

E2District

Deliverable 2.1 Detailed Specifications on Interoperability

Dissemination Level: PU

This project is partially funded by the European Commission under the H2020 Framework Programme - Grant Agreement no. 696009

Project Number	:	696009
Project Title	:	Energy Efficient Optimised District Heating and Cooling – E ² District
Deliverable Dissemination Level	:	PU

Deliverable Number	:	D2.1
Title of Deliverable	:	Detailed Specifications on Interoperability
Nature of Deliverable	:	R
Internal Document Number	:	E2D_D2.1_WP2
Contractual Delivery Date	:	m8-30/9/2016
Actual Delivery Date	:	31/04/2017
Work Package	:	WP2
Author(s)	:	V. Partenay, G. Manganini, D. Mouquet, J. Fournier, T. Lecomte, C. Beder, G. Naves-Maschietto, S. Couturier
Total number of pages (including cover)	:	38

Abstract

This document presents technical specification for interoperability between the different simulation platforms developed in the project that will interact together via web-services

Keyword list

Interoperability, Cloud based platform, Simulation tool, Optimized Controller, Predictive scheduling

Document History

Date	Revision	Comment	Author/Editor	Affiliation
01/07/2016	0	Draft structure	V. Partenay	CSTB
28/09/2016	1	Draft update	V. Partenay	CSTB
09/11/2016	1	Specs from UTRC	G. Manganini	UTRC
21/11/2016	1	Contributions	D. Mouquet, J. Fournier	VERI
22/11/2016	1	Contributions	T. Lecomte	VERI
25/11/2016	2	Merge/Update	V. Partenay	CSTB
20/12/2016	3	New structure	V. Partenay	CSTB
24/01/2017	3	Specs from CIT	C. Beder	CIT
06/02/2017	3	Specs from VERI	G. Naves-Maschietto	VERI
16/03/2017	4	Merge/ First Final Version	V. Partenay	CSTB
24/03/2017	4	Amendments from VERI	S. Couturier	VERI
27/03/2017	4	Final Document	V. Partenay	CSTB
30/03/2017	4	Final Review	M. Cychowski	UTRC
03/04/2017	4	Final Comments and changes page 12	K. Kouramas	UTRC
04/04/2017	4	Last amendments from VERI	G. Naves-Maschietto	VERI
07/04/2017	5	Integration of final comments/ adding summary	V. Partenay	CSTB

Executive Summary

This report outlines the Detailed Specifications on Interoperability between tools from different partners involved in E²District project.

The global objective of E²District project is to deploy, validate and demonstrate a cloud-based district management and decision support framework for DHC (District Heating & Cooling) systems in order to improve the energy, environmental and economic performance. This framework shall cover the whole lifecycle of such systems, including concept, design and operation. That leads to a high complexity of potential interactions between the various key stakeholders tools or methodology within the DHC value chain. These interactions need to be taken into account even more consistently when a collateral feature of the project is to literally “connect” several processes through a common web platform.

This report associated to Task 2.1 addresses this harmonization process for the specific case of the different solicitations that District Simulation Platform (developed within WP2) shall have to support throughout the project. As illustrated on Figure 1 this work has been closely done with WP3 requirements and aims to prepare as best as possible the integration of these simulation features in WP4.

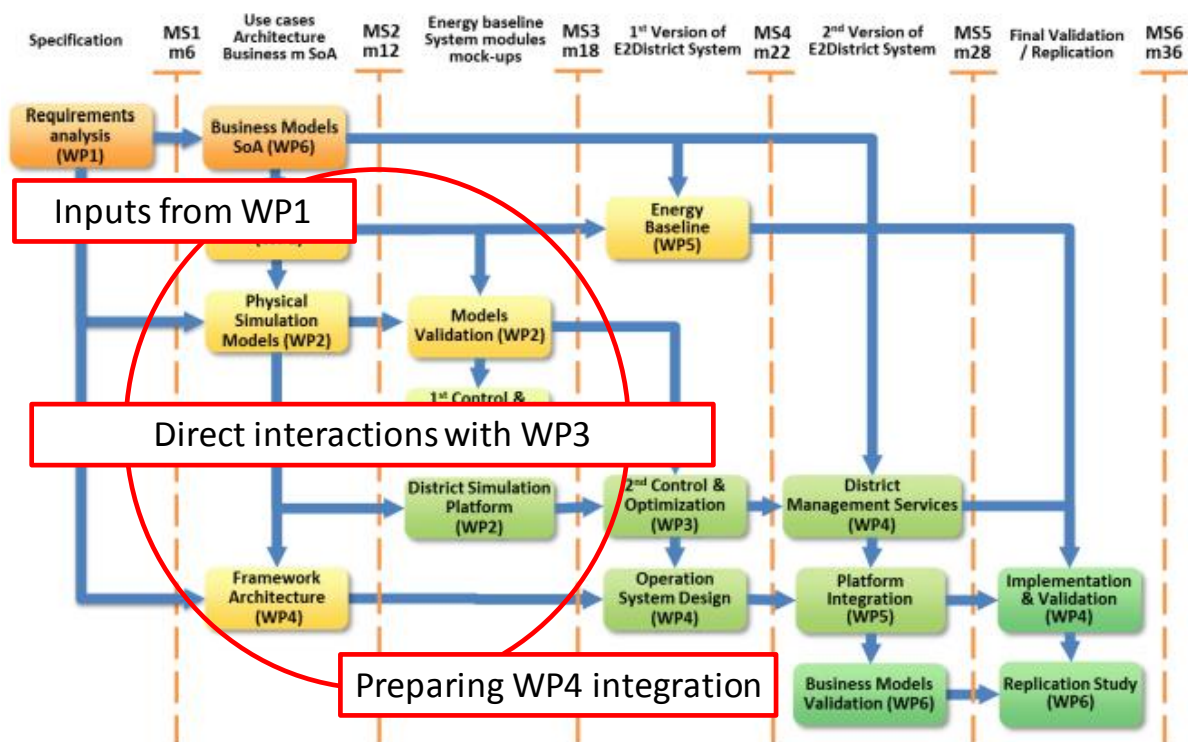


Figure 1: Position of WP2 into E2District Workflow

Detailed technical specifications are specified into this report for interactions between the following different platform/interfaces:

- District Simulation Platform (**DSP**), developed by CSTB
- Supervisory Controller (**SC**), developed by UTRC
- Production Scheduling Optimizer (**PSO**), developed by VERI
- District Operation System (**DOS**) & integration in Nicore, developed by CIT

Table of Contents

1	Introduction – Objectives of Task 2.1	2
2	Overview of involved platforms	3
2.1	District Simulation Platform (DSP)	3
2.1.1	Model Instantiation	4
2.1.2	Co-simulation	5
2.1.3	Calculated Variables - Results	6
2.2	Supervisory Controllers (SC)	7
2.2.1	Tasks	7
2.2.2	Inputs/Outputs	8
2.2.3	Time scale – Sampling Time	9
2.3	Production Scheduling Optimizer (PSO)	10
2.3.1	Tasks	10
2.3.2	Inputs/Outputs	10
2.3.3	Time scale (time steps, time horizon)	11
2.4	Cloud-based integrated platform DOS (CIT - NiCore)	12
3	Specifications on Interoperability	13
3.1	Global Architecture	13
3.2	Description of Use Cases	14
3.2.1	SC using DSP as a testbed	14
3.2.2	PSO using DSP as a thermal hub load forecast engine	15
3.2.3	Interactions through DOS	16
3.2.4	Cross platform summary	17
3.3	Data Format	18
3.3.1	District Information File	18
3.3.2	Weather	18
3.3.3	Data Exchange Format between Platforms	19
3.3.3.1	Use Case SC/DSP	19
3.3.3.2	Use Case PSO/DSP	21
3.3.3.3	Request for simulation and initialisation	22
4	Conclusion	23
5	Appendix A (CIT example)	24
6	Appendix B: Correspondence with project KPIs (T1.3)	33

1 Introduction – Objectives of Task 2.1

A District Simulation Platform is being developed and adapted, as a centralized physical simulation engine for buildings and DHC (District Heating & Cooling) networks that will be used in other work packages in E2District Project. Though this platform already presents several capabilities in terms of dynamic energy simulation, with an existing library of basic energy systems configurations, the main objective of WP2 is to develop and implement innovative components more closely related to DHC, and well-suited for E2District technical solutions.

Nevertheless, if the elaboration of new physical models remains the core of this work package, an inherent transversal objective consists in its ability to operate with other platforms developed in the project, in particular to avoid redundant development of separated simulation tools. Then the challenge is to develop and deploy a physical simulation engine so that it can be made available to the other external platforms that would make use of it as a service.

Task 2.1 directly addresses this issue, and the aim of this report is to provide detailed technical specifications on the different interfaces to be considered between the following platforms :

- Supervisory Controller (**SC**), developed by UTRC
- Production Scheduling Optimizer (**PSO**), developed by VERI
- District Simulation Platform (**DSP**), developed by CSTB

These platforms shall interact through a common managing cloud-based platform developed by CIT (Nicore - **DOS**).

Furthermore, interoperability specifications tend to cover the three following dimensions:

➤ *Model Instantiation*

Simulations will be carried out on different districts that present their own characteristics (building geometry, thermal properties, energy systems, piping network, weather, etc...). Model instantiation regroups all these sets of parameters that shall be gathered in a common global semantic-based input file

➤ *Simulation (or co-simulation)*

A major expected feature of the project is to allow testing of optimized control strategies directly with the DSP before applying them on real sites. As sophisticated as they could be, control strategies all rely on the collection of various states to deliver the most appropriate set of commands in real-time. Interoperability here deals with the possibility of data exchange (co-simulation) at each time-step between DSP and control algorithms.

➤ *Post-processed Results*

Simulations results consist in several time-series (building thermal loads, gas and electricity consumptions, electricity production, etc...) that need to be post-processed to be consistent with the expected KPIs of the project. In the same manner than for model instantiation, results have to respect a common-shared format that also requires external conversion values such as primary energy conversion factor, energy costs or grid calculations for CO₂ emissions. All the post processing routines will be implemented according the project KPIs equations as described in Deliverable 1.3.

This report first describes the different platforms (DSP, SC & PSO), their “standalone” capabilities and potential applications. Then focus is made on their main interactions, more specifically through Nicore, the latter turning out to be a real conductor between platforms. All along the document a description of data that will be shared is also presented.

2 Overview of involved platforms

The objective of this section is not to provide a fully detailed description of the involved platforms but to mark out their main frontiers from an interoperability point of view.

2.1 District Simulation Platform (DSP)

District Simulation Platform allows to perform dynamic simulation of buildings (from one up to thousands of buildings) and heating networks at the district level, taking into account their main physical interactions. At building level, each thermal zone is modelled as a R-C model (R20C10) and can integrate several heating, cooling, ventilation, and/or domestic hot water systems. At district level, it is possible to pre-process solar masks between buildings and landscape, building adjacencies, to simulate central heat production (with eventually additional storage, backup) and also pressure/temperature for piping networks connecting the buildings.

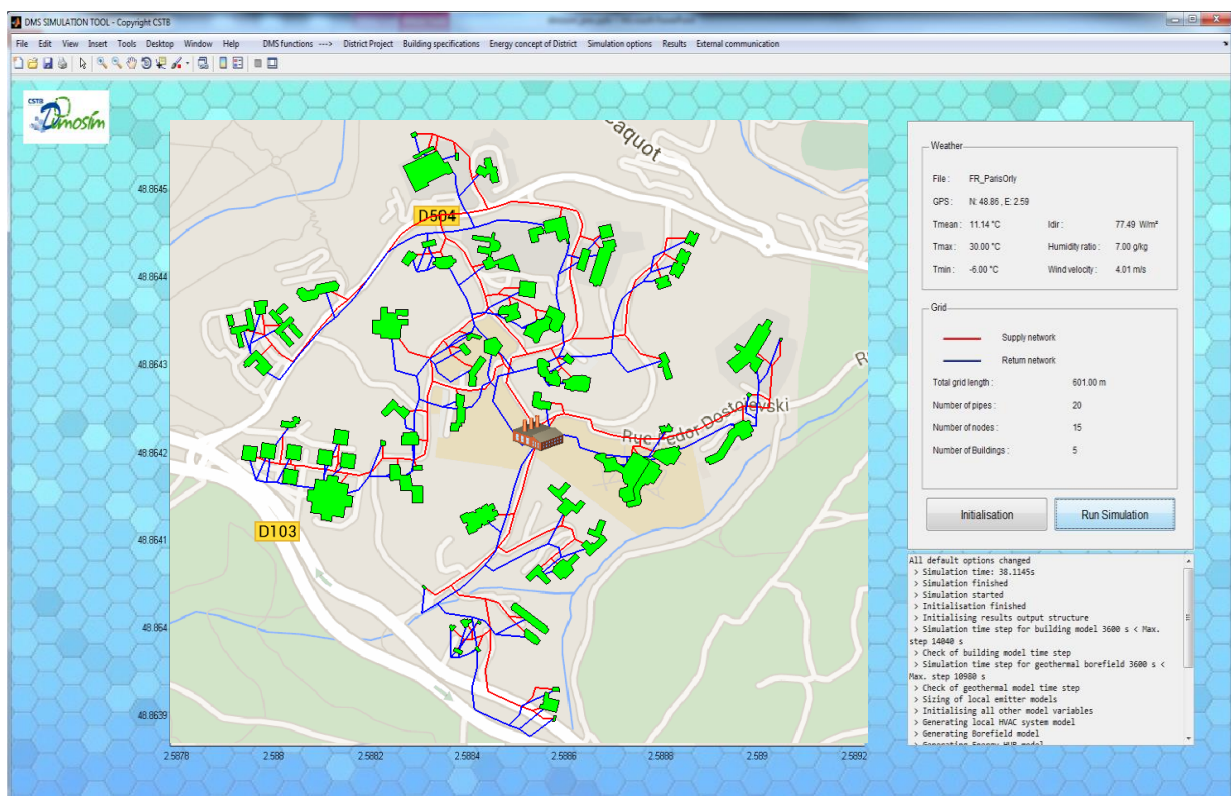


Figure 2: Overview of District Simulation Platform (local GUI)

Simulations can be carried out at different time steps depending on the final objective of the associated study, typically from 1 hour for annual energy performance assessment, down to 1 second for taking into account finer controls with very low time constants. DSP also integrates functionalities such as sizing (systems, emitters, storages) or network balancing, and feeding modules (to generate occupancy profiles and associated gains and electrical loads), that will be presented in more details in deliverable 2.2.

Focus is made hereunder on the three interoperability dimensions introduced previously, which are model instantiation, co-simulation and post-processed results.

2.1.1 Model Instantiation

A configuration that can be considered as ready to simulate can be defined as a parsimonious set of input data. Some of these data are intrinsic parameters (building geometry coordinates, thermal properties, type of heating system ...) that are not time dependent. The other part consists in profiles or time series (such as weather, occupancy ...).

District simulation Platform can handle several input data formats, from the fully harmonised digital mockup (citygml or geojson - BIM equivalent at district level) to a combination of user-defined tables (.xlsx, .mat) for profiles and standardized readers for weather (.tmy, .epw). In “real life”, collection of data often comes out on a wide set of different files, the ideal situation being logically when all data are gathered in one semantic-based digital mockup file, as illustrated on Figure 3:

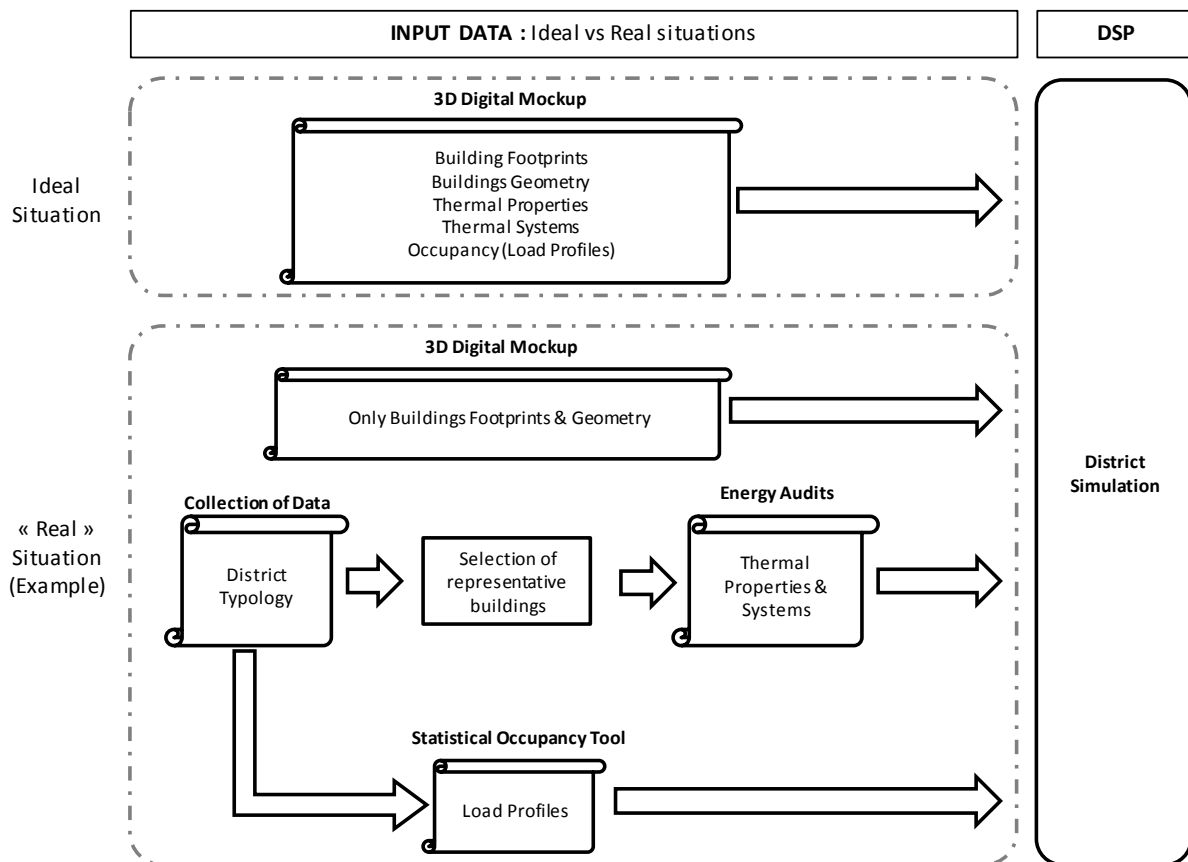


Figure 3: Various sets of input data for District Simulation

Regarding interoperability, the chosen approach shall be based on a solution as close as possible to this “ideal” situation. Even if the work carried out for data collection is not clearly BIM-based, only one particular set of input data will be used throughout the project. Consequently, for each configuration of a district there will be one associated input file, named “district information file” hereafter in the document.

2.1.2 Co-simulation

DSP already integrates basic control functions (Hysteresis, PI, Heating laws) for the management of implemented energy devices, typically dedicated to turning on/off systems, modulating powers or flow rates or building setpoint scheduling, as illustrated on Figure 4:

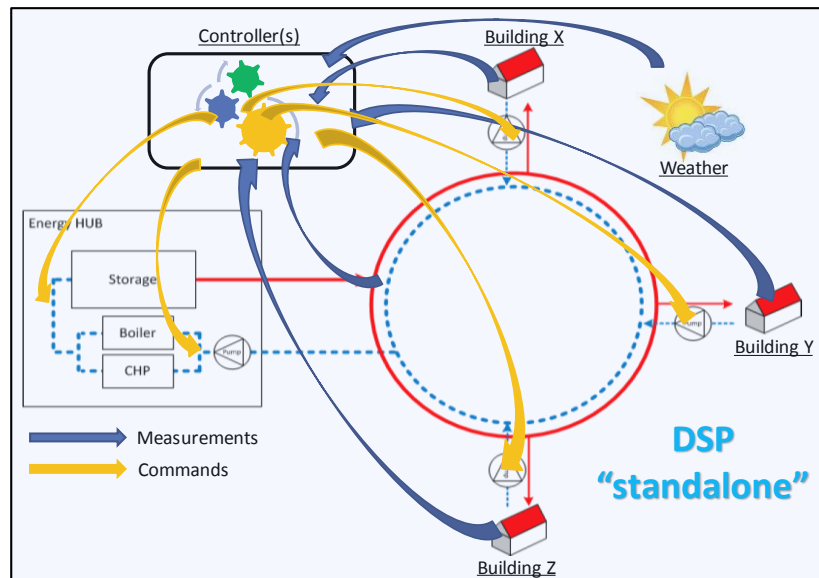


Figure 4: District Simulation Model Example (Integrated Control)

On Figure 4 we can observe that the controller waits for measurements (eg current building temperature, piping network flowrate, ...) as inputs. More accurately, the term “measurements” has to be understood here as the states of the different models at a given time step, similarly as in a real installation.

When it comes to test control algorithms that are implemented in another environment, District Simulation Platform has to be configured as “open within time loop”, meaning that the simulation sequences cannot be run in a row anymore and a specific communication port has to be accessible at each time step for data exchange: District Simulation Platform behaves then as a district emulator, sending measurements (states) to the control algorithms and waiting for the calculated commands before incrementing to the following time step.

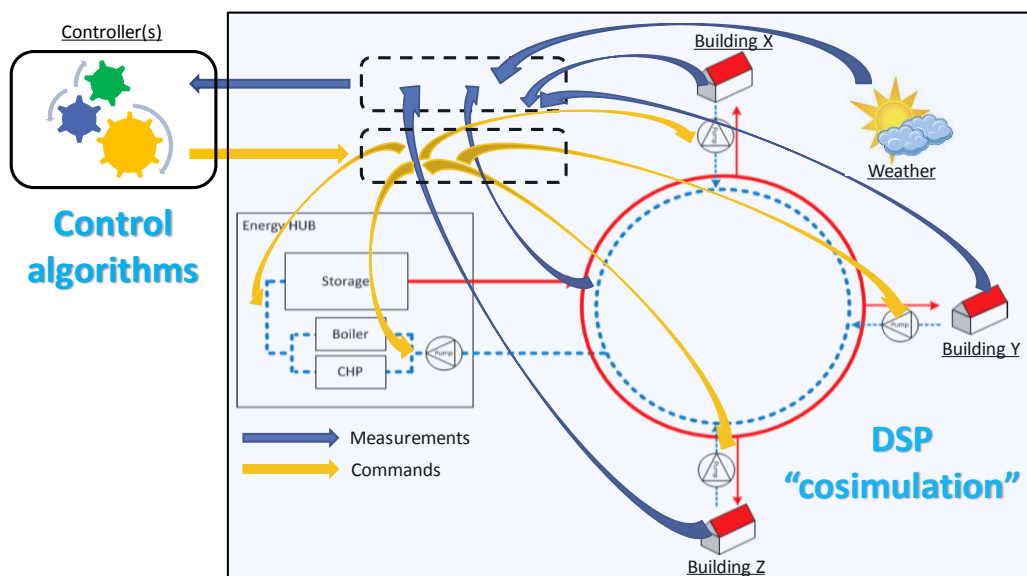


Figure 5: District Simulation Model Example (External Control)

2.1.3 Calculated Variables - Results

District Simulation Platform generates raw results as time series representing key states and fluxes for each building, piping network and/or centralized production. The main global outputs are :

- For each building
 - Zone air/operative temperatures [°C]
 - Heating demand [W]
 - Cooling demand [W]
 - Domestic hot water draw-off/demand [W]
 - Electrical consumption [W]
 - Gas consumption [W]
 - Fuel consumption [W]
 - Biomass consumption [W]
 - Electrical production [W]
 - Renewable thermal production [W]
- For DHC system
 - Piping segments temperatures [°C]
 - Piping segments pressures [Pa]
 - Hub thermal load [W]
 - Electrical consumption [W]
 - Gas consumption [W]
 - Fuel consumption [W]
 - Biomass consumption [W]
 - Electrical production [W]
 - Renewable thermal production [W]

The listing above only summarizes the global time-series, but District Simulation Platform also delivers all possible derived-values for the consumptions, meaning it is possible to get the part of electrical consumption of building n°X only specifically to domestic hot water production for instance.

The list of variables required for KPI computation has to be consistent with DSP platform (electrical and thermal production/consumption must be available for all systems). In the same way, variables required for KPIs computation must be available from the real DHC site in DOS platform.

From these time-series the elementary final energy or efficiency involved at various levels of the simulated district can be directly deducted. In a second time, when dealing with indicators such as primary energy or energy costs, these “raw results” can be integrated in the relevant equations for KPIs (as described in Deliverable 1.3) that will require also a data base for all related external factors such as primary energy conversion rate or price per energy unit., or grid calculation for CO₂ emission. The variables needed for KPIs calculations are summarized in Appendix B.

2.2 Supervisory Controllers (SC)

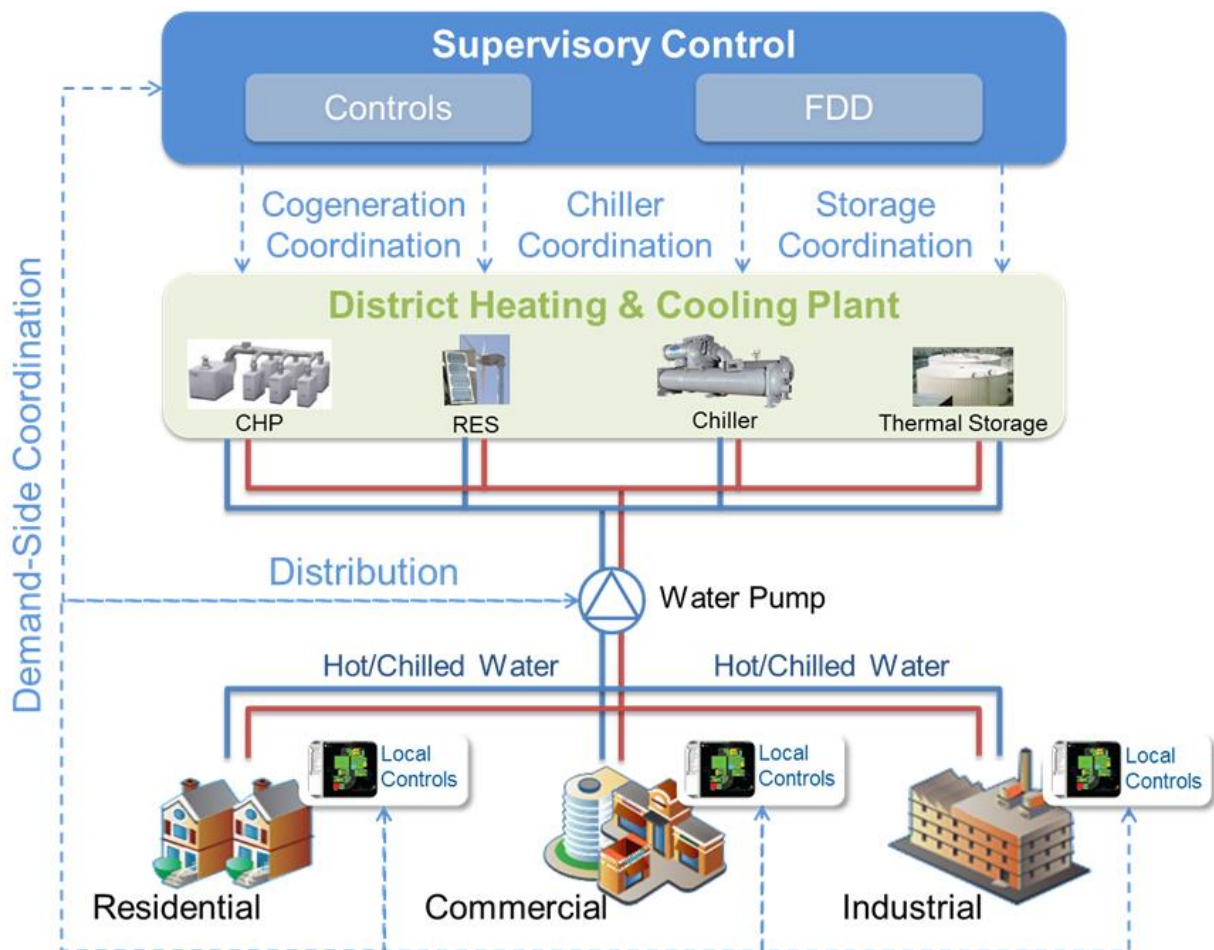


Figure 6: District Simulation Model Example (External Control)

The control algorithm developed is an intelligent, self-learning supervisory control algorithm for optimal operation and coordination of district heating/cooling plant equipment (such as CHPs, heat-pumps, chillers, boilers), demand (heat exchanger load) and energy storage.

The supervisory controller features self-learning capabilities to “learn” the behavior and operation patterns of DHC subsystems in order to ensure that most efficient and cost-effective operation of the overall DHC system is maintained.

2.2.1 Tasks

In contrast with traditional control system (e.g. fixed equipment operating schedules, weather compensated set-point strategies and local actuator controllers) which focus on individual subsystems of the DHC plant and network, intelligent supervisory control account for the interaction between different DHC plant generation equipment, the DHC plant, the heating network (distribution) and end-users.

The coordination of the wide variety of equipment and load is carried out by varying the schedules and set-points in real-time and in an optimal manner, ensuring the overall energy efficiency and cost-effective operation of the DHCN to varying climate conditions, load demand, end-user preferences/requirements, or operational anomalies.

The main tasks accomplished by the algorithm can be summarized as follows:

- Intelligent coordination of on-site generation and demand
- Plant-wide heating and cooling optimization for improved energy efficiency
- Adaptive behaviour to demand, weather and operational variations
- Increased efficiency and constraints satisfaction

2.2.2 Inputs/Outputs

The intelligent supervisory control algorithm proposed is an on-line closed-loop control algorithm. The main ingredients of the closed-loop control strategy depicted in Figure 7 are the following:

- The algorithm measures system states/outputs in “real-time” to detect deviations from desired performance (monitoring)
- The controller reacts by changing set-points or key-input variables to improve system performance (computing)
- Control signals are applied to the systems actuators either directly or by means of local controllers (actuating)
- The monitoring-computing-actuating process is repeated continuously at regular time-intervals in an automated way

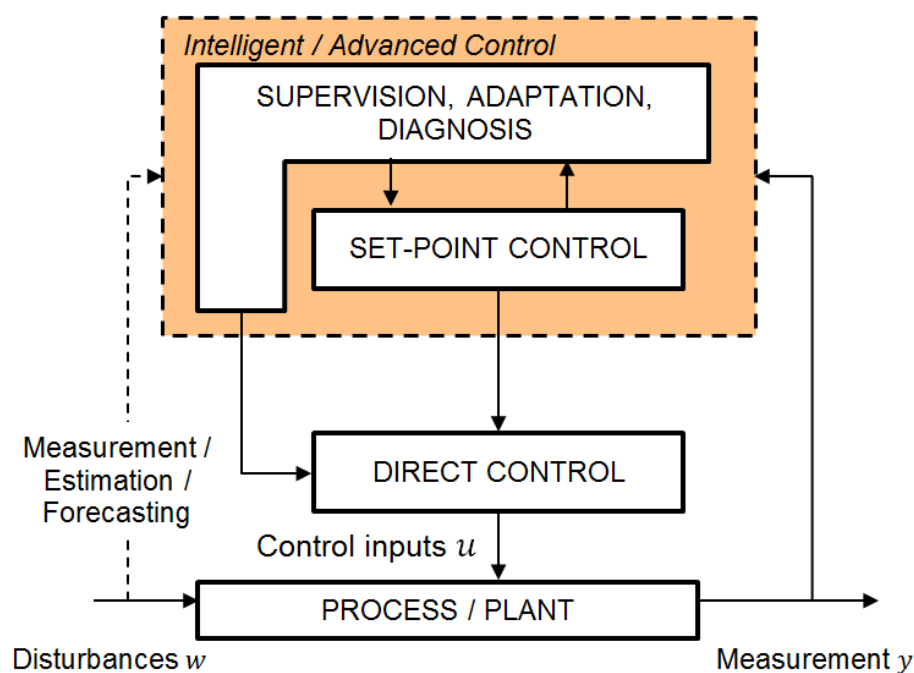


Figure 7: Multi-layer Control Architecture

The supervisory controller, knowing the control objectives and the tasks as summarized above, makes decisions on the control inputs, based on the values of the measurable system outputs and measured or estimated disturbances acting on it. In particular, in the proposed architecture, the Intelligent Control Algorithm decides set-point values for direct controllers, which have direct access to the real plant.

The **main inputs** to the algorithm are all the real-time (collected in the BMS) and historical data from DHCN, and optimal scheduled energy profiles coming from the Production Plant Scheduler, like:

- real-time data, collected from the system sensors and low level controller such as:
 - Outside, building and supply/return water temperatures
 - Equipment operational set-points and status
 - Flows and valves set-points
 - Pressures and pumps set-points
 - Thermal storage charge/discharge set-points
 - Heat and power meters
- historical data from DHCN
- energy price
- weather, load and heat/cooling demand measurement and forecast
- DHCN operational anomalies
- forecast of energy production and distribution
- state of behavioral flexibility
- state of virtual storage proposed by all building inertia

The **main outputs** of the supervisory controller are all those decision variables that may act on the DHC system and change its behaviour, namely all the optimized set-point signals to low level DHCN controllers and actuators through the BMS (see Figure 7):

- Main Distribution Heating Circuit and valves
- Boilers and associated pumps (water temperature set-point, flow-rate pump set-point)
- CHP and associated pumps (load set-point, start-up time, flow-rate pump set-point)
- Thermal storage modulating valve, charge/discharge temperature set-point, flow-rate pump set-point

As an example, a summary of the measurements and set-points available for the CIT demo-site are reported in.

Table 3 (in Appendix) presents all the data expected to be controlled/measured when deploying the Supervisory Controller on CIT DHN

2.2.3 Time scale – Sampling Time

The feedback controller reads, on a regular sampling time (order of minutes), the real-time measurements coming from the DHCN, and possibly also historical data on it (still to define whether the historical data has to be stored directly in its memory or can be imported directly as a time series from external source). Then the controller makes decisions on the control signals to be applied to the system in order to guarantee its optimal performance and energy efficiency operations.

2.3 Production Scheduling Optimizer (PSO)

Production Scheduling Optimizer allows to determine an optimized operation schedule to be applied to the district operation.

On the day before (D-1), the production scheduling model optimizes day (D) operations of a district heating and cooling network. It takes decisions on the day before (D-1) and recommends actions to (D). Here are some definitions:

“(D) operations” means the thermal and electrical production schedule and the optimization of the supply temperatures needed to satisfy the heat demand.

“(D-1) decisions” means the decisions that must be taken on day (D-1) for the day (D), e.g. the electricity selling or features of the heat business strategy, such as the heat consumption shifting on the time horizon.

In this way, the Production Scheduling Optimizer recommends on (D-1) the most efficient and cost-effective decisions to take on (D-1) and on (D). So that, the implemented recommendations of (D-1) constraint the DHCN operator or the Supervisory Controller decisions on (D).

2.3.1 Tasks

As the Supervisory Control (SC), the Production Scheduling Optimizer (PSO) considers the features of different thermal and electrical production equipment, the district heating and cooling distribution network and the end-users. But it mainly considers the electrical market and business strategy features.

The electricity selling and business strategy features optimization ensure the overall plant day ahead efficiency and cost-effective operations. The day ahead optimization of the DHCN is subjected to climate conditions, heat demand and electricity prices forecast, as well as the operational constraints, electrical market and business features constraints.

The main tasks accomplished by the production scheduling model can be summarized as follows:

- Intelligent prediction of electricity selling
- Optimization of heating and cooling production and distribution to increase operational efficiency and constraints satisfaction
- Optimization of the electricity production to improve the electricity selling prediction quality
- Optimization of the heat selling subjected to the business strategy features

2.3.2 Inputs/Outputs

Before presenting the main inputs and outputs to/from the PSO's algorithm, we define the following terms:

Production zone: the area that comprises production and storage units, such as Cogeneration heat and power (CHP), boilers and accumulators.

Distribution network: the distribution network infrastructure, such as the pipes and equipment that connect the production zone to the substations.

Demand forecast: end-users demand forecast (it does not consider the heat losses between the production zone and the final use sites).

Load forecast: demand forecast transposed to the interface between the production and the distribution network, we can consider it as the sum of the network heat losses and the end-users demand forecast

Production load: the power production scheduled on the production zone's units.

The **main inputs** to the PSO's algorithm are forecasted data, real-time indicators and historical data from DHCN:

- Forecasted data, collected from the DSP tool (possibly provided by external source):
 - load forecast
 - DHN's thermal power auto consumption
 - DHN's electrical power auto consumption
 - Heat losses rate between the production zone and the substations
- Forecasted data, collected from external sources :
 - electricity and/or fuel prices
- Real time indicators, such as plant measurements: equipment availability (DHCN operational anomalies), maintenance schedules, state of heat storage, etc.
- Operational indicators: unit production efficiency, production zone and distribution network characterization; etc.
- Historical data from DHCN
- State of behavioral flexibility (changes on heat consumption behaviour characterization)
- State of building flexibility (building inertia)

The **main outputs** of the production scheduling optimizer are all the decision variables that may act on the DHCN and may change its behaviour. These variables indicate:

- boilers' switch-on/off signal, production load
- CHPs' switch-on/off signal, production load, heat recovery set-point
- on/off discharge and thermal charge/discharge flow
- supply temperature
- flexibility schedule (behavioural, building inertia, heat storage)

The information about CHP's electrical production on day (D) helps the decision about the amount of electricity to sell on day (D-1). In the same way, the thermal production information on (D) helps the decisions about the customers' relationship decisions to take on day (D-1) to attain the business strategy specifications.

The decisions taken on day (D-1) by the PSO must be respected on day D by the operations of the DHCN. For example, the decisions about either over or under produce can be taken by the DHCN operator or SC, based on real time measurements, and on the (D-1) PSO's decisions, forecasts, penalties, etc.

2.3.3 Time scale (time steps, time horizon)

At each day (D-1), the PSO reads the DHCN measurements, historical data and forecasts. At the end of the day (D-1), the PSO calculates the optimal operational schedule of the DHCN for the day D. This optimal operational schedule will be based on fixed time steps, in a range from 10min up to 1 hour depending on the simulated case.

2.4 Cloud-based integrated platform DOS (CIT - NiCore)

In order to facilitate the communication with the live district as well as between the individual modules a cloud based district operation system (DOS) is to be used.

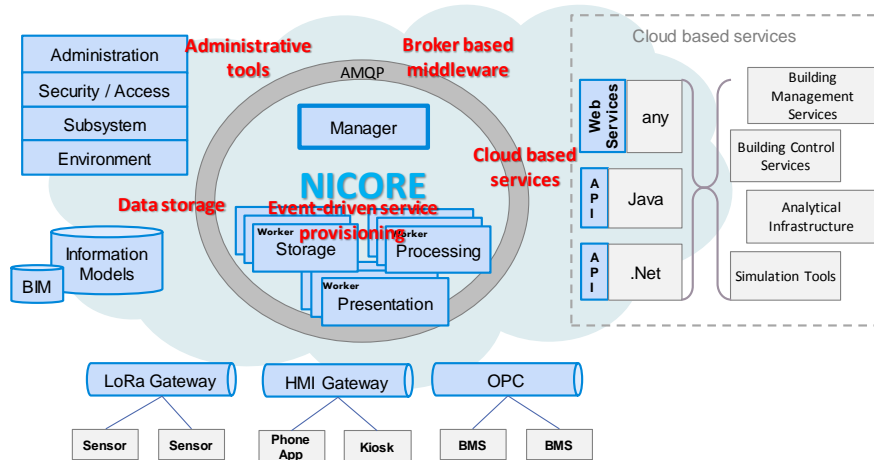


Figure 8: E2District District Operation System Architecture

At the core of the DOS is the AMQP broker-based middleware NiCore implementing a distributed and scalable infrastructure for all components to be connected to. The first key role of the NiCore middleware is the abstraction of the underlying district subsystems. For instance, in the pilot district on the CIT campus these subsystems are :

- The currently operating building management systems (BMS), which are connected using the automation system industry standard OPC
- Additional IoT sensors where required, which are connected using the LoRa communication standard
- The HMI gateways for building occupant engagement, which are implemented as part of the NiCore system

All data collected by subsystems connected to the NiCore middleware is made available to potential subscribers as live data stream in order to facilitate real time processing and control. Further to that, all data is recorded and archived in a MongoDB database to facilitate bulk processing based on historical data. Scalability is achieved through an event-driven service provisioning mechanism that will scale out the processing based on the amount of data to be processed. The platform will also monitor the health of all connected components, so that system faults or performance bottlenecks are detected and mitigated if necessary.

Further to the live data streams directly received from the message broker, the DOS implements a couple of APIs to facilitate accessing the data from the district by E2District modules and other third party applications. These APIs include

- Web Services (SOAP and REST) for general purpose communication with the platform, including a user-friendly frontend GUI based on these
- A .NET API for integrating the district into existing .NET applications
- A Java API for integrating the district into existing Java application. This Java API also integrates with MATLAB through appropriate wrapper functions and allows the integration of the live district into existing MATLAB applications.

Finally the DOS is configured and managed through a set of administrative tools, some of which are desktop applications, some of which are web based content management systems (CMS) catering for the different stakeholders in the system and the resulting different levels of access required.

3 Specifications on Interoperability

3.1 Global Architecture

As described in the previous section, three main platforms (SC, PSO and DSP) are to be coupled through the DOS (Nicore). These platforms being hosted at different locations, the data exchange is based on web service interfaces as illustrated on Figure 9 :

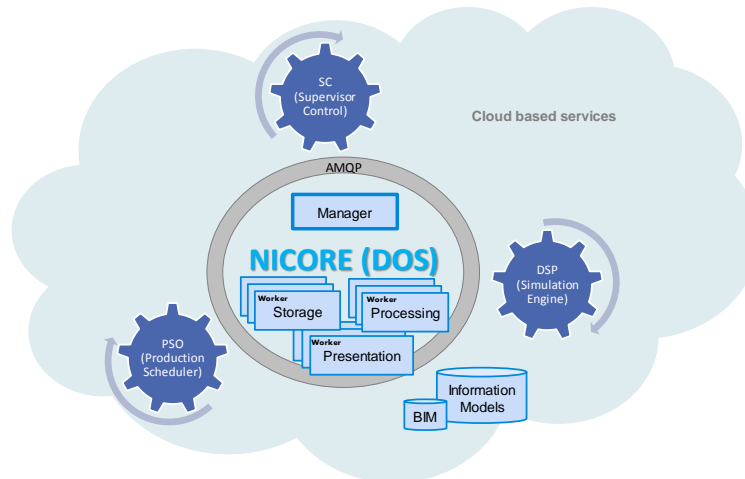


Figure 9: Overview of web based platforms architecture

Though all the platforms shall interoperate together it is important here to clearly define the main use cases* that are required for this project since the interactions between DSP and SC or between DSP and PSO will serve distinct purposes. They will be then totally different from interoperability point of view. There are two identified use cases as follows :

- Control strategies testing and validation on a simulation-based testbed before field deployment
- Optimisation of production scheduling based on hub thermal load forecast

The first case (which involves SC and DSP) is based on co-simulation while the second one (which involves PSO and DSP) relies on predictive simulation over a time horizon for further optimization of production schedule.

For both cases, a preliminary phase consists in calibrating a district model. So even if the interactions are intrinsically different, a common requirement from SC and PSO platform is the ability to call through DSP a district model as representative as possible of the real district. All the data gathered along the project (mock-up, energy audits, historical data, ...) will be stored on Nicore database and for each district a “sim-ready” district information model will be generated to feed simulation platform accordingly.

Choice has been initially made to perform all data exchange through Nicore. Though a question of optimal computational time (especially for co-simulation between DSP and SC) can logically be risen, it was decided to take advantage of all pre-existing data management/access structure offered by Nicore. Moreover a major constraint of intellectual property for each partner’s platform makes complicated to share them all on a common machine (though that would be for sure the best solution in terms of efficiency).

* not to be confused with “use cases” as defined in WPI, here it focuses only on software interactions

3.2 Description of Use Cases

The two use cases introduced above are described here. They are both relying on the use of a physical model (DSP). For each demonstration site, a district simulation model will be preliminary elaborated from all the technical specifications, historical data and eventually energy audits that will be at disposal.

This process will be carried out manually, calibration and available data being very dependent of each demonstration site, it cannot practically be generically automated. However the resulting district information models ("simready") will be stored in DOS and made accessible to SC and PSO when needed.

3.2.1 SC using DSP as a testbed

This first use case involving SC and DSP consists in control strategies testing on the simulation platform, before operating directly on a real district as described on Figure 10 :

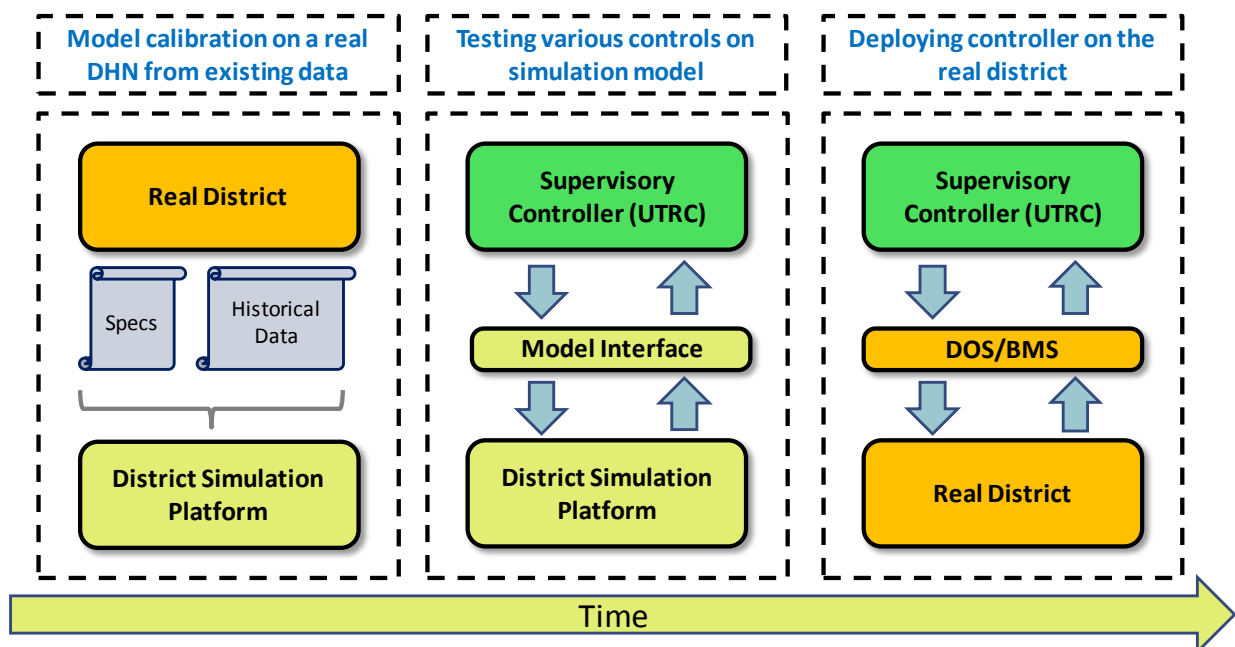


Figure 10: DSP/SC use case schematic & timeline

Once calibrated, DSP is used here by SC platform in order to perform tests for various control strategies as if it was operating on the real district. DSP will be configured in co-simulation mode so that SC can directly interacts with simulation engine within each time-steps by sending commands to DSP, that will take them into account and returns the associated measurements for the next time step. Perform weekly, seasonal or even annual simulations it will allow to assess controller behaviour before deploying it on the real site.

In parallel, a specific attention will be paid on these co-simulation variables exchanged through the "model interface (cf Figure 10))so that it can be as consistent as possible to "DOS/BMS" interface, aiming at deploying SC as plug & play as possible.

3.2.2 PSO using DSP as a thermal hub load forecast engine

This second use case involves DSP and PSO. In this case, DSP is called by PSO as a thermal hub load forecast tool. The objective is to allow PSO to provide a set of scheduled setpoints to the platform over a time horizon of N days (from 2 up to 5 days) so that the simulation engine can assess the thermal hub profile and return it to PSO.

DSP will be then called as many times as the optimisation loop of PSO requires to satisfy as best as possible an operational objective (likely maximizing electricity sellings back to the grid and/or minimizing operation costs).

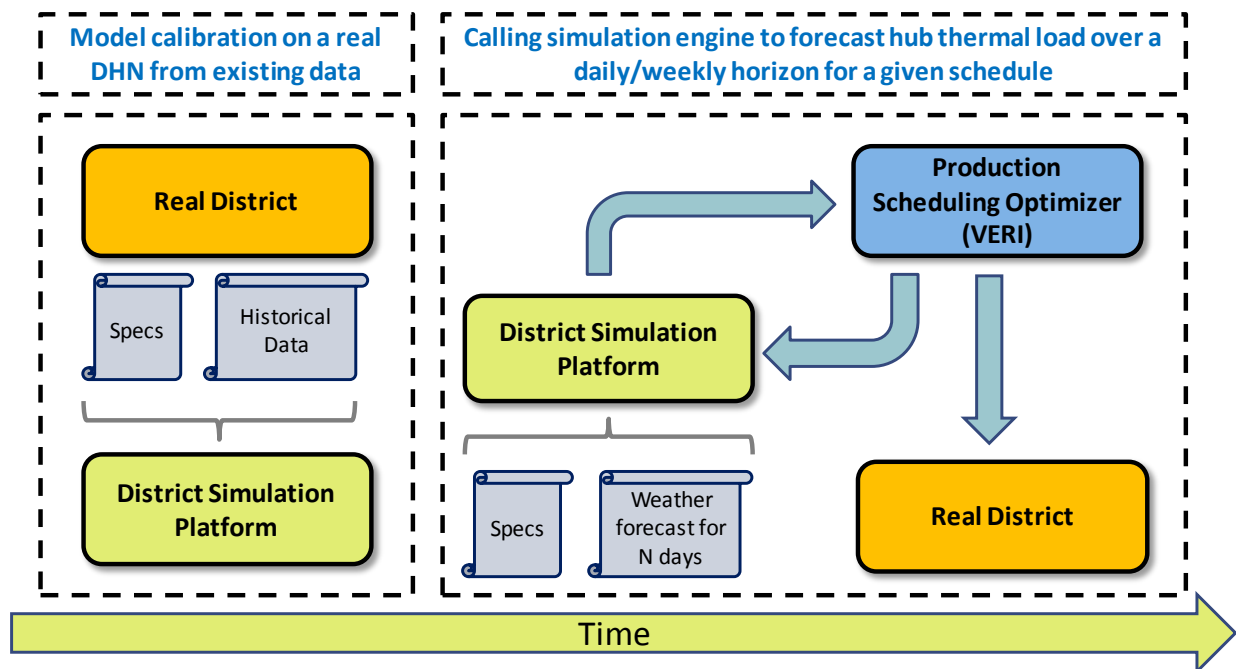


Figure 11: DSP/PSO use case schematic & timeline

DSP won't provide weather forecast but will take it as input and then will generate load forecast on building side and also on central production side (ie it calculates the thermal demand that will be imposed on the generator side taking into account smoothing effects of pipe inertia, storage for one given weather forecast data).

The main difference with the use case involving SC in terms of interoperability is that DSP won't be called in co-simulation mode, meaning at each time step, but will be executed on a certain period (couple days as described above) and shall be called each day for a global optimization.

3.2.3 Interactions through DOS

As presented in the previous section, the two controller platforms SC and PSO will interact with DSP according to their specific use. This global process will be conducted around DOS as illustrated on Figure 12:

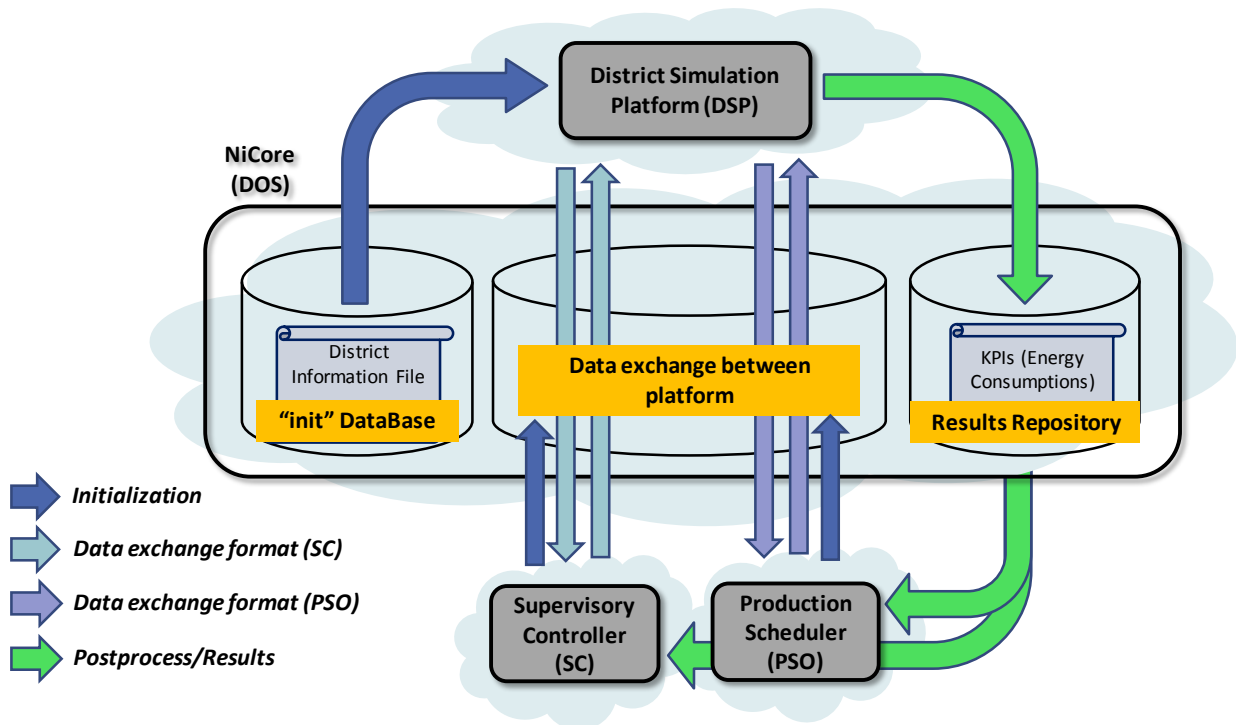


Figure 12: Data flow between DSP, SC and PSO platforms through DOS

Three main consecutive stages of interactions are to be considered:

➤ **Initialization**

Either SC or PSO platform* requests the use of DSP for a specific site. The corresponding district information file (refer to 3.3.1) from DOS Database is then sent (with associated files such as weather) to DSP and open a specific process to initiate a simulation.

➤ **Data exchange (Co-simulation)**

Using dedicated formats, this stage refers to the period to perform calculations, exchanging data at each time step over one simulation for DSP/SC use case and exchanging simulation input/outputs for multiple simulations for DSP/PSO use case.

➤ **Post-Process**

Once the calculations are performed, besides the values (and relative results) exchanged between DSP, SC or PSO, a global set of results (KPIs) is to be stored on DOS database.

** It is presented here as either one platform or another as exclusive, but it has to be noted that multiple separated processes of DSP (and then simultaneous simulations) could be also considered*

3.2.4 Cross platform summary

For the two use cases, DSP will act as a **server** that can “serve” (indeed !) the two potential **clients** that SC and PSO are. This relationship will condition the implementation of the platform deployment since DSP will be listening for requests that will be initiated at any time by SC and/or PSO.

The global procedure managed by the DOS will follow a sequence as :

- Client (PSO/SC) sends a “PUT” request to DSP via Nicore with all required input data
- Server (DSP) launches the requested simulation after creating an associated job ID
- Client (PSO/SC) waits till DSP returns job ID to indicates that calculations are done through Nicore and then access results (“GET”)

The same procedure will be applied for both use cases. All the differences in terms of inputs/outputs are summarized in Table 1:

Table 1: Summary for DSP/SC and DSP/PSO interactions

DSP interactions	UTRC Controller (SC)	VERI Scheduler (PSO)
Objective	DSP as a “real-time” testbed	DSP as a hub load forecaster
Using Calibrated Models	yes	yes
Time Consideration	DSP is called at each time step (~1-10min) by the controller for a given testing period (season, year)	DSP is called as many time as required by optimization loop with over a time horizon of 2-5 days
Required External Data	District specs (mockup) Annual weather file	District specs (mockup) Weather forecast file
Inputs to DSP	Set of commands calculated by the controller	Setpoint schedules over the given time horizon
Outputs from DSP	States (measurements) required from DSP for calculating next commands	Hub Thermal loads System Consumptions
Interactions with Nicore	To get considered district specs at initialization To use a “BMS like” server to run co-simulation	To get considered district specs at initialization To use a repository to download schedules and upload results for one simulation call

3.3.3 Data Exchange Format between Platforms

Data exchange between DSP and SC or PSO will be carried out through two distinct files (json format), one from DSP towards SC/PSO and one from SC/PSO towards DSP.

The two files to be considered for the use case **SC/DSP** are:

- sc2dsp.json (from SC towards DSP)
- dsp2sc.json (from DSP towards SC)

And for the use case **PSO/DSP**:

- pso2dsp.json (from PSO towards DSP)
- dsp2pso.json (from DSP towards PSO)

These files will first present a header “Project Info” containing the name of the considered district information file (“CIT_districtmodel.citygml” in the following figures) to indicate the relationship with the district to be simulated.

Even if presented as “file”, it can be observed that the file itself shouldn’t be seen as a file to be stored somehow on a hard drive, but as the structure that will be serialized and communicated between web-based platforms.

3.3.3.1 Use Case SC/DSP

The structure of these files is illustrated on Figure 15 and Figure 16:

```

1  {
2    "sc2dsp": {
3      "ProjectInfo": {
4        "DemonstrationSite": "Cork Institute of Technology (CIT)",
5        "DistrictInformationFile": "CIT_districtmodel.citygml"
6      },
7      "Variable": [
8        {
9          "description": "CHP Enable",
10         "value": "True",
11         "unit": "bool",
12         "bms_tag": "CHPE-BH-01"
13       },
14       {
15         "description": "Thermal Store Charge Setpoint",
16         "value": 65.3,
17         "unit": "°C",
18         "bms_tag": "CSP-TS-BH-01"
19       },
20       {
21         "description": "Boiler No 1 Enable",
22         "value": "False",
23         "unit": "bool",
24         "bms_tag": "BRE-BH-01"
25       },
26       {
27         "description": "Boiler No 2 Enable",
28         "value": "False",
29         "unit": "bool",
30         "bms_tag": "BRE-BH-02"
31       }
32     ]
33   }
34 }
```

Figure 15: Data exchange file – sc2dsp.json

In this example (Figure 15) are shown the variables that will be controlled by SC algorithms, such as “CHP Enable” to turn on/off the CHP (Combined Heat Power system) or set the thermal storage setpoint on the production side of the DHN. Besides the values themselves that are to be sent to DSB, additional descriptive labels (eg “description”, “unit”, “bms_tag”) are also given for each variable.

The relative BMS tag is given for preparing as much as possible the global interoperability. In other words, since the aim of co-simulation between DSP and SC is to test and optimize the controller before deploying it on a real district, a special attention is paid on associating each controlled variable to a real corresponding BMS tag (all BMS Tags taken into consideration for the CIT Demonstration site are listed in Appendix, Table 3). That should help to ensure SC and BMS match when they will be coupled.

Figure 16 shows the data exchange file that will be returned by DSP towards SC :

```

1 {
2   "dsp2sc": {
3     "ProjectInfo": {
4       "DemonstrationSite": "Cork Institute of Technology (CIT)",
5       "DistrictInformationFile": "CIT_districtmodel.citygml"
6     },
7     "Variable": [
8       {
9         "description": "Thermal Store Temperature",
10        "value": 56.2,
11        "unit": "°C",
12        "bms_tag": "TT-ST-01"
13      },
14      {
15        "description": "CHP Flow Temperature",
16        "value": 65.3,
17        "unit": "°C",
18        "bms_tag": "TT-CHP-F"
19      },
20      {
21        "description": "CHP Return Temperature",
22        "value": 48.2,
23        "unit": "°C",
24        "bms_tag": "TT-CHP-R"
25      },
26      {
27        "description": "Outside Air Temperature",
28        "value": 10.3,
29        "unit": "°C",
30        "bms_tag": "TT-OAT"
31      }
32    ]
33  }
34 }

```

Figure 16: Data exchange file – dsp2sc.json

The structure is very similar to the “sc2dsp.json” file, and contains all emulated variables such as CHP temperature, outside air temperature, ...) required by SC to perform its calculations at each time step.

NB : for clarity purpose, Figure 15 and Figure 16 only illustrate the concept through a subset of control/emulated measurements variables required for CIT demonstration site. The exhaustive listing can be found in Table 3

3.3.3.2 Use Case PSO/DSP

The structure of data exchange between PSO and DSP is very similar to the one between SC and DSP. The major difference deals with time series consideration: in this use case, platforms are not “discussing” at each time step of the simulation but along a certain period. Then the values to be exchanged are not scalars anymore but time based-vectors. Figure 17 shows the structure for scheduled data to be sent from PSO towards DSP:

```

1  {
2    "pso2dsp": {
3      "ProjectInfo": {
4        "DemonstrationSite": "Cork Institute of Technology (CIT)",
5        "DistrictInformationFile": "CIT_districtmodel.citygml"
6      },
7      "Time": {
8        "description": "Simulation Time",
9        "value": [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,],
10       "unit": "hours"
11     },
12     "ProductionSchedule": {
13       "CHP_setpoint": {
14         "description": "DHN supply setpoint",
15         "value": [64,63,61,61,65,60,62,61,65,64,63,62,60,64,60,],
16         "unit": "°C"
17       }
18     },
19     "WeatherForecast": {
20       "Outside_temperature": {
21         "description": "Predicted outside temperature",
22         "value": [6.2,7.5,8.7,9.9,11,11.9,12.8,13.5,14.1,14.5,],
23         "unit": "°C"
24       }
25     }
26   }
27 }

```

Figure 17: Data exchange file – pso2dsp.json

As explained previously, PSO will send a time-based vector for scheduling the production temperature set point for example. Then time (which is expressed in hours in the illustrative figure) needs to be also specified. Additional information required to perform loads calculations such as weather forecast will be also sent through this file.

Figure 18 shows the structure of data that will be sent back from DSP :

```

1  {
2    "dsp2pso": {
3      "ProjectInfo": {
4        "DemonstrationSite": "Cork Institute of Technology (CIT)",
5        "DistrictInformationFile": "CIT_districtmodel.citygml"
6      },
7      "Time": {
8        "description": "Simulation Time",
9        "value": [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,],
10       "unit": "hours"
11     },
12     "Production": {
13       "SupplyTemperature": {
14         "description": "DHN Supply Temperature",
15         "value": [64.4,63.3,64.2,63.2,64.3,63.9,66,64.1,64,63.],
16         "unit": "°C"
17       }
18     }
19   }
20 }

```

Figure 18: Data exchange file – dsp2pso.json

For time consistency, even if DSP platforms performs calculations at a different time-step (e.g. 10 minutes), results will be resampled if needed according to the shared time basis (e.g. hours).

3.3.3.3 Request for simulation and initialisation

To run a district simulation, the first function to be executed is a request, with all the parameters that will define this simulation. This request shall contain all necessary information (or link to additional information) to perform calculations

Global parameters will be assigned to each simulator variables (cf Table 2), and once all the values are imported, simulation can be carried out.

These global simulation parameters are illustrated in Table 2.

Table 2: Global simulation parameters example

Parameter	Type	Unit	Example value / format
District Project	string	-	"Cork Institute of Technology (CIT)"
District Information File	string	-	"CIT_districtmodel.citygml"
Project Configuration	string	-	"Baseline" or "Scenario n°i"
Simulation start time	string	-	2016-01-01-00-00-00 (YYYY-MM-DD-hh-mm-ss)
Simulation stop time	string	-	2016-12-31-23-59-59 (YYYY-MM-DD-hh-mm-ss)
Simulation time step	float	s	3600
Weather data filename	string	-	"CorkWeather.epw"
SC to DSP filename*	string	-	"sc2dsp.json"
DSP to SC filename*	string	-	"dsp2sc.json"
DOS "get" commands	string	-	"get(IPaddress,...,CorkWeather.epw)"
DOS "put" commands	string	-	"put(IPaddress,...,Results.txt)"

* example for use case between SC and DSP

This first "header" part illustrated in Table 2 contains basic simulation parameters such as time step or simulation time, and also includes filenames that will be used by DSP and SC for exchanging data at each time step.

In the case of the use case between PSO and DSP, simulation start time will be set to the beginning of the next day "D", according to PSO definitions (see 2.3), and simulation stop time to, at least, the end of day "D".

Finally are presented DOS get/put commands in a very general description that is still to be defined later in the project (WP4) when implementing the process in DOS.

4 Conclusion

In this report are described the specifications for interoperability of the different platforms developed and adapted throughout E2District project. They can be summarized as follows :

- **DSP** (District Simulation Platform)
 - A physical simulation engine for district energy performance dynamic calculations
- **SC** (Supervisory Controller)
 - An intelligent, self-learning supervisory control algorithm for optimal operation and coordination of district heating/cooling plant equipment
- **PSO** (Production Scheduling Optimizer)
 - An optimization algorithm for decision support of DHC dispatcher

These three platforms will interact through cloud-based platform **DOS** (Nicore) in the form of two identified use cases:

- ➔ Test of SC on a simulation-based emulator (DSP) before field deployment
- ➔ Scheduling optimisation (PSO) based on hub thermal load forecast (DSP)

First presenting the intrinsic capabilities and potential applications of these platforms, a particular attention has been paid on their main interactions and also on the data format that will be shared by all the partners. For each demonstration site a District Information File will be generated in a consistent structure that will be stored on DOS database (Nicore) and made available to all calculation plugins.

Finally, specific data format have been elaborated (co-simulation data for two different use cases interactions), the latter being potentially adjusted depending on technical barriers (as computational time issues) and/or optimisation possibilities that could be met when under development.

5 Appendix A (CIT example)

As we already have a lot of information on CIT DHN, the idea of this section is to illustrate the global interoperability specs presented in this document by writing down the set of measurements and commands that will be involved in the co-simulation or production scheduling processes, accordingly to the case of CIT.

Example of data exchange between DSP and SC for co-simulation via json strings :

From SC to DSP

```
{
  "sc2dsp": {
    "ProjectInfo": {
      "DemonstrationSite": "Cork Institute of Technology (CIT)",
      "DistrictInformationFile": "CIT_districtmodel.citygml"
    },
    "Variable": [
      {
        "description": "CHP Enable",
        "value": "True",
        "unit": "bool",
        "bms_tag": "CHPE-BH-01"
      },
      {
        "description": "Thermal Store Charge Setpoint",
        "value": 65.3,
        "unit": "°C",
        "bms_tag": "CSP-TS-BH-01"
      },
      {
        "description": "Boiler No 1 Enable",
        "value": "False",
        "unit": "bool",
        "bms_tag": "BRE-BH-01"
      },
      {
        "description": "Boiler No 2 Enable",
        "value": "False",
        "unit": "bool",
        "bms_tag": "BRE-BH-02"
      }
    ]
  }
}
```

BMS Tags are red-coloured to underline correspondence with real BMS values (described in Table 3)

From DSP to SC

```
{
  "dsp2sc": {
    "ProjectInfo": {
      "DemonstrationSite": "Cork Institute of Technology (CIT)",
      "DistrictInformationFile": "CIT_districtmodel.citygml"
    },
    "Variable": [
      {
        "description": "Thermal Store Temperature",
        "value": 56.2,
        "unit": "°C",
        "bms_tag": "TT-ST-01"
      },
      {
        "description": "CHP Flow Temperature",
        "value": 65.3,
        "unit": "°C",
        "bms_tag": "TT-CHP-F"
      },
      {
        "description": "CHP Return Temperature",
        "value": 48.2,
        "unit": "°C",
        "bms_tag": "TT-CHP-R"
      },
      {
        "description": "Outside Air Temperature",
        "value": 10.3,
        "unit": "°C",
        "bms_tag": "TT-OAT"
      }
    ]
  }
}
```

...

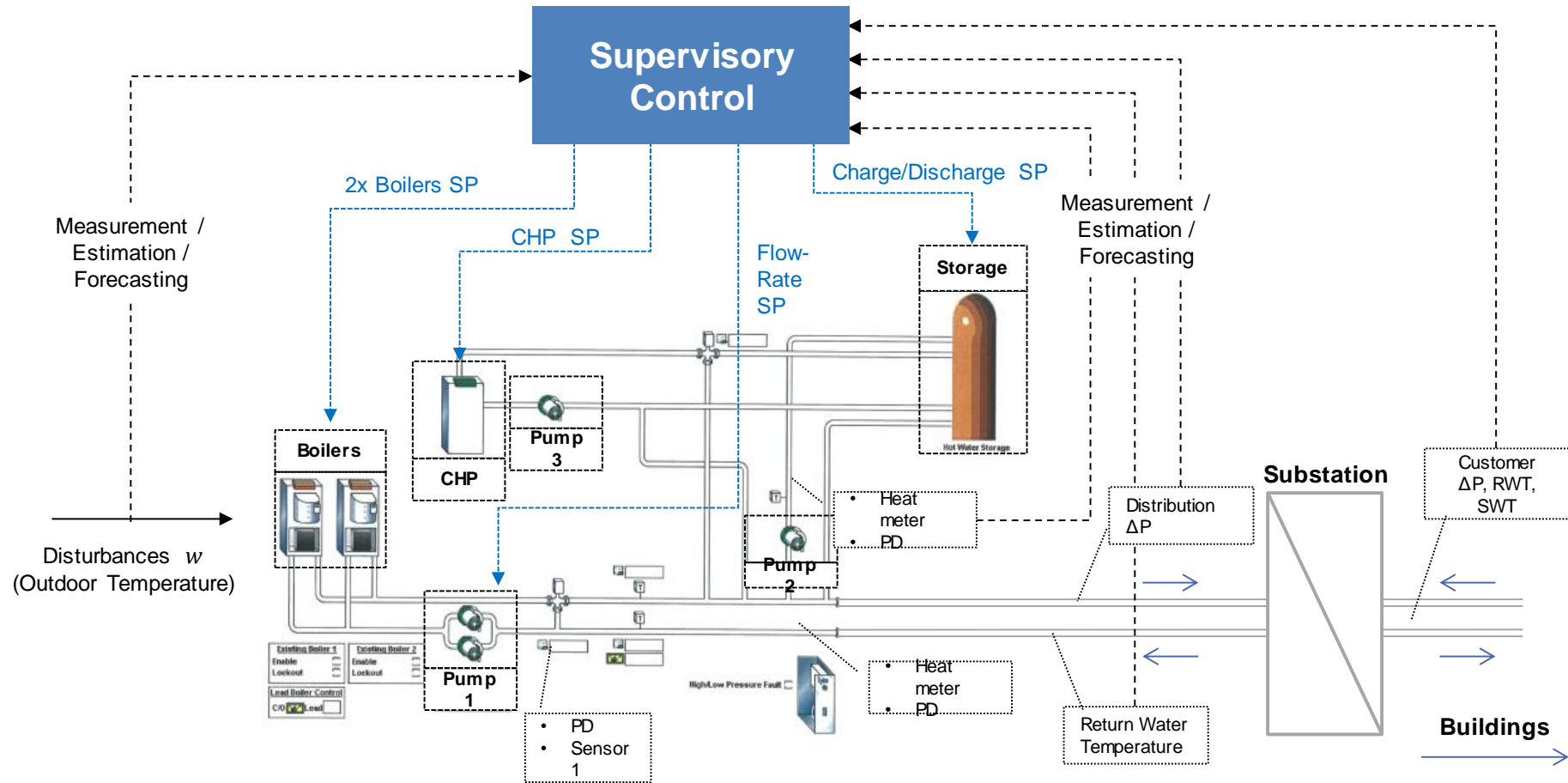


Figure 19: Schematic of DHN in CIT with SC interactions

Table 3 presents all the data expected to be controlled/measured when deploying the Supervisory Controller on CIT DHN

Table 3 : Supervisory Controller: measurements and set-points.

Subsystem/Equipment	Measurements		Control inputs	
	Description	BMS tag	Description	BMS tag
<u>Storage Tank</u>	Thermal Store Pump Run	TSPR-BH-01	Thermal Store Pump Enable	TSPE-BH-01
	Thermal Store Pump Diff Pressure Transmitter	DPT-BH-02	Thermal Store Pump Speed	TSPS-BH-01
	Storage Tank Flow Temperature Sensor No1, No2, No3	TT-ST-01, TT-ST-02, TT-ST-03	Thermal Store Discharge Setpoint	DSP-TS-BH-01
			Thermal Store Charge Setpoint	CSP-TS-BH-01
	Storage Tank High Level Immersion Sensor	Sensor 3		
	Thermal Store Circuit Heat Meter Flow Temp	FT-HM-TS		
	Thermal Store Circuit Heat Meter Return Temp	RT-HM-TS		
	Thermal Store Circuit Heat Meter Flow Rate	FR-HM-TS		
	Thermal Store Circuit Heat Meter Energy	E-HM-TS		
	Thermal Store Circuit Heat Meter Power	P-HM-TS		

<u>CHP</u>	CHP Flow Temperature Sensor	TT-CHP-F		
	CHP Return Temperature Sensor	TT-CHP-R		
			CHP Mixing Actuator Override	VO-MV-BH-02
			CHP Mixing Actuator No 2	MV-BH-02
			CHP Enable	CHPE-BH-01
			CHP Load Setpoint	CHP-LSP-01
			CHP Startup Time	CHP-ST-HRS
			Main Header Return Setpoint CHP	MHSP-CHP
<u>Boilers</u>			Pump Set P-03	
			Boilers On/Off Valve Override	VO-MV-BH-01
			Boilers On/Off Valve Actuator	MV-BH-01
			Boiler No 1 Enable	BRE-BH-01
			Boiler No 2 Enable	BRE-BH-02
			Main Header Return Setpoint Boilers	MHSP-BH

<u>Plant</u>	Outside Air Temperature Sensor	TT-OAT		
	Main Header Flow Temperature Sensor	TT-MH-F		
	Main Header Return Temperature Sensor	TT-MH-R		
	Main Header Pump Diff Pressure Transmitter	DPT-BH-01		
	Main Heating Circuit Heat Meter Flow Temp	FT-HM-MH		
	Main Heating Circuit Heat Meter Return Temp	RT-HM-MH		
	Main Heating Circuit Heat Meter Flow Rate	FR-HM-MH		
	Main Heating Circuit Heat Meter Energy	E-HM-MH		
	Main Heating Circuit Heat Meter Power	P-HM-MH		

Example of data exchange between DSP and PSO via json strings :From PSO to DSP (example for a 6 hours period with 1 hour time step)

```

{
  "pso2dsp": {
    "ProjectInfo": {
      "DemonstrationSite": "Cork Institute of Technology (CIT)",
      "DistrictInformationFile": "CIT_districtmodel.citygml"
    },
    "Time": {
      "description": "Simulation Time",
      "value": [1,2,3,4,5,6],
      "unit": "hours"
    },
    "ProductionSchedule": {
      "CHP_enable": {
        "description": "CHP enable (in [0 1])",
        "value": [1,1,0,0,0,0],
        "unit": "-"
      },
      "CHP_setpoint": {
        "description": "DHN supply setpoint",
        "value": [65,62,62,62,65,62],
        "unit": "°C"
      }
    },
    "WeatherForecast": {
      "Outside_temperature": {
        "description": "Predicted outside temperature",
        "value": [5.6,9.2,10.6,9.2,5.6,0.6],
        "unit": "°C"
      },
      "Direct_horizontal_radiation": {
        "description": "Predicted direct radiation on horizontal plane",
        "value": [328,511,578,511,328,78],
        "unit": "W/m2"
      },
      "Diffuse_horizontal_radiation": {
        "description": "Predicted diffuse radiation on horizontal plane",
        "value": [169.5,279.3,319.5,279.3,169.5,19.5],
        "unit": "W/m2"
      },
      "Sky_temperature": {
        "description": "Predicted effective sky temperature",
        "value": [-3.9,-3.8,-4.3,-4.4,-3.7,-2.3],
        "unit": "°C"
      }
    }
  }
}

```

From DSP to PSO (example for a 6 hours period with 1 hour time step)

```

{
  "dsp2psa": {
    "ProjectInfo": {
      "DemonstrationSite": "Cork Institute of Technology (CIT)",
      "DistrictInformationFile": "CIT_districtmodel.citygml"
    },
    "Time": {
      "description": "Simulation Time",
      "value": [1,2,3,4,5,6],
      "unit": "hours"
    },
    "Production": {
      "SupplyTemperature": {
        "description": "DHN Supply Temperature",
        "value": [65.5,65.8,65.3,63.2,65.3,64.7],
        "unit": "°C"
      },
      "ReturnTemperature": {
        "description": "DHN Return Temperature",
        "value": [48.8,47.9,48.1,47.2,49.8,48.4],
        "unit": "°C"
      },
      "FlowRate": {
        "description": "DHN Flow Rate",
        "value": [26005.9,26213.4,26024,25987,25586.1,25468.1],
        "unit": "kgVhr"
      }
    },
    "Buildings": [
      {
        "Name": "Block Old Cert Area",
        "Temperature": {
          "value": [20.7,20.3,21.6,20.6,21.1,20.3],
          "unit": "°C"
        }
      },
      {
        "Name": "A141",
        "Temperature": {
          "value": [21.2,20.5,21.3,21.4,21.5,20.9],
          "unit": "°C"
        }
      },
      {
        "Name": "B158",
        "Temperature": {
          "value": [20.2,20.5,21.8,20.3,21.7,21.1],
          "unit": "°C"
        }
      },
      {
        "Name": "B178",
        "Temperature": {

```

```
        "value": [22,20.2,20.9,20.2,21.9,20],
        "unit": "°C"
      },
    {
      "Name": "C126",
      "Temperature": {
        "value": [21.5,21.6,21.7,20.2,20.8,20.5],
        "unit": "°C"
      }
    },
    {
      "Name": "C158",
      "Temperature": {
        "value": [21.6,20.9,21.8,20.4,20.5,20.3],
        "unit": "°C"
      }
    },
    {
      "Name": "D136",
      "Temperature": {
        "value": [20.3,21.7,21.2,21.1,20.3,21.7],
        "unit": "°C"
      }
    }
  ]
}
```

6 Appendix B: Correspondence with project KPIs (T1.3)

Table 4: Variables needed for the KPIs calculation (task1.3)

System	Variables	Time step	KPI				
			Technical	Environmental	Economical	Quality of service	Social
Boilers	Energy and power	10 minutes (1 minute if possible)	x				x
	Energy input (kWh LHV)		x	x	x		x
	Interruption period (hour)					x	
Renewable and recuperation energy	Energy and power available	Hour or less	x				x
Cogeneration	Heat production	Hour or less	x				x
	Electricity production		x	x	x		x
	Energy input (kWh LHV)		x	x	x		x
	Number of running hour		x				x
All auxiliary (pump, valve,...) for production and energy distribution	Electrical consumption	Hour or less	x				
Substations	Energy delivered	10 minutes (1 minute if possible)	x	x	x	x	x
	Number of running hour						x
Buildings	Indoor temperature	10 minutes (but can depend on sensor)	x			x	x
	Window state						x
	Door state						x
	Heat appliance settings						x
	Occupancy						x
	Humidity						x
	CO2						x
Other	Outdoor temperature	10 minutes	x				