Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of the Environment, Transport, Energy and Communications DETEC

Swiss Federal Office of Energy SFOE
Energy Research and Cleantech

Yearly Report October 29, 2025

RESULTES

Reliable pre-feasibility and design tool for quick assessment of large TES integrated with renewable energy sources and DHN

Publisher:

Swiss Federal Office of Energy SFOE
Energy Research and Cleantech
CH-3003 Bern
www.energy-research.ch

Subsidy recipients:

OST - Ostschweizer Fachhochschule
SPF Institut für Solartechnik
Oberseestr. 10
CH-8640 Rapperswil
www.spf.ch

Authors:

Ignacio Gurruchaga, ignacio.gurruchaga@ost.ch
Dr. Daniel Carbonell, DCarbo Energy Consulting S.L., daniel.carbonell@dcarbo.es
Damian Birchler, damian.birchler@ost.ch

SFOE project coordinators:

Andreas Eckmanns, Andreas.Eckmanns@bfe.admin.ch
Stephan A. Mathez, stephan.a.mathez@solarcampus.ch

SFOE Contract number: SI/502780-01

Contents

List of Acronyms

ATES	Aquifer Thermal Energy Storage
BTES	Borehole Thermal Energy Storage
CFD	Computational Fluid Dynamics
DH	District heating
DHN	District Heating Network
GUI	Graphical User Interface
HP	Heat Pump
HX	Heat Exchanger
KPI	Key Performance Indicator
PTES	Pit Thermal Energy Storage
PV	Photovoltaic
TES	Thermal Energy Storage
TTES	Tank Thermal Energy Storage

Abstract

The analysis of the integration of large TES in District Heating Networks (DHNs) and renewable sources requires dynamic simulation to address the fluctuating behaviour of sources and sinks. The development of dynamic simulation energy system models requires experience, data for validation, and, above all, it is time-consuming. The need to evaluate many possibilities with a small budget in early project phases, such as pre-design, sees planners use simplistic methodologies which might lead to inaccurate results and even sometimes the non-consideration of a given technology if an easy-to-use evaluation tool is not available. Furthermore, in later project stages, only few planners use dynamic simulations, e.g. using TRNSYS, for the evaluation of the TES integrated into the whole thermal system. This project aims to change that by developing a reliable quick assessment tool targeting both the pre-feasibility and design phase of large-scale TES, unifying the calculation processes of all project stages in a single methodology. We propose an easy-to-use, user-friendly planning tool based on dynamic simulations models already developed within the TRNSYS-based pytrnsys https://pytrnsys.readthedocs.io/en/latest/index.html open-source environment.

Zusammenfassung

Die Analyse der Integration von großen Wärmespeichern (TES) in Fernwärmenetze (District Heating Networks (DHNs)) mit erneuerbaren Energiequellen erfordert dynamische Simulationen, um das fluktuierende Verhalten von Quellen und Senken ausreichend genau berücksichtigen zu können. Die Entwicklung dynamischer Simulationsmodelle, wie z.B. zur Integration von TES mit erneuerbaren Energiequellen zur Wärmeversorgung, erfordert Erfahrung, Daten zur Validierung und vor allem viel Zeit. Insbesondere in frühen Projektphasen, wie z.B. in der Vorplanung, verwenden Planer aus Mangel an Ressourcen in der Regel einfache und schnelle Berechnungsmethoden, die weder auf dynamischen Simulationen basieren noch standartisiert sind. Dies kann zu entsprechend ungenauen Ergebnissen führen. Auch in späteren Projektphasen nutzen aufgrund des hohen Aufwands nur wenige Planer dynamische Simulationsprogramme wie TRNSYS, um den TES in das gesamte thermische System zu integrieren. Dieses Projekt zielt darauf ab, diese Lücke durch die Entwicklung eines zuverlässigen Planungstools zur schnellen Bewertung von grossen TES sowohl für die Vorplanung als auch für spätere Planungsschritte zu schliessen. Dabei werden die notwendigen Berechnungen für alle Projektphasen in eine einzige Methodik auf Basis von dynamischen Simulationsmodellen integriert. Dazu wird die TRNSYS-basierten Open-Source-Umgebung pytrnsys https://pytrnsys.readthedocs.io/en/latest/index.html verwendet.

1 Introduction

1.1 Problem description

The decarbonization of the energy supply of District Heating Networks (DHNs) involves the adoption of energy efficiency measures, such as waste heat recovery from industrial processes or co-generation, or integrating different renewable sources such as solar thermal energy, heat pumps or biomass. TES play an important role in a smoother and more flexible coupling of different renewable energy sources in DHNs due to the capability to overcome mismatches between renewable energy production and heat demand.

Depending on the project's constraints, different TES must be considered: Tank Thermal Energy Storage (TTES), Pit Thermal Energy Storage (PTES), Borehole Thermal Energy Storage (BTES) or Aquifer Thermal Energy Storage (ATES) thermal energy storages (Dahash, Ochs, Janetti, & Streicher, 2019; Dahash, Ochs, Tosatto, & Streicher, 2020; Dahash, A., Ochs, F., & Tosatto, A., 2020). For a higher storage performance (good stratification, higher energy density, low thermal losses) to respond to fluctuating demands, large-scale TTES and PTES, from several hundred to hundreds of thousand cubic meters, have been implemented in different DHNs in Europe. However, PTES and TTES have the disadvantage of higher investment cost compared to BTES and ATES. Therefore, proper planning and operation are essential for their economic feasibility.

The interaction between renewable or waste heat sources and demands with the TES exhibits a dynamic and time-dependent behaviour: solar energy has a diurnal cycle with a higher seasonal concentration in summer, and most of the demand for space heating occurs in winter (Dahash et al., 2019).

The operation of air-source heat pumps can be driven by different criteria (efficiency, renewable Photovoltaic (PV) production or market electricity prices), and certain waste energy sources can have very specific utilisation cycles. The dynamic nature of all these phenomena will entail a continuous variation of TES temperatures and, in turn, the efficiency of renewable energy sources such as solar thermal or heat pumps. In extreme cases, this could limit the exploitable capacity of these heat sources. Analysis of peak demand reduction also requires dynamic simulations.

The development of engineering planning projects integrating large-scale TES, before their future procurement, construction, and operation, is carried out in different phases: from pre-feasibility analysis, via a design phase, to optimization and detailed engineering (Dahash et al., 2019; Ochs, Dahash, Tosatto, & Bianchi Janetti, 2020). In each of these phases, different issues are addressed that normally need to be solved using different methodologies or tools. The investment of resources in the development of the project, as shown in figure. 1, is normally proportional to its progress: the pre-design phase is usually allocated relatively little resources, while more detailed phases are more resource-consuming.

1.1.1 Pre-design phase

In the pre-design phase, TES pre-feasibility and its preliminary integration with demands and with different renewable energy options must be assessed. At this stage, little or very little data is normally available both about the demands and renewable heat sources, and tools are best kept simple in order to decide TES cost competitiveness accordingly to this limited available data. In this phase the TES type (TTES, PTES, etc.) and capacity of TES (indeed related to the type) should be analyzed for their cost-effectiveness (Dahash et al., 2020).

In this phase, planners usually use simple tools. These turn often turn out to be too simple and are not standardised. Usually, static calculations, and not dynamic simulations, as would be desirable, are carried out in these simple tools because of a misconception regarding the resources needed to use dynamic tools such as TRNSYS. Unreliable tools or methodologies may have led to a TES integration solution being rejected when it could have been cost-effective and to the acceptance of a TES that might have been discarded with more reliable tools.

1.1.2 Design phase

Once the suitability of a TES has been established and the technology chosen, the conceptual design and integration of the different renewable or waste heat energy sources should be approached in accordance with

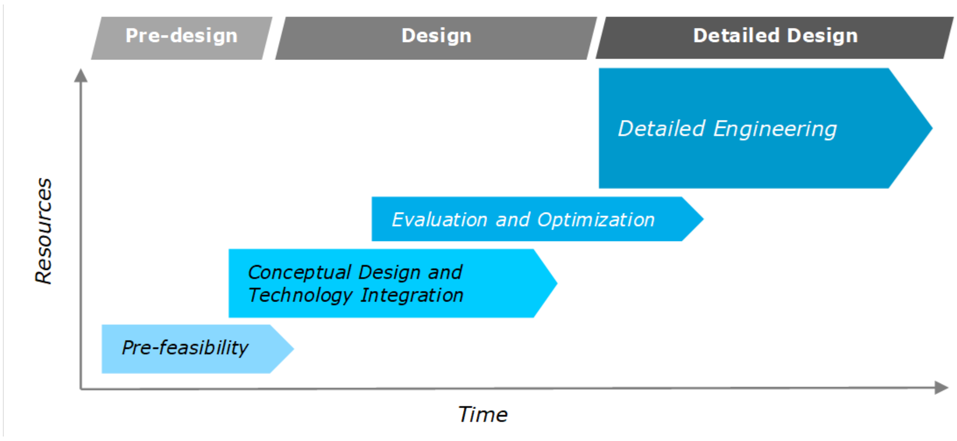


Figure 1: Different phases in a TES project distributed on Time and its available Resources

the TES and foreseen demands.

The first step is to analyse the suitability of the integration of different energy sources (either solar thermal energy, heat pumps, biomass, waste heat or a combination of the previous ones). At this stage, more detailed specifications about the TES might be needed: volume, geometry, maximum thermal stratification, maximum pressure, number of ports, insulation and location (buried or not). As the thermal system behaves as a whole, changes to the renewable generation technologies or the demands affect the design of the TES. The opposite is also true: modifications in the design of the TES will, in turn, affect the compliance with the demands or the performance and operation of the renewable generation technologies.

This is why, in the design phase, the entire system is also analysed as a whole, from the generation of the energy sources to the demands, considering the TES as well as the DHN. In this design phase, the focus must be placed on the integration details. For example, regarding the TES, it is necessary to analyse

1) the number and positions of port inlets or outlets to increase the performance or profitability of the tank

2) the control strategy of the different renewable energy sources and their prioritisation to optimize the cost-competitiveness of the whole system

In this phase, some advanced planners use dynamic simulation tools such as TRNSYS for the evaluation of the global system performance (Gauthier, 2020; Jensen, 2015; McDaniel & Kosanovic, 2016; Ochs et al., 2022; Pan et al., 2022; Schmidt et al., 2018; Sibbitt et al., 2012; Sørensen & Schmidt, n.d.). Furthermore, TRNSYS components for TES have been successfully compared with experimental data (Schmidt & Sørensen, n.d.) and validated (Gauthier, 2020; Raab, Mangold, & Müller-Steinhagen, 2005; Xie et al., 2021).

1.1.3 Detailed design phase

Even though an interdisciplinary team of energy, environmental, construction, electrical, materials and computational engineers are working on developing the different aspects of the energy project, the focus on the Key Performance Indicators (KPIs) established in the earlier design phases must not be lost.

Therefore, the use of dynamic simulation tools with a focus on the whole system would continue to be necessary to assess the impact of eventual changes in advanced engineering decisions, thereby minimizing a decrease of the energetic performance and/or cost competitiveness.

On the other hand, when optimizing the TES itself, Computational Fluid Dynamics (CFD) would be preferable. Examples where CFD is necessary are: the design of diffusors of inlet water ports or for analyzing de-stratification as a result of convective flows generated by thermal losses as well as the interaction of the ground and the TES. (Chang, Nie, et al., 2017; Chang, Wu, et al., 2017; Dahash et al., 2020; Dahash,

A. et al., 2020; Ochs et al., 2020, 2022; Panthalookaran, El-Amin, Heidemann, & Müller-Steinhagen, 2008; Panthalookaran, Heidemann, & Müller-Steinhagen, 2011; Tosatto, Ochs, Dahash, & Muser, 2022). Dynamic simulation tools, such as TRNSYS or Modelica, have in some instances been coupled with CFD for detailed co-simulation analysis.

1.1.4 Conclusion

There is a lack of standardized methodologies and tools spanning the different design phases for large-scale TES projects. Dynamic simulation tools are often passed over because of the relatively long time - perceived or real - required for working with them.

1.2 Solution

To fill this gap, a reliable and accurate pre-feasibility and design tool, named RESULTES, for the quick assessment of large-scale TES is proposed. RESULTES shall unify all calculation processes for the different stages of project development in a single methodology based on dynamic simulations. This common methodology will improve and unify the chain of TES planning using a single tool.

The tool will use TRNSYS as its simulation engine but will present a different front end shaped and adapted to the different phases of the planning process, considering the different levels of information available in each. The different front-ends and models for the different design phases are summarized in Tab. 1.

Phase	RESULTES User interface / front-end	RESULTES Model
Pre-design	Web-based GUI	Based on similar TRNSYS / pytrnsys model. See Tab. 2 for more details.
Design	Web-based GUI	
Detailed design	pytrnsys GUI	

Table 1: Comparison of different RESULTES' front-ends and models for different design phases

1.2.1 pytrnsys

All dynamic simulation models will be developed in pytrnsys which is an open-source Python framework for carrying out and post-processing TRNSYS simulations. In pytrnsys, one draws a system's hydraulic scheme in order to define it. This facilitates the finding of errors and the visualization of results as calculated mass flow rates and temperatures can be shown overlaid onto the hydraulic diagram. Figure 3 shows a hydraulic diagram as it is being developed in the pytrnsys GUI. In addition, pytrnsys is especially well suited to running parametric studies by varying many parameters at once and automatically running the different variations in parallel.

1.2.2 Pre-design phase

The TRNSYS model for pre-feasibility analysis in pre-design phase will be a simplified system using a detailed TES with little input data required from the user. This is important since, in the design phase, little data is usually available regarding both the demand and renewable supply side. For this reason, the TRNSYS model foreseen for the pre-feasibility analysis consists of a detailed TES model connected to charging and discharging profiles that can be freely chosen by the user.

The charging profiles emulate the renewable sources while discharging profiles the demands. The control implemented in the simulation model will decide if and when it will make use of the sources available. Maintaining a detailed TES model will provide the user with a first idea about TES temperature evolution, the profitability of the potential renewable energy sources and the compliance with the demands. Also, this model will provide a first impression of the TES stratification, thermal losses, underground temperature development and the sensibility of all the above to the main cost parameter – its volume.

1.2.3 Design- and detailed design phase

On the basis of the same open-source framework pytrnsys, more detailed dynamic simulation models in TRNSYS, mostly for renewable energy sources and their integration into TES, will be developed for the design phase. These detailed models for different renewable energy sources - solar thermal, heat pump, biomass or waste heat energy sources - will be included in different hydraulic schemes in order to have a sufficient representation of different TES integrations into DHN.

1.2.4 Comparison of different models

The different models for the different design phases are summarized in table 2.

Phase	System model	TES model	Source models	Demand model
Pre-design	Simplified	Detailed [1]	Profile or simplified model	Profile
Design	Detailed, not modifiable by user	Detailed	Detailed	
Detailed design	Detailed, modifiable by user			

Table 2: Comparison of different models for different design phases

[1] The actual inclusion in the system might be simplified, e.g. the number of ports into the storage.

1.2.5 TES cost models

The RESULTES tool will give an estimate of the cost of the TES used within a simulated system. To this end, two cost models will be developed for each system within the project: a simplified and a more detailed version. The former will be used in the pre-design phase model and the latter for the two remaining, more detailed models.

1.2.6 RESULTES user interface ("front-end")

In the pre- and design phase, the user interacts with the models via a graphical, web-based user interface, also called "the front-end" further down. This common front end will unify both pre-design and design interfaces in order to be more consistent. In the detailed design phase, where it is possible to modify the system's hydraulic configuration, the user will not use the RESULTES GUI but pytrnsys' [1]. To transition between interfaces, the web-based user interface will enable the user to export the system as a pytrnsys project.

The RESULTES user interface is shown in figure 2.

A screenshot of the pytrnsys GUI with the TTES system opened is shown in figure 3.

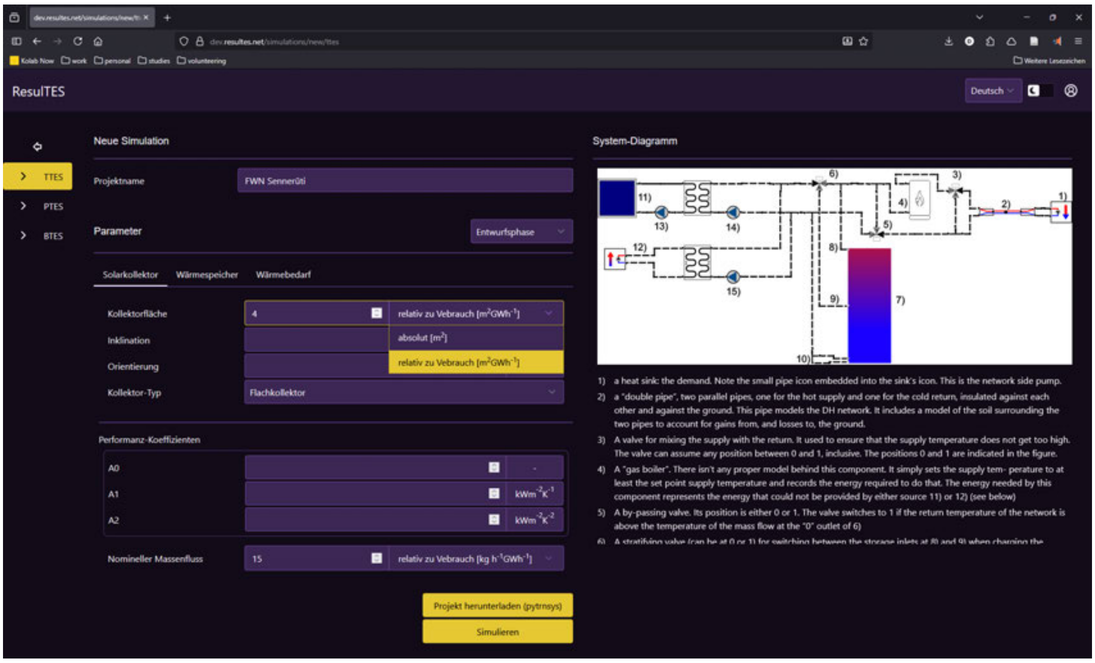
Both pytrnsys / TRNSYS models and cost models will be used by the front-end.

For the pre-design phase interface, the user will need to provide demand profiles and the potential profiles of renewable sources. The TRNSYS dynamic simulation model will use these profiles to evaluate the TES's thermal behaviour. For example, for simulating a solar thermal system, the user must provide a potential yearly energy yield production profile, i.e., a maximum energy yield profile. The control in the simplified TRNSYS model will ensure that TES will not exceed operating conditions like the TES temperature.

For the design phase interface, as detailed TRNSYS models for each renewable energy source are foreseen, more detailed input data regarding the main parameters will be necessary in order to run a full dynamic system-level simulation. Default values will be provided in case there is no specific data available to the user. For example, for simulating a solar thermal system, the user must provide the main solar thermal parameters: aperture area, coefficient of performance, IAM, etc. A more detailed control will ensure that TES and, in this case, the solar thermal source, will not exceed the operation conditions.

---

[1] The pytrnsys GUI is a separate from the RESULTES GUI. However, the user can download the pytrnsys Graphical User Interface (GUI) project from the RESULTES GUI.



Projektname: FWN Sennerüti

Parameter: Entwurfsphase

Kollektorfläche: 4 relativ zu Vebrauch [$m^2 GWh^{-1}$]

absolut [m^2]

relativ zu Vebrauch [$m^2 GWh^{-1}$]

Inklination

Orientierung

Kollektor-Typ: Flachkollektor

A0: -

A1: $kWm^{-2}K^{-1}$

A2: $kWm^{-2}K^{-2}$

Nomineller Massenfluss: 15 relativ zu Vebrauch [$kg\ h^{-1}GWh^{-1}$]

1) a heat sink: the demand. Note the small pipe icon embedded into the sink's icon. This is the network side pump.
2) a "double pipe", two parallel pipes, one for the hot supply and one for the cold return, insulated against each other and against the ground. This pipe models the DH network. It includes a model of the soil surrounding the two pipes to account for gains from, and losses to, the ground.
3) A valve for mixing the supply with the return. It used to ensure that the supply temperature does not get too high. The valve can assume any position between 0 and 1, inclusive. The positions 0 and 1 are indicated in the figure.
4) A "gas boiler". There isn't any proper model behind this component. It simply sets the supply tem- perature to at least the set point supply temperature and records the energy required to do that. The energy needed by this component represents the energy that could not be provided by either source 11) or 12) (see below)
5) A by-passing valve. Its position is either 0 or 1. The valve switches to 1 if the return temperature of the network is above the temperature of the mass flow at the "0" outlet of 6)
6) A stratifying valve (can be at 0 or 1) for switching between the storage inlets at 8) and 9) when charging the

Figure 2: Setup of a new TTES simulation in the RESULTES GUI

Weather data from Switzerland will be included per default with the front-end but it will be possible to change the location by the user by means of an external file containing the weather data. Once the simulation, or simulations in case of running parametric analysis, have run, the interface will provide the results to the user.

The pre-design tool results will include, among others, the renewable energy source profitability (how much of the available energy could actually be used), compliance with the demand, storage tank hourly temperatures and thermal losses. Other results, such as the reduction of peak demands on the TES, could also be provided as standard results.

In addition to the results of the pre-design tool, the design tool's results would also include the performance and hourly temperatures of the different renewable energy sources.

The front-end for the pre-design and design phases will be simple and quick to use and will not require any TRNSYS or programming knowledge in order to perform a dynamic simulation in TRNSYS of a TES system integrated with renewable energy sources.

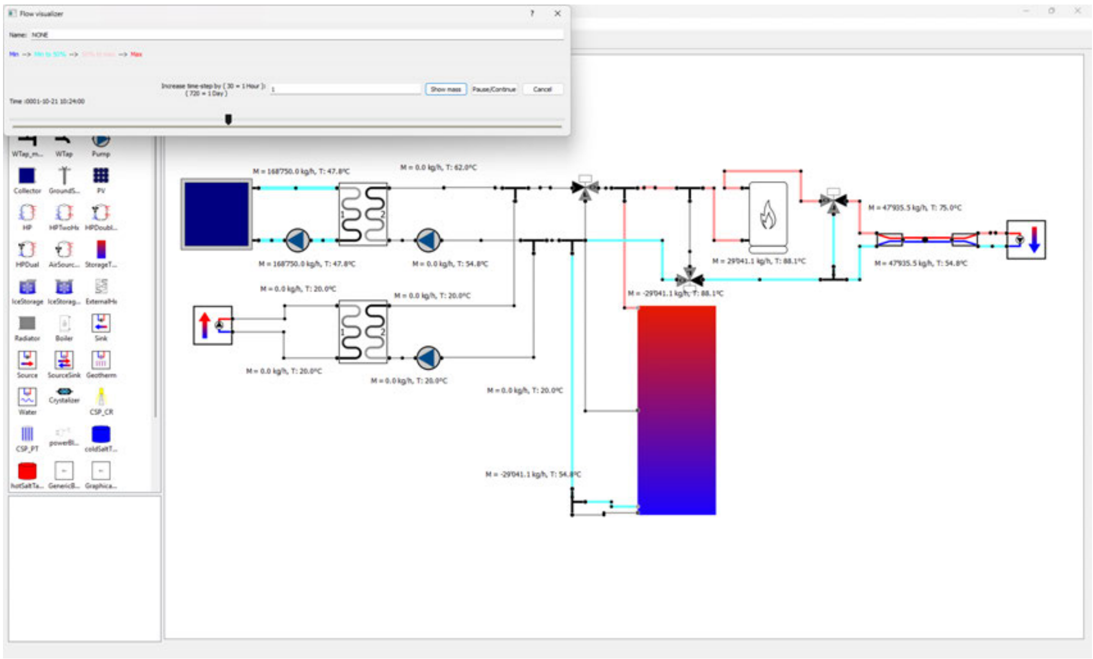
For the detailed design phase, users may need to make slight changes in the dynamic simulation models in TRNSYS. For that reason, in the detailed design phase, users will use the open-source pytrnsys tool directly. Starting from a functioning, detailed and validated system model, they will be able to perform all the necessary changes needed in the their project's detailed design phase. Making changes to the pytrnsys model will require knowledge of TRNSYS, and, preferable, to work with pytrnsys, some knowledge of the Python programming language.

An overview of the required inputs and results for each design phase is given in table 3.

1.2.7 RESULTES tool's main features

To summarize, the main features of the RESULTES tool will be:

Consistency and reliability For all three phases, the RESULTES tool will rely on dynamic simulations and a shared methodology and models. Verified and validated models for the storage types will be used.



Name: NONE

Increase time-step by (30 = 1 Hour): (720 = 1 Day) 1

Time :0001-10-21 10:24:00

Figure 3: The TTES system opened in the pytrnsys GUI

Phase	Source inputs	TES inputs	Demand inputs	Results
Pre-design	Capacity profile	Volume	Profile	1) profitability of sources 2) compliance with demand 3) TES pre-feasibility 4) TES hourly temperatures 5) TES thermal losses 6) TES performance [1]
Design	Main parameters	All parameters	Profile	1) profitability of sources 2) compliance with demand 3) TES cost-analysis 4) TES hourly temperatures 5) TES thermal losses 6) TES performance [1] 7) TES soil temperatures [2]
Detailed design	All parameters	All parameters	Profile	

Table 3: Comparison of different models for different design phases

[1] For parametric runs
[2] If applicable

Speed The user is presented with a working model. As such, the most time-consuming part of running a dynamic simulation, i.e. the creation of the model, has been taken care of already. It should take the user less than half an hour to have pre-design results when starting from scratch.

Usability The level of information required to run a given simulation is suited to the phase the project is in: little information in the pre-design phase and progressively more in the later phases. The tool is maintained long-term by the tool's developers.

2 Project description

2.1 Project goals/research questions

The project's general objective is to improve the technical design and dimensioning capabilities for the planning of large thermal heat storage systems through the development of a reliable pre-feasibility and design tool for quick assessment based on TRNSYS dynamic simulations.

The specific objectives of the project are:

1) Develop TRNSYS models and hydraulic schemes based on pytrnsys / TRNSYS that will be used as the background calculation engine of the RESULTES tool.

2) Develop one simplified TES cost model and one detailed multi-parametrical cost model that will be integrated into the RESULTES tool.

3) Develop the RESULTES tool user interface based on the pytrnsys / TRNSYS engine which includes the interface development for running simulations and parametric analyses and the post-processing of results.

4) The validation of the RESULTES tool

5) Dissemination of the RESULTES tool

2.2 Work packages and tasks

The project's work packages and tasks are listed in table 4. The work packages must be completed for each system. As explained later, instead of completing one whole work package after the other, we chose instead to proceed on a system-by-system basis, i.e. completing all the work packages for the first system, before restarting at WP2) again for the second system, and so forth.

Task	Description	Work package
T1)	Project coordination	WP1)
T2.1)	Define how the heat sources, sinks and storages are integrated into the whole system, i.e. define the hydraulic setup of the system.	WP2)
T2.2)	Having defined the system's components and how they connect to each other, implement the simulation of the system. This includes setting up the component models and implementing the control of the system, i.e. when and in what way pumps and valves are actuated.	
T3)	Implement a cost model for the system to calculate the system cost given a set of relevant system parameters (e.g. collector area, storage size, etc.).	WP3)
T4.1)	Implement the user-facing web-based front-end for the system.	WP4)
T4.2)	Set up the infrastructure for running simulations of the system, managing result data and processing it to generate reports on the user's behalf on SPF's servers.	
T5)	Validate "the final user-facing product", i.e. all the above steps.	WP5)
T6)	Dissemination of work in the form of tutorials, presentation at seminars, congresses or working groups and/or publications	WP6)

Table 4: Tasks and work packages

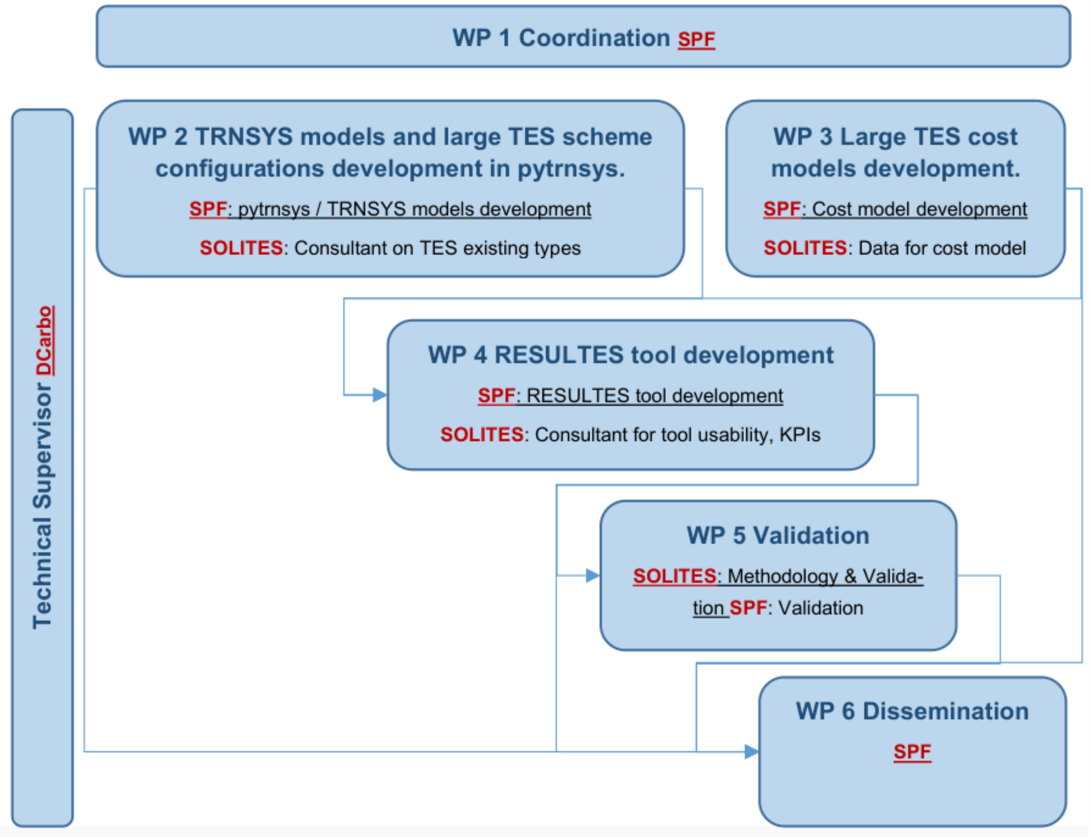


Figure 4: The work packages including involved and responsible (underlined) parties.

2.3 National and international collaboration

The project is done in collaboration with an external partner, DCarbo Energy Consulting (DCarbo), based in Barcelona, Spain. Solites (Steinbeis Research Institute for Solar and Sustainable Thermal Energy Systems), based in Stuttgart, Germany, will support the project work as an external expert.

2.4 Responsibilities of involved parties

SPF is responsible for the overall project management and will be the main party working on the project. Solites is the responsible party for the validation of the tool developed within the project. DCarbo will have a supporting and consulting role throughout the project. The responsibilities are shown graphically in figure 4.

2.5 Project Planning

At the kickoff meeting with the SFOE held on June 25, 2024, it was agreed to proceed system by system: we will first focus on the TTES system, implement the simulation, the web interface, and the infrastructure to run and post-process and present the results on the web to the user. Only then will we turn our attention to the PTES system. Finally, after completing the PTES system, we will start working on the BTES system.

This new approach differs from the approach lined out in the project proposal. A redefinition of the project milestones was therefore proposed and accepted at the kick-off meeting.

In addition, because the project started six months later a project extension of three months till June 2026 was asked for and granted.

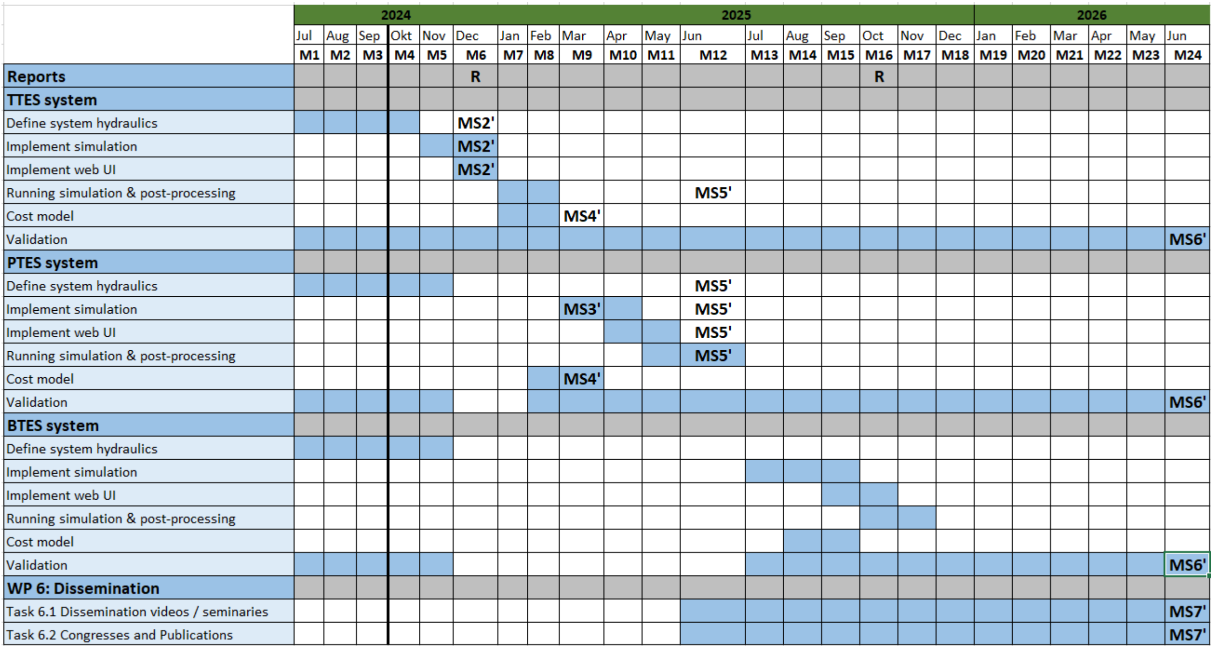
The new milestones (indicated with a prime (') like, e.g., MS2') and their deadlines are described in the next section.

2.6 Milestones

The milestones are summarized in table 5. The schedule for the milestones is visualized in figure 5. The time starting from December 2026 will be used for the second iteration of the RESULTES tool. We aim to have a working but basic implementation of the tool for each of the systems early on, i.e. by December 2026. From then on, based on the user experience and feedback that we get, we'd like to correct errors, improve and possibly extend the tool developed so far.

Milestone	**System**	**Task**	**Deadline**
MS2'	S1) TTES	T2.1) System hydraulics T2.2) Simulation and control T4.1) Web-based front-end (inputs only)	December 8, 2024
R1	-	Yearly report 2024	December 8, 2024
MS3'	S2) PTES	T2.1) System hydraulics T2.2) Simulation and control	March 2025
MS4'	S1) TTES S2) PTES	T3) Cost model	March 2025
MS5'	S1) TTES S2) PTES	T4.1) Web-based front-end T4.2) SPF server side infrastructure	March 2025
R2	-	Yearly report 2025	October 2025
MS6'	S1) BTES	T2.1) System hydraulics T2.2) Simulation and control T3) Cost model T4.2) Web-based front-end T4.2) SPF server-side infrastructure	June 2026
MS7'	S1) TTES S2) PTES S3) BTES	T5) Validation	June 2026
MS8'	S1) TTES S2) PTES S3) BTES	T6) Dissemination	June 2026
R3	-	Final report	June 2026

Table 5: Project milestones and associated tasks.



	2024						2025												2026					
	Jul	Aug	Sep	Okt	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
	M1	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**	**M9**	**M10**	**M11**	**M12**	**M13**	**M14**	**M15**	**M16**	**M17**	**M18**	**M19**	**M20**	**M21**	**M22**	**M23**	**M24**
Reports						**R**										**R**								
TTES system																								
Define system hydraulics	🟦	🟦	🟦	🟦		**MS2'**																		
Implement simulation					🟦	**MS2'**																		
Implement web UI						**MS2'**																		
Running simulation & post-processing							🟦	🟦				**MS5'**												
Cost model							🟦	🟦	**MS4'**															
Validation	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	**MS6'**
PTES system																								
Define system hydraulics	🟦	🟦	🟦	🟦	🟦							**MS5'**												
Implement simulation									**MS3'**	🟦		**MS5'**												
Implement web UI										🟦	🟦	**MS5'**												
Running simulation & post-processing											🟦	**MS5'**												
Cost model								🟦	**MS4'**															
Validation	🟦	🟦	🟦	🟦	🟦			🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	**MS6'**
BTES system																								
Define system hydraulics	🟦	🟦	🟦	🟦	🟦																			
Implement simulation													🟦	🟦	🟦									
Implement web UI															🟦	🟦								
Running simulation & post-processing																🟦	🟦							
Cost model														🟦	🟦									
Validation	🟦	🟦	🟦	🟦	🟦								🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	**MS6'**
WP 6: Dissemination																								
Task 6.1 Dissemination videos / seminaries												🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	**MS7'**
Task 6.2 Congresses and Publications												🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	**MS7'**

Figure 5: Schedule for the milestones

3 Progress report

3.1 Overview

Tasks T2.1), T2.2) and T4.1) for the TTES (S1) system have been completed. This completes milestone MS2'. T2.1) has been completed for the other two systems, i.e. work on MS3', MS4' and MS6' has already begun. The current state of completion for the milestones is summarized in table 6.

3.2 WP1) Project coordination

As part of work package one, the following fundamental questions regarding the project have been investigated.

3.2.1 RESULTES tool

In consultation with the project partners, it was decided that the RESULTES tool should be accessible online via a web browser. That way, the user will not have to install any software to start their investigation and they need not be concerned with running the simulations: the simulations will be run on the server(s) provided by SPF. The user will be notified when a simulation has finished, and post-processing and storage of the results will be taken care of by the RESULTES software.

For maximum flexibility — typically at a later stage in a project's development, e.g., in the detailed design phase —, the user can also choose to download the pytrnsys simulation. Having done so, they are free to change any system parameter, the control of the system or even the hydraulic connections. In short, they can change every aspect of the system or even choose the system as the starting point for building a new system. They might also choose to download simulation results to perform their own post-processing and analysis using a tool of their own choosing (e.g. Excel or R).

Milestone	**System**	**Task**	**Deadline**	**Status**
MS2’	S1) TTES	T2.1) System hydraulics T2.2) Simulation and control T4.1) Web-based front-end	December 8, 2024	Completed
R1	N/A	Yearly report 2024	December 8, 2024	Completed
MS3’	S2) PTES	T2.1) System hydraulics	March 2025	Completed
		T2.2) Simulation and control		
MS4’	S1) TTES S2) PTES	T3) Cost model	March 2025	First model available
MS5’	S1) TTES S2) PTES	T4.1) Web-based front-end	June 2025	Prototype online
		T4.2) SPF server side infrastructure		Completed
R2	N/A	Yearly report 2025	October 2025	Completed
MS6’	S1) BTES	T2.1) System hydraulics T2.2) Simulation and control	June 2026	Completed
		T3) Cost model		First model available
		T4.2) Web-based front-end		Prototype online
		T4.2) SPF server-side infrastructure		Completed
MS7’	S1) TTES S2) PTES S3) BTES	T5) Validation	June 2026	Initial work started
				-
MS8’	S1) TTES S2) PTES S3) BTES	T6) Dissemination	June 2026	Ongoing
R3	N/A	Final report	June 2026	-

Table 6: Completion status of project tasks. Tasks with entry “-” in the **Status** column have not been started.

3.2.2 Storage technologies and systems

In discussion with our partners, we concluded that we would like to focus on three widely used, representative storage technologies within the field of DHN: TTES, PTES and a BTES. For each of these three storage technologies, we will provide a simulation model of the storage embedded in a wider system (called systems S1), S2) and S3), respectively, from now on). A system includes the storage, heat producers and consumers, hydraulic components (pipes, tee-pieces, valves and pumps of plants and the network), the ground, the weather and the control of the pumps and valves in the system. For RESULTES to be the most useful, the three systems should be representative of widely deployed existing systems.

3.2.3 Project organization

The strategy of implementing the simulation, web interface and server infrastructure for each system one after the other outlined in the 2024 yearly report has been relaxed: the PTES and BTES system simulations were developed in parallel with the server infrastructure of all three systems. This was done to better take advantage of available personnel, to be able to better parallelize work and to speed up project progress.

3.3 WP2) Implement system model using pytrnsys and TRNSYS

s

3.3.1 T2.1) System hydraulics

S1) TTES system The tank storage system is the simplest of the three systems considered. Figure 21 shows the system schematic in *pytrnsys*. It contains the following components:

1) a heat sink: the demand. Note the small pipe icon embedded into the sink's icon. This is the network side pump.
2) a "double pipe", two parallel pipes, one for the hot supply and one for the cold return, insulated against each other and against the ground. This pipe models the District heating (DH) network. It includes a model of the soil surrounding the two pipes to account for gains from, and losses to, the ground.
3) A valve for mixing the supply with the return is used to ensure that the supply temperature does not get too high. The valve can assume any position between 0 and 1, inclusive. The positions 0 and 1 are indicated in the figure.
4) A "gas boiler". There isn't any proper model behind this component. It simply sets the supply temperature to at least the set point supply temperature and records the energy required to do that. The energy needed by this component represents the energy that could not be provided by either source 11) or 12) (see below)
5) A by-passing valve. Its position is either 0 or 1. The valve switches to 1 if the return temperature of the network is above the temperature of the mass flow at the "0" outlet of 6)
6) A stratifying valve (can be at 0 or 1) for switching between the storage inlets at 8) and 9) when charging the storage. The valve position will be chosen such that the temperature at the storage inlet is closer to the supply temperature going into the valve.
7) The TES tank
8) The tank inlet for charging with high temperatures / The tank outlet for discharging.
9) The tank inlet for charging with lower temperatures.
10) The cold tank inlet/outlet. Due to technical limitations of the simulation framework, storage ports must come in pairs. In reality, only one port would be needed instead of the two used at 10). Therefore, thet are immediately combined using the T-piece.

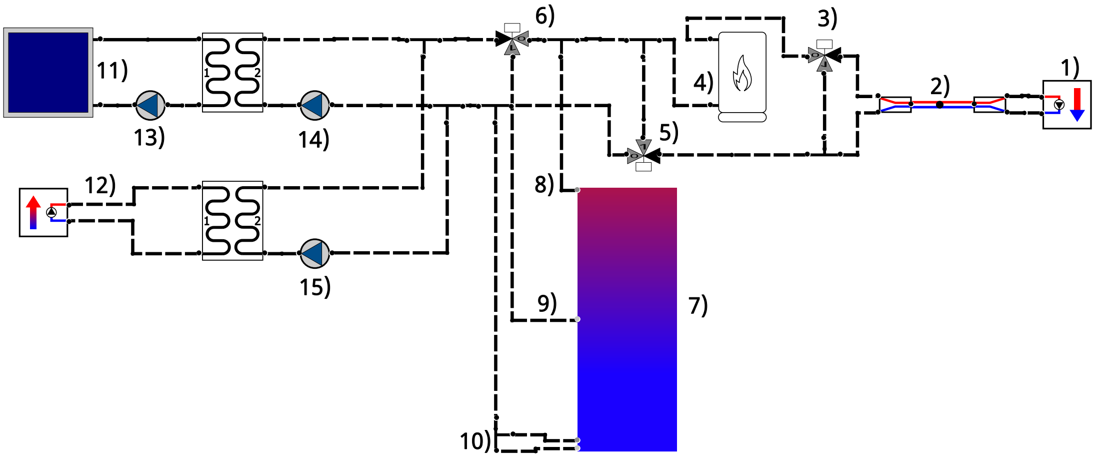


Figure 6: The hydraulic scheme for the TTES system in *pytrnsys*.

11) The solar thermal collector field heat source
12) The user-defined heat source
13) The pump for the primary loop of the solar thermal system
14) The pump for the secondary loop of the solar thermal system
15) The pump for the secondary loop of the user-defined source (the pump for the primary loop is "built into" the source as is indicated by the small pump inside the source's icon)

Note that the storage can be charged via the port at 9) while being discharged at 8). This can occur when the sources provide energy to the system, but not enough to reach the supply temperature. Then, the supply temperature is provided from the top of the storage while lower-temperature heat is injected into the middle of the storage.

Depending on the volume of the storage, it can serve as a buffer tank for peak loads or as a seasonal storage tank, or anything between these two extremes.

The working fluid is plain water throughout the system.

Any difference in mass flow rate between the secondary source pumps and the network pump (shown as embedded in the sink 1) in 21 will flow through the storage. In this way, the storage can absorb energy in case of surpluses. It can also provide additional energy in case of a lack of supply.

S2) PTES system The hydraulic scheme of the PTES system is given in figure 7. The differences with respect to the TTES system are the pit storage at 4) and the additional heat exchanger between the three-way valves at 1) and heat pump between the three-way valves at 2). In a first step, the supply flow from the renewable or user-defined sources and storage goes through the heat exchanger at 1) to preheat the DH network's return flow. In so doing, the supply flow is cooled down for a first time. It then goes into the heat pump's cool side to further heat the pre-heated return flow, which passes through the heat pump's hot side to reach the required supply temperature.

Both the heat exchanger and/or the heat pump can be skipped in the above scenario by switching both valves at 1) or both valves at 2), respectively. All the valves concerned are either at position 0 or 1.

As in the TTES system, there is a valve to control the inlet position into the pit storage for improved stratification within the storage. And, like in the TTES system, the mass flow rate into or out of the storage is determined by the difference in the secondary source pumps and the primary supply pump at 3). The latter's mass flow rate is set to be equal to the mass flow rate of the network pump.

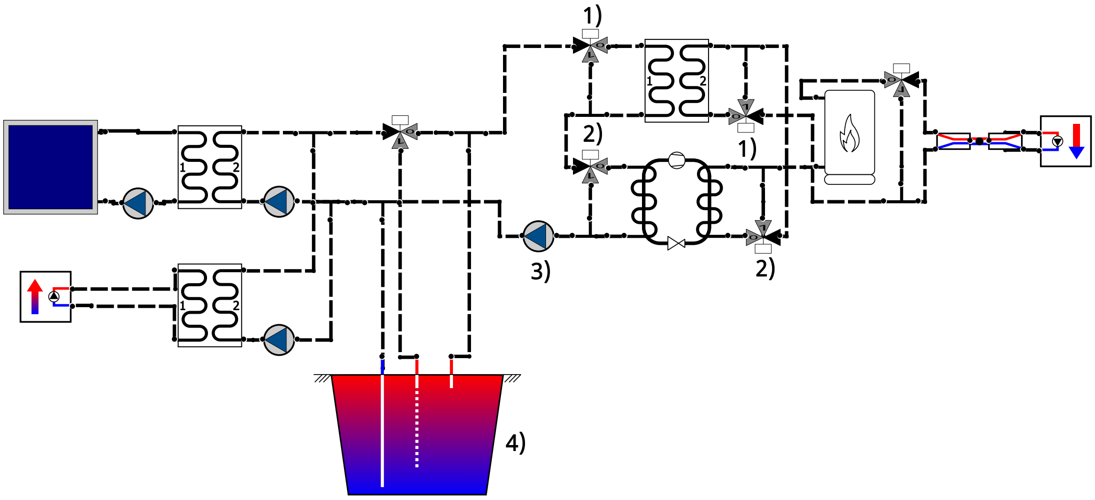


Figure 7: The hydraulic scheme for the PTES system in *pytrnsys*.

Figures 8, 10 and 9 show the different modes of the PTES system. The pipes' colors must be disregarded, as explained above. In all these figures the additional source is active, but there is no pre-mixing in the supply.

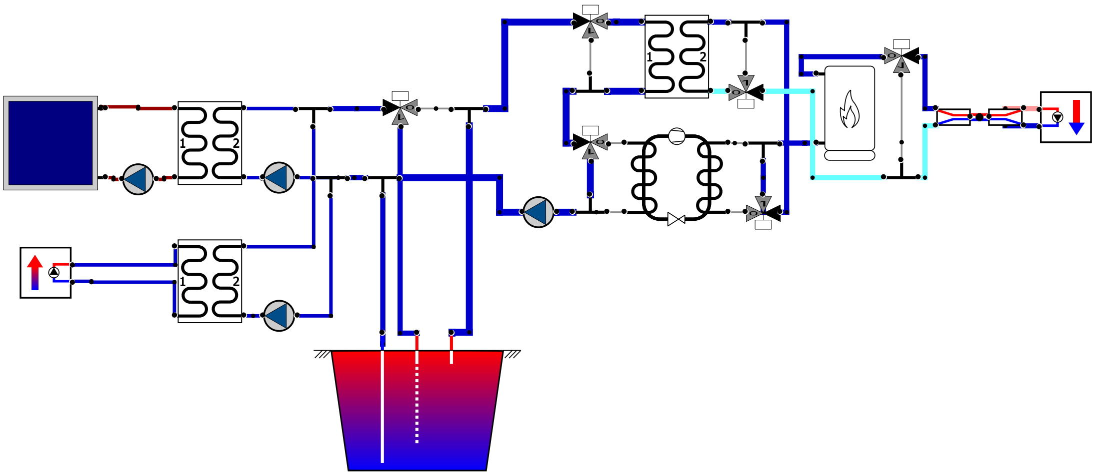


Figure 8: The solar thermal field and the additional source supply energy to meet the demand while the DHN demand is active in the PTES system. As the storage temperature is high enough for the supply, only the HX is in use and the HP is bypassed.

S3) BTES system Figure 11 shows the hydraulic arrangement of the BTES system. This system is quite different from the TTES and PTES systems. It has the following components that are not part of the previous two systems:

1) A small buffer tank. It is necessary because the power extractable from the borehole storage is limited. This tank ensures the required power is available to the sources.
2) A pump for the condenser side of the heat pump

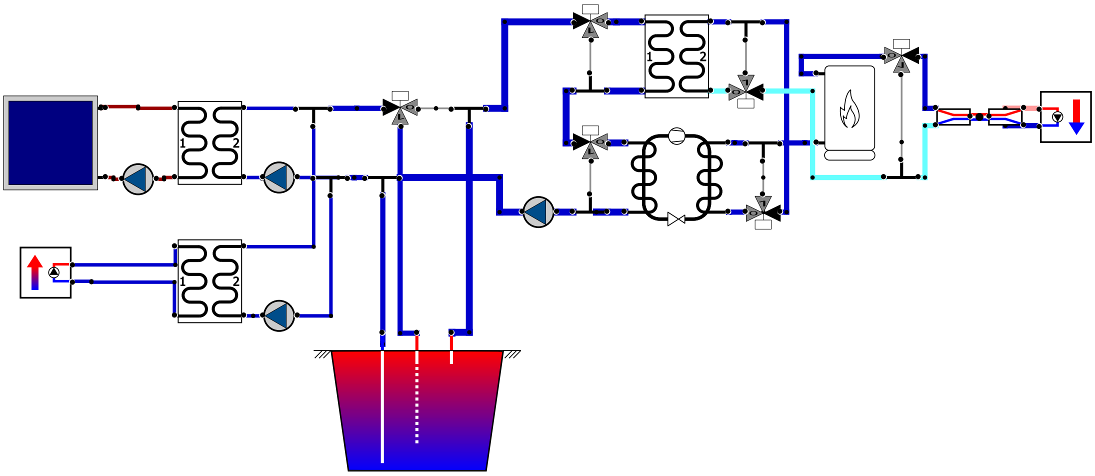


Figure 9: The solar thermal field and the additional source supply energy to meet the demand while the DHN demand is active in the PTES system. The storage temperature isn't high enough to achieve the supply set-point temperature directly, but it is used to pre-heat the return flow. The pre-heated return flow is then passed through the HP to achieve the desired set-point temperature.

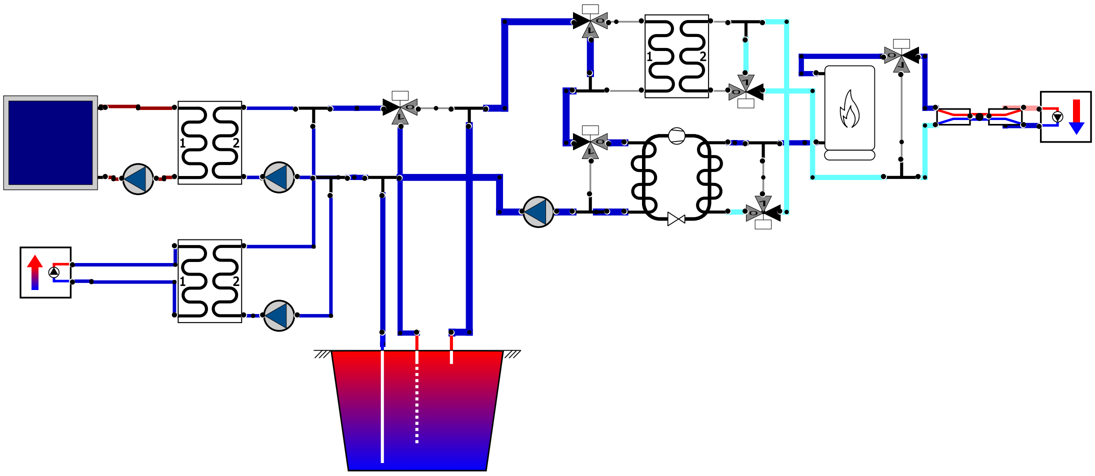


Figure 10: The solar thermal field and the additional source supply energy to meet the demand while the DHN demand is active in the PTES system. Since the storage temperatures are too low even to preheat the return flow, only the HP is in use.

3) A heat pump to lift the temperature coming from the borehole field storage

4) A valve to switch between charging the borehole storage (position 1, coming from the buffer tank) or discharging it (position 0, going into the cold side of the heat pump)

5) A mixing valve to ensure that the borehole field storage is not entered with too hot a temperature lest the borehole pipes get damaged

6) A pump to drive the mass flow rate into the borehole storage field

7) The borehole storage field

BTES system has different operational modes. Figures 12 and 13 show two. Again, the pipes' colours must be disregarded.

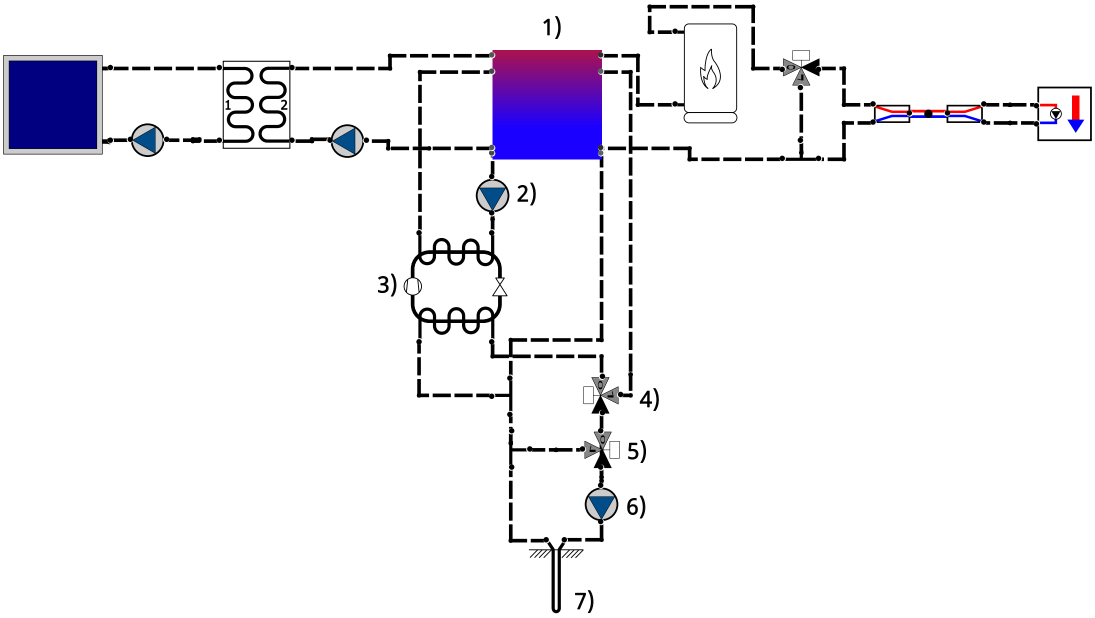


Figure 11: The hydraulic scheme for the BTES system in *pytrnsys*.

3.3.2 T2.2) System control

S1) TTES and S2) PTES The control of the TTES and PTES systems are very similar except for the control of the extra heat exchanger and the additional heat pump in the PTES system.

The system control ensures that:

1) Set point temperatures are reached (when possible).

2) The system is run safely, i.e. values are kept at all times within the safety ranges stipulated throughout the system.

The default set points and extrema assumed for the system are given in table 7.

A "match flow" control strategy for the primary loop of the solar collector is used: the mass flow rate through the collector is continuously adjusted to achieve a collector output temperature as specified by S4) in table 7, if possible. Minimum and maximum mass flow rates are respected (see E6) in table 7. If the collector output temperature rises above the maximum admissible temperature (E7) in table 7 it is assumed that the

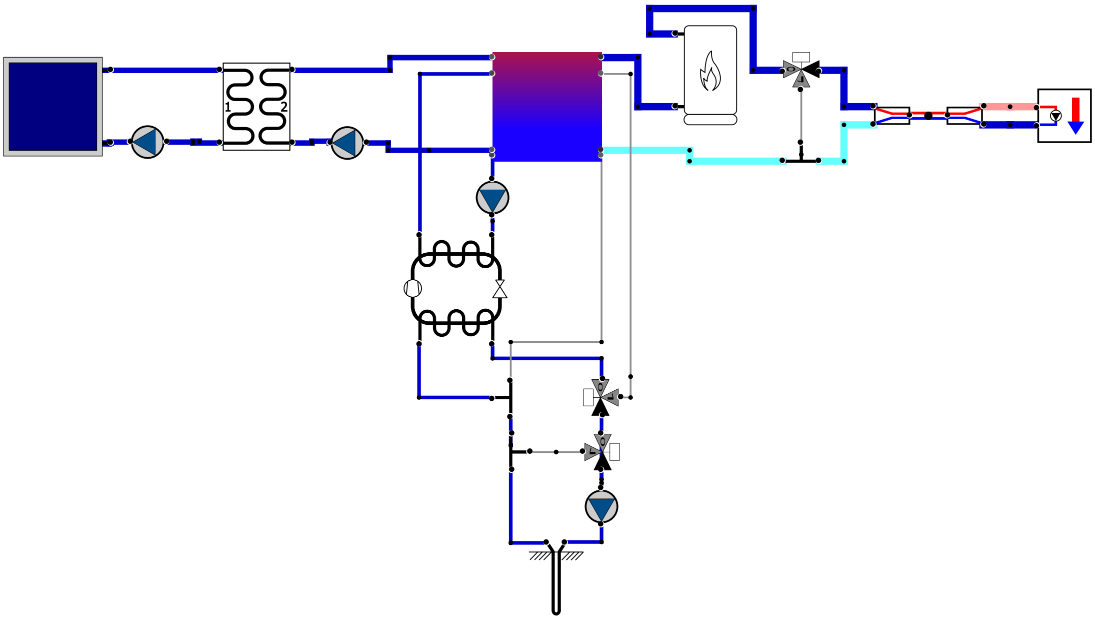


Figure 12: A mode of the BTES system. The BTES is discharged while the buffer tank is also charged via the solar collector field and discharged to the DHN demand.

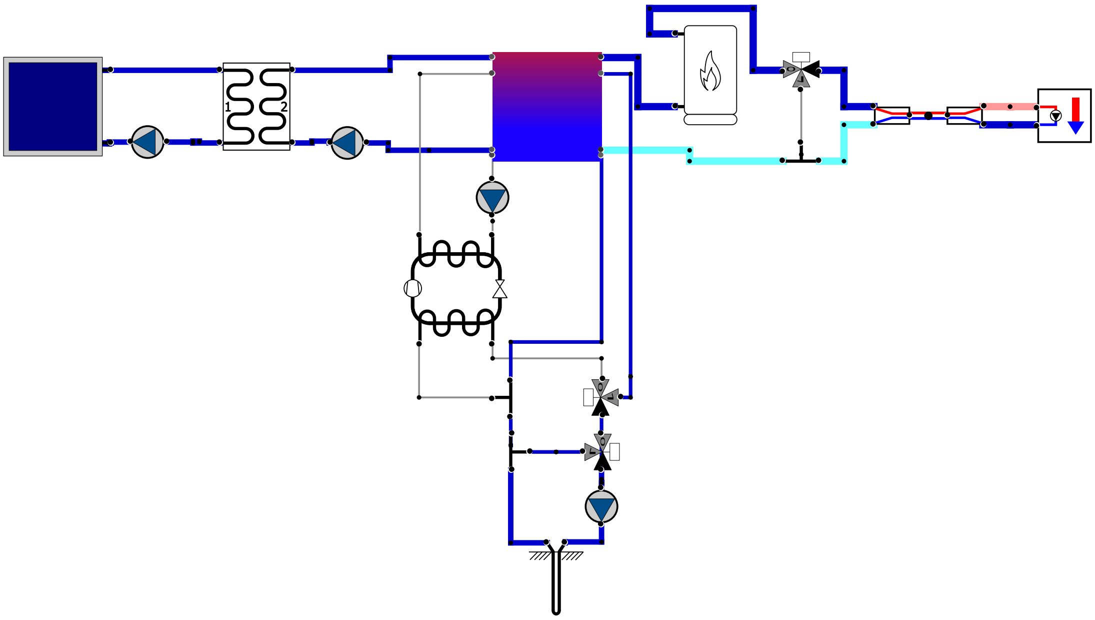


Figure 13: A mode of the BTES system. The BTES is charged while the buffer tank is also charged via the solar collector field and discharged to the DHN demand.

Requirement	**Name**	**Description**	**Value**
E1)	Minimum incident irradiation per area	The collector primary loop is only run if the irradiation is above this value	200 W/m2
S2)	Network supply set point temperature	Set point temperature of mass flow rate going into supply pipe of double pipe	75 °C
E3)	Storage top temperature	Average storage temperature from 85% to 95% of storage height	90 °C
S4)	Solar collector primary loop set point temperature	–	95 °C
E6)	Maximum collector primary loop mass flow rate	Twice the collector field nominal mass flow rate (dependent on collector field area)	–
E7)	Solar collector primary loop maximum temperature	If the maximum temperature is reached the pump at 13) in figure. 21 will be turned off until at least midnight	120 °C

Table 7: Requirements for TTES control

collector field enters saturation and the pump for the collector field (13) in figure 21 will be shut down until at least the next midnight.

The pump for the secondary loop of the collector field (14) in figure 21 is activated if there is supply from the primary loop and the storage's top temperature isn't too high. Its mass flow rate is set equal to the mass flow rate on the primary side. More specifically, it activates if E3) in table 7 is respected and the temperature going into the heat exchanger on the collector primary loop is at least 5 °C higher than both the temperature at the bottom of the storage and the network return temperature.

The position of the mixing valve for the supply at 3) in figure 21 is set in such a way that the output temperature matches the network supply set point temperature (S2) in table 7. Valves 5) and 6) in figure 21 are controlled as described in table 7.

Figures 14 and 15 show two different operation "modes"[2] of the TTES system. In these figures, the colors must be disregarded. The only information conveyed is the magnitude of the mass flow rate: the thicker a pipe is shown, the higher the mass flow through it.[3]

In all the modes shown, the additional source 12) (cf. figure 21) is active, there is mixing at 3) to cool down the supply and the by-pass at 5) is not active.

In the PTES system, the extra heat exchanger and heat pump also needs to be controlled. The valve pairs at 1) and 2) in figure 7 are controlled as follows:

1) If the network return at the right-hand-side 1) is higher than the supply at the left-hand-side 1) then the heat exchanger is by-passed and only the heat pump is active.

2) If the network return temperature can be raised to the supply temperature by going through the heat exchanger alone the heat pump is by-passed.

---

[2] We call a given set of prescribed pump mass flow rates and valve positions a "mode". A mode often corresponds to higher-level concepts such as "cooling mode" or "regeneration mode", etc.

[3] These figures are produced solely on the basis of prescribed pump mass flow rates and valve positions. The "inner life" of components is not considered at this step. I.e., the temperatures of the fluid(s) in the system are not known, only the mass flow rates. Hence, there isn't any temperature information to show. This is not the case when loading full simulation results into the *pytrnsys* GUI. Then, the pipes are colored according to the temperature of the fluid flowing through them.

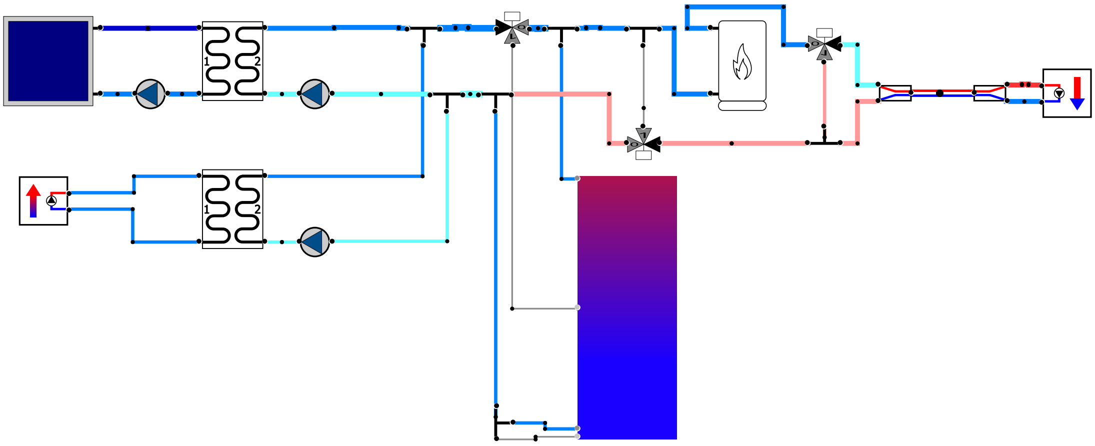


Figure 14: The solar thermal field and the additional source supply energy to meet the demand while the DHN demand is active in the TTES system. When being charged, hot fluid coming from the collector enters the TES' top port and cool fluid is pushed out through the bottom port. When the storage is discharged, the cool DHN return enters the storage at the bottom port and hot fluid is pushed out through the top port. The difference in the mass flow rates of the sources and the demand determines the direction of the flow through the storage and, by extension, whether the storage is charged or discharged.

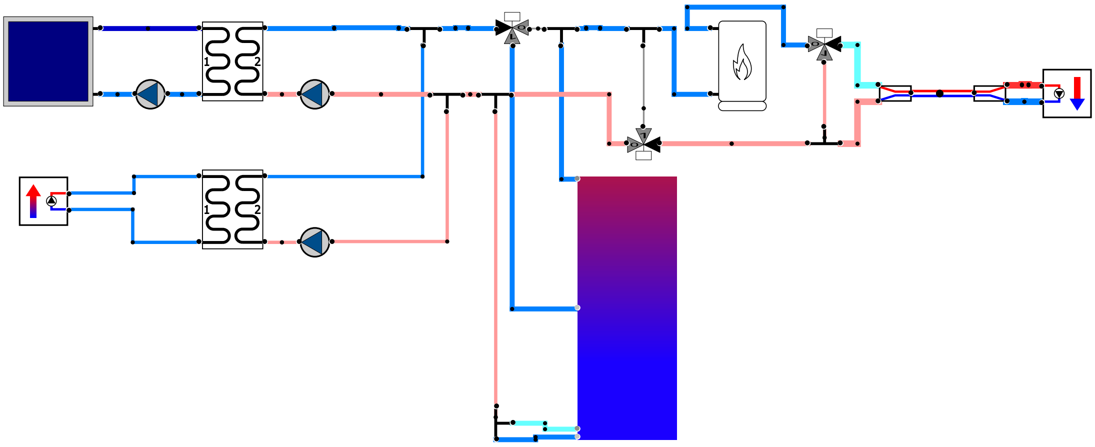


Figure 15: The solar thermal field and the additional source supply energy at the middle warm port of the TES to meet the demand while the DHN demand is active in the TTES system. The TES is charged by introducing hot fluid from the sources at the middle warm port. At the same time, the storage is discharged by withdrawing hot fluid through the top port. Depending on the relative sizes of the mass flow rates at which warm fluid is introduced and hot fluid withdrawn, cold fluid is either withdrawn to or introduced from the DHN return at the bottom port (warm mass flow rate greater or small than hot mass flow rate, respectively).

Requirement	Name	Description	Value
S8)	Borehole inlet set point temperature	Set point temperature of the fluid going into the borehole storage	75 °C
S9)	Condenser outlet set point temperature	—	75 °C

Table 8: Requirements for BTES control

3) Otherwise, both the heat exchanger and the heat pump are in use.

The heat pump is modeled as a variable drive heat pump: its output power is varied according to the demand, i.e. the mass flow rate and input temperatures. The minimum power at which the heat pump is run is 20 % of its nominal power. When the heat pump is required to provide power below that minimum it is shut down instead.

S3) BTES The control of the system that includes the BTES differs from the strategy used in the other two systems. Due to the nature of the storage — it has a lower power capacity, especially when discharging —, an extra buffer tank is needed. The heat source (the solar collector field) and the demand (the district heating network) are connected to this smaller storage and not directly to the seasonal storage. The BTES is also charged and discharged from the buffer tank. The heat pump increases the temperature of the fluid coming from the BTES when it is being discharged.

The same control for the solar collector field and for achieving the supply temperature (boiler and mixing valve) as in the other two systems is used in the BTES system.

The values presented in table 7 also apply to the BTES system. In addition, the setpoints outlined in table 8 also apply.

The main difference in the control of the BTES system compared to the TTES and PTES system is the control to do with directing the energy flows between the buffer tank and the borehole field. When to charge or discharge the borehole field is decided based on the actual temperature of the buffer tank and its setpoint.

When the temperature in the intermediate part of the tank is higher than its setpoint value, the pump in the inlet of the boreholes is turned on and the seasonal storage is charged (as seen in figure 13). The mixing valve at 5) in figure 11 is controlled to avoid temperatures above 75 °C in the inlet of the boreholes.

Conversely, when the upper temperature of the tank is lower than the design setpoint, energy is extracted from the boreholes. Under this operating mode, the heat pump is used to increase the temperature of the water coming from the boreholes (as seen in figure 12). The capacity of the heat pump is controlled (i.e., it is modeled as a variable-drive heat pump) to reach the desired temperature in the buffer tank.

The activation of the charge and discharge modes is carried out by making use of separate dead-bands in order to avoid undesirable on/off behaviors in the system.

3.4 WP3) TES cost model development

For the development of the TES cost model, preliminary cost correlations for TTES, PTES, and BTES systems were derived based on data from the latest IEA DHC Annex XII Project 03 report IEA DHC Annex XII Project 03 (2020). This initial model is currently undergoing validation to incorporate the most recent investment cost data provided by our collaborators.

The IEA DHC Annex XII report integrates investment costs for various thermal energy storage (TES) technologies, including ATES. The source data (see figure 16) represent the specific investment cost per cubic meter of water equivalent as a function of the total storage volume in water equivalent. According to the report caption, these costs include all expenditures associated with the construction of the storage unit itself, but exclude design, interconnection piping, heating plant equipment, and value-added tax (VAT).

From these data, the following methodological steps were carried out to derive cost functions for each TES technology:

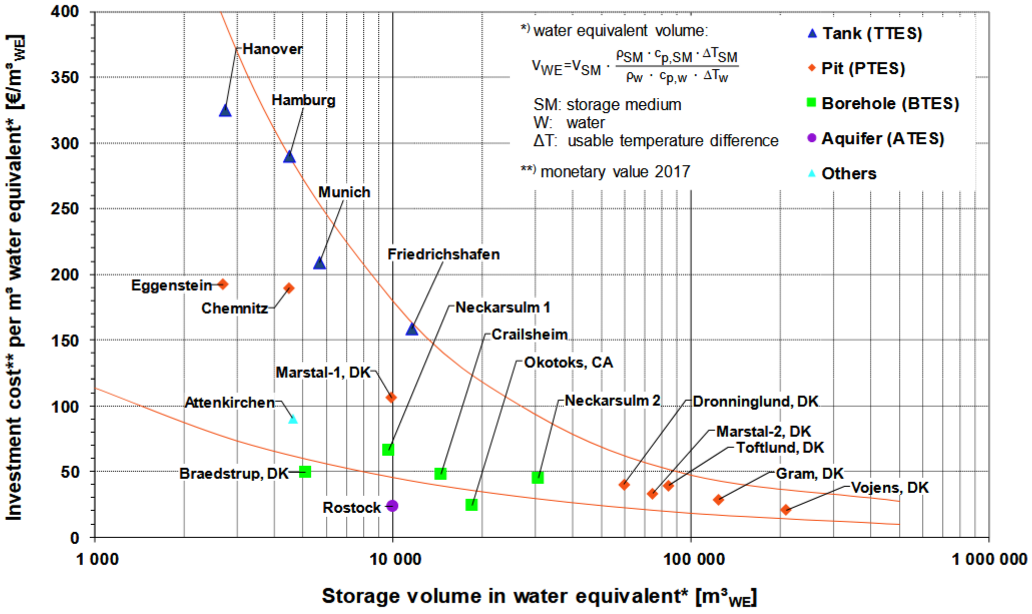


Figure 16: Investment costs for various thermal energy storage (TES) technologies from IEA DHC Annex XII Project 03 (2020).

1) **Data extraction:** Numerical values were extracted from the referenced graph, yielding the raw dataset (see columns in Tab. 3.4).

2) **Data organisation and cost updating:** The extracted data were categorized by TES technology and updated using the Harmonised Index of Consumer Prices (HICP) published by the European Central Bank European Central Bank (2025), corresponding to a yearly mean inflation factor of 2.85 % for the period 2017–2025.

3) **Function fitting:** Several functional forms were tested, with the most suitable correlation identified as a power-law (potential) relationship:

$$y = a\,x^b$$

   This formulation achieved coefficients of determination (R^2) greater than 0.90 for all technologies except BTES, where higher data dispersion was observed.

The resulting derived cost function for TTES in €/$\mathrm{m}^3_{\mathrm{WE}}$, currently under validation, is formulated as:

$$y = 27102\,x^{-0.527}, \quad R^2 = 0.914$$

The derived cost function for PTES in €/$\mathrm{m}^3_{\mathrm{WE}}$ (also under validation), reads as:

$$y = 19142\,x^{-0.539}, \quad R^2 = 0.9678$$

The derived cost function for BTES in €/$\mathrm{m}^3_{\mathrm{WE}}$ (also under validation), is expressed as:

$$y = 472.54\,x^{-0.225}, \quad R^2 = 0.1844$$

This relatively low coefficient of determination indicates a high dispersion of the available cost data for BTES systems, suggesting greater variability in reported investment costs than for other TES technologies.

Technology	**Storage Volume** (m^3_{WE})	**Investment Cost (2017)** (€/m^3_{WE})	**Investment Cost (2025)** (€/m^3_{WE})
TTES	2746	322.8	404.9
	4506	287.7	360.8
	5684	206.9	259.6
	11628	156.5	196.3
PTES	2704	192.6	241.6
	4489	189.7	238.0
	9884	106.0	133.0
	59757	39.6	49.7
	74216	32.9	41.2
	83999	38.9	48.8
	123687	28.3	35.5
	205782	20.0	25.1
BTES	5080	49.8	62.4
	9657	66.4	83.3
	14442	48.1	60.3
	18358	24.4	30.6
	30595	45.1	56.5

Table 9: Investment cost data for large-scale TES technologies (2017–2025)

3.5 WP4) RESULTES tool

3.5.1 Progress overview

The “SPF server infrastructure” for all three systems has been completed. This completes milestones MS2’, MS5’ and MS6’ of task T4.2)). Prototypes for the user interfaces for the TTES and PTES systems are online and available at https://dev.resultes.net (MS2’ and MS5’: T4.1)). To create a user account the interested reader can write to *damian.birchler@ost.ch* to receive a registration code. The user interface prototypes for all systems need to be fully fleshed out in coordination with the project partners in the remainder of the project. An example of an energy balance for a simulation run shown in the RESULTES GUI is shown in figure 17.

In the remainder, the software architecture and development process of the RESULTES tool will be described. The emphasis is mostly on the “SPF server infrastructure” instead of on the web user interfaces as the latter will be changed and extended substantially in the remainder of the project.

The remaining subsections in WP4)’s section are mostly taken from a conference paper submitted to the SolarPaces congress which took place in autumn 2025.

3.5.2 *pytrnsys*

RESULTES uses the open-source, Python-based *pytrnsys* software for modeling and simulating thermal energy systems. *pytrnsys* is a wrapper around the well-known TRNSYS modeling software. *pytrnsys* allows one to define the hydraulics of a system graphically and makes it easier to re-use components and find errors in a simulation. It also gives the user a way to semi-automatically generate plots and reports — for individual simulations but also across multiple simulations — based on the simulation results. Finally, it can perform parametric sweeps over the parameter space and run the resulting parameterized simulations in parallel.

When running a parametric sweep, *pytrnsys* creates a “deck file” for each individual case to be run. “Deck file” is the name for the file format in which TRNSYS simulations are defined, i.e. they are the

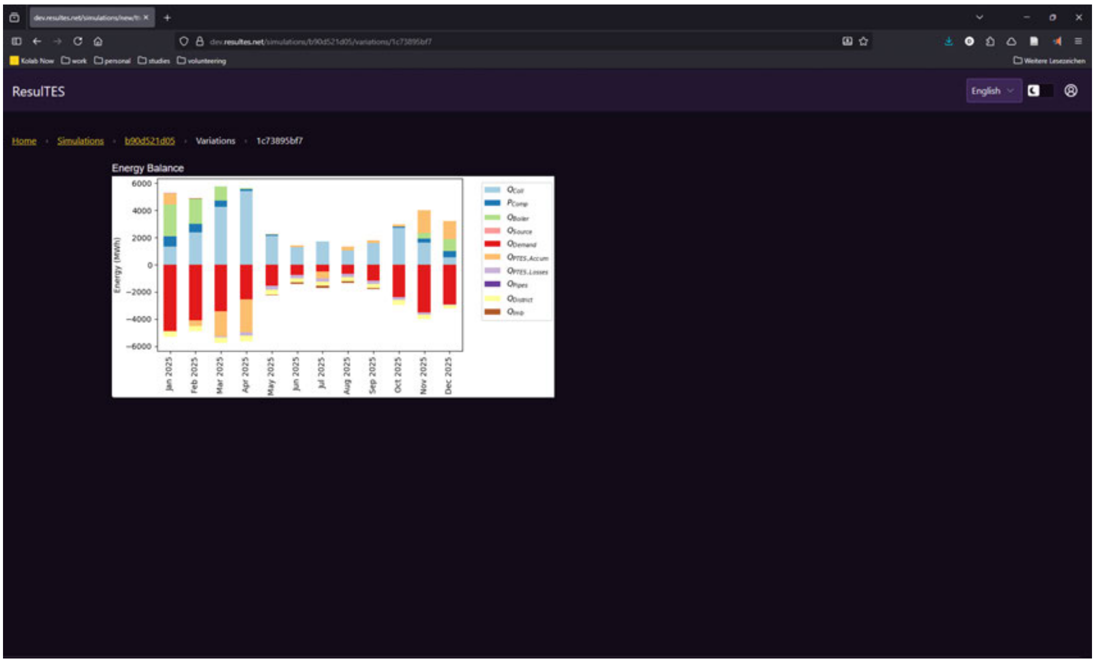


Figure 17: The life cycle of a simulation request.

input files to TRNSYS. However, because *pytrnsys* is a single-user application and designed to be run on a single machine, *pytrnsys*' existing functionality for running cases in parallel cannot be used within RESULTES. Instead, RESULTES implements its own way of running simulations of multiple users in parallel.

3.5.3 Software architecture

For design phases other than the detailed design phase, RESULTES is a web-based application: it is accessed by the user via the Web using a browser. With their web-browser, the user can, once logged in, define and start new simulations or look at the results of previously run simulations.

RESULTES is composed of various components (set in cursive further down). The component directly exposed to the user is the *client server* (or simply *client*). It serves RESULTES' website. The website is not static but runs JavaScript code inside the user's browser. This type of website is often more accurately called a web application. The data comprising the *web application* consists of HTML, CSS, images and the JavaScript code to run within the user's browser, on their computer. The *client* might send the *web application*'s data to the user in multiple pieces as the user navigates the website.

Once loaded into the user's browser, the *web application* itself does not communicate with the *client* but with another RESULTES component: the *external server*. The *external server* exposes a REST API to the Web. The *web application* can, e.g., start a new simulation via this API when the user hits the appropriate button. The *external server* is also responsible for authenticating users to make sure only authorized user can start simulations or access simulation results.

The *external server* can be thought of as a wrapper around RESULTES' relational *data base*. It "translates" from RESULTES' domain model (simulation parameters, parametric simulation runs, results, etc.) to the generic relational database model of data tables. However, it is the *data base* which ultimately holds, and is responsible for the persistence and consistency of, the data. Of the components described so far, the *data base* is the only one which has state. The other ones could crash or be restarted without resulting in a loss of data[4].

---

[4] Bar the partial results of running simulations.

The *client*, *web application*, *external server* and *data base* component are sufficient for a user to be able to submit a simulation for running, for instance: when they click the "simulate" button in their web browser a request to start a simulation together with the simulation parameters is sent to the *external server*. The *external server*

1) validates the request data
2) authorizes the user
3) translates the request into the required rows in the needed tables which encode the request
4) asks the *data base* to insert the rows and save the tables

At this point, however, the request simply sits there in the *data base* waiting to be picked up. Further components are needed to run the simulations.

Like the *external server*, the *internal server* exposes the *data base*'s contents via a REST Web API. Unlike the *external server*, however, the *internal server* does not listen to requests from the public Web, but only from requests originating from RESULTES' private network. Authentication is not required to access the *internal server*'s API and the API can be used to retrieve information across all users (e.g. which simulation out of all simulations, from all users, waiting to run to run next) or to update the state of RESULTES' business objects (e.g. set the state of a parametric run from "waiting" to "running").

The *internal server* is used by the *scheduler*, one of RESULTES' most central components. The *scheduler* schedules the creation of individual simulation runs for parametric studies as well as the running and post-processing of individual runs. It is also concerned with managing the *runners* on which the simulations will actually be run. If all the *runners* are currently busy running simulations, the *scheduler* will start a new runner and, once it is ready, submit the simulation to run to it. Conversely, it will shut down idle *runners*. The *runners* are the last remaining component to discuss. It is them that do the having lifting of actually running the simulations using TRNSYS. The *runners* expose a JSON-RPC Web API to RESULTES' internal network, which is used by the *scheduler* to communicate with them.

For larger data items, OpenStack's object store is used for communicating between the *scheduler* and the *runners*. More on that further below.

In the following, we'll encounter terms like public cloud, OpenStack and Kubernetes. We'll give a brief overview of these terms first.

Public Cloud

Also commonly referred to as just "cloud", a public cloud is a virtual datacenter which offers the paying public the possibility to rent computers (CPU and RAM memory), disk (and object) storage and network bandwidth and possibly other resources on demand, in an automated way, via the Internet. The billing of the resources used is quite granular, often on a per-minute or even finer basis.

OpenStack

OpenStack is a set of services which can be used to implement the software side of a public cloud (physical machines, storage, etc., are of course also required). Unlike many competing products, like those running Microsoft's Azure or Amazon Web Service, OpenStack is open-source and used by many small to medium-sized cloud providers. With OpenStack, users can provision virtual machines running a desired operating system image and attach disk storage and (virtual) network interfaces to them, if needed.

Docker

Docker is a containerization software. It can create so-called software "container images" and run them as containers. A software container can be thought of as an isolated "process" (the comparison is not entirely accurate as multiple processes can run within one container). Two containers running next to each other on the same machine will not know of each other's existence. To an application running in one container, it will look as if it's the only application running on a given machine: the file system, network stack, process tree, etc., of the two containers are completely separated from each other. One advantage of containerized applications is that they bring their own software requirements without any danger of them interfering with one another.

Kubernetes

While the level of abstraction of OpenStack is virtual machines, hard disks, and network interfaces, Kubernetes is concerned with more high-level “software services” defined by Docker containers. In fact, Kubernetes is referred to as a “container orchestration system”. Many of the RESULTES’ components are good examples of such services. Take the *external server* component. For RESULTES to function properly, the component needs to have its required software installed (Python and the Python libraries needed serve the API, connect to the *data base*, . . .), be able to access a network interface exposed to the public Web and know at what IP address to access the *data base*. It must also not be starved off CPU power and have enough memory. Other details, however, like what machine it runs on and which other processes are running on that machine are not important to the functioning of the component. Because Kubernetes is concerned with software services and not plain machines, it also offers functionality like automatically scaling out services (i.e. starting another copy of a service) if a given instance of a service risks being overwhelmed or restarting a service if it has crashed, among other things.

Block and Object Storage

In cloud computing two types of persistent storage are often encountered: block storage and object storage. Block storage can be thought of as hard disks akin to the ones in personal computers. Block storages are almost always formatted with a file system and mounted into a virtual machine, and can then be accessed like an internal hard disk. Object storage is quite different. It does not have a file system. Instead, it is more like a key-value store. Data can be stored under a given key. E.g., a profile picture could be stored under /users/damian/profile. The data under a key cannot be edited like a traditional file, but only downloaded or uploaded as a whole. I.e., to modify an entry, the new version must be re-uploaded. Typically, data is accessed via the Hypertext Transfer Protocol (HTTP), the same protocol that is used to access web sites. Object storage is often slower than block storage and does not allow “random access” (i.e. only the whole object can be accessed and not individual bytes). It is, however, often significantly cheaper to rent than block storage. Therefore, it is the recommended option to store large quantities of date over long periods of time.

Except for the *web application* and *runners*, all the components run as dockerized Linux services within a public Kubernetes-as-a-service cloud. The *web application* runs in the user’s browser — as previously described and the *runners* cannot run under Linux as they must be able to run TRNSYS, which is only available on Windows. Because the public cloud which the RESULTES project is using does not support Windows nodes (“nodes” are the machines running dockerized applications, so-called “pods” in Kubernetes) the *runners* cannot be run as Kubernetes pods. Instead, each runner is a fully fledged OpenStack Windows virtual machine which is created and destroyed as needed by the *scheduler*.

3.5.4 Development process

RESULTES’ code is stored on the GitHub code hosting service and can be found at https://github.com/orgs/resultes-net/repositories. It should be noted, however, that most repositories are currently private, because they hold credentials which should not be shared with the public. Before completion of the project, those repositories’ code will be refactored such that the secrets are not kept within the repositories and the repositories made public. The source code is tracked using the git version management software.

Each RESULTES component has its own GitHub repository (except for the *client* and *web application* which share one). If a change to a component’s source code is submitted, a GitHub workflow is started that creates a Docker container containing the newest version of the code plus its required software dependencies and stores it in the container registry offered by GitHub, the GitHub container registry. The registry is monitored by a Keel service instance running in the RESULTES Kubernetes cluster. Keel is an open-source third-party software which can be installed into a Kubernetes cluster to ensure that all pods always run the latest version of their software containers. The Keel instance periodically checks the container registry for newer versions and updates the running Kubernetes pods if a new version is found. A route from the public Internet into the RESULTES cluster has been setup: the *client* can be reached at the URI https://dev.resultes.net while the *external server* is reachable under https://dev.resultes.net/api . The *dev* prefix stands for development. The production-ready version will be reachable at https://www.resultes.net once it’s available.

The runner component is an exception to the above because it does not run as a dockerized application. Submitting a change to the runner's source code will result in the creation of a new Extendable Virtual Hard Disk image (.vhdx). This is a special file format by which the content of virtual hard disk for use by common virtualization software can be specified. The newly created image will then be uploaded into RESULTES' public cloud provider's OpenStack image registry. An OpenStack image registry stores virtual hard disk images (in contrast to a container registry which stores container images). When a new runner virtual machine instance is started, the *scheduler* attaches a virtual hard disk containing the disk image's content to the machine. As part of its start-up routine, the runner runs a start-up script from its attached disk, which in turn starts up the runner's software. This way, the runner virtual machine image can remain unchanged when changes to the runner's source code are made, and only the disk image needs to be updated. Creating a full machine image would take half an hour and result in a large image which is more expensive to store whereas creating the disk image takes a few minutes and the image is significantly smaller. The *web application* is written in the TypeScript programming language using the Svelte web framework and runs on NodeJS. The *data base* is a PostgreSQL instance. All the other RESULTES components are written in Python. REST APIs are implemented using the Python FastAPI library. The Pydantic Python library is used for communicating structured data (in the JSON format) between components.

3.5.5 Life Cycle of a Request

Figure 18 shows the life cycle of a simulation request. The *runners* are shown as dashed boxes in contrast to the other components to highlight the fact that they are ephemeral and are created and destroyed as needed. All other components are long-running.

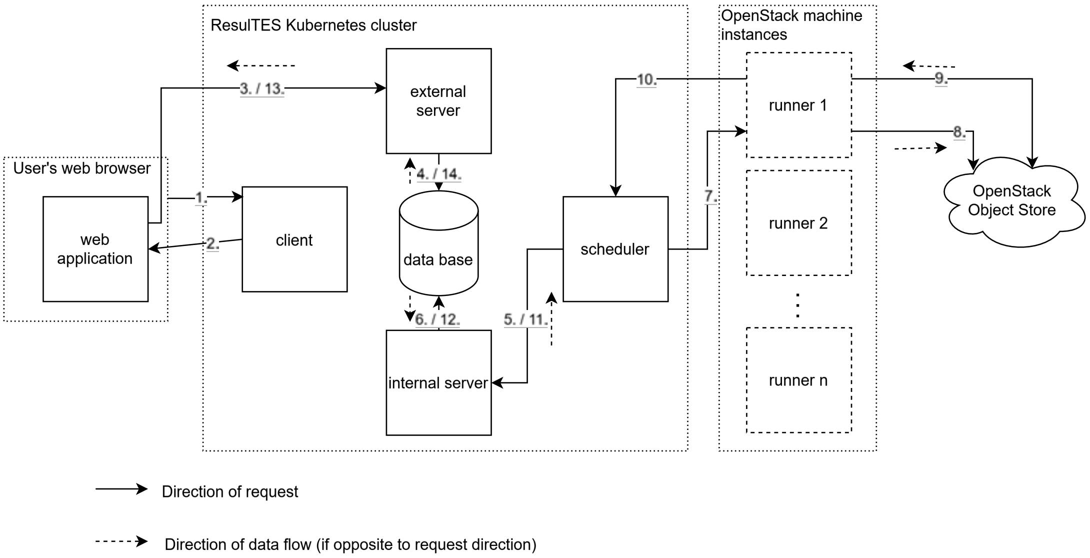


Figure 18: The life cycle of a simulation request.

The life cycle of a simulation request is as follows:

1) The user types RESULTES' URI into their web browser address bar and hits enter. The browser then queries the *client* for the *web application*'s data.

2) The data is sent to the browser and the *web application* loaded into browser.

3) The user hits the "simulate" button in the browser causing the *web application* to send the request to the *external server*.

4) The *external server* translates the requests into appropriate data base commands that it sends to the *data base*.

5) The *scheduler* periodically queries the *internal server* for simulation requests.

6) The *internal server* translates the query into the respective data base queries and queries the *data base*. It translates the *data base*'s response and sends the reply to the *scheduler*.

7) The *scheduler* considers the request for a simulation. It then decides which runner should run the given simulation. If all the available *runners* are busy — or if there isn't any runner running at the moment — it requests the creation of a new runner via the OpenStack API. If there are multiple pending requests for simulations, the *scheduler* also decides which of the possible simulations should be run next.

8) The runner downloads the *pytrnsys* project from the OpenStack object store and starts running the simulation with the parameters provided to it by the *scheduler* via the JSON-RPC Web API.

9) Once the simulation has finished, the runner uploads the results (raw data [hourly and monthly values] and post-processing results [JSON files and graphical plots]) to the object store.

10) The runner notifies the *scheduler* that it has completed the simulation run.

11) The *scheduler* updates the status of the simulation in the *data base* via the *internal server*.

12) The *internal server* issues the data base commands to update the simulation status to the *data base*.

13) The *web application* periodically checks with the *external server* for changes in the simulation runs' statuses to inform the user about completed simulations.

14) The *external server* translates the queries for simulation run status into data base commands and sends them to the *data base*.

3.5.6 Generation of Deck Files for Parametric Sweeps

The generation of deck files for a given set of parameter combinations is a job that is scheduled by the *scheduler*, much like it schedules jobs for running a given parametric run. A job to create deck files results in various new jobs, one for each deck file. This is illustrated in figure 19.

The steps shown in figure 18 are as follows:

1) The *scheduler* instructs the runner to generate the deck files for a given parametric simulation request.

2) The runner downloads the *pytrnsys* project template from the object store and generates the deck files for the given set of parameter combinations.

3) It then uploads the *pytrnsys* project now containing the deck files to the object store.

4) The runner informs the *scheduler* that the deck files have been generated.

5) The *scheduler* creates a "run simulation" request for each generated deck file through the *internal server*.

6) The *internal server* translates the requests to data base commands which it sends to the *data base*. The *data base* persists the requests to be picked up by the *scheduler* at some point

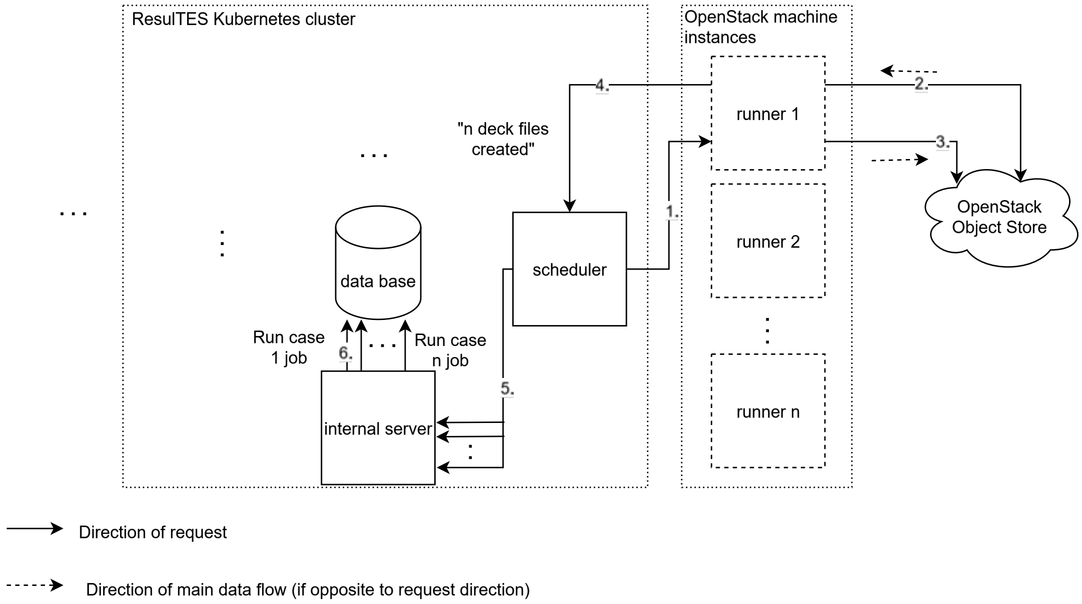


Figure 19: The workflow for generating the different *deck* files for a parametric study.

3.6 WP5) Validation

To validate the simulation models developed within the RESULTES project, simulations' results will be compared to the results of Solites' state-of-the-art models of similar systems.

Validation work has been started for the PTES system. SPF and Solites are drafting a set of simulation parameters to use in their respective simulations to be able to compare the results. Table 10 lists the preliminary set of agreed-upon parameters for the PTES validation. These are based on the TES-only reference cases elaborated in the IEA-ES Task 39, Task 45's predecessor.

Name	**Value**	**Unit**	**Remark**
Simulation parameters			
Start time	01.01.2025	—	
End time	31.12.2028	—	
Evaluation year	2028	—	Year whose simulation results to calculate KPIs on
Solar collector			
TRNSYS type	1375	—	
Area	30 000	m^2	
Slope	45	°	
Location	Zurich, Switzerland	—	
Weather data set	Meteonorm, *CH-Zuerich-Klotten-66700.tm2*	—	
Model	TVP MT-Power_v4	—	

c_0	0.737	—	
c_1	0.504	$\mathrm{W\,m^{-2}\,K^{-1}}$	
c_2	0.006	$\mathrm{W\,m^{-2}\,K^{-2}}$	
c_3	0	$\mathrm{J\,m^{-3}\,K^{-1}}$	
c_4	0	—	
c_5	15.32	$\mathrm{kJ\,m^{-2}\,K^{-1}}$	
c_6	0	$\mathrm{s\,m^{-1}}$	
K_d	0.957	—	
PTES			
Volume	100 000	$\mathrm{m^3}$	
Slope at sides	27	°	
Heigth	16	m	
Number of nodes	50	—	
Top insulation U-value	0.1	$\mathrm{W\,m^{-2}\,K^{-1}}$	
Horizontal extension	5	m	Insulation's horizontal extension beyond the edge of the storage volume
Heat transfer coefficient to soil	90	$\mathrm{W\,m^{-2}\,K^{-1}}$	At bottom and sides, irrespective of height
Maximum allowed temperature	85	°C	
Initial temperature	10	°C	Storage content (water) and soil
Soil thermal gradient	0	$\mathrm{K\,m^{-1}}$	Thermal gradient is ignored
Soil thermal conductivity	1.8	$\mathrm{W\,m^{-1}\,K^{-1}}$	
Soil thermal capacity (volumetric)	2.8	$\mathrm{MJ\,m^{-3}\,K^{-1}}$	
Heat pump			
TRNSYS type	977	—	Developed by SPF
Nominal capacity	3	MW	
Nominal COP	4.68	—	
Refrigerant	RA410A	—	
Demand			
Yearly demand	30	GW h	
Demand profile	See here.	—	
Supply temperature	80	°C	Into network piping
Return temperature	50	°C	From demand
Heat exchanger			
In discussion			

Table 10: Parameters used in validation of PTES system

3.7 WP6) Dissemination

A paper outlining RESULTES has been submitted and a poster was presented at the SolarPACES congress in Spain in autumn 2025. A poster will also be presented at the Solar World Congress in Brazil in November 2025. See apendix for the posters. Furthermore, ResulTES was presented in February 2025 at the IEA-ES Task 45 experts meeting in February in Stuttgart, with a follow-up presentation on-line at the task meeting in October the same year. Finally, RESULTES will be presented at the Wärmepumpensymposium at the Eastern Switzerland University of Applied Sciences OST in the beginning of November 2025.

4 National/International cooperation

This project involves two partners, SPF and DCarbo Energy Consulting (Spain), who are developing the tool, and an external expert in designing, planning, and implementing large-scale thermal storage systems: Solites, based in Germany. Solites ensures the alignment of the RESULTES tool with planners' requirements and contributes expertise to enhance the tool's applicability and effectiveness.

Additionally, we have proposed our participation in **Task 45** (`http://iea-es.org/task-45/`) of the IEA's Energy Storage Technology Collaboration Programme, which focuses on advancing large thermal energy storage (LTES) systems for district heating and industrial applications. Task 45 aims to overcome implementation barriers, standardize performance assessments, and improve tools for feasibility and cost analysis. The RESULTES tool aligns with Task 45's goals by offering system-level modelling capabilities to assist stakeholders in designing LTES integration into district heating networks. This collaboration would provide the RESULTES project with access to valuable TES monitoring data, supporting the validation of RESULTES models and enhancing their alignment with stakeholder needs.

5 Evaluation of 2025 and outlook of 2026

In the project's second (this time full) year the simulation models for the remaining two systems (PTES and BTES) have been implemented. The SPF server infrastructure for running simulations on-demand in the cloud has been deployed and can run simulations. Also completed and deployed were prototype versions of the web user interfaces for the TTES and PTES systems. First models for the cost calculation has been developed and validation activity has started. Outreach activities have been ongoing with two conferences, one symposium and two IAE-ES task experts' meetings attended.

DCarbo Energy Consulting and Solites With some interruptions, weekly meetings where held with DCarbo Energy Consulting and biweekly meetings with Solites throughout the year.

5.1 Outlook 2026

The next step for the project until the project end in June 2026 will be:

1) Complete validation of system simulation models
2) Complete cost calculation functionality for all three systems
3) Bring prototype web user interfaces to production level
4) Write and submit the final report

References

Chang, C., Nie, B., Leng, G., Li, C., She, X., Peng, X., & Deng, J. (2017, December). Influences of the key characteristic parameters on the thermal performance of a water pit seasonal thermal storage. *Energy Procedia*, *142*, 495–500. Retrieved 2023-07-06, from `https://linkinghub.elsevier.com/retrieve/pii/S1876610217358058` doi: 10.1016/j.egypro.2017.12.077

Chang, C., Wu, Z., Navarro, H., Li, C., Leng, G., Li, X., . . . Ding, Y. (2017, December). Comparative study of the transient natural convection in an underground water pit thermal storage. *Applied Energy*, *208*, 1162–1173. Retrieved 2023-07-06, from `https://linkinghub.elsevier.com/retrieve/pii/S0306261917313089` doi: 10.1016/j.apenergy.2017.09.036

Dahash, A., Ochs, F., Janetti, M. B., & Streicher, W. (2019, April). Advances in seasonal thermal energy storage for solar district heating applications: A critical review on large-scale hot-water tank and pit thermal energy storage systems. *Applied Energy*, *239*, 296–315. Retrieved 2023-07-06, from `https://linkinghub.elsevier.com/retrieve/pii/S0306261919301837` doi: 10.1016/j.apenergy.2019.01.189

Dahash, A., Ochs, F., Tosatto, A., & Streicher, W. (2020, December). Toward efficient numerical modeling and analysis of large-scale thermal energy storage for renewable district heating. *Applied Energy*, *279*, 115840. Retrieved 2023-07-06, from `https://linkinghub.elsevier.com/retrieve/pii/S0306261920313179` doi: 10.1016/j.apenergy.2020.115840

Dahash, A., Ochs, F., & Tosatto, A. (2020, September). Simulation-based design optimization of large-scale seasonal thermal energy storage in re-newable-based district heating systems. Retrieved 2023-07-06, from `https://diglib.tugraz.at/download.php?id=60585e6f774d7&location=datacite` (Publisher: Verlag der Technischen Universität Graz) doi: 10.3217/978-3-85125-786-1-12

European Central Bank. (2025). *Harmonised index of consumer prices (hicp).* `https://www.ecb.europa.eu/stats/macroeconomic_and_sectoral/hicp/html/index.en.html`. (Accessed: 2025-10-29)

Gauthier, G. (2020, November). Benchmarking, and improving models of subsurface heat storage dynamics.

IEA DHC Annex XII Project 03. (2020). *Report task a: Cost functions for thermal energy storage technologies* (Tech. Rep.). International Energy Agency (IEA) District Heating and Cooling Programme. Retrieved from `https://www.iea-dhc.org/fileadmin/documents/Annex_XII/2020.03.09_Report_Task_A_IEA_DHC_Annex_XII_Project_03.pdf` (Accessed: 2025-10-29)

Jensen, M. V. (2015, June). *IEA SHC Task45. Seasonal Thermal Energy Storages. Report on state of the art and further R+D.*

McDaniel, B., & Kosanovic, D. (2016). Modeling of combined heat and power plant performance with seasonal thermal energy storage. *Journal of Energy Storage*, *7*(Supplement C), 13 – 23. Retrieved from `http://www.sciencedirect.com/science/article/pii/S2352152X1630055X` doi: https://doi.org/10.1016/j.est.2016.04.006

Ochs, F., Dahash, A., Tosatto, A., & Bianchi Janetti, M. (2020, May). Techno-economic planning and construction of cost-effective large-scale hot water thermal energy storage for Renewable District heating systems. *Renewable Energy*, *150*, 1165–1177. Retrieved 2023-07-07, from `https://linkinghub.elsevier.com/retrieve/pii/S0960148119316982` doi: 10.1016/j.renene.2019.11.017

Ochs, F., Dahash, A., Tosatto, A., Reisenbichler, M., O'Donovan, K., Gauthier, G., . . . Schmidt, T. (2022). Comprehensive Comparison of Different Models for Large-Scale Thermal Energy Storage:. Global Online Event, Germany. Retrieved 2023-07-07, from `https://www.atlantis-press.com/article/125971003` doi: 10.2991/ahe.k.220301.005

Pan, X., Xiang, Y., Gao, M., Fan, J., Furbo, S., Wang, D., & Xu, C. (2022, August). Long-term thermal performance analysis of a large-scale water pit thermal energy storage. *Journal of Energy Storage*, *52*, 105001. Retrieved 2023-07-07, from `https://linkinghub.elsevier.com/retrieve/pii/S2352152X22010064` doi: 10.1016/j.est.2022.105001

Panthalookaran, V., El-Amin, M. F., Heidemann, W., & Müller-Steinhagen, H. (2008, June). Calibrated models for simulation of stratified hot water heat stores. *International Journal of Energy Research*, *32*(7), 661–676. Retrieved 2023-07-07, from `https://onlinelibrary.wiley.com/doi/10.1002/er.1423` doi: 10.1002/er.1423

Panthalookaran, V., Heidemann, W., & Müller-Steinhagen, H. (2011, November). The Effects of Boundary

Design on the Efficiency of Large-Scale Hot Water Heat Stores. *Journal of Solar Energy Engineering*, *133*(4), 041007. Retrieved 2023-07-07, from `https://asmedigitalcollection.asme.org/solarenergyengineering/article/doi/10.1115/1.4004472/469318/The-Effects-of-Boundary-Design-on-the-Efficiency` doi: 10.1115/1.4004472

Raab, S., Mangold, D., & Müller-Steinhagen, H. (2005, November). Validation of a computer model for solar assisted district heating systems with seasonal hot water heat store. *Solar Energy*, *79*(5), 531–543. Retrieved 2023-07-07, from `https://linkinghub.elsevier.com/retrieve/pii/S0038092X05000289` doi: 10.1016/j.solener.2004.10.014

Schmidt, T., Pauschinger, T., Sørensen, P. A., Snijders, A., Djebbar, R., Boulter, R., & Thornton, J. (2018, September). Design Aspects for Large-scale Pit and Aquifer Thermal Energy Storage for District Heating and Cooling. *Energy Procedia*, *149*, 585–594. Retrieved 2023-07-07, from `https://linkinghub.elsevier.com/retrieve/pii/S1876610218305198` doi: 10.1016/j.egypro.2018.08.223

Schmidt, T., & Sørensen, P. A. (n.d.). Monitoring Results from Large Scale Heat storages for District Heating in Denmark.

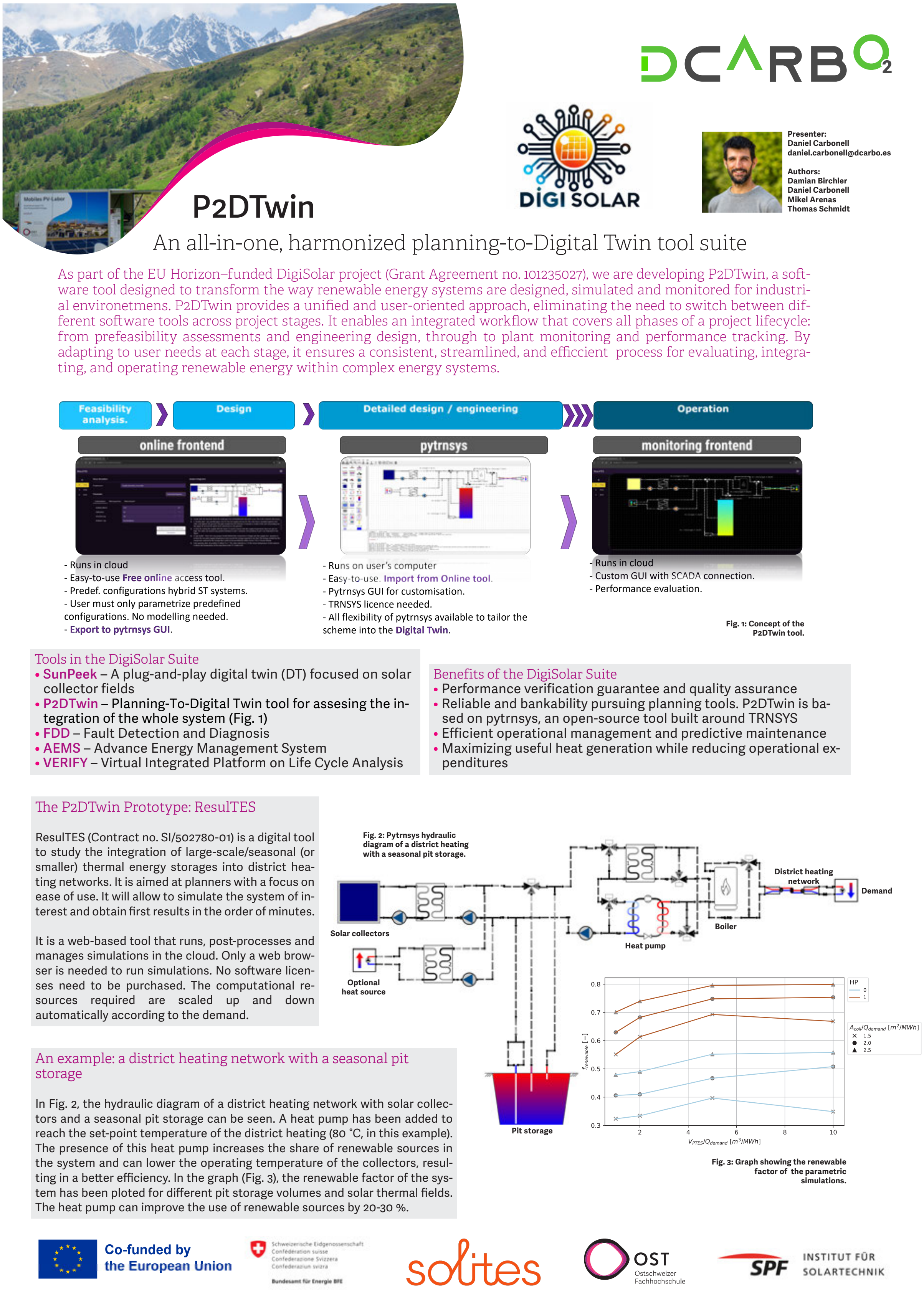
Sibbitt, B., McClenahan, D., Djebbar, R., Thornton, J., Wong, B., Carriere, J., & Kokko, J. (2012). The Performance of a High Solar Fraction Seasonal Storage District Heating System – Five Years of Operation. *Energy Procedia*, *30*(Supplement C), 856 – 865. Retrieved from `http://www.sciencedirect.com/science/article/pii/S187661021201613X` doi: https://doi.org/10.1016/j.egypro.2012.11.097

Sørensen, P. A., & Schmidt, T. (n.d.). Design and Construction of Large Scale Heat Storages for District Heating in Denmark. Retrieved from `https://planenergi.eu/wp-content/uploads/2023/08/Soerensen-and-Schmidt_Design-and-Construction-of-Large-Scale-Heat-Storages-12.03.2018-004.pdf`

Tosatto, A., Ochs, F., Dahash, A., & Muser, C. (2022). The Challenge of Planning and Constructing Large-Scale Hot Water TES for District Heating System: A Techno-Economic Analysis:. Global Online Event, Germany. Retrieved 2023-07-07, from `https://www.atlantis-press.com/article/125970989` doi: 10.2991/ahe.k.220301.006

Xie, Z., Xiang, Y., Wang, D., Kusyy, O., Kong, W., Furbo, S., & Fan, J. (2021, August). Numerical investigations of long-term thermal performance of a large water pit heat storage. *Solar Energy*, *224*, 808–822. Retrieved 2023-07-07, from `https://linkinghub.elsevier.com/retrieve/pii/S0038092X21005120` doi: 10.1016/j.solener.2021.06.027

A Appendix



P2DTwin

An all-in-one, harmonized planning-to-Digital Twin tool suite

Presenter:
Daniel Carbonell
daniel.carbonell@dcarbo.es

Authors:
Damian Birchler
Daniel Carbonell
Mikel Arenas
Thomas Schmidt

As part of the EU Horizon–funded DigiSolar project (Grant Agreement no. 101235027), we are developing P2DTwin, a software tool designed to transform the way renewable energy systems are designed, simulated and monitored for industrial environetmens. P2DTwin provides a unified and user-oriented approach, eliminating the need to switch between different software tools across project stages. It enables an integrated workflow that covers all phases of a project lifecycle: from prefeasibility assessments and engineering design, through to plant monitoring and performance tracking. By adapting to user needs at each stage, it ensures a consistent, streamlined, and efficient process for evaluating, integrating, and operating renewable energy within complex energy systems.

Feasibility analysis. › **Design** › **Detailed design / engineering** ››› **Operation**

online frontend

- Runs in cloud
- Easy-to-use **Free online** access tool.
- Predef. configurations hybrid ST systems.
- User must only parametrize predefined configurations. No modelling needed.
- **Export to pytrnsys GUI**.

pytrnsys

- Runs on user's computer
- Easy-to-use. **Import from Online tool**.
- Pytrnsys GUI for customisation.
- TRNSYS licence needed.
- All flexibility of pytrnsys available to tailor the scheme into the **Digital Twin**.

monitoring frontend

- Runs in cloud
- Custom GUI with SCADA connection.
- Performance evaluation.

Fig. 1: Concept of the P2DTwin tool.

Tools in the DigiSolar Suite

- **SunPeek** – A plug-and-play digital twin (DT) focused on solar collector fields
- **P2DTwin** – Planning-To-Digital Twin tool for assesing the integration of the whole system (Fig. 1)
- **FDD** – Fault Detection and Diagnosis
- **AEMS** – Advance Energy Management System
- **VERIFY** – Virtual Integrated Platform on Life Cycle Analysis

Benefits of the DigiSolar Suite

- Performance verification guarantee and quality assurance
- Reliable and bankability pursuing planning tools. P2DTwin is based on pytrnsys, an open-source tool built around TRNSYS
- Efficient operational management and predictive maintenance
- Maximizing useful heat generation while reducing operational expenditures

The P2DTwin Prototype: ResulTES

ResulTES (Contract no. SI/502780-01) is a digital tool to study the integration of large-scale/seasonal (or smaller) thermal energy storages into district heating networks. It is aimed at planners with a focus on ease of use. It will allow to simulate the system of interest and obtain first results in the order of minutes.

It is a web-based tool that runs, post-processes and manages simulations in the cloud. Only a web browser is needed to run simulations. No software licenses need to be purchased. The computational resources required are scaled up and down automatically according to the demand.

Fig. 2: Pytrnsys hydraulic diagram of a district heating with a seasonal pit storage.

Fig. 3: Graph showing the renewable factor of the parametric simulations.

An example: a district heating network with a seasonal pit storage

In Fig. 2, the hydraulic diagram of a district heating network with solar collectors and a seasonal pit storage can be seen. A heat pump has been added to reach the set-point temperature of the district heating (80 °C, in this example). The presence of this heat pump increases the share of renewable sources in the system and can lower the operating temperature of the collectors, resulting in a better efficiency. In the graph (Fig. 3), the renewable factor of the system has been ploted for different pit storage volumes and solar thermal fields. The heat pump can improve the use of renewable sources by 20-30 %.

Co-funded by the European Union

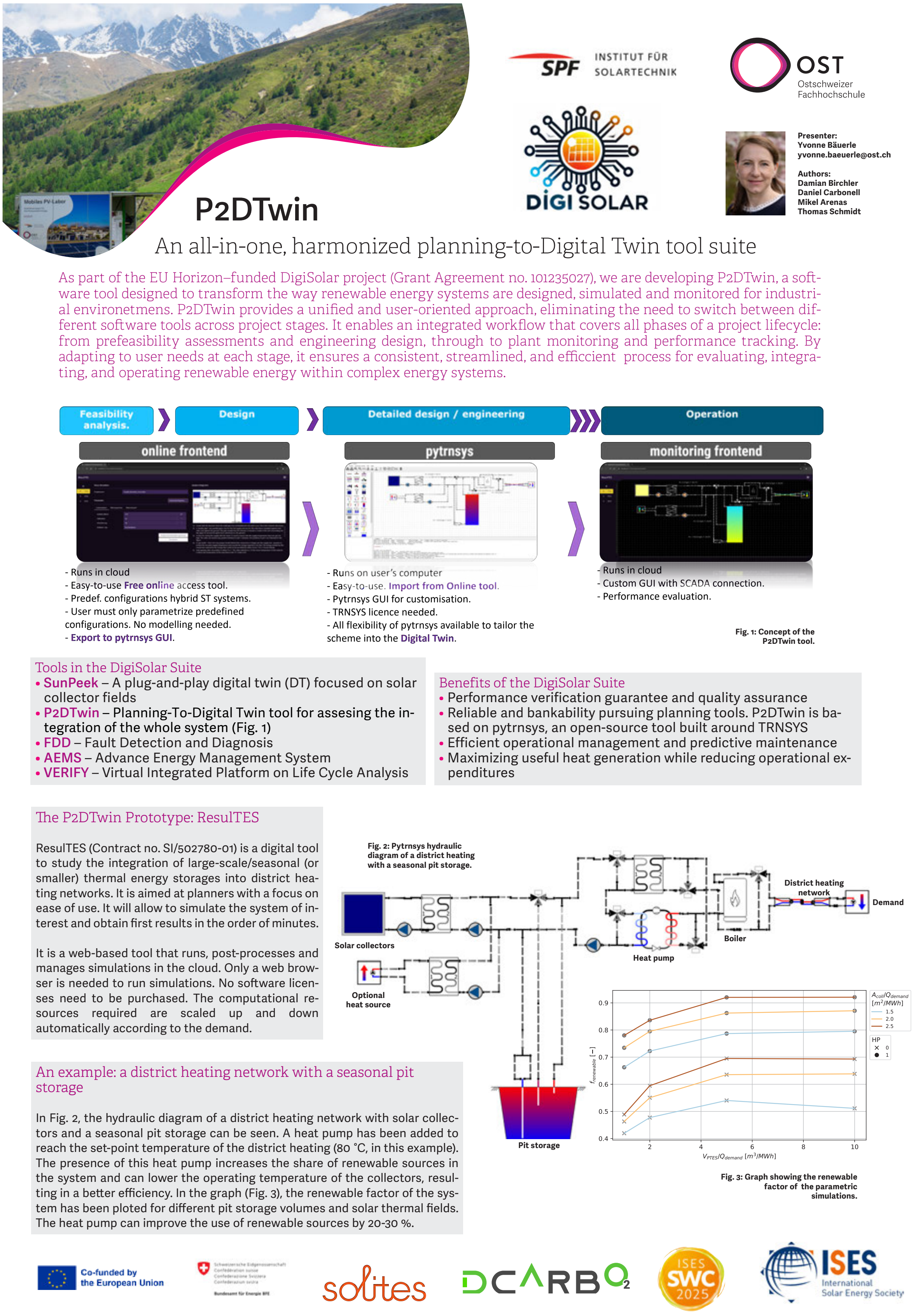
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Bundesamt für Energie BFE

solites

OST Ostschweizer Fachhochschule

SPF INSTITUT FÜR SOLARTECHNIK

Figure 20: Poster presented at the SolarPaces 2025 conference.



INSTITUT FÜR SOLARTECHNIK

OST
Ostschweizer
Fachhochschule

DiGi SOLAR

Presenter:
Yvonne Bäuerle
yvonne.baeuerle@ost.ch

Authors:
Damian Birchler
Daniel Carbonell
Mikel Arenas
Thomas Schmidt

P2DTwin

An all-in-one, harmonized planning-to-Digital Twin tool suite

As part of the EU Horizon–funded DigiSolar project (Grant Agreement no. 101235027), we are developing P2DTwin, a software tool designed to transform the way renewable energy systems are designed, simulated and monitored for industrial environetmens. P2DTwin provides a unified and user-oriented approach, eliminating the need to switch between different software tools across project stages. It enables an integrated workflow that covers all phases of a project lifecycle: from prefeasibility assessments and engineering design, through to plant monitoring and performance tracking. By adapting to user needs at each stage, it ensures a consistent, streamlined, and effficient process for evaluating, integrating, and operating renewable energy within complex energy systems.

- Runs in cloud
- Easy-to-use **Free online** access tool.
- Predef. configurations hybrid ST systems.
- User must only parametrize predefined configurations. No modelling needed.
- **Export to pytrnsys GUI**.

- Runs on user's computer
- Easy-to-use. **Import from Online tool**.
- Pytrnsys GUI for customisation.
- TRNSYS licence needed.
- All flexibility of pytrnsys available to tailor the scheme into the **Digital Twin**.

- Runs in cloud
- Custom GUI with SCADA connection.
- Performance evaluation.

Fig. 1: Concept of the P2DTwin tool.

Tools in the DigiSolar Suite

- **SunPeek** – A plug-and-play digital twin (DT) focused on solar collector fields
- **P2DTwin** – Planning-To-Digital Twin tool for assesing the integration of the whole system (Fig. 1)
- **FDD** – Fault Detection and Diagnosis
- **AEMS** – Advance Energy Management System
- **VERIFY** – Virtual Integrated Platform on Life Cycle Analysis

Benefits of the DigiSolar Suite

- Performance verification guarantee and quality assurance
- Reliable and bankability pursuing planning tools. P2DTwin is based on pytrnsys, an open-source tool built around TRNSYS
- Efficient operational management and predictive maintenance
- Maximizing useful heat generation while reducing operational expenditures

The P2DTwin Prototype: ResulTES

ResulTES (Contract no. SI/502780-01) is a digital tool to study the integration of large-scale/seasonal (or smaller) thermal energy storages into district heating networks. It is aimed at planners with a focus on ease of use. It will allow to simulate the system of interest and obtain first results in the order of minutes.

It is a web-based tool that runs, post-processes and manages simulations in the cloud. Only a web browser is needed to run simulations. No software licenses need to be purchased. The computational resources required are scaled up and down automatically according to the demand.

An example: a district heating network with a seasonal pit storage

In Fig. 2, the hydraulic diagram of a district heating network with solar collectors and a seasonal pit storage can be seen. A heat pump has been added to reach the set-point temperature of the district heating (80 °C, in this example). The presence of this heat pump increases the share of renewable sources in the system and can lower the operating temperature of the collectors, resulting in a better efficiency. In the graph (Fig. 3), the renewable factor of the system has been plotted for different pit storage volumes and solar thermal fields. The heat pump can improve the use of renewable sources by 20-30 %.

Fig. 2: Pytrnsys hydraulic diagram of a district heating with a seasonal pit storage.

Fig. 3: Graph showing the renewable factor of the parametric simulations.

Co-funded by the European Union

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Bundesamt für Energie BFE

solites

DCARBO2

ISES SWC 2025

ISES International Solar Energy Society

Figure 21: Poster presented at the Solar World Conference 2025.