has a GPS sensor to provide WGS-84 coordinates of its location.
Actors	GPS Network
Pre-conditions	- AWS has to be turned on.
Primary Scenario	1. The GPS sensor gets the information from the global GPS network. 2. The sensor outputs a digital output with the position data. 3. The data packet is sent for the communication device.
Post-conditions	- The sensor output shows de right value of the right measure (having a certain delay in consideration); - The communication device has the data packet with the position information (with no errors).

Table 8 - Scenarios of the AWS UC4 – Read Position.

ID and Name	UC5 – Receive Power
Summary	The battery receives the energy from the external font.
Rationale	Each AWS gets its energy from an external source.
Actors	Power Source.
Pre-conditions	-The remaining battery level is not 100%; -At least, one of possible power sources has energy to provide to the battery.
Primary Scenario	1. A power source provides energy. 2. The energy is sent from the power source. 3. The battery receives the energy.
Post-conditions	Battery level of energy increased.
Alternative Scenario "From AC Grid"	1. The power source is an AC power grid.
Alternative Scenario "From Solar Energy"	1. The power source is a photovoltaic cell array.
Alternative Scenario "From Wind Energy"	1. The power source is a wind turbine.
Alternative Scenario "Low level of energy"	2. When the battery reaches the energy level of 5%, the battery is turned off, but continues being charged. (The AWS is turned on, when the battery energy level reaches 10% again)

Table 9 - Scenarios of the AWS UC5 – Receive Power.

ID and Name	UC6 - Send Weather Readings
Summary	Environment readings are sent as data packets over a GPRS network
Rationale	---
Actors	GPRS Network
Pre-conditions	- There are readings in the communication device to be sent
Primary Scenario	1. The communication device handles all the GPRS connection, authentication, encryption. 2. The data packets are sent over a GPRS mobile network
Post-conditions	- The data packets were sent.

Table 10 - Scenarios of the AWS UC6 – Send Weather Readings.

Artefact #6 – BWS context model

Similarly to the AWS <-> GPRS Network interaction, the BWS is only focused on the information coming from the GPRS system (not concerning on who send the data there).

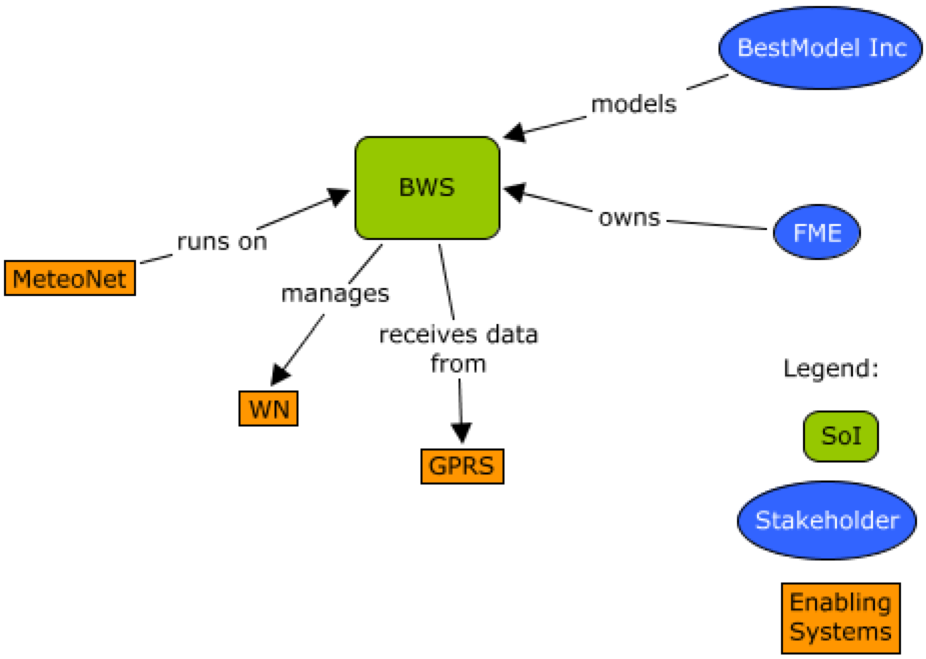

Figure 3 - BWS context model.

Below is shown the Context Model of a Weather Network.

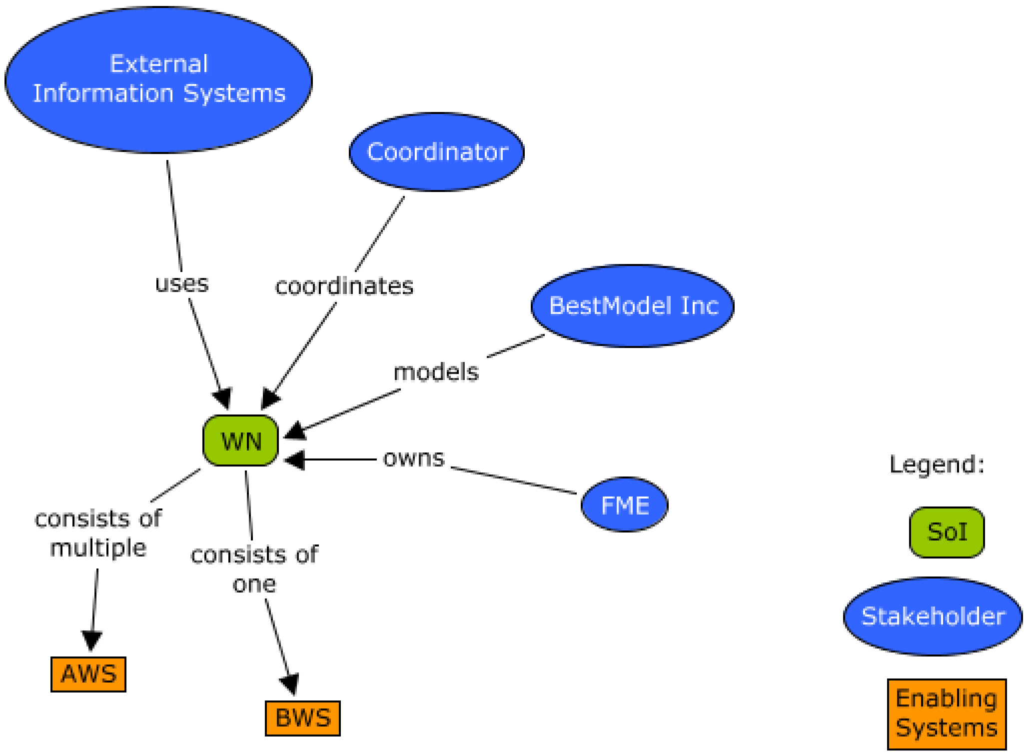

Figure 4 - WN context model.

Artefact #7 – MeteoNet context model

Being the software that runs in the BWS, it is the system which allows the access to the information contained in the BWS so it has all the other systems that needs that information as enabling systems.

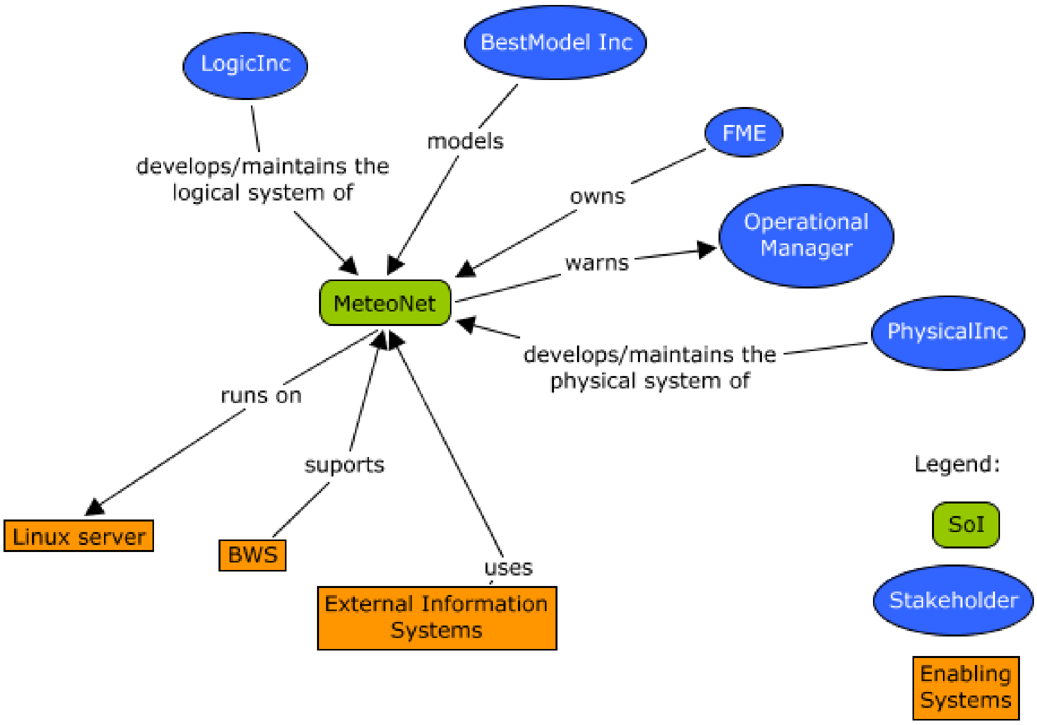

Figure 5 - MeteoNet context model.

Artefact #8 – MeteoNet requirements

ID	Source	Type	Description
1	Line 165	F	MeteoNet is always prepared to receive data packets from the WN.
2	Lines 168-169	F	MeteoNet sends one e-mail for each error code identified in the packets.
3	Line 170	NF	MeteoNet maintains a record for each AWS within the WN.
3.1	Line 171	NF	Each record has the AWS ID.
3.2	Line 172	NF	Each record has a list of all sensors installed on the AWS along.
3.2.1	Lines 172-173	NF	The list of all sensors installed on the AWS comprises the type of each sensor, its serial number, provider, date of installation, and correspondent sampling period.
3.3	Line 174	NF	Each record has a list of all problems related to a sensor.
3.3.1	Line 174	NF	The list of all problems related to a sensor comprises the timestamp, sensor identifier and error type.
3.4	Line 175	NF	Each record has records of all readings.
3.4.1	Lines 175-176	NF	A record of a reading comprises the timestamp, the message and the correspondent content (structured in a proper logical information structure).
3.5	Line 177	NF	Each record has the configuration of the AWS.
3.6	Line 178	NF	Each record has the geographical location.
ID	**Source**	**Type**	**Description**
4	Line 182 - 185	NF	MeteoNet has a functional interface that must be compatible with the external information systems: DALI, PICASSO and MATISSE.
4.1	Lines 188-189	F	The external information systems must be able to set the state of the MeteoNet.
4.1.1	Lines 188-189	F	The external information systems can add stations.
4.1.2	Lines 188-189	F	The external information systems can remove stations
4.2	Lines 188-189	F	The external information systems must be able to get the state of the MeteoNet.
4.2.1	Lines 188-189	F	MeteoNet can provide a list of all AWS and their attributes.
4.2.2	Lines 188-189	F	MeteoNet can provide the measurements from a single station.
4.2.3	Lines 188-189	F	MeteoNet can provide the measurements (within a time interval) from a type of sensor.
4.2.4	Lines 188-189	F	MeteoNet can provide the measurements (within a time interval and area) from a type of sensor can be listed
4.2.5	Lines 188-189	F	MeteoNet can provide a list of silent stations (within a time interval).
4.2.6	Lines 188-189	F	MeteoNet can provide a list of sensor errors (within a time interval).

Table 11 - MeteoNet requirements list.

ID	Source	Type	Description
4.2.7	Lines 15 - 17	F	MeteoNet system must have a new service, called GetDomain, that must save, in the BWS local file system, all the internal domain of the system in that moment.

Table 12 - Additional MeteoNet requirements list

Artefact #9 – MeteoNet use cases

All the systems that use the measurements contained on the BWS (PICASSO, DALI and MATISSE) interact with MeteoNet in the same way because their interest in this software is similar. The other entities involved on the use cases are those which manage and control the system.

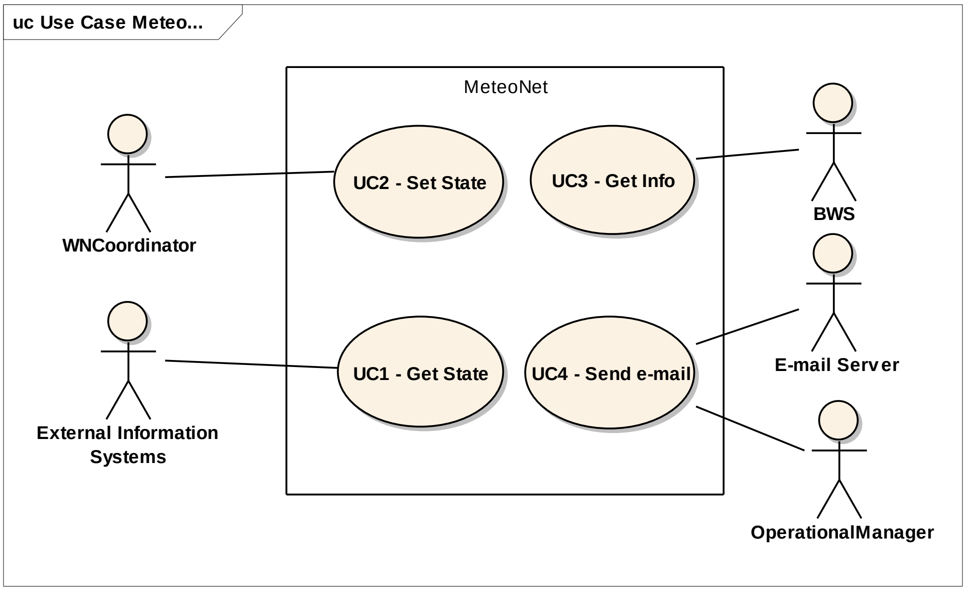

Figure 6 - Use Case diagram for the MeteoNet system.

Artefact #10 – MeteoNet use cases scenarios

ID and Name	UC1 - Get State
Summary	External Information Systems use MeteoNet services to get various readings.
Rationale	In order to get the necessary readings specific parameters (sensor id and a time interval) have to be passed to the functions provided by the functional interface. The output is a list of (aws id, measurements)
Actors	External Information Systems (DALI, MATISSE, PICASSO)
Pre-conditions	Some External Information System needs readings information in order to accurately determine current weather.
Primary Scenario	The External Information System uses the functional interface provided by MeteoNet and calls GetWNReadings to get the information needed.
Post-conditions	The MeteoNet state remains unchanged.
Alternative Scenario "GetStations"	The External Information System uses The functional interface provided by MeteoNet and calls GetStations to get The Information needed. Depending on the service, the input and ouput parameters differ.
Alternative Scenario "GetStationReadings"	The External Information System uses The functional interface provided by MeteoNet and calls GetStationReadings to get The Information needed. Depending on the service, the input and ouput parame-ters differ.
Alternative Scenario "GetAreaReadings"	The External Information System uses the functional interface provided by MeteoNet and calls GetAreaReadings to get the information needed. Depending on the service, the input and ouput parameters differ.
Alternative Scenario "GetSilentStations"	The External Information System uses the functional interface provided by MeteoNet and calls GetSilentStations to get the information needed. Depending on the service, the input and ouput parameters differ.
Alternative Scenario "GetSensorErrors"	The External Information System uses the functional interface provided by MeteoNet and calls GetSensorErrors to get the information needed. Depending on the service, the input and ouput parameters differ.
Alternative Scenario "GetDomain"	The External Information System uses the functional interface provided by MeteoNet and calls GetDomain in order to have the current MeteoNet state saved in the file system.

Table 13 - Scenarios of the AWS UC1 – Get State.

ID and Name	UC2 - Set State
Summary	Install a new AWS.
Rationale	In order to add the new station the AWS ID must be passed as the argument of the AddStation service. The output is an error code.
Actors	WNCoordinator
Pre-conditions	The WNCoordinator decides to install a new AWS.
Primary Scenario	The WNCoordinator decides to install a new AWS to expand the WN covered area. He/she calls the service AddStation provided by the MeteoNet functional interface.
Post-conditions	The MeteoNet state is changed if the AWS is successfuly added.
Alternative Scenarios "RemoveStation"	The WNCoordinator decides to remove an AWS. He/she calls the service RemoveStation provided by the MeteoNet functional interface. The input and output parameters are similar to the AddStation service parameters.

Table 14. Scenarios of the AWS UC2 – Set State.

ID and Name	UC3 - Receive Data
Summary	The MeteoNet software receives data from the BWS.
Rationale	The data sent from the BWS is received through a GPRS network and is made available to MeteoNet using an AWS driver.
Actors	BWS
Pre-conditions	The BWS receives data packets from the GPRS network.
Primary Scenario	The BWS sends the data from the whole Weather Network to the MeteoNet software to be processed.
Post-conditions	The MeteoNet state is updated.

Table 15. Scenarios of the AWS UC3 – Receive Data

ID and Name	UC4 - Send e-mail
Summary	The MeteoNet software sends an e-mail reporting an error.
Rationale	The e-mail includes information like the AWS id, sensor id and the data packet timestamp.
Actors	E-mail server, OperationalManager
Pre-conditions	Error code is identified in data packet.
Primary Scenario	For each error code, MeteoNet sends an e-mail to the OperationalManager through an e-mail server reporting a sensor error.
Post-conditions	The MeteoNet state is changed.

Table 16. Scenarios of the AWS UC4 – Send e-mail

Artefact #11 – AWS structure

Building both bdd and ibd, it is possible to build an image about the final system, along with the interactions that will occur between the multiple parts which compose the system of interest.

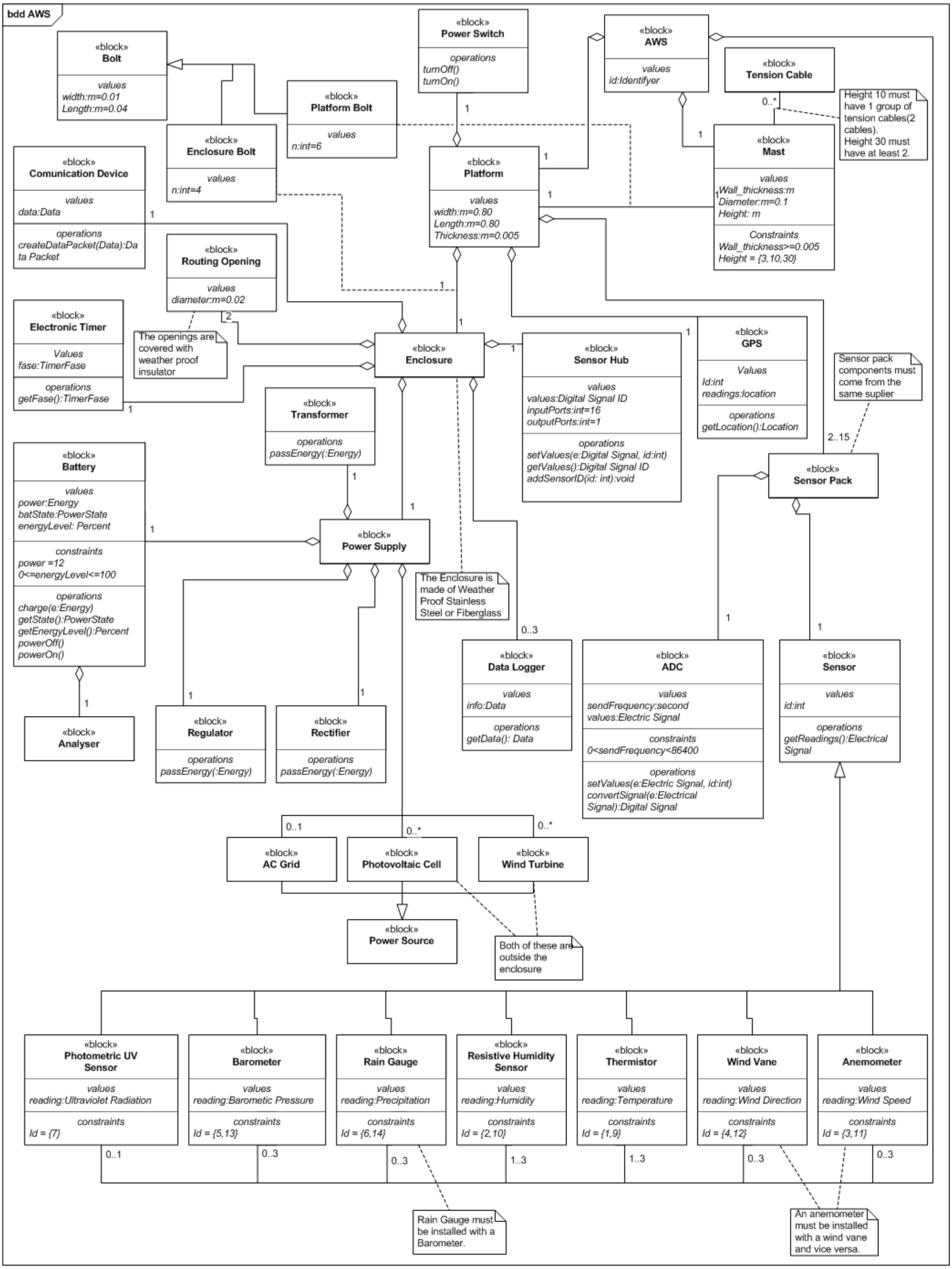

Figure 7 - Block definition diagram for the AWS system.

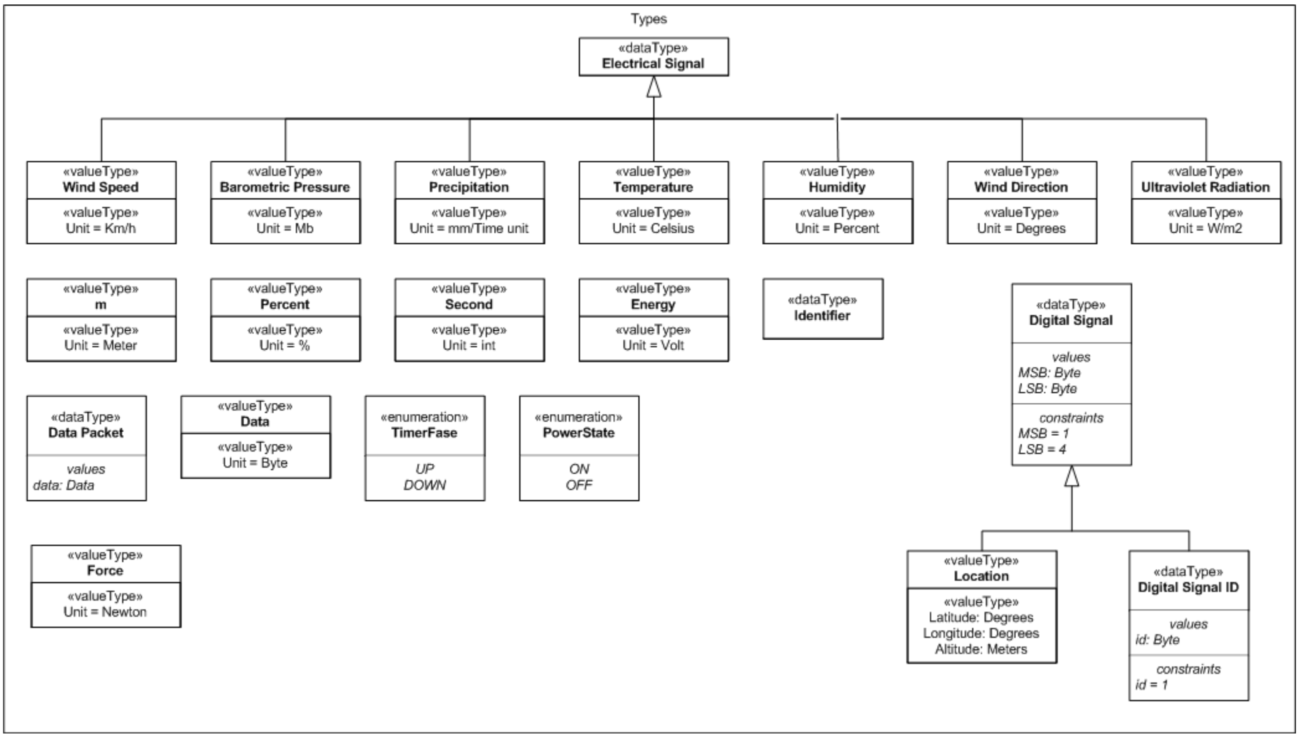

Figure 8 - Block definition diagram for the AWS system types.

In both alocation diagrams only the ports, blocks and flows relevant to that allocation are shown.

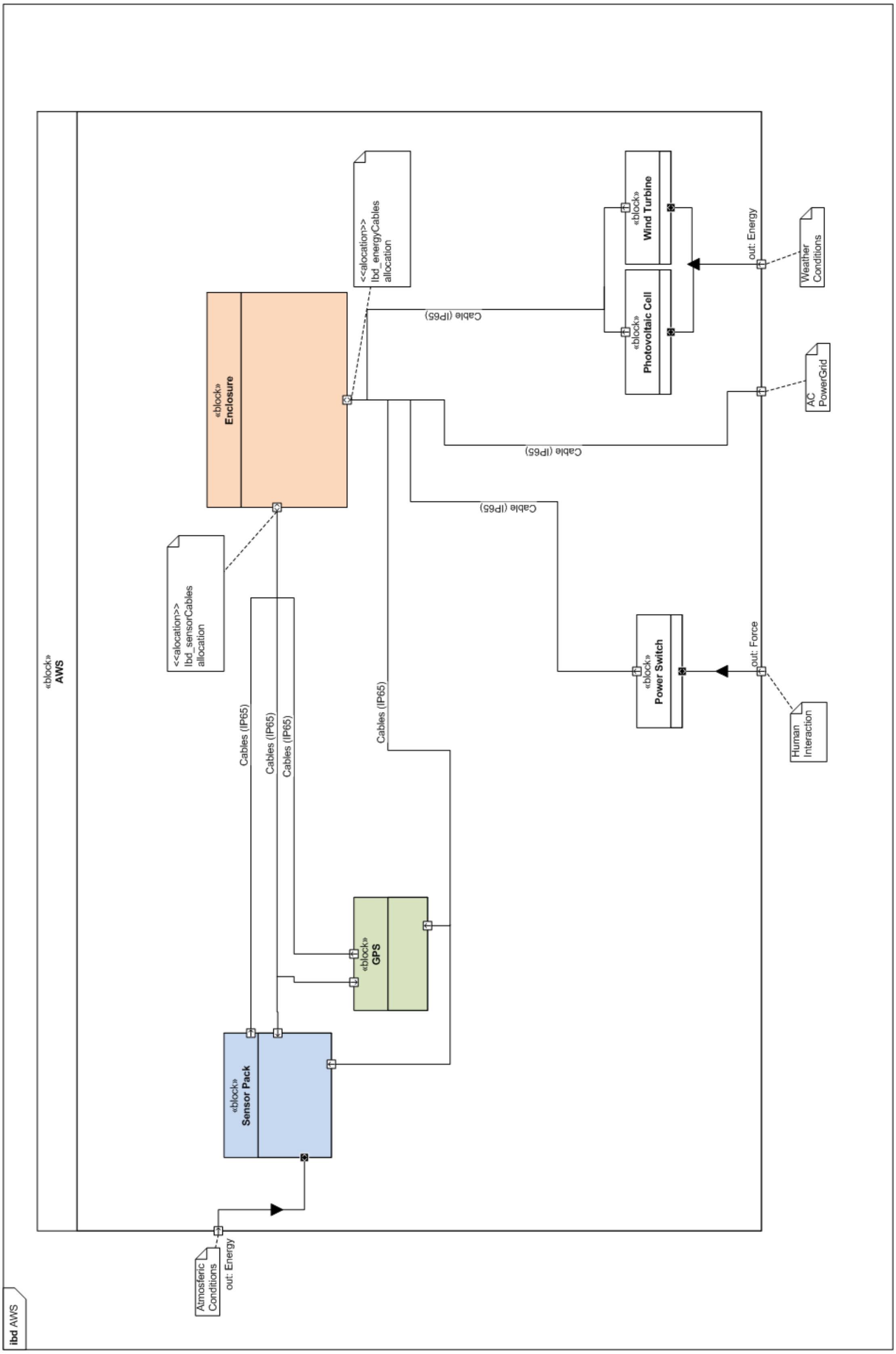

Figure 9 - Internal Block diagram for the AWS system.

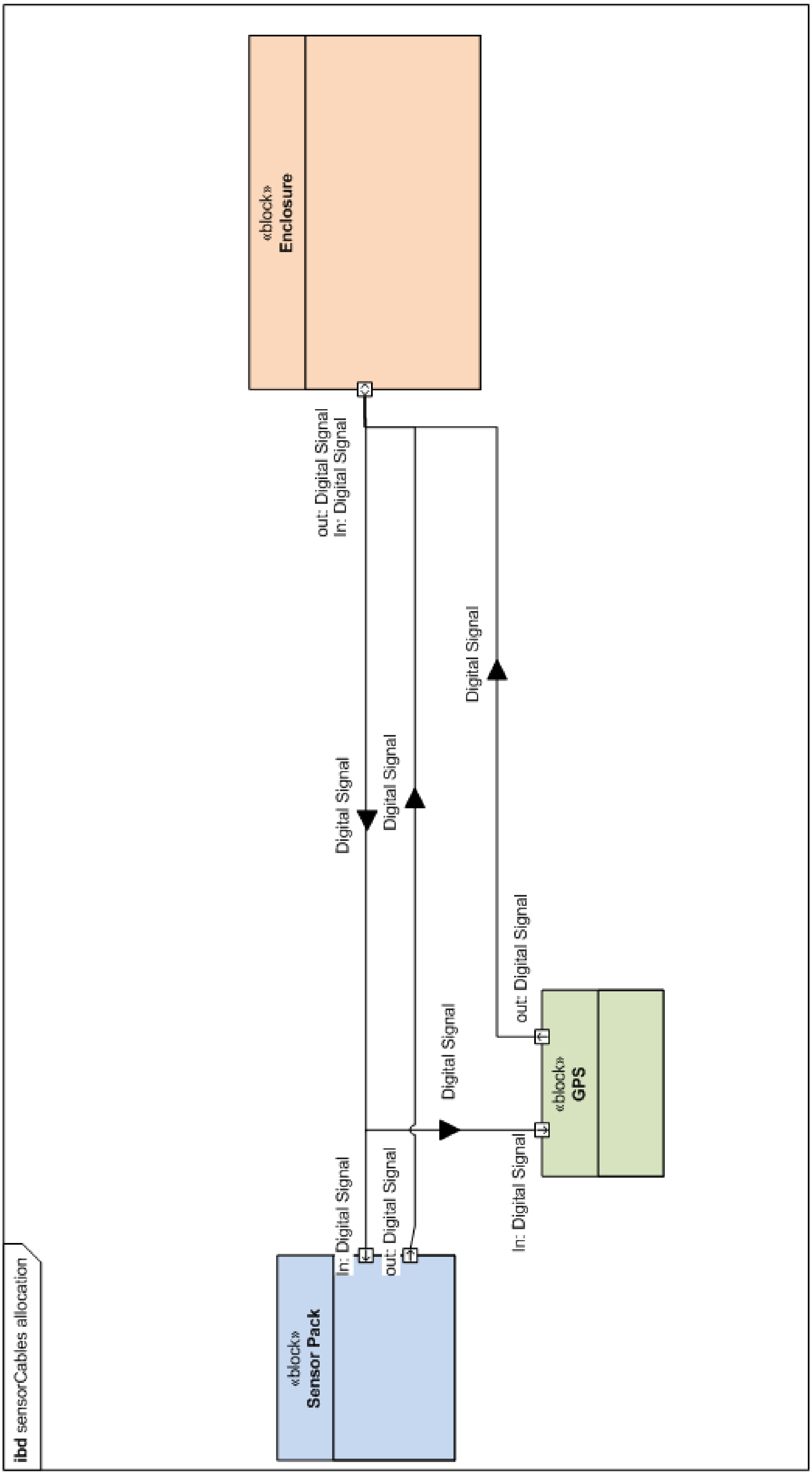

Figure 10 - Internal Block diagram for the SensorCables Allocation

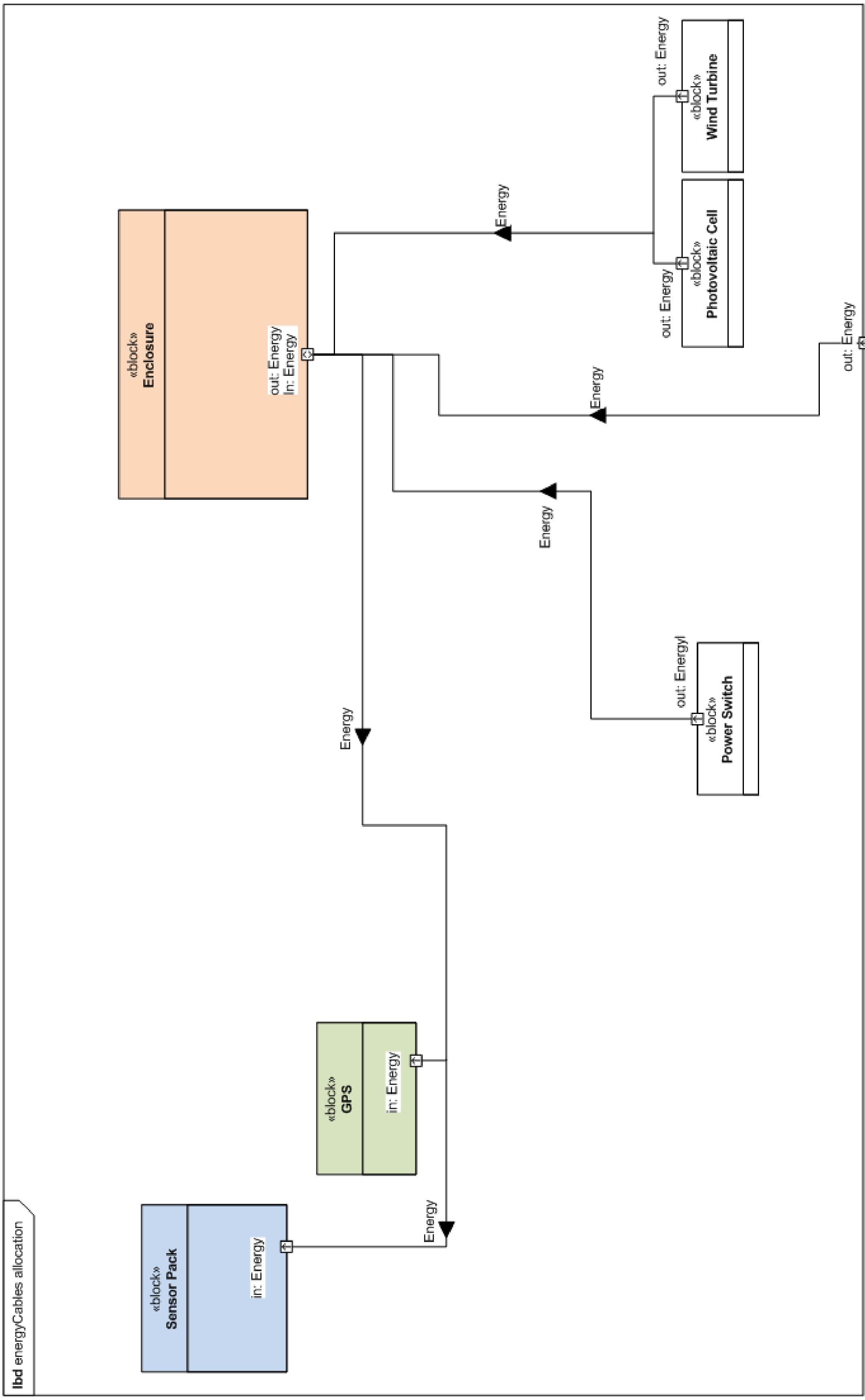

Figure 11- Internal Block diagram for the EnergyCables Allocation.

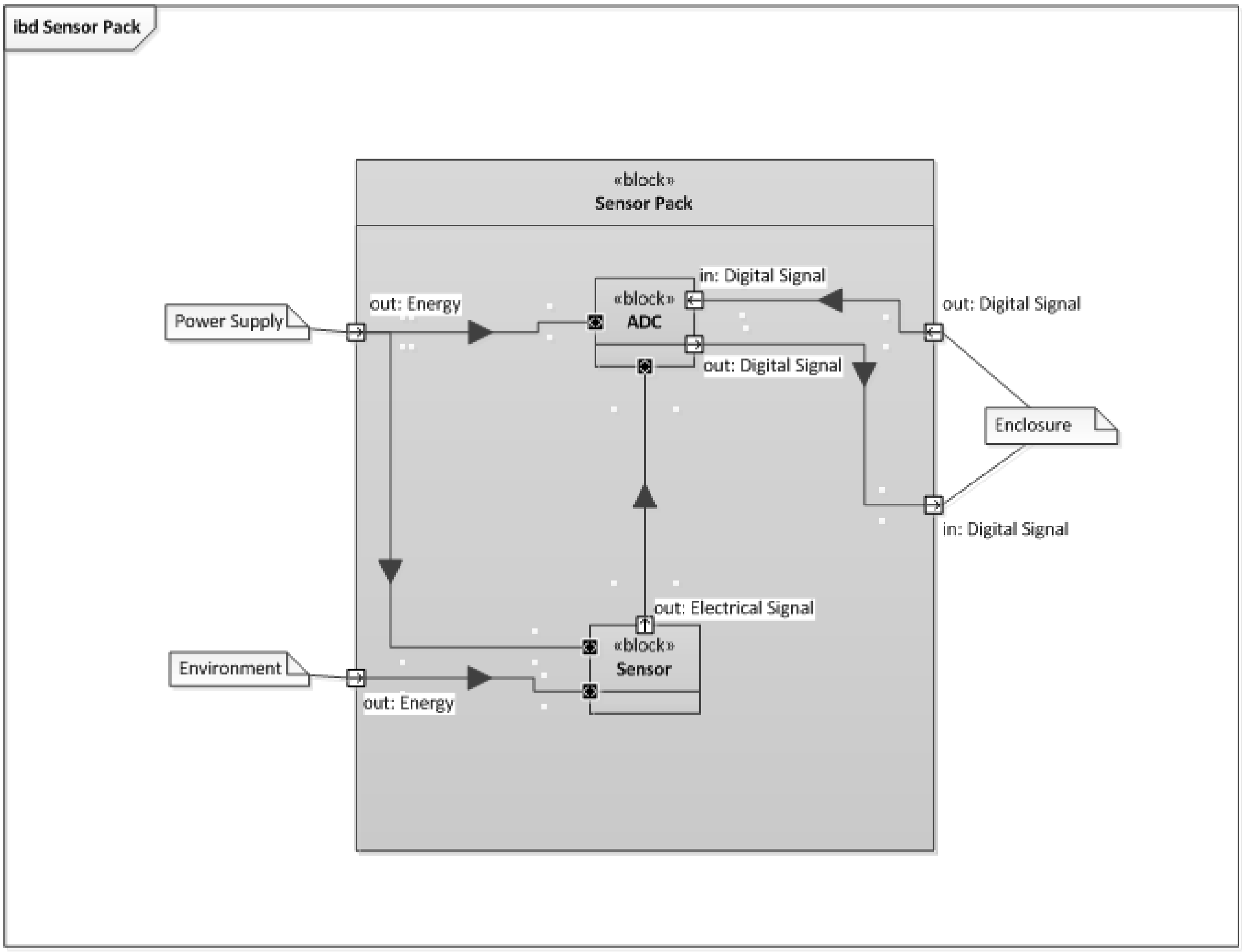

Figure 12- Internal Block diagram for the Sensor Pack.

On the next Diagram the comments have different colours with different meanings. A comment in red (GPS and Sensor Pack) represents that this Block handles the input and output of the Digital Signal port and only the input of the Energy Port they are connected to. For example GPS is red, so this means that it receives the input of the Digital Signal Port, and it send Digital Signal to the port, the output of the port. Also GPS receives Energy from the second port, input.

A green comment means that this is only responsible for the port output, in other words they only give energy to the enclosure, it does not receive.

A comment in blue is only to show to which of the blocks represented in the green comments that port is connected.

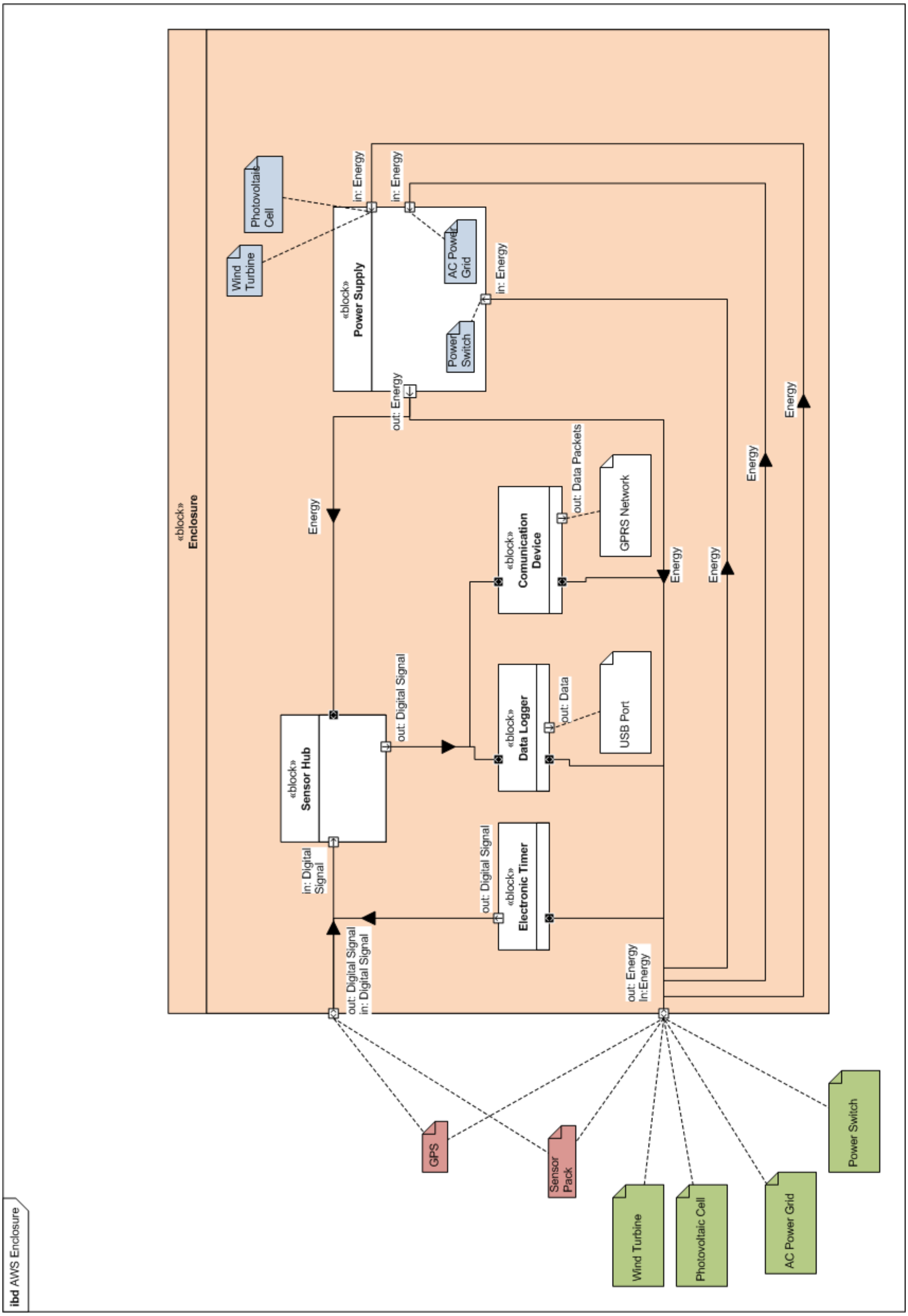

Figure 12- Internal Block diagram for the Enclosure.

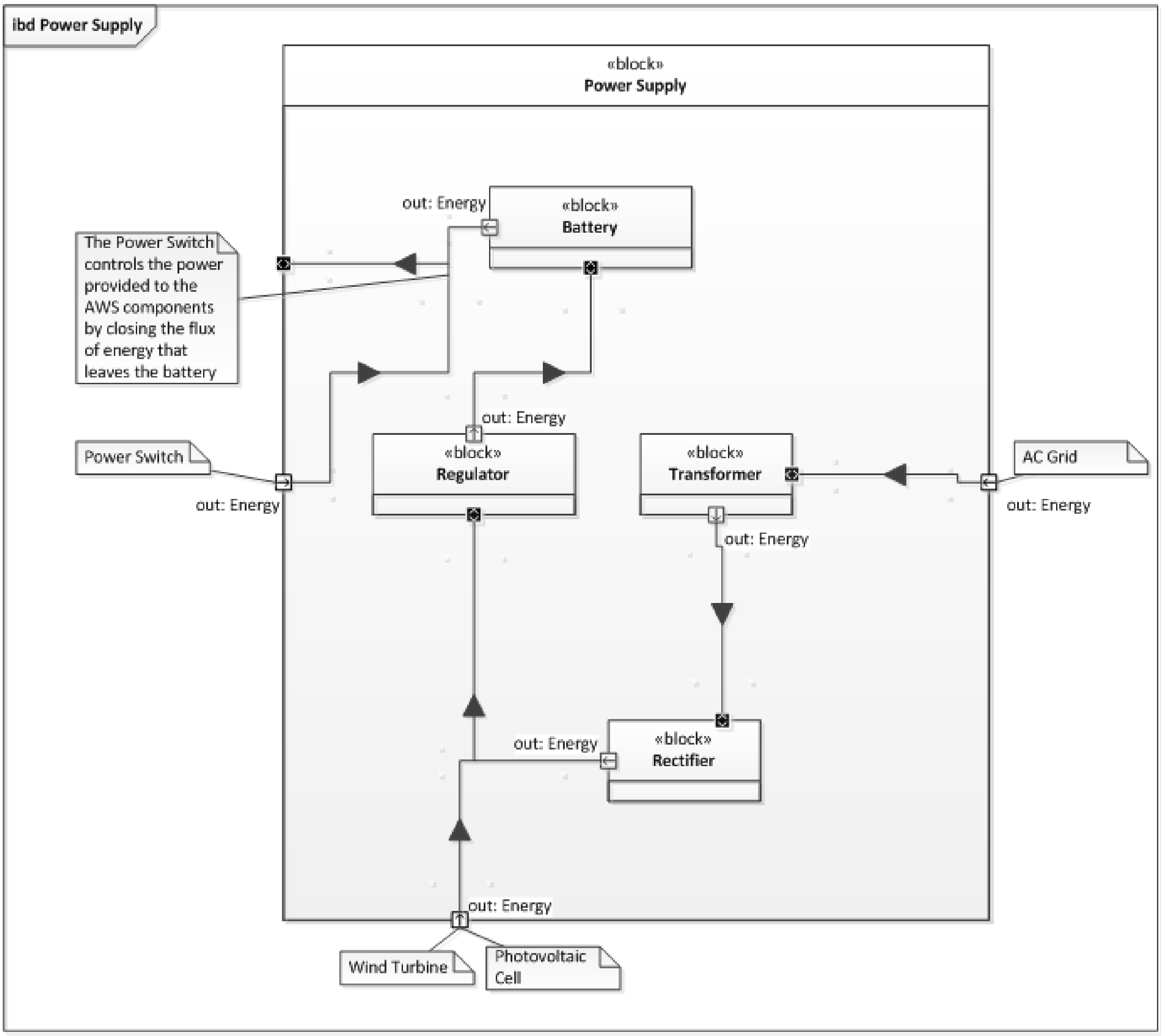

Figure 134- Internal Block diagram for the Power Supply.

Artefact #12 – MeteoNet classes

Similarly to the last artefact, here we have the model of the software system, MeteoNet, allowing the reader to understand how it must be formed, and how to relate the parts with each other.

The data type classes used by the MeteoNet software are represented in a different diagram.

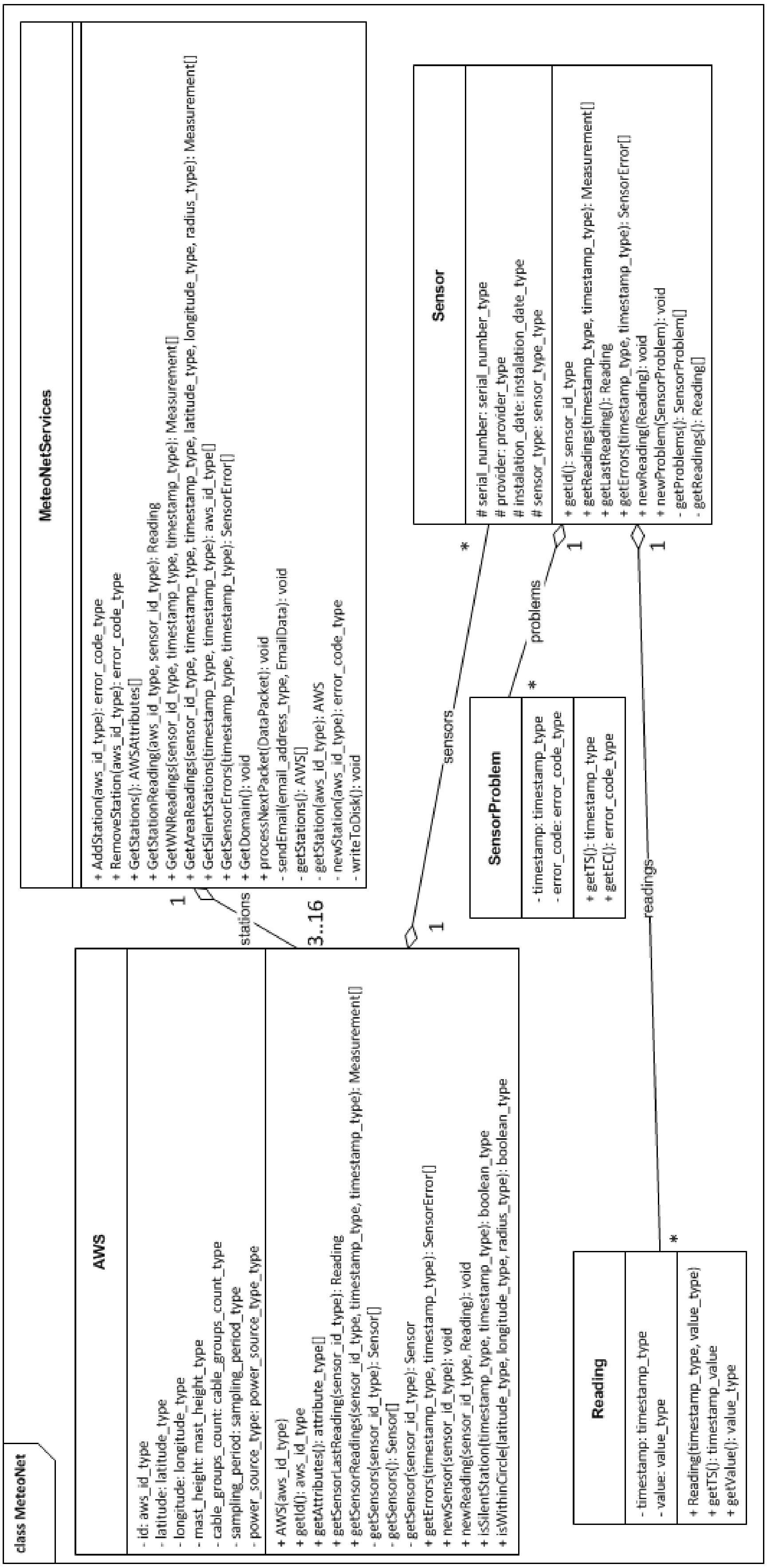

Figure 14 - Classes diagram for the MeteoNet system.

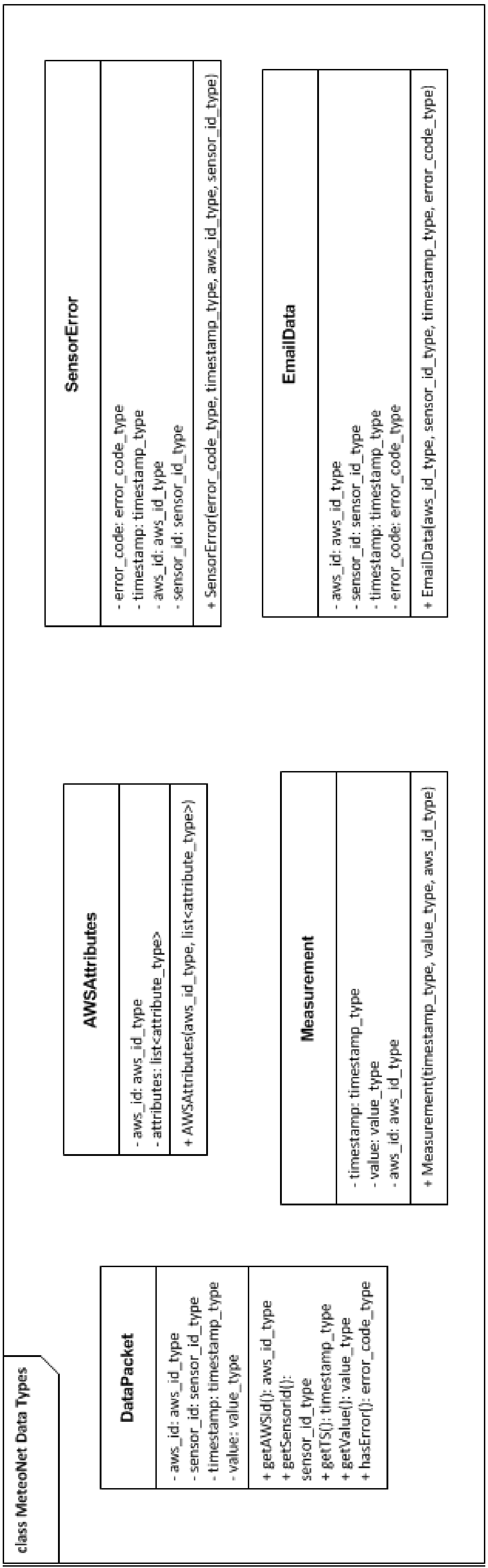

Figure 15 - Classes diagram (data types) for the MeteoNet system.

Artefact #13 – MeteoNet components

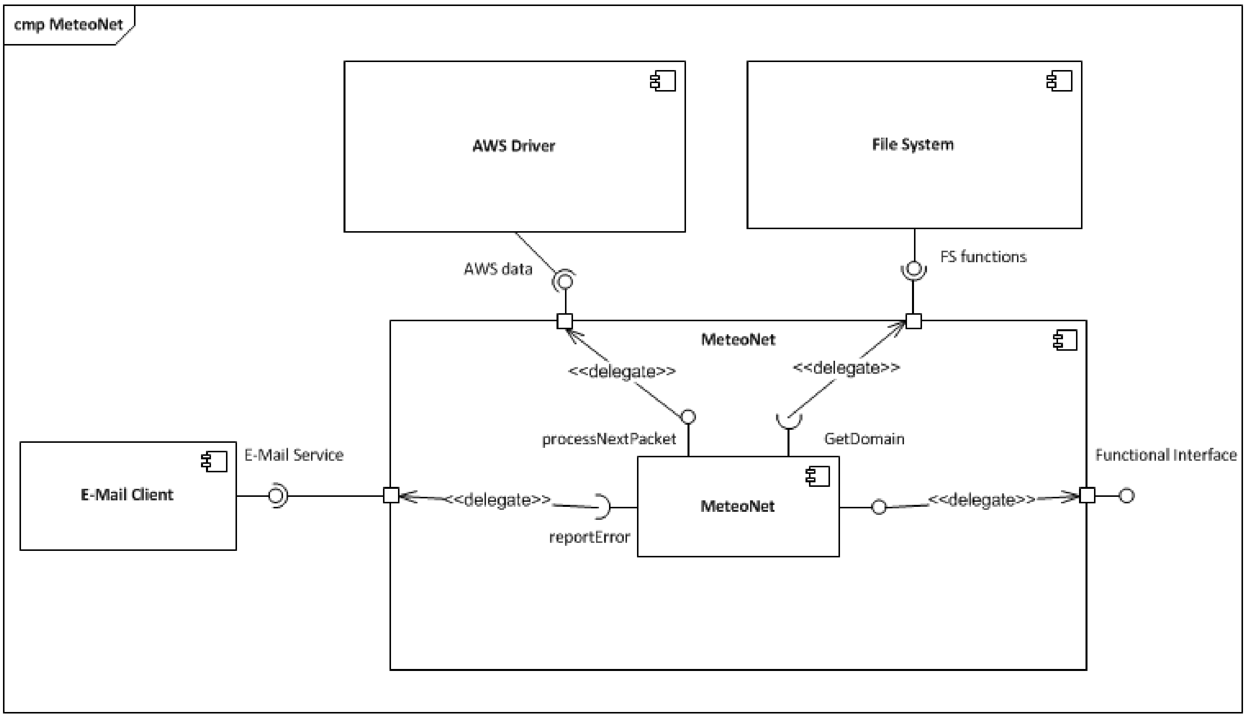

Figure 16 - Component diagram for the MeteoNet system.

Artefact #14 – MeteoNet deployment

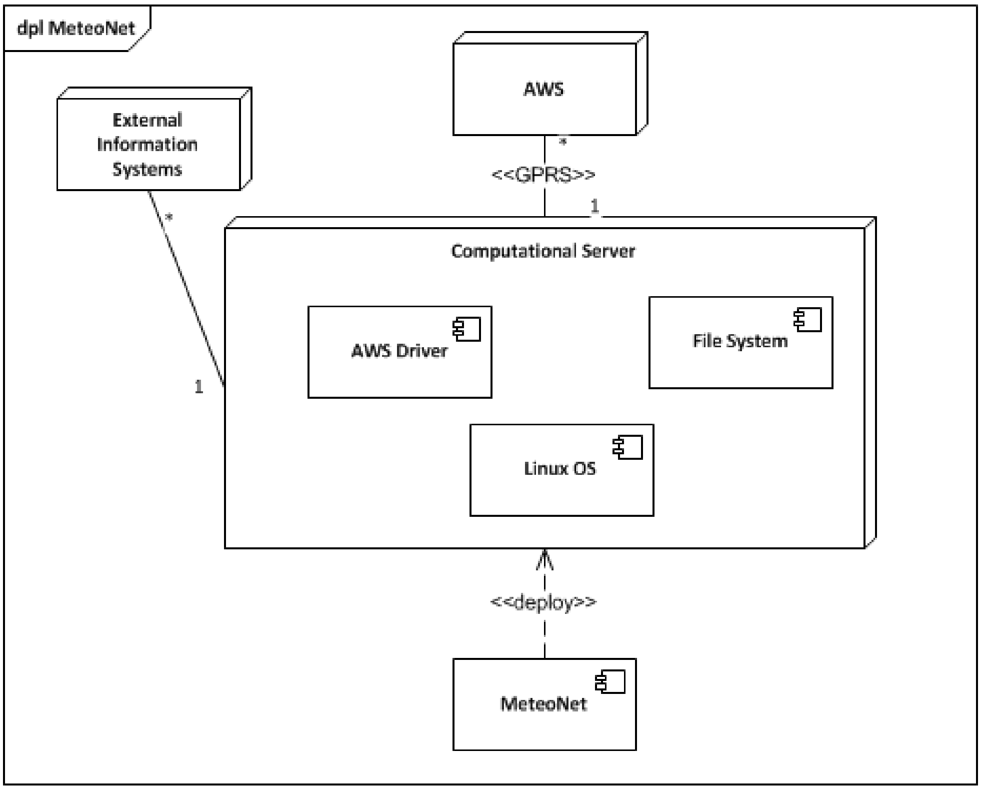

Figure 17 - Deployment diagram for the MeteoNet system.

Artefact #15 – AWS behavior

In the perform measurements and read Location diagrams, the actions done in parallel are: the sensor always reading the environment, or the location, the ADC sending the values read to the sensor Hub, but only doing this when the timer is in the correct phase and the other two are the communication device and the data loggers getting that sensor information, but only when the timer phase is right.

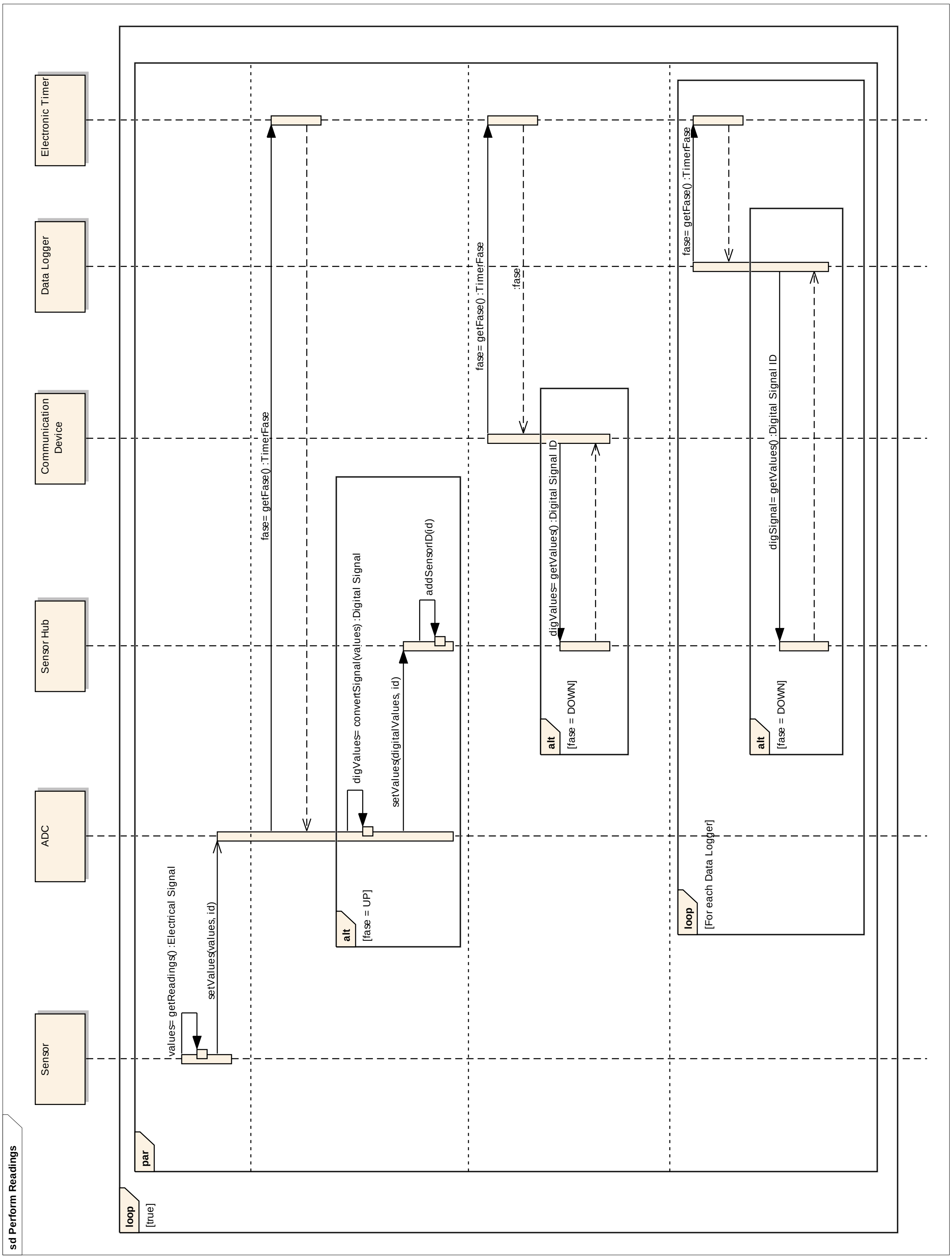

Figure 18 - Sequence diagram for the Perform Measurements Use Case.

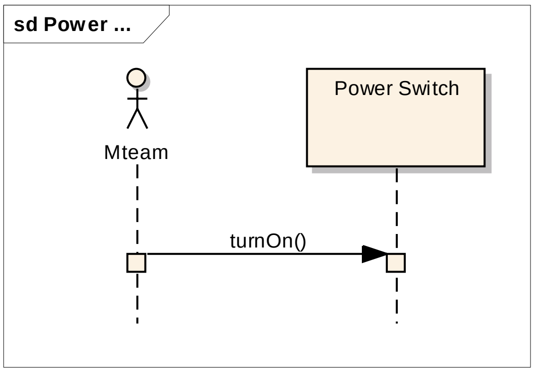

Figure 19 - Sequence diagram for the Power On Use Case.

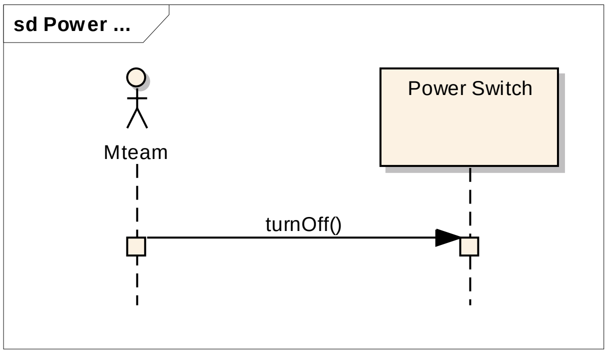

Figure 20- Sequence diagram for the Power Off Use Case.

Both Diagrams present in this page, Power On and Power Off, represent the Mteam turn the Power Switch On(Figure 17) or Off(figure 18), that way turning the AWS on or off respectively.

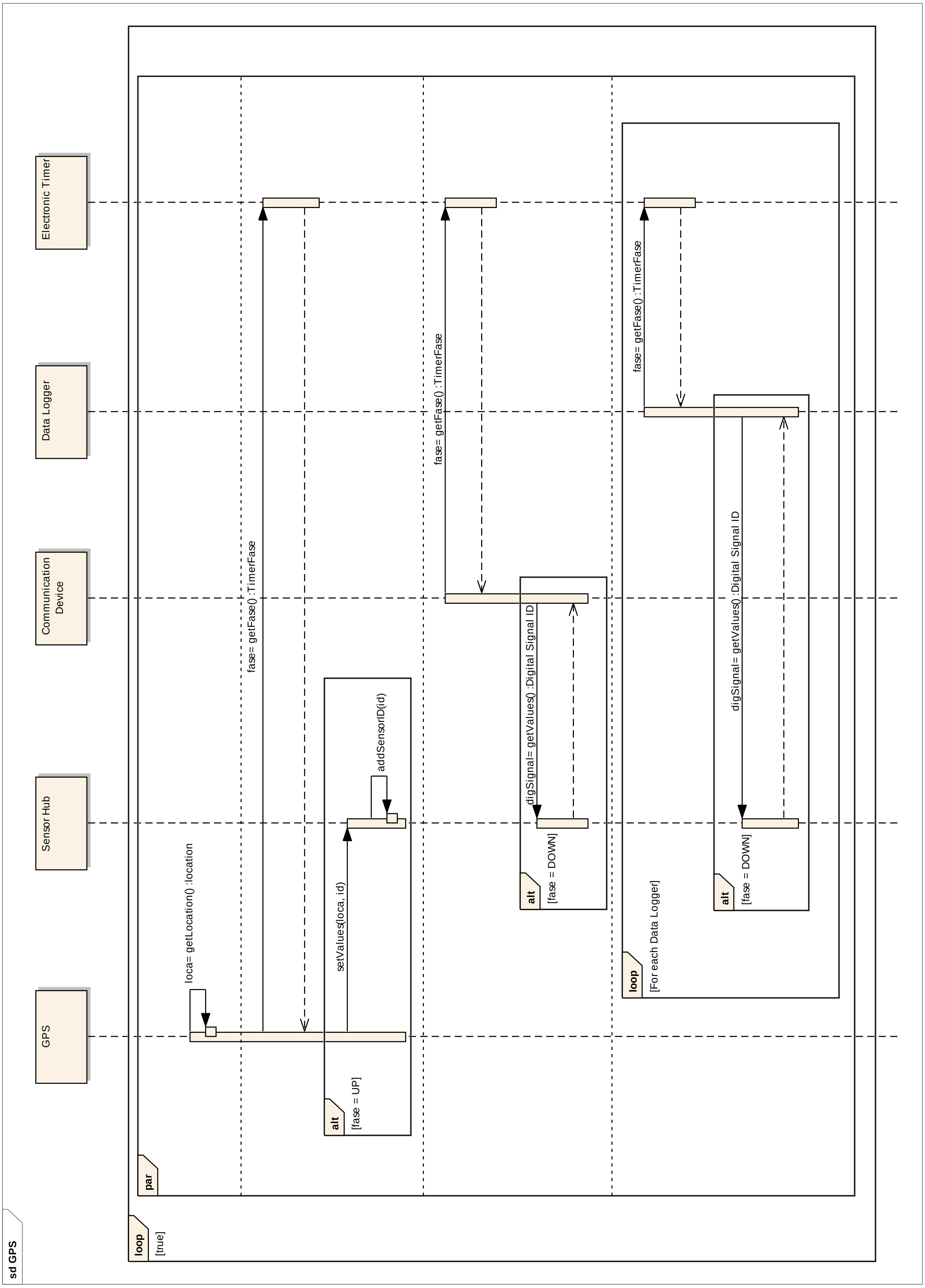

Figure 21- Sequence diagram for the Read Position Use Case.

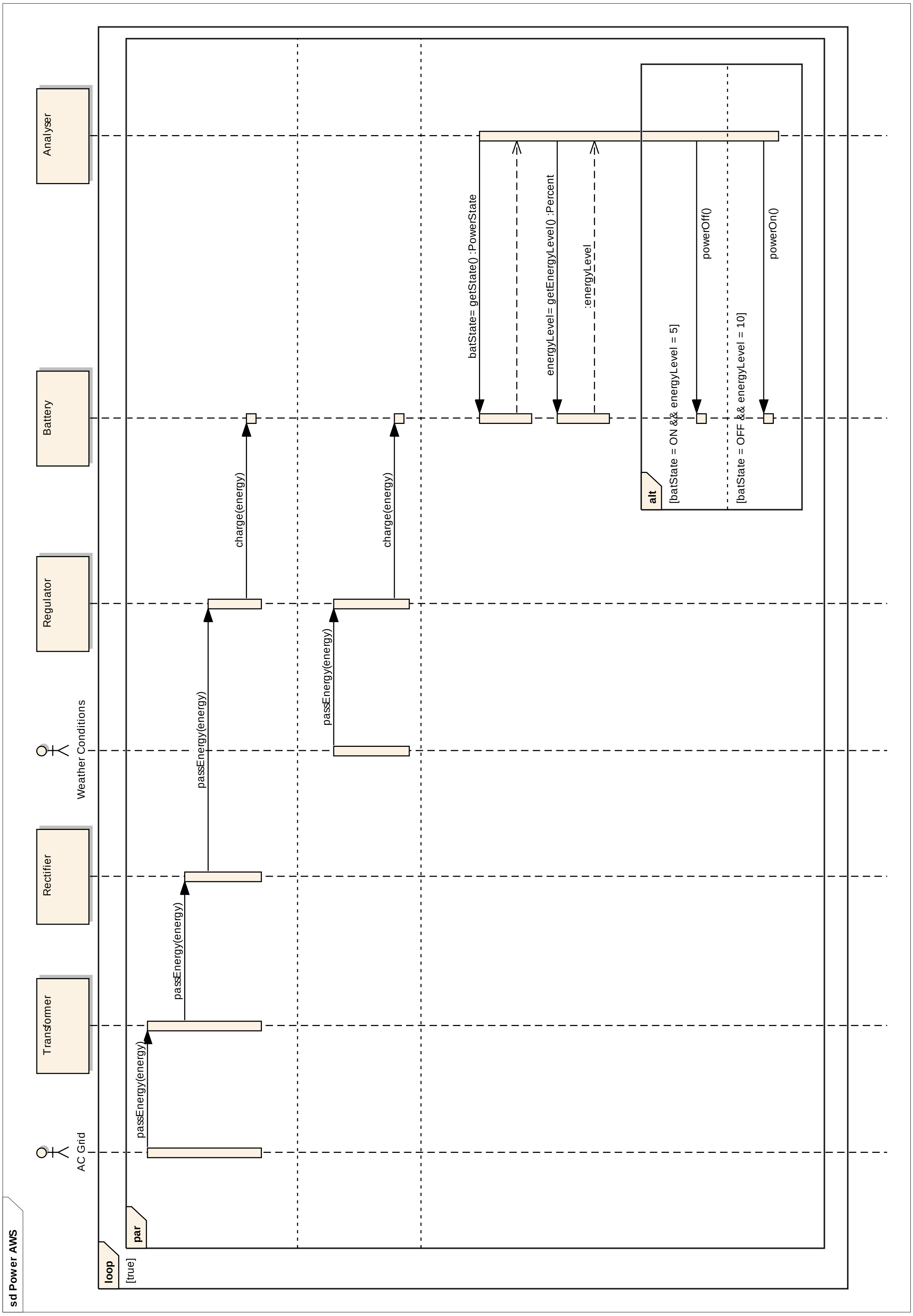

Figure 22 - Sequence diagram for the Receive Power Use Case.

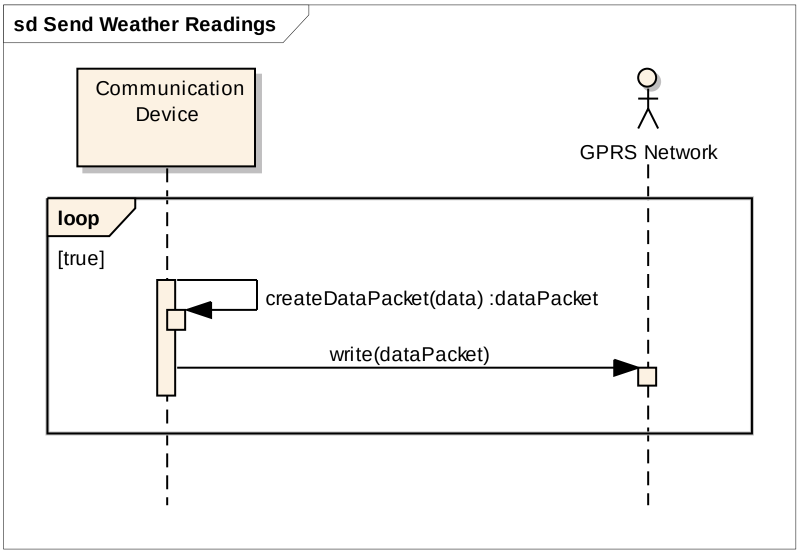

Figure 23- Sequence diagram for the Send Weater Readings Use Case.

Artefact #16 – AWS states

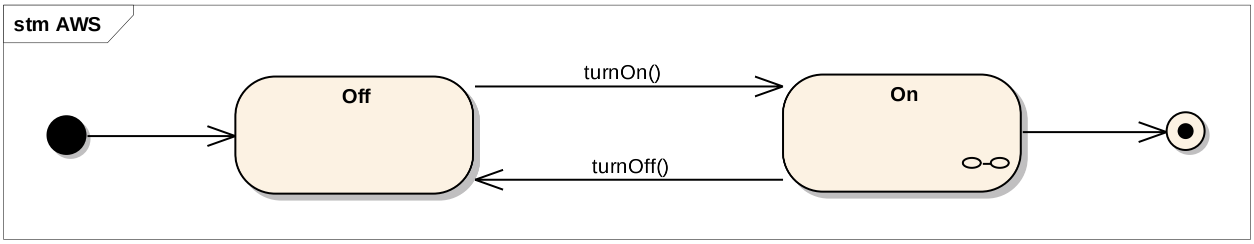

Figure 24- State Machine diagram for the AWS System.

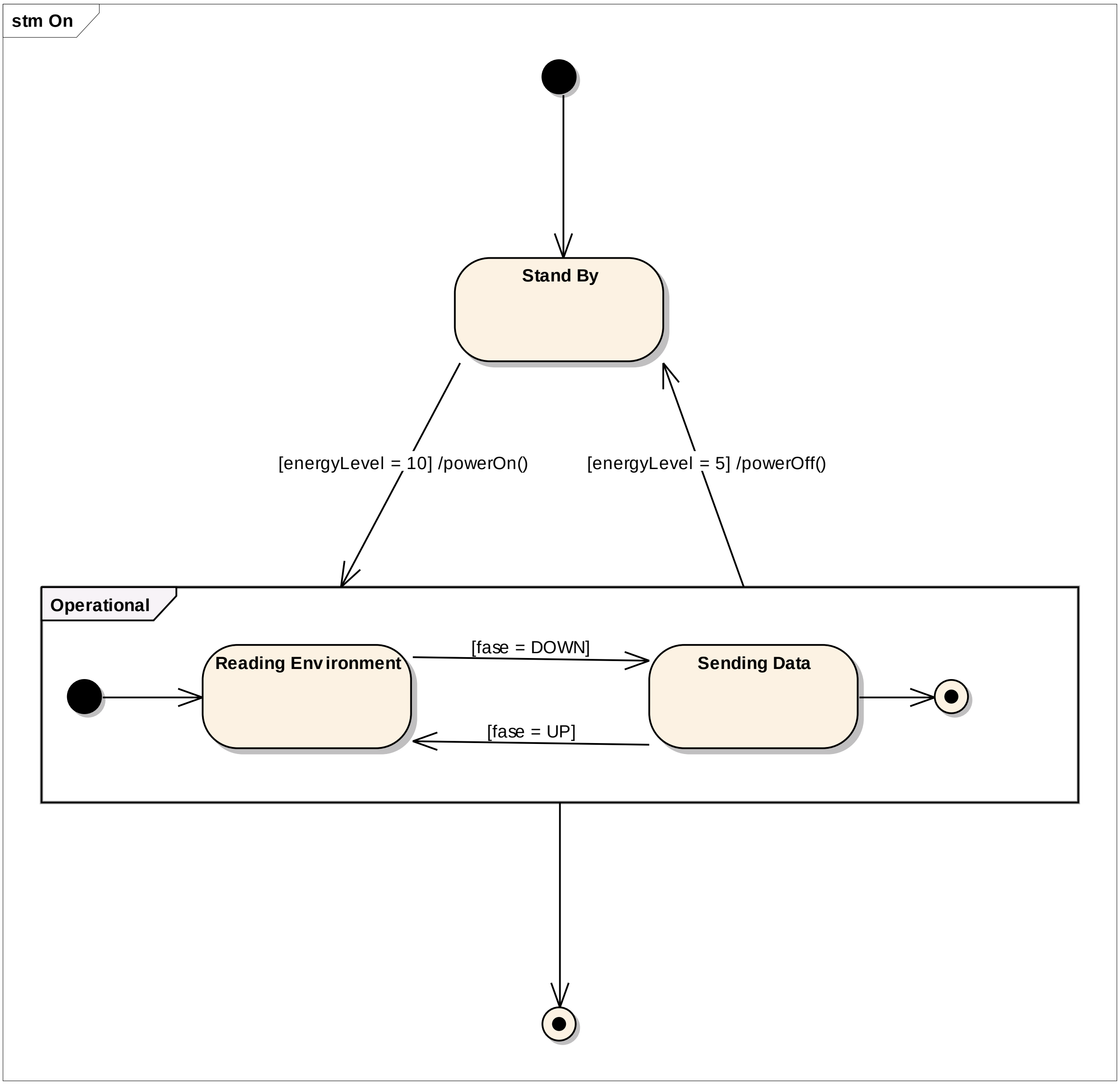

Figure 25- State Machine diagram for the AWS On state.

Artefact #17 – MeteoNet behavior

The following sequence diagrams are related to the UC1, GetState.

The alternative scenario which uses the service GetAreaReadings was not represented as a sequence diagram because it is similar to the GetWNReadings. The only difference is that only the AWS that are within the circle specified by the parameters are taken into account when populating the list to return as the service return value.

NO_ERROR is a constant that represents an error code indicating no error.

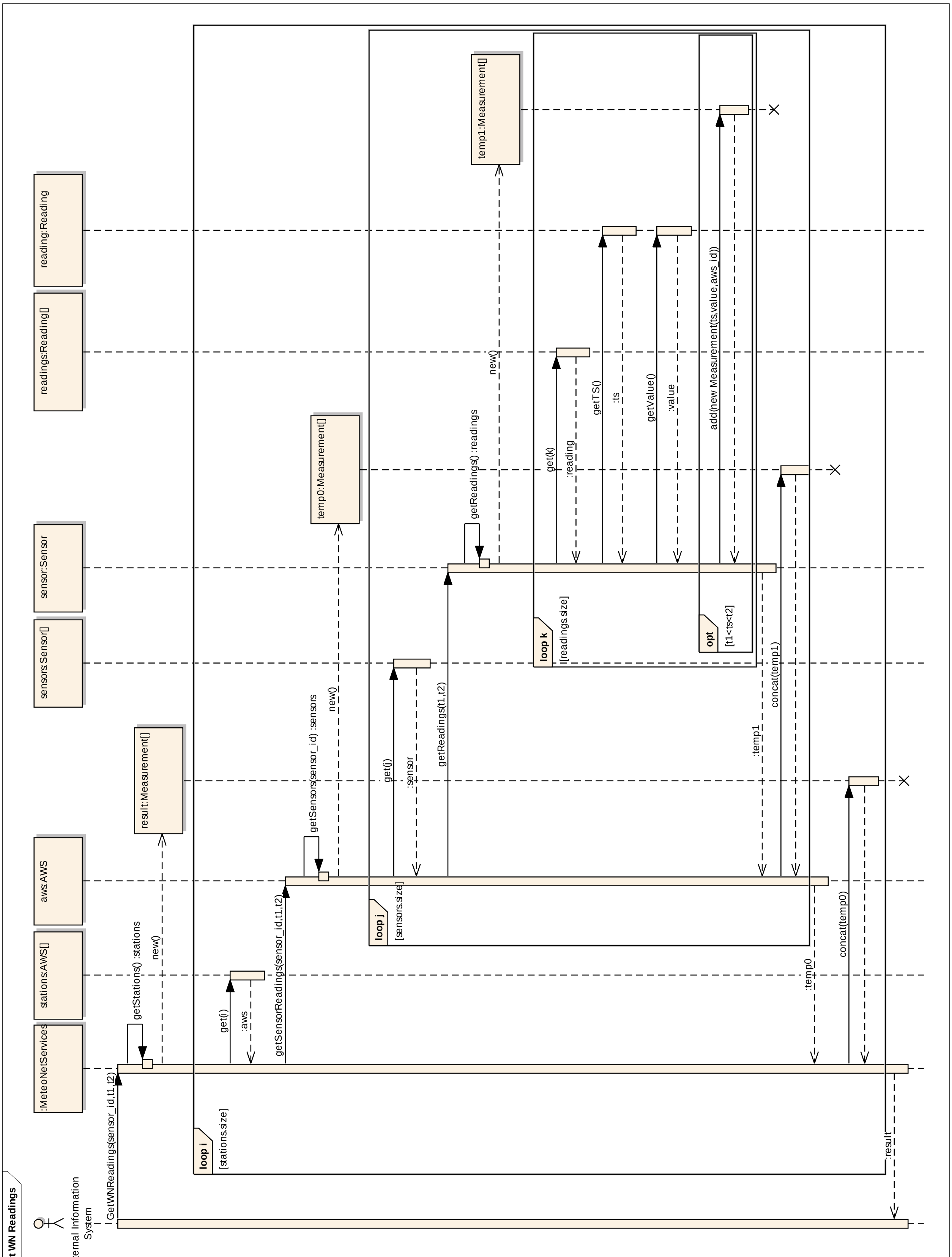

Figure 26- Sequence diagram for the Get WN Readings scenario.

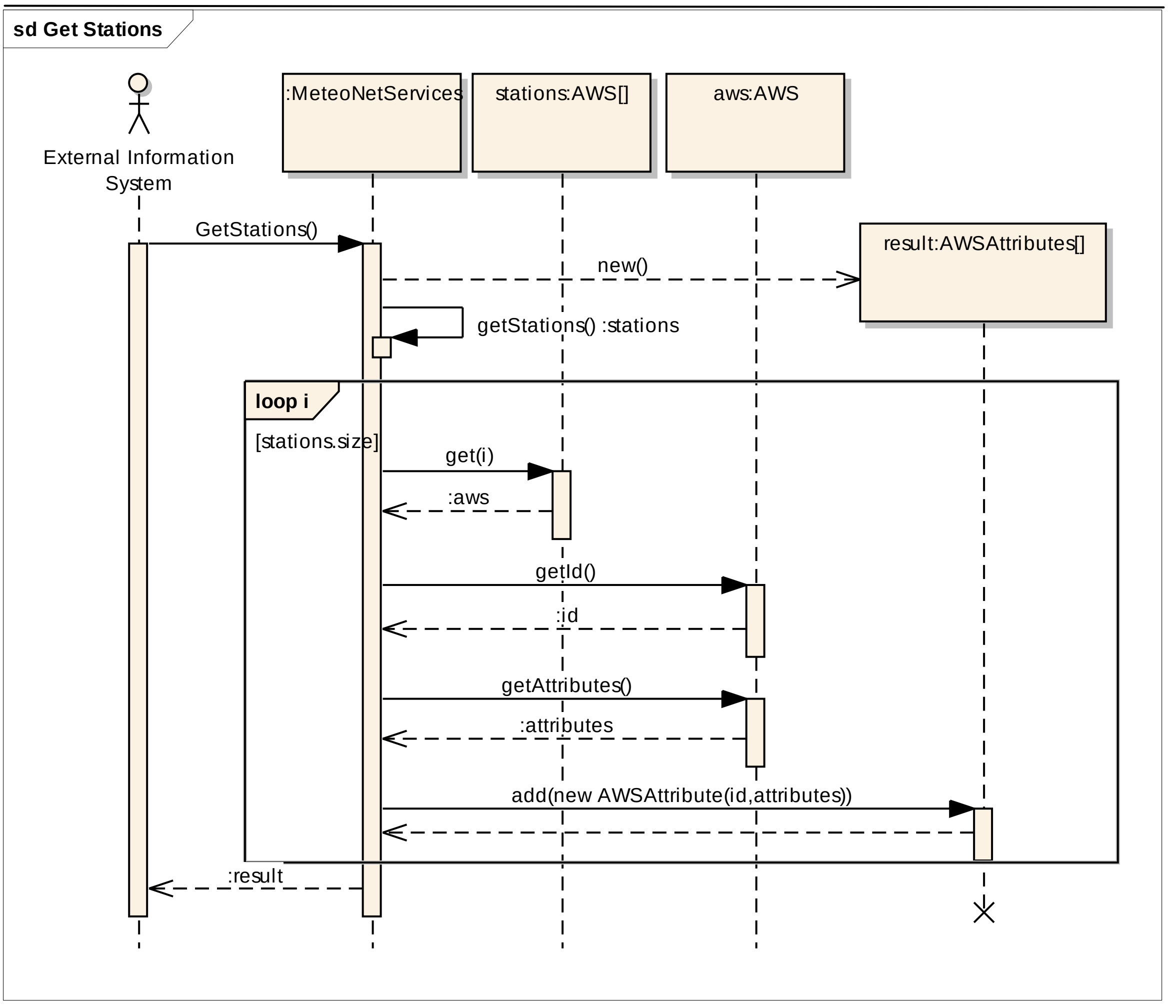

Figure 25– Sequence diagram for the GetStations scenario.

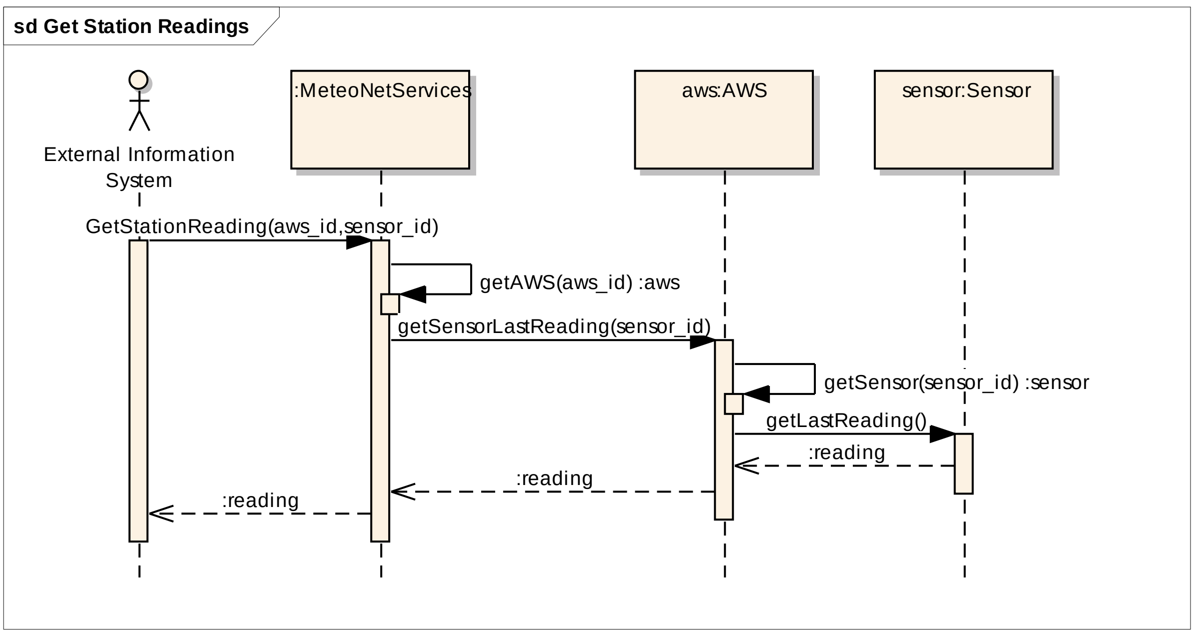

Figure 28- Sequence diagram for the Get Station Readings scenario.

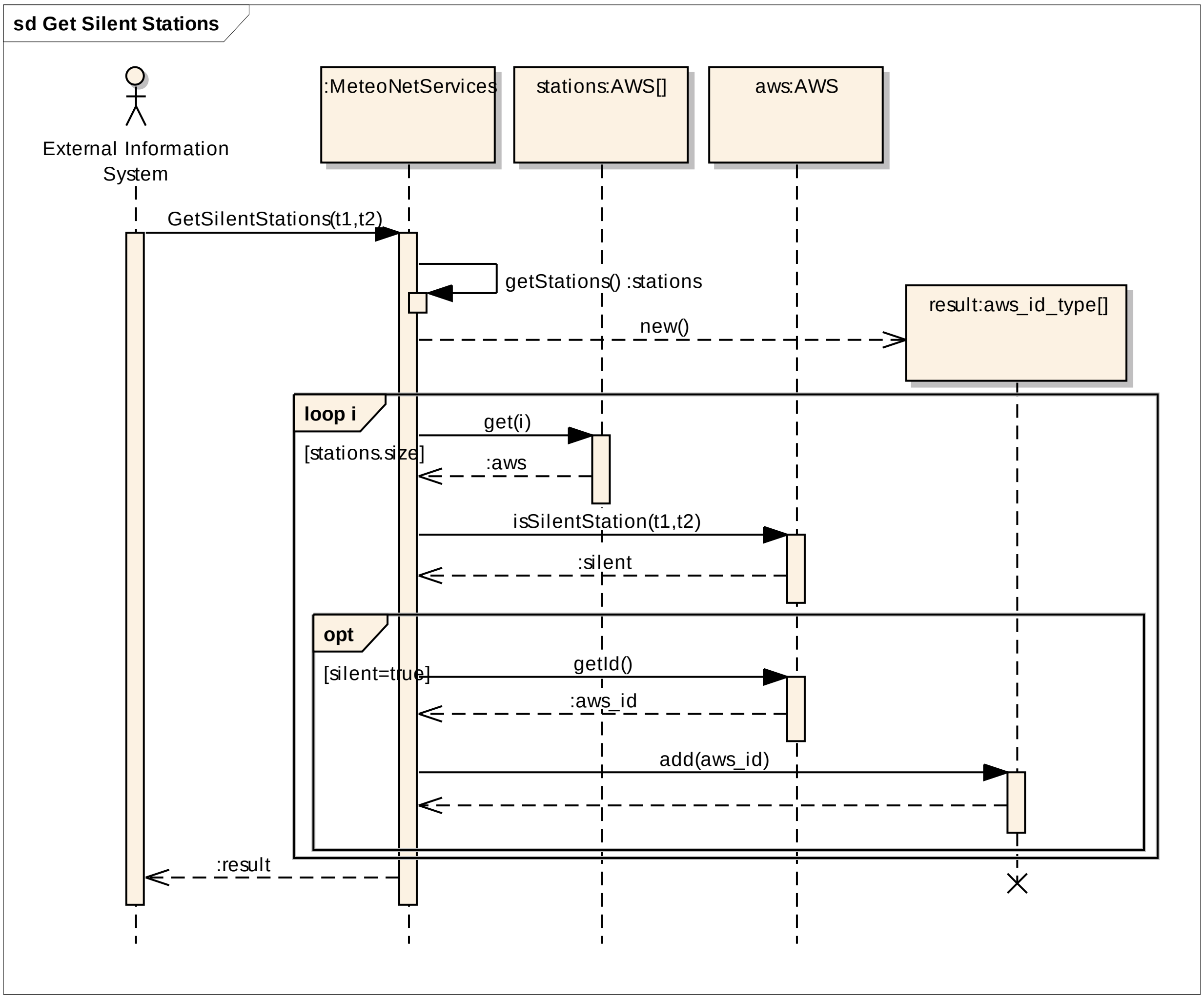

Figure 29- Sequence diagram for the Get Silent Stations scenario.

Figure 30- Sequence diagram for the Get Silent Stations scenario.

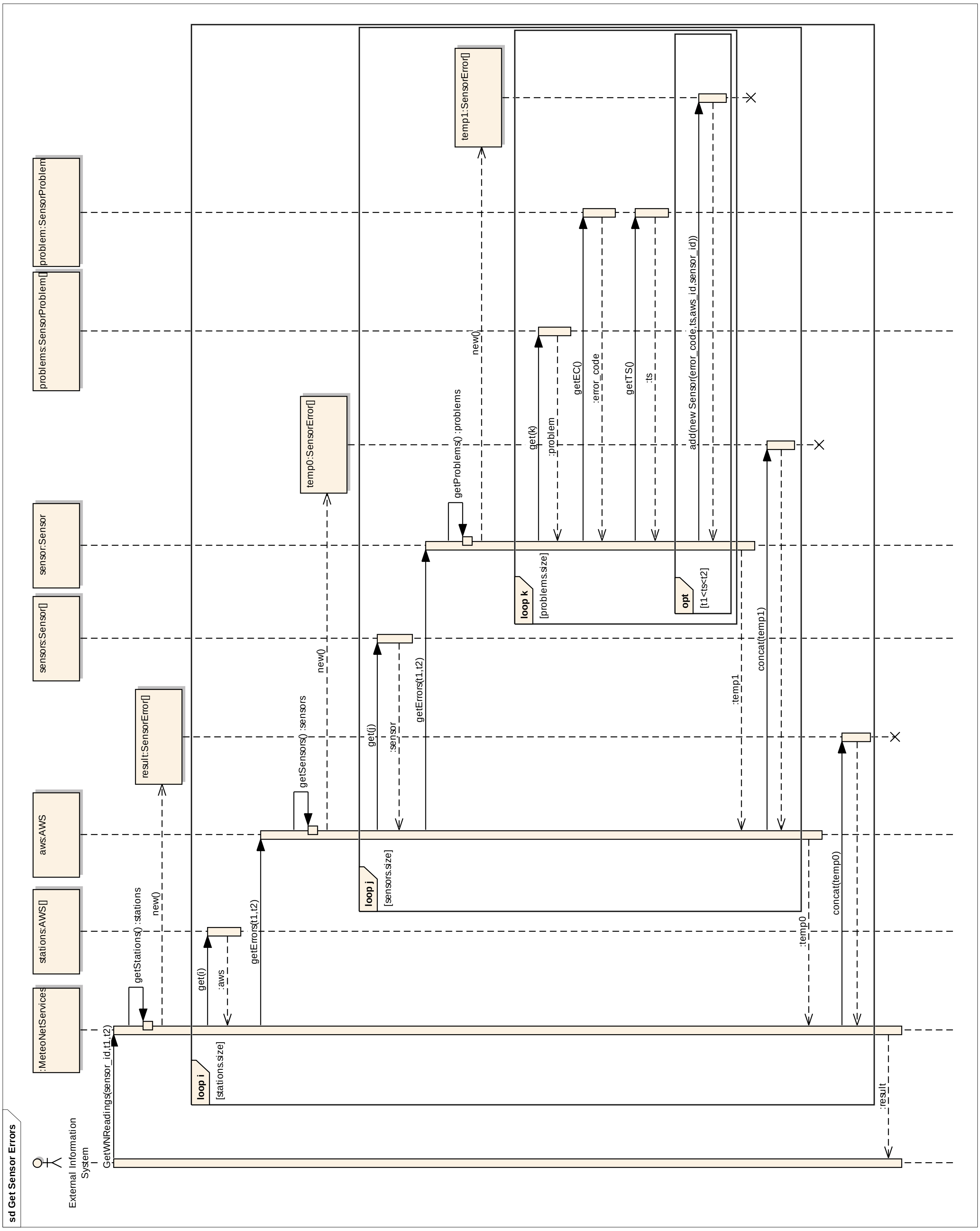

Figure 31 - Sequence diagram for the Get Sensor Errors scenario.

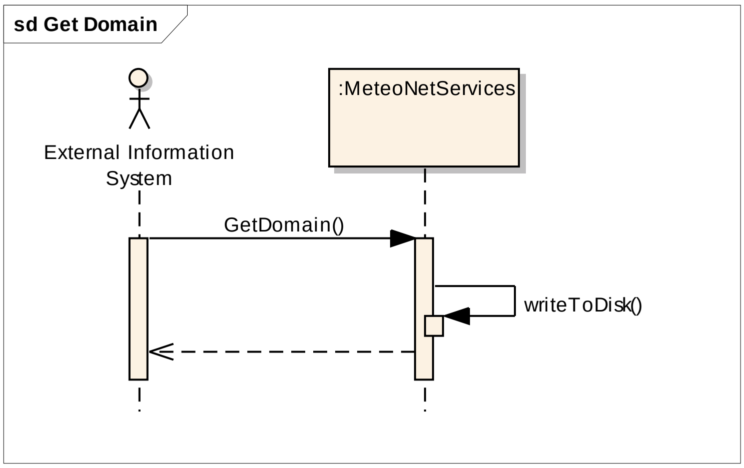

Figure 32- Sequence diagram for the Get Domain scenario.

The following sequence diagrams are related to the UC2-Set state.

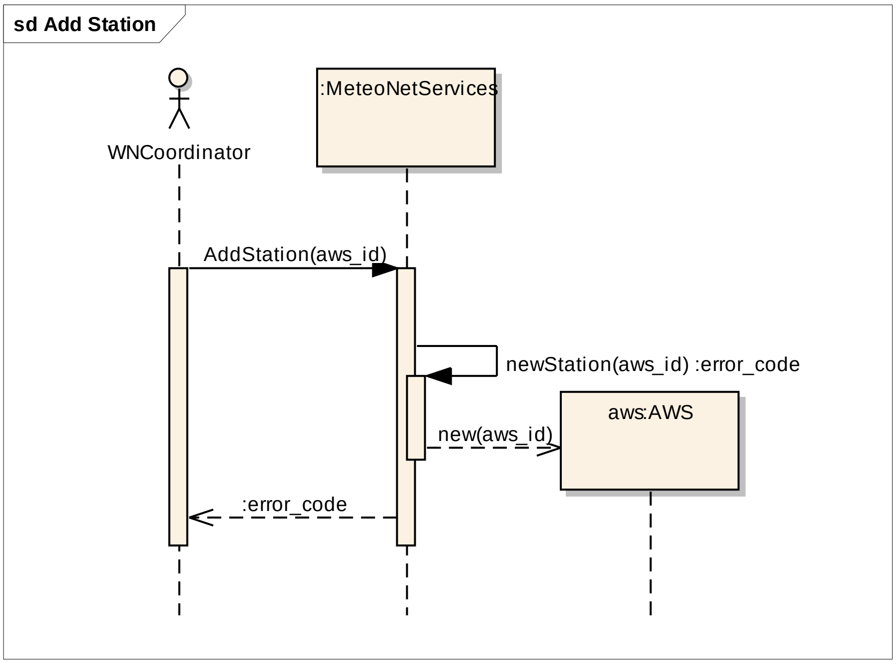

Figure 33- Sequence diagram for the Add Station scenario.

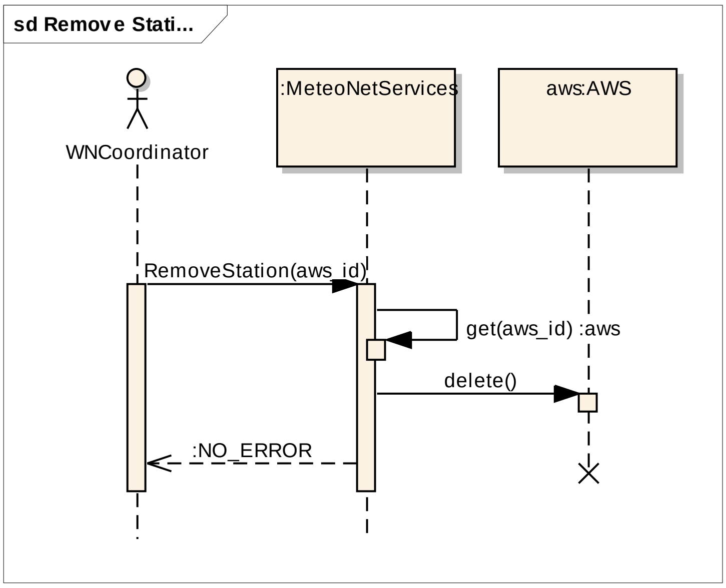

Figure 34 - Sequence diagram for the Remove Station scenario.

The following sequence diagram is related to the UC3-GetInfo.

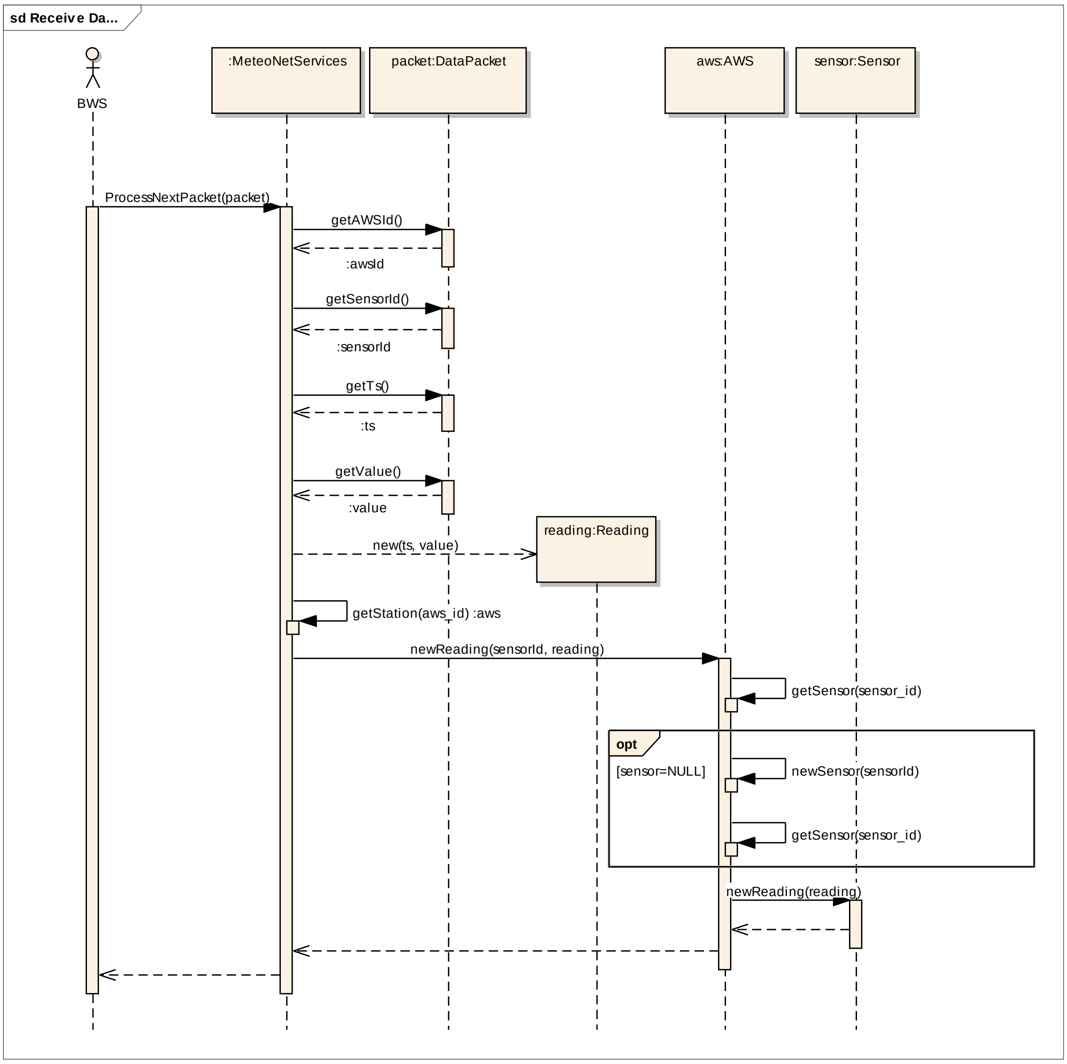

Figure 35- Sequence diagram for the Receive Data scenario.

The following sequence diagram is related to the UC4-SendEmail.

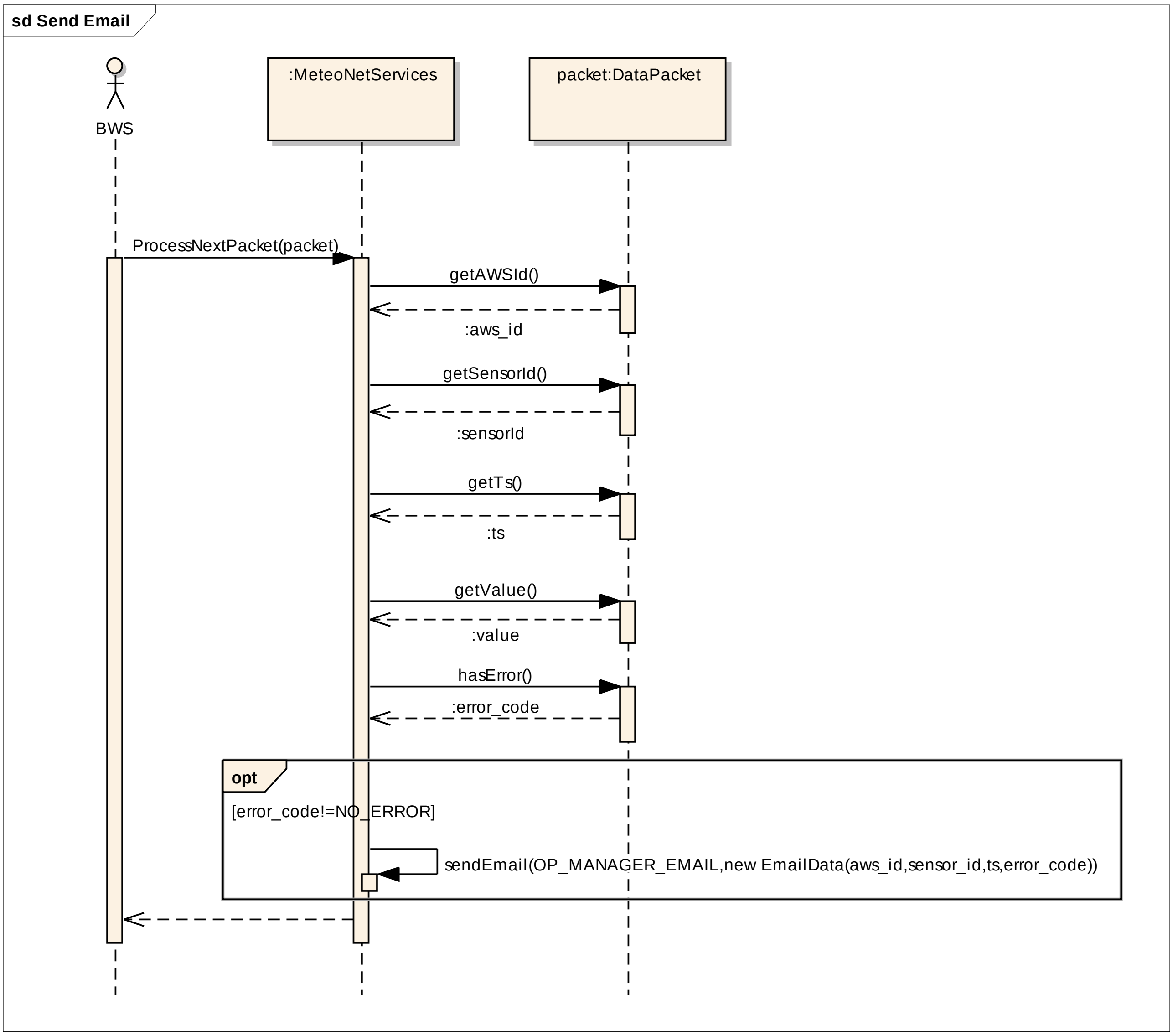

Figure 36- Sequence diagram for the Send Email scenario.

Artefact #18 – MeteoNet states

All the transitions in this diagram are triggered ou guarded by methods represented in the class diagram or in the abstract containers:

- processNextPacket(DataPacket) is a MeteoNetServices method;
- hasError() is a DataPacket method;
- newReading(Reading) is an AWS method;
- newReading(Reading) is a Sensor method;
- newProblem(SensorProblem) is a Sensor method;
- sendEmail(email_address_type, EmailData) is a MeteoNetServices method.

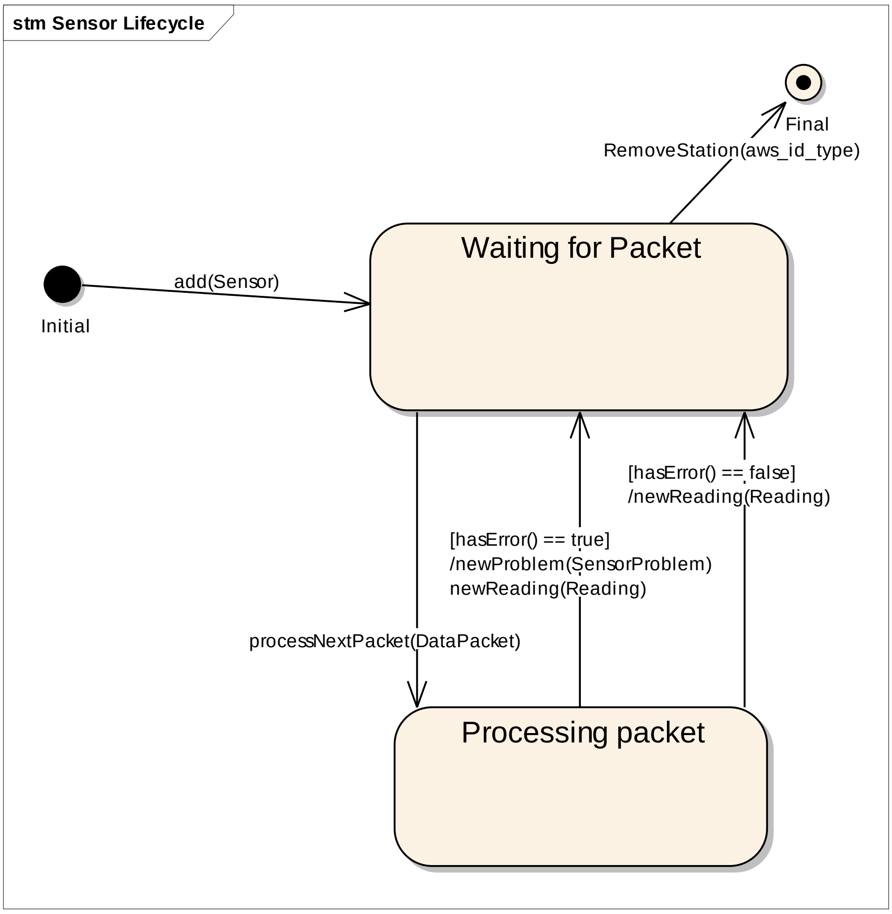

Figure 37- State Machine diagram for the Sensor Lifecycle.

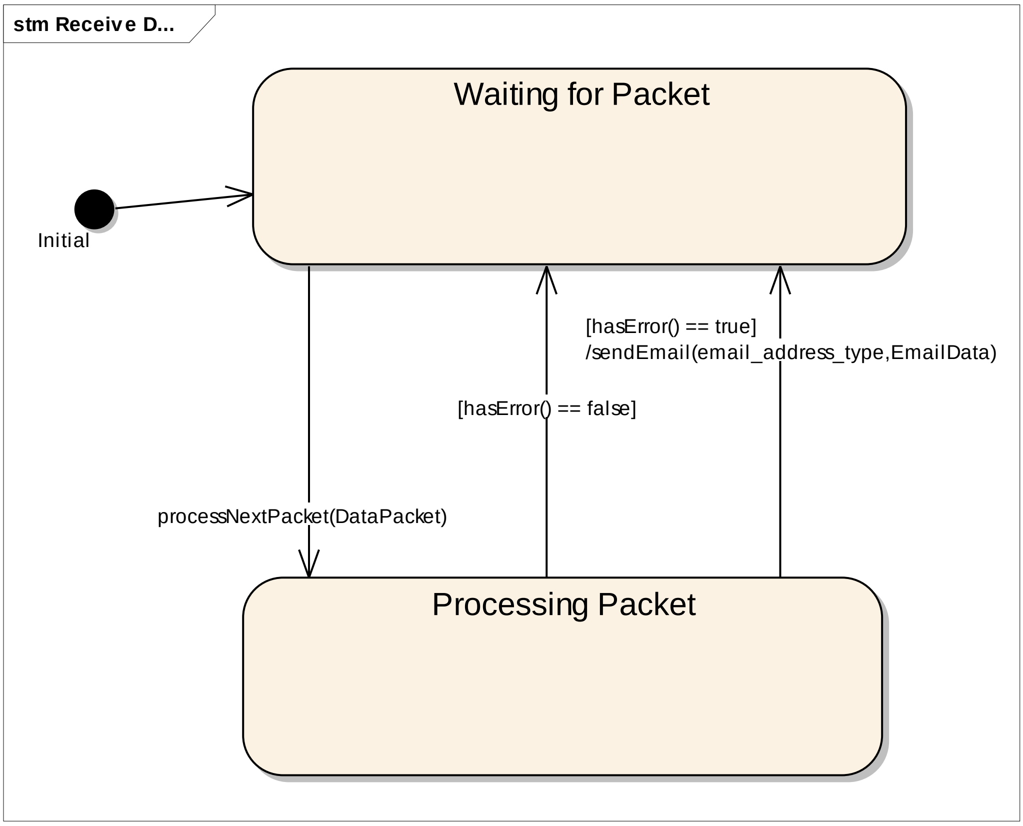

Figure 38- State Machine diagram for Receive Data

Artefact #19 – DWP management process

The WNCoordinator decides when he wants to install or remove an AWS.

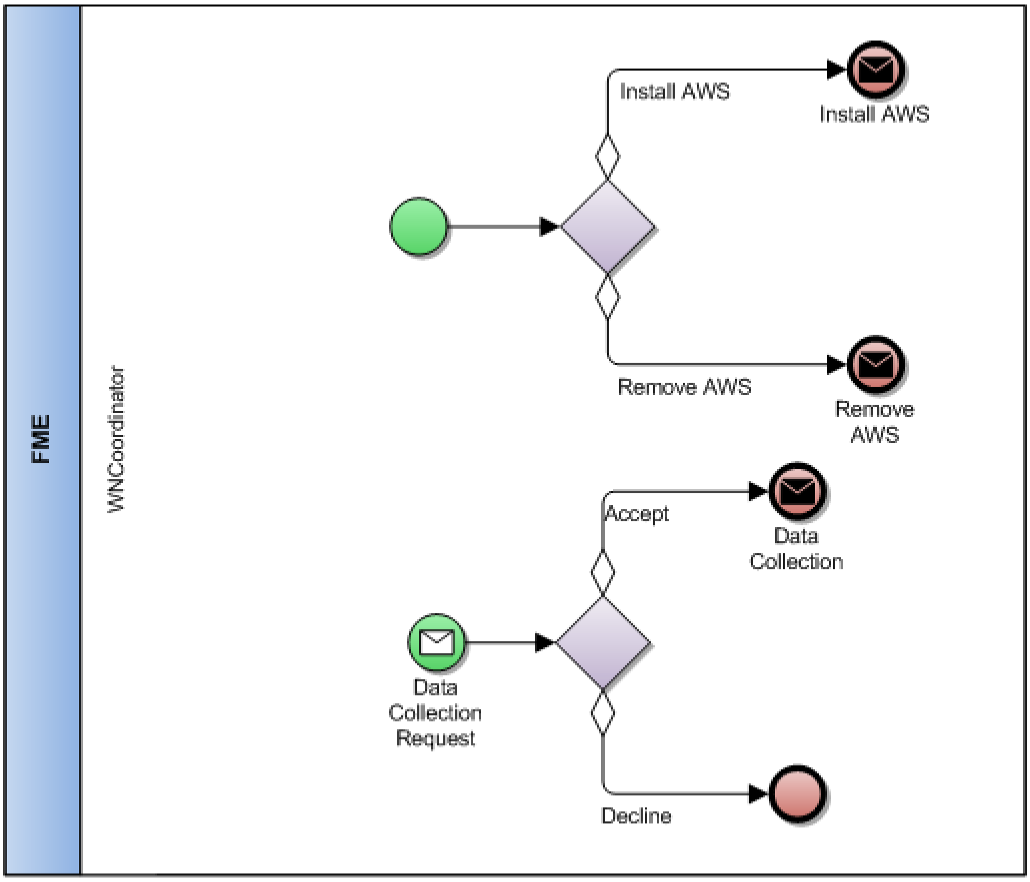

Figure 39- FME pool for DWP management process BPMN diagram.

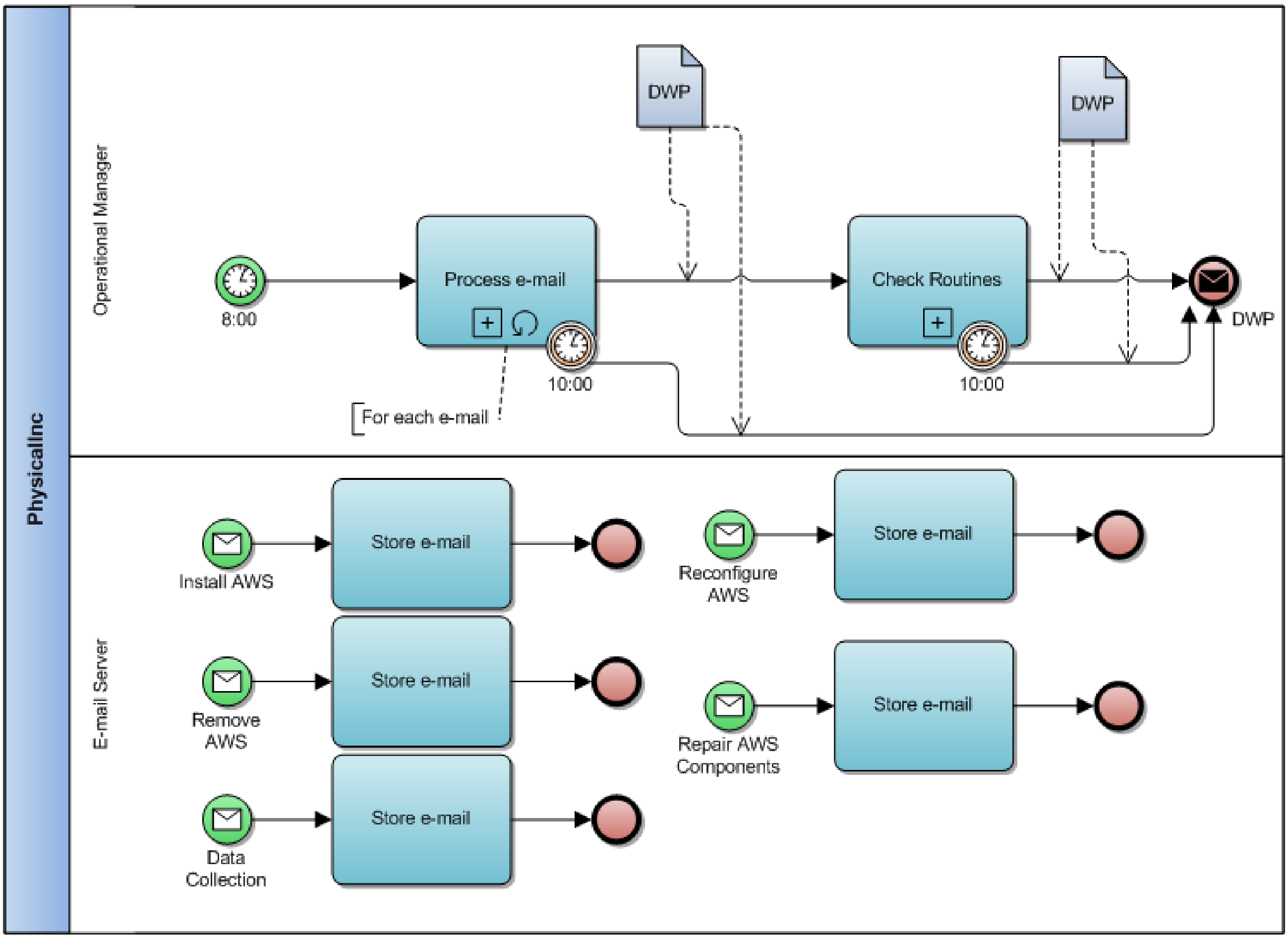

Figure 40- PhysicalInc pool for DWP management process BPMN diagram.

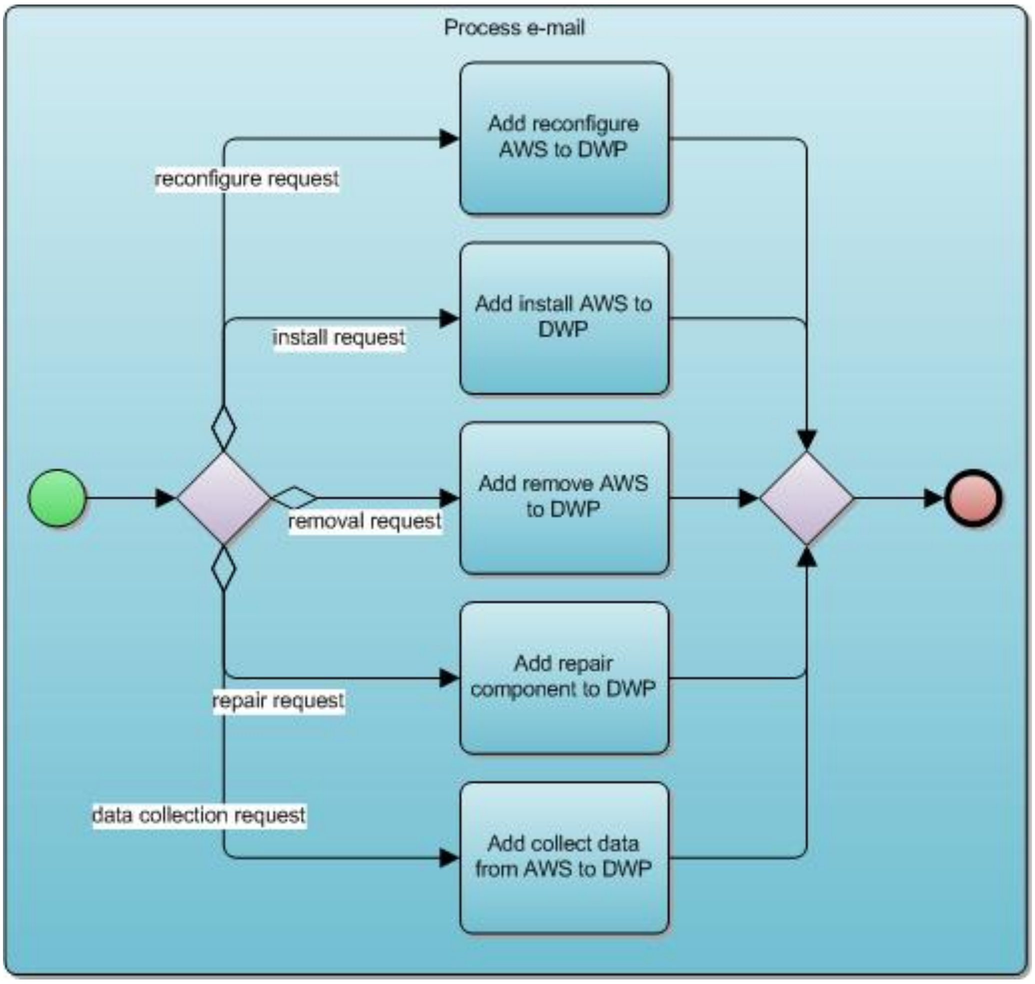

Figure 41- Detailed Process e-mail

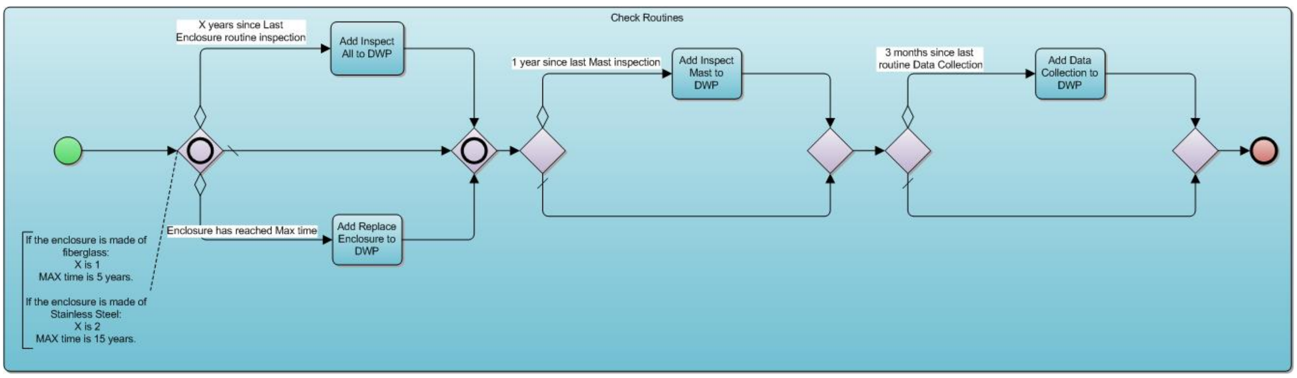

Figure 42 - Detailed Check Routines

Artefact #20 – Installation process

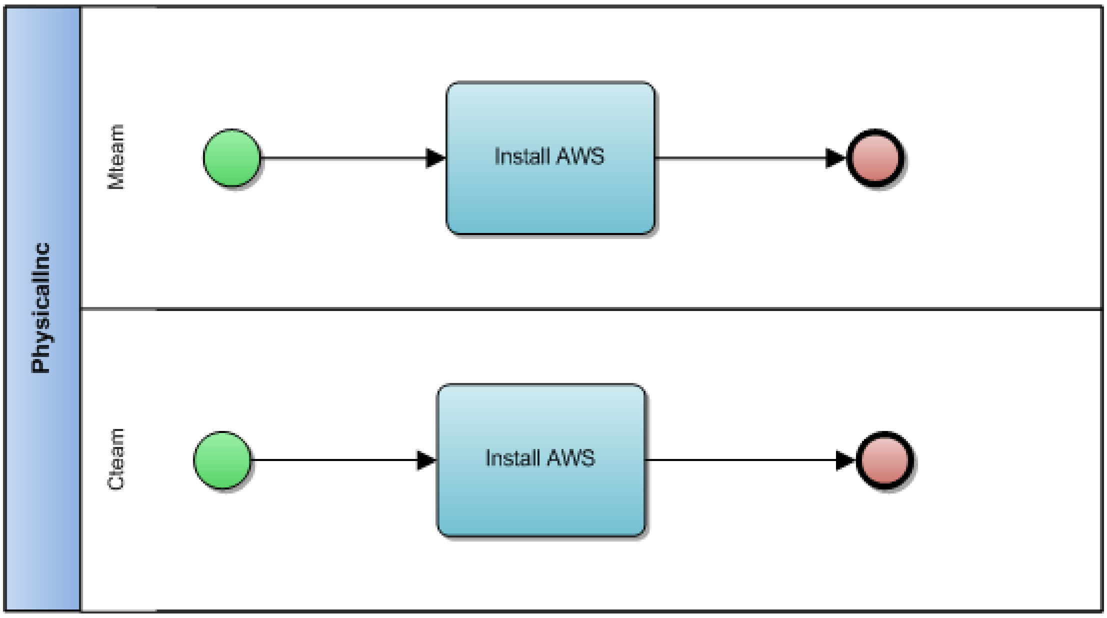

Figure 43- PhysicalInc pool for Instalation process BPMN diagram.

Artefact #21 –Reconfiguration process

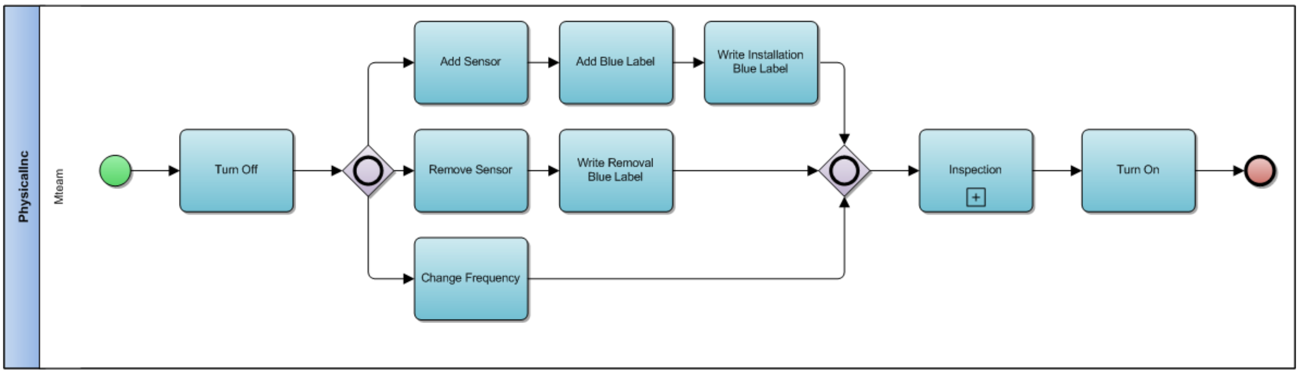

Figure 44 - PhysicalInc pool for Reconfiguration process BPMN diagram.

Artefact #22 – Data collection process

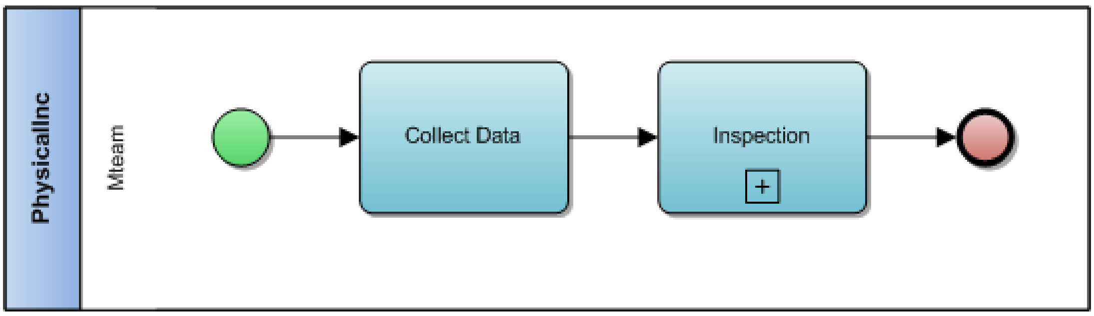

Figure 45 - PhysicalInc pool for Data collection process BPMN diagram.

Artefact #23 – Inspection process

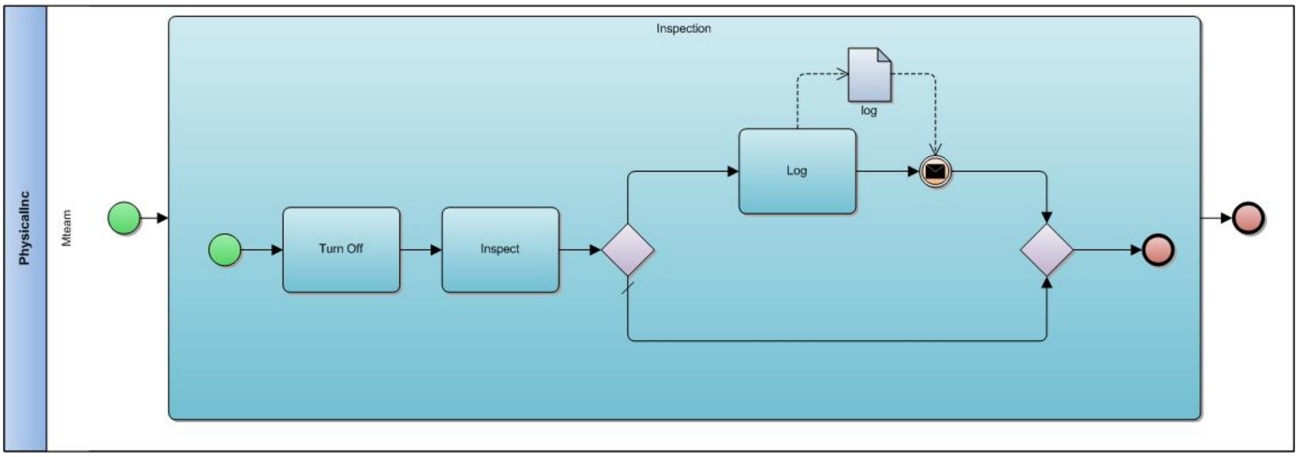

Figure 46 - PhysicalInc pool for Inspection process BPMN diagram and Detailed Inspection.

Artefact #24 – Repair process

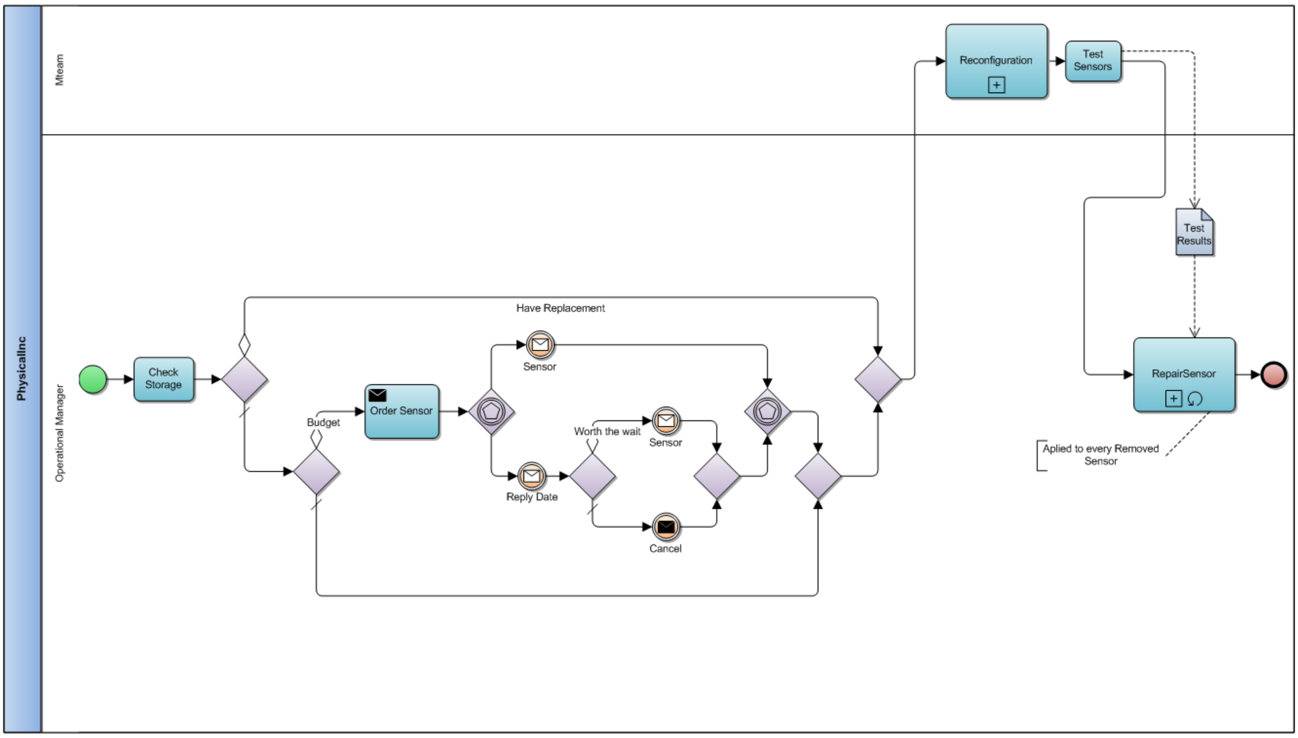

Figure 47-PhysicalInc pool for Repair process BPMN diagram.

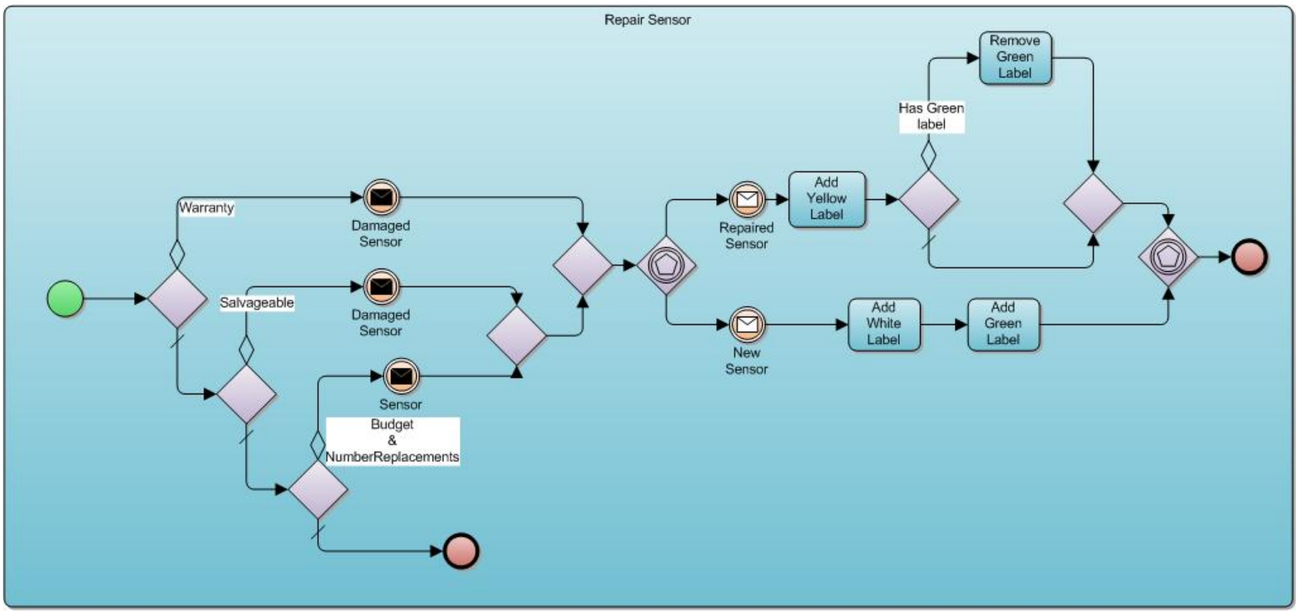

Figure 48- Detailed Repair Sensor

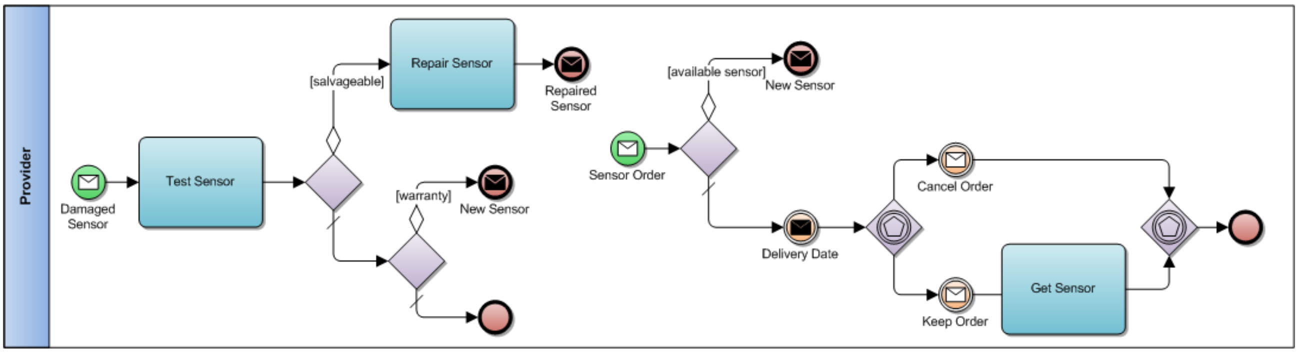

Figure 49 - Provider pool for Repair process BPMN diagram.

Artefact #25 – Removal process

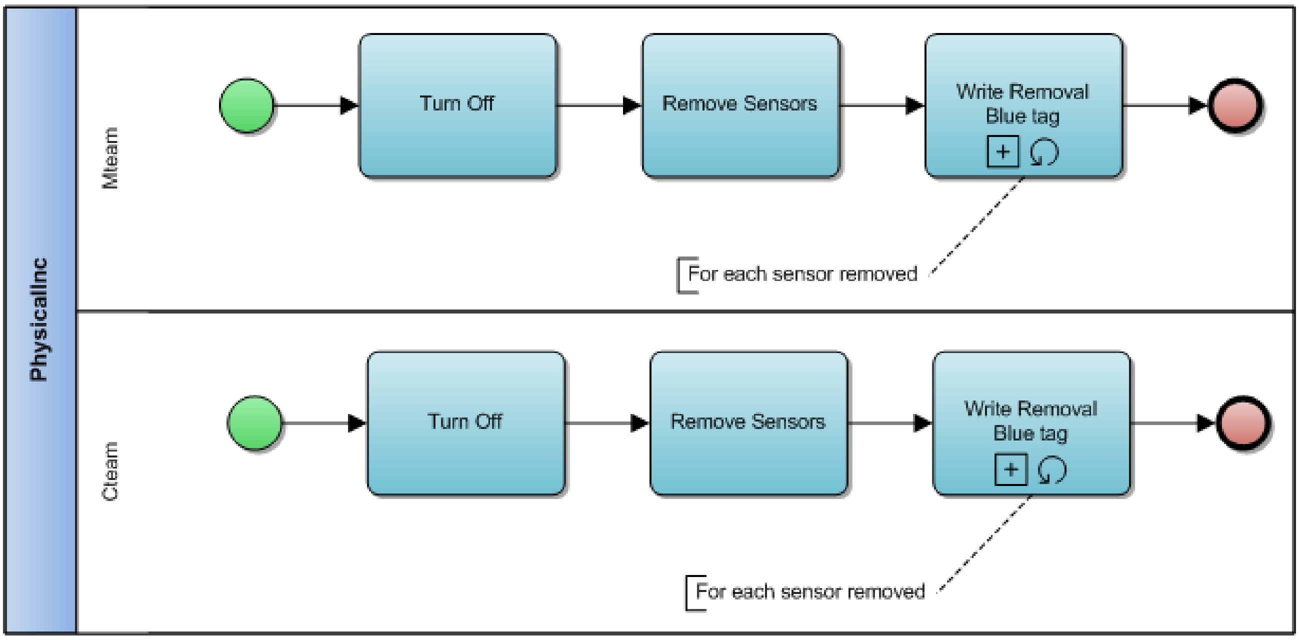

Figure 50 - PhysicalInc pool for Removal process BPMN diagram.

Artefact #26 – AWS traceability matrix

Analysing step by step each use case defined on the other sections of the document, it is possible to draw the following table, showing the relation between the system requirements previously obtained and the use cases that have been imagined and supposed to happen. Note that one use case can use several requirements, but other use cases may only use one or two, although it is fundamental to follow all of them during the implementation to obtain a fully operational final system.

Req	UC1	UC2	UC3	UC4	UC5	UC6
2.9.2					X	
3.1	X			X		
3.2.3	X					
3.2.4	X					
4.1						X
7.1		X	X			
7.2		X	X			

Table 14 - Use cases refining each requirement (traceability matrix) for the AWS.

Req ID	AC Grid	ADC	Analyser	Anemometer	AWS	Barometer	Battery	Bolt	Comuniction Device	Data Logger	Electronic Timer	Enclosure	Enclosure Bolt	GPS	Mast	Photovoltaic Cell	Photometric UV Sensor	Platform	Platform Bolt	Power Source	Power Supply	Power Switch	Rain Gauge	Rectifier	Regulator	Resistive Humidity Sensor	Routing Opening	Sensor	Sensor Hub	Sensor Pack	Tension Cable	Thermistor	Transformer	Wind Turbine	Wind Vane
1					X										X			X																	
1.1															X																				
1.1.1															X																X				
1.1.2															X																X				
1.2																		X																	
1.3								X							X			X																	
2					X							X																							
2.1												X									X														
2.1.1							X														X														
2.1.2																					X				X										
2.1.3																				X	X														
2.2										X		X																							
2.3												X																	X						
2.4												X															X								
2.4.1																											X								
2.4.2																											X	X							
2.4.3																					X						X								
2.4.4																											X	X							
2.6		X									X	X																							
2.7												X																	X						
2.7.1		X																											X						
2.7.2										X																			X						
2.8										X		X																							
2.8.1										X																									
2.8.2										X																									
2.9							X					X												X	X								X		
2.9.1							X																												
2.9.3																X									X										
2.9.4																									X									X	
2.9.5	X																								X										
2.9.6	X																																X		
2.9.7												X				X																		X	
2.9.8			X				X																												
2.9.8.1			X																																
2.9.8.2			X																																
2.12												X	X																						
3					X																							X							
3.1.1				X																															X
3.1.2						X																	X												
3.1.3														X																					
3.2		X																										X							
3.2.1		X																																	
3.2.2		X																											X						
3.2.5																														X					
3.3																												X							
3.4				X	X	X								X									X			X									X
3.5														X			X																		
3.6												X																X							
4					X				X																										
4.2									X																				X						
7					X																	X													

Table 15 - Blocks refining each requirement (traceability matrix) for the AWS.

Artefact #27 – MeteoNet traceability matrix

As explained on the last artefact, but now for the MeteoNet software interactions.

Req ID	UC1	UC2	UC3	UC4
1			X	
2				X
4,1		X		
4.1.1		X		
4.1.2		X		
4.2	X			
4.2.1	X			
4.2.2	X			
4.2.3	X			
4.2.4	X			
4.2.5	X			
4.2.6	X			
4.2.7	X			

Table 16 - Use cases refining each requirement (traceability matrix) for the MeteoNet.

Req ID	AWS	Sensor	SensorProblem	Reading	MeteoNetServices
3	X				X
3.1	X	X			
3.2		X	X		
3.2.1		X			
3.3		X	X		
3.3.1			X		
3.4		X		X	
3.4.1				X	
3.5	X				
3.6	X				
4					X

Table 176 - Classes refining each requirement (traceability matrix) for the MeteoNet.

When more than one class meets a requirement it means that the structural relation between those classes meets the requirement.

Artefact #28 – Architectural Description

Artefact 1 – Effort and work-breakdown

Artifact	1 - Effort and work-breakdown
System of Interest	---
Concern(s)	Organize project development
Viewpoint	Set of rules to read and create tables
View	Table
Stakeholder(s)	BestModelInc
Technique	Table
Notation	Excel Table
Architect	Miguel Aragão

Table 18- Architectural Description of artefact 1

Artefact 2 – AWS context model

Artifact	2 - AWS context
System of Interest	AWS
Concern(s)	Support the making of other artifacts, help understanding the AWS context
Viewpoint	Set of rules to read and create conceptual maps
View	Conceptual map
Stakeholder(s)	BestModelInc, PhysicalInc
Technique	Concept Map
Notation	Cmap Tools 5.04.02 Lite
Architect	Cláudia Henriques

Table 19 - Architectural Description of artefact 2

Artefact 3 – AWS requirements

Artifact	3 - AWS Requirements
System of Interest	AWS
Concern(s)	Support the making of other artifacts, support AWS installation
Viewpoint	English grammar rules
View	Text
Stakeholder(s)	BestModelInc, PhysicalInc
Technique	Table
Notation	Excel Table
Architect	Daniel Cardoso

Table 20 - Architectural Description of artefact 3

Artefact 4 – AWS use cases

Artifact	4 - AWS use cases
System of Interest	AWS
Concern(s)	Support the making of other artifacts
Viewpoint	SysML 1.2 specification
View	SysML use case diagram
Stakeholder(s)	BestModelInc
Technique	SysML use case diagram
Notation	SysML,template EA
Architect	Cláudia Henriques

Table 21- Architectural Description of artefact 4

Artefact 5 – AWS use cases scenarios

Artifact	5 - AWS use cases scenarios
System of Interest	AWS
Concern(s)	Support the making of other artifacts
Viewpoint	Set of rules to read and create tables
View	Table
Stakeholder(s)	BestModelInc
Technique	Table
Notation	Excel Table
Architect	Daniel Cardoso

Table 22- Architectural Description of artefact 5

Artefact 6 – BWS context model

Artifact	6 – BWS context model
System of Interest	BWS
Concern(s)	Support the making of other artifacts, help understanding the BWS context
Viewpoint	English grammar rules
View	Text
Stakeholder(s)	BestModelInc
Technique	Table
Notation	Excel Table
Architect	Miguel Aragão

Table 23- Architectural Description of artefact 6

Artefact 7 – MeteoNet context model

Artifact	7 - MeteoNet context model
System of Interest	MeteoNet
Concern(s)	Support the making of other artifacts, help understanding the MeteoNet context
Viewpoint	Set of rules to read and create conceptual maps
View	Conceptual map
Stakeholder(s)	BestModelInc, LogicInc
Technique	Concept Map
Notation	Cmap Tools 5.04.02 Lite
Architect	Francisco Raposo

Table 24- Architectural Description of artefact 7

Artefact 8 – MeteoNet requirements

Artifact	8 - MeteoNet requirements
System of Interest	MeteoNet
Concern(s)	Support the making of other artifacts, support MeteoNet design and implementation
Viewpoint	English grammar rules
View	Text
Stakeholder(s)	BestModelInc, LogicInc
Technique	Table
Notation	Excel Table
Architect	Miguel Aragão

Table 25- Architectural Description of artefact 8

Artefact 9 – MeteoNet use cases

Artifact	9 - MeteoNet use cases
System of Interest	MeteoNet
Concern(s)	Support the making of other artifacts
Viewpoint	UML 2.2 specification
View	UML use case diagram
Stakeholder(s)	BestModelInc
Technique	SysML use case diagram
Notation	UML 2.2, template EA
Architect	Francisco Raposo

Table 26- Architectural Description of artefact 9

Artefact 10 – MeteoNet use cases scenarios

Artifact	10 - MeteoNet use cases scenarios
System of Interest	MeteoNet
Concern(s)	Support the making of other artifacts
Viewpoint	Set of rules to read and create tables
View	Table
Stakeholder(s)	BestModelInc
Technique	Table
Notation	Excel Table
Architect	Miguel Aragão

Table 27- Architectural Description of artefact 10

Artefact 11 – AWS structure

Artifact	11 - AWS structure
System of Interest	AWS
Concern(s)	Support AWS construction/installation
Viewpoint	SysML 1.2 specification
View	SysML bdd, ibd diagrams
Stakeholder(s)	BestModelInc, PhysicalInc
Technique	BDD/ IBD diagrams
Notation	SysML 1.2, template Visio
Architect	Daniel Cardoso

Table 28- Architectural Description of artefact 11

Artefact 12 – MeteoNet classes

Artifact	12 - MeteoNet classes
System of Interest	MeteoNet
Concern(s)	Design MeteoNet software
Viewpoint	UML 2.2 specification
View	UML classes diagram
Stakeholder(s)	BestModelInc, LogicInc
Technique	UML class diagram
Notation	UML 2.2, template Visio
Architect	Francisco Raposo

Table 29- Architectural Description of artefact 12

Artefact 13 – MeteoNet components

Artifact	13 - MeteoNet components
System of Interest	MeteoNet
Concern(s)	Help understanding MeteoNet components interaction
Viewpoint	UML 2.2 specification
View	UML components diagram
Stakeholder(s)	BestModelInc, LogicInc
Technique	UML components diagram
Notation	UML 2.2, template Visio
Architect	Cláudia Henriques

Table 30- Architectural Description of artefact 13

Artefact 14 – MeteoNet deployment

Artifact	14 - MeteoNet deployment
System of Interest	MeteoNet
Concern(s)	Help understanding MeteoNet deployment context
Viewpoint	UML 2.2 specification
View	UML deployment diagram
Stakeholder(s)	BestModelInc, LogicInc
Technique	Table
Notation	Excel Table
Architect	Miguel Aragão

Table 31- Architectural Description of artefact 14

Artefact 15 – AWS behaviour

Artifact	15 - AWS behaviour
System of Interest	AWS
Concern(s)	Help understanding AWS blocks interaction
Viewpoint	SysML 1.2 specification
View	SysML sequence diagram
Stakeholder(s)	PhysicalInc
Technique	SysML sequence diagram
Notation	SysML 1.2, template EA
Architect	Daniel Cardoso

Artefact 16 – AWS states

Artifact	16 - AWS States
System of Interest	AWS
Concern(s)	Help understanding AWS states behaviour
Viewpoint	SysML 1.2 specification
View	SysML state machine diagram
Stakeholder(s)	PhysicalInc
Technique	SysML state machine diagram
Notation	SysML 1.2, template EA
Architect	Miguel Aragão

Artefact 17 – MeteoNet behaviour

Artifact	17 - MeteoNet behaviour
System of Interest	MeteoNet
Concern(s)	Help understanding MeteoNet software classes interaction with each other
Viewpoint	UML 2.2 specification
View	UML sequence diagram
Stakeholder(s)	LogicInc
Technique	UML sequence diagram
Notation	UML 2.2, template EA
Architect	Cláudia Henriques

Artefact 18 – MeteoNet states

Artifact	18 - MeteoNet States
System of Interest	MeteoNet
Concern(s)	Help understanding MeteoNet software behaviour (states/transitions)
Viewpoint	UML 2.2 specification
View	UML state machine diagram
Stakeholder(s)	LogicInc
Technique	UML state machine diagram
Notation	UML 2.2 , template EA
Architect	Francisco Raposo

Artefact 19 -DWP management process

Artifact	19 -DWP management process
System of Interest	AWS
Concern(s)	Help understanding AWS DWP management process
Viewpoint	BPMN 2.0 specification
View	BPMN diagram
Stakeholder(s)	LogicInc
Technique	BPMN diagram
Notation	BPMN 2.0, template VISIO
Architect	Daniel Cardoso

Artefact 20 -Installation Process

Artifact	20 -Installation Process
System of Interest	AWS
Concern(s)	Help understanding AWS installation process
Viewpoint	BPMN 2.0 specification
View	BPMN diagram
Stakeholder(s)	LogicInc
Technique	BPMN diagram
Notation	BPMN 2.0, template VISIO
Architect	Cláudia Henriques

Artefact 21 -Reconfiguration Process

Artifact	21 -Reconfiguration Process
System of Interest	AWS
Concern(s)	Help understanding AWS reconfiguration process
Viewpoint	BPMN 2.0 specification
View	BPMN diagram
Stakeholder(s)	LogicInc
Technique	BPMN diagram
Notation	BPMN 2.0, template VISIO
Architect	Cláudia Henriques

Artefact 22 -Data Collection Process

Artifact	22 -Data Collection Process
System of Interest	AWS
Concern(s)	Help understanding AWS data collection process
Viewpoint	BPMN 2.0 specification
View	BPMN diagram
Stakeholder(s)	LogicInc
Technique	BPMN diagram
Notation	BPMN 2.0, template VISIO
Architect	Miguel Aragão

Artefact 23 -Inspection Process

Artifact	23 -Inspection Process
System of Interest	AWS
Concern(s)	Help understanding AWS inspection process
Viewpoint	BPMN 2.0 specification
View	BPMN diagram
Stakeholder(s)	LogicInc
Technique	BPMN diagram
Notation	BPMN 2.0, template VISIO
Architect	Miguel Aragão

Artefact 24 -Repair Process

Artifact	24 -Repair Process
System of Interest	AWS
Concern(s)	Help understanding AWS repair process
Viewpoint	BPMN 2.0 specification
View	BPMN diagram
Stakeholder(s)	LogicInc
Technique	BPMN diagram
Notation	BPMN 2.0, template VISIO
Architect	Francisco Raposo

Artefact 25 -Removal Process

Artifact	25 -Removal Process
System of Interest	AWS
Concern(s)	Help understanding AWS removal process
Viewpoint	BPMN 2.0 specification
View	BPMN diagram
Stakeholder(s)	LogicInc
Technique	BPMN diagram
Notation	BPMN 2.0, template VISIO
Architect	Daniel Cardoso

Artefact 26 – AWS traceability matrix

Artifact	26 - AWS traceability matrix
System of Interest	AWS
Concern(s)	Verify if requirements are met
Viewpoint	Set of rules to read and create tables
View	Table
Stakeholder(s)	BestModelInc
Technique	Table
Notation	Excel Table
Architect	Daniel Cardoso

Artefact 27 – MeteoNet traceability matrix

Artifact	27 - MeteoNet traceability matrix
System of Interest	MeteoNet
Concern(s)	Verify if requirements are met
Viewpoint	Set of rules to read and create tables
View	Table
Stakeholder(s)	BestModelInc
Technique	Table
Notation	Excel Table
Architect	Francisco Raposo

Artefact 29 – Change Log

Artifact	29 - Change log
System of Interest	---
Concern(s)	Trace changes occurred during project development
Viewpoint	English grammar rules
View	Text
Stakeholder(s)	BestModelInc
Technique	Text
Notation	Word Text
Architect	Cláudia Henriques

Artefact #29 – Change Log

First to Second:

<u>Artefact 2</u>

- Some minor adjustments in the context diagram.

<u>Artefact 3</u>

- Some of the AWS Requirements were changed as well as some added.

<u>Artefact 4</u>

- Use Cases were also changed: some of the previous Use Cases were merged in a new one; some Use Cases, that hadn't been considered, were added.

<u>Artefact 5</u>

- As we changed the use cases we had to change the use case scenarios.

Artefact 6

- The WN Context model was added as requested.

Artefact 7

- The MeteoNet context model was incomplete so it was changed.

Artefact 8

- Some requirements were changed/added.

Artefact 9

- We were defining the uses cases in the wrong level, so we changed that. The previous UCs were merged in two new ones.

Artefact 10

- As we changed the use cases we had to change the use case scenarios.

Artefact 26

- As the requirements and uses chases were changed, the traceability matrix had to be changed.

Artefact 27

- As the requirements and uses chases were changed, the traceability matrix had to be changed.

Artefact 28

- We opted to model this artefact as a table set as recommended.

Second to Third:

Artefact 2

- Some minor adjustments.

Artefact 3

- Some requirements are now just one requirement.

Artefact 4

- Some changes regarding use cases (names, actors...).

Artefact 5

- As the use cases were changed, these were also corrected.

Artefact 6

- Some minor adjustments (in both diagrams).

Artefact 7

- Some minor adjustments.

Artefact 8

- Two requirements were removed due to the Project Clarifications v2.

Artefact 9

- Use cases added.

Artefact 10

- Same as for Artefact 5.

Artefact 11

- Changes regarding components, operations, attributes, flowports, types and allocations.

Artefact 12

- Some changes in the overall structure and several changes regarding types and classes' attributes and methods.

Artefact 13

- MeteoNet components added.

Artefact 14

- File System was missing, now added.

Artefact 26

- Added traceability to non-functional requirements and minor changes regarding the traceability between use cases and functional requirements.

Artefact 27

- Same as for Artefact 26.

Artefact 28

- Added new rows for each table as suggested.

Third to Forth(Final):

<u>Artefact 2</u>

.

<u>Artefact 4</u>

<u>Artefact 5</u>

.

<u>Artefact 6</u>

<u>Artefact 7</u>

<u>Artefact 8</u>

<u>Artefact 9</u>

<u>Artefact 10</u>

<u>Artefact 11</u>

- Allocations corrected as well as some flowPorts.

<u>Artefact 12</u>

- Data types in a separate diagram. Also, to represent a collection of objects we use abstract arrays (notation Type[]) instead of specifying if it is a map or a list. This change had several effects regarding changes in other MeteoNet artefacts.

<u>Artefact 13</u>

- The MeteoNet software is now considered to be a unique component itself.

<u>Artefact 14</u>

- The MeteoNet software is the only component that is deployed in the server. The file system, AWS Driver and the Linux OS are already inside the server.

<u>Artefact 15:</u>

- Corrected when blocks called method on actors. Variables and Return types corrected. Added loop to get the data to more than one Data logger, if it exists.

<u>Artefact 16:</u>

- Corrected the composite states ON and Operational and presentation of the stm in report improved.

Artefact 17:

- Changes made according to the artefact 12 changes.

Artefact 18:

- Now, we only consider 2 states in each one of the state machine diagrams.

Artefact 26

- Tables aspect in report improved.

Artefact 27

Artefact 28

- Added new rows for each table as suggested.