EUROPEAN COMMISSION

HORIZON EUROPE PROGRAMME – TOPIC: HORIZON-CL5-2022-D2-01

FASTEST

Fast-track hybrid testing platform for the development of battery systems

Deliverable D1.3: Requirements and Specifications for Digital Twins

Primary Author [Marco Rodrigues]

Organization [INEGI]

Date: [06/03/2024]

Doc.Version: [V1.1]

Funded by the European Union

Funded by the European Union under grant agreement N° 101103755. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA) Neither the European Union nor CINEA can be held responsible for them.

Document Control Information	
Settings	**Value**
Work package:	1 – Specifications, requirements and use cases definition
Deliverable:	1.3
Deliverable Type:	R – Document, report
Dissemination Level:	PU - Public
Due Date:	29.02.2024 (Month 9)
Actual Submission Date:	06.03.2024
Pages:	28
Doc. Version:	V1.1
GA Number:	101103755
Project Coordinator:	Alvaro Sanchez | ABEE (alvaro.anquela@abeegroup.com)

Formal Reviewers		
Name	**Organization**	**Date**
Alvaro Sanchez	ABEE	22.02.2024
Aitor Barrutia	IKERLAN	22.02.2024
Antonio Passaro	COMAU	22.02.2024

Document History			
Version	**Date**	**Description**	**Author**
V0.2	02.02.2024	First draft	Marco Rodrigues (INEGI)
V1.0	23.02.2024	Updates according to feedback received	Marco Rodrigues (INEGI)
V1.1	06.03.2024	Updates after European Project Officer's feedback	Marco Rodrigues (INEGI)

Project Abstract

Current methods to evaluate Li-ion batteries safety, performance, reliability and lifetime represent a remarkable resource consumption for the overall battery R&D process. The time or number of tests required, the expensive equipment and a generalized trial-error approach are determining factors, together with a lack of understanding of the complex multiscale and multi-physics phenomena in the battery system. Besides, testing facilities are operated locally, meaning that data management is handled directly in the facility, and that experimentation is done on one test bench.

The FASTEST project aims develop and validate a fast-track testing platform able to deliver a strategy based on Design of Experiments (DoE) and robust testing results, combining multi-scale and multi-physics virtual and physical testing. This will enable an accelerated battery system R&D and more reliable, safer and long-lasting battery system designs. The project's prototype of a fast-track hybrid testing platform aims for a new holistic and interconnected approach. From a global test facility perspective, additional services like smart DoE algorithms, virtualized benches, and DT data are incorporated into the daily facility operation to reach a new level of efficiency.

During the project, FASTEST consortium aims to develop up to TRL 6 the platform and its components: the optimal DoE strategies according to three different use cases (automotive, stationary, and off-road); two different cell chemistries, 3b and 4 solid-state (oxide polymer electrolyte); the development of a complete set of physic-based and data driven models able to substitute physical characterization experiments; and the overarching Digital Twin architecture managing the information flows, and the TRL6 proven and integrated prototype of the hybrid testing platform.

LIST OF FIGURES

LIST OF ABBREVIATIONS, ACRONYMS AND DEFINITIONS

Acronym	Name
API	Application Programming Interface
DoE	Design of Experiments
DT	Digital Twin
ETL	Extract, Transform, Load
FMI	Functional Mock-up Interfaces
FMU	Functional Mock-up Units
FTPS	File Transfer Protocol Secure
HTTPS	Hyper Text Transfer Protocol Secure
LIMS	Laboratory Inventory Management System
MQTT	Message Queuing Telemetry Transport
NoSQL	Not-Only Structured Query Language
SSB	Solid State Batteries
SQL	Structure Query Language
URL	Uniform Resource Locator
UUT	Unit Under Test
WP	Work Package

Table of Contents

1. EXECUTIVE SUMMARY

The FASTEST project is a groundbreaking initiative in battery technology, prioritizing innovation for sustainable progress. It seeks to boost the efficiency and safety of lithium-ion batteries while expediting research and development through a state-of-the-art fast-track hybrid testing platform. This revolutionary approach combines physical testing and virtual simulations, offering an advanced system to assess the safety, reliability, and lifespan of batteries.

Deliverable 1.3 provides a meticulous look at the internal components of the Digital Twin, as well as defines and specifies its requirements, functions and technical demands that reside at the center of FASTEST's core structure. The Digital Twin composition and requirements are not just technically relevant for the completion of the project but also acts as the enabler and facilitator of a smooth integration between external systems, as well as the guarantee of virtual test execution. From the upload of Physics-based Simulations and Data-driven Models, which virtually replicate real test execution in various scenarios, to the storage of its outputs creating a Digital Thread overtime and allowing for further Advanced Data Analysis to extract the most valuable knowledge from data, each component represents a small part and has been designed to contribute to the project's success.

Another relevant aspect described in this document is the expected data flow for the execution of each feature of the Digital Twin, where the data comes from and what the destination is. This is particularly important for FASTEST since the system as a whole works through a set of integrated sub-systems exchanging data. These elements are not only important to the operation of the project but also serve as a model for future R&D initiatives in similar fields.

Ultimately, this document intends to be a complete list of requirements and specifications with all the information gathered so far, bearing in mind that in the future adjustments or adaptations may be necessary due to the innovative nature of the project. It also highlights the importance of considering robust, standardized and flexible approaches when dealing with a multitude of systems and technologies.

2. INTRODUCTION

This document starts with the identification of the Digital Twin internal modules, thus giving an overview of its internal composition and how all blocks interact and exchange data. These blocks represent the skeleton of the Digital Twin, conceived and designed to provide solutions capable of working with large amounts of data from physical and virtual tests, as well as virtually representing their behavior with high fidelity by running the simulations and Toolchain models.

Subsequently, all identified requirements for the Digital Twin are listed and categorized in order to specify what the Digital Twin will do and how.

First, the functional requirements are identified. These define the specific functionalities, features, and capabilities that the Digital Twin must provide to meet the needs and objectives of FASTEST, and describe what the Digital Twin is supposed to do.

Next, non-functional requirements for the Digital Twin are identified. These define the characteristics and qualities that describe how the system should perform its functions. Performance, security or communication/interfaces metrics and protocols are addressed. Despite not being possible yet to define specific values or intervals for the metrics, these are still requirements for the proper functioning of the system.

Moreover, Data Requirements with Data Flow Diagrams are presented to illustrate how data should flow for each Functional Requirement. Testing and Validation Requirements are, as well, identified, defining the criteria and processes to verify that the Digital Twin meets its intended specifications and functions correctly. These requirements ensure the quality, reliability, and correctness of the Digital Twin modules, are essential for a structured and systematic approach providing reliability on the developments, and help stakeholders assess the software's readiness level.

Due to the innovative nature of the project, there may be a need for alterations and/or adjustments and/or new features over the course of the project. These situations should be analyzed together with the rest of the consortium, guaranteeing validation for their development/integration.

In summary, this deliverable offers a detailed comprehension of the Digital Twin's requirements and specifications. This includes an in-depth examination of its internal modules, outlining their functionalities and implementation methods, ensuring seamless integration with neighboring systems within FASTEST scope.

3. DIGITAL FLOW FOR VIRTUAL TESTING

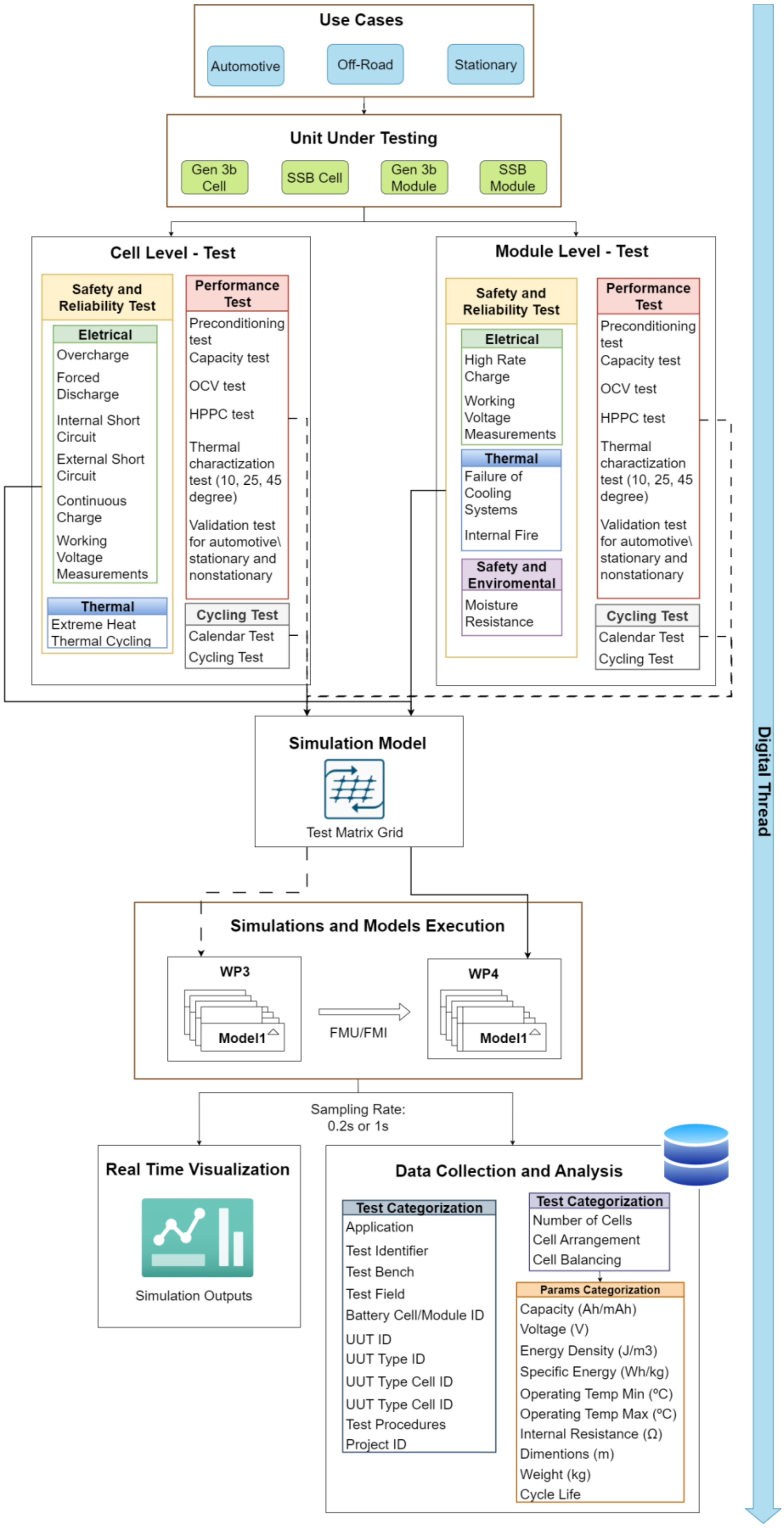

Figure 1 - Fastest Digital Flow.

As part of the development of a fast-track hybrid testing platform, FASTEST project intends to deliver a combination of multi-scale and multi-physics virtual and physical testing to accelerate battery systems R&D. This innovative platform is grounded in three different use cases, automotive, stationary and off-road, and its digital data flow is illustrated in Figure 1, as it is segmented in several parts.

The core of the project revolves around text execution for a specific Unit Under Test (UUT), between Gen3b cell, SSB cell, Gen3b module or SSB module, either physical or virtual, targeting a specific level, cell or module, and a test category, from Safety and Reliability to Performance Tests. Each of these consists in several tests, and match a specific level, as the tests designated for the cell level encompass electrical, thermal, cycling tests, aging and performance, and safety and environmental, and for the module level, the tests include electrical, aging and performance, and cycling tests. It is noteworthy that, at this stage, the same tests are planned across all use cases.

The determination of the use case, unit under test and test type allows to start the digital flow by processing this information within the simulation models block, where a test matrix grid translates the UUT, test specification and use case into a simulation model developed by either WP3 or WP4. Dashed lines in the diagram represent the tests simulated in WP3, while solid lines correspond to WP4. Following the selection and execution of simulations with all the testing procedures, its outputs are presented in real-time on a dedicated component running a real-time data with a customized user interface, with sampling rates ranging from 1 second for slower simulations to 0.2 seconds for faster ones, being as well persistently recorded in a database system for long-term storage.

The database, fed by the interconnected digital flow, will create a digital thread for battery testing overtime and shall be organized into three main parts. The first part considers Test Categorization, recording all aspects that define the test from the application and UUT ID to the test procedures. The second part for Parameter Categorization, which remains consistent across all UUTs and records parameters based on the specific test conducted. Common parameters such as capacity and voltage are included. The third part, Module Categorization, differs from cell categorization and, thus, is allocated in a distinct column. It includes all elements of cell categorization plus additional parameters such as the number of cells, cell arrangement and cell balancing, which are crucial for comprehensive module characterization.

This well-defined data structure and data management procedures will allow the creation and further instantiation of the discrete (also referred to as component) [1] twins based on the lowest level of abstraction for the asset representation, which in FASTEST project are the cells, and the composite twins representing the module, that relies within the combination of discrete digital twins and other resources transposed to a higher level of abstraction. Figure 2 illustrates the combination of discrete twins to create a composite twin.

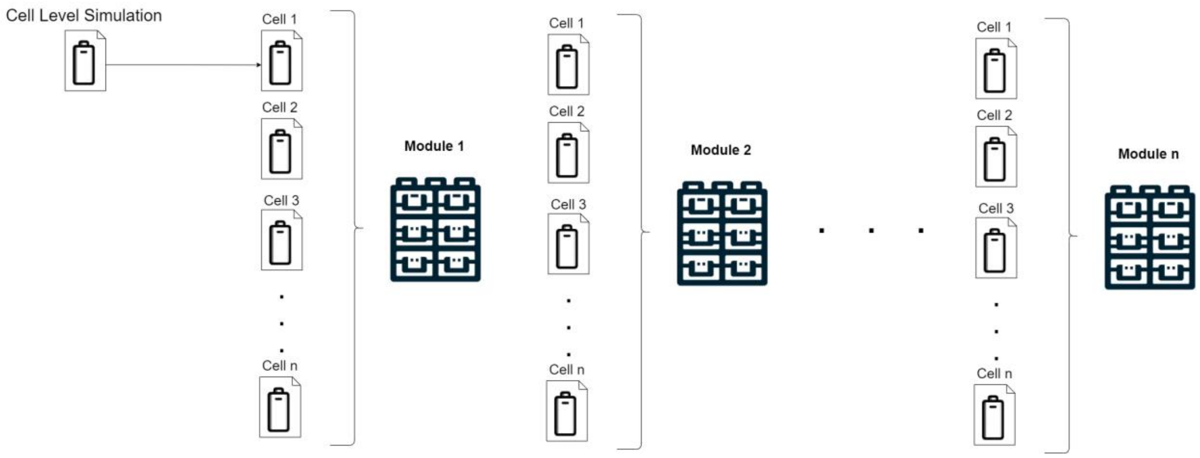

Figure 2 - Discrete Twins and Composite Twins

For FASTEST project, implementing the Digital Twin and its internal components is crucial for several reasons. Firstly, its a cornerstone in the digitalization of tests, accelerating the testing processes. By creating a virtual representation of physical assets as simulations and virtual models, physics-based and data-driven, the digital environment will provide the possibility to virtually execute these tests reducing the need for costly and time-consuming physical tests. Additionally, the interoperability among the different digital components within the digital twin, including the simulations, enables a seamless data exchange and further storage. Moreover, this capability of the Digital Twins over time is essential for accumulating historical data. By capturing and storing data from ongoing operations, valuable insights or anticipation of potential issues regarding asset behaviour in different tests may be provided, ultimately enhancing efficiency, reliability, and decision-making processes.

It is important to acknowledge that the scope of the FASTEST project is subject to continuous development, and the current framework may evolve in the future.

4. DIGITAL TWINS MODULES

In the FASTEST project is vital to create a detailed composition of all active sub-systems, specifying its core modules, features and requirements as much as possible. As part of a forefront initiative in revolutionizing battery testing through digitalization, the Digital Twin should be composed of a well-defined internal architecture to guarantee a smooth integration with neighbouring elements, such as LIMS, DoE or XMOD. This architecture will be implemented and deployed in Microsoft Azure platform, with containerization technologies (e.g., Docker) and orchestration platforms (e.g., Kubernetes) for efficient deployment and management of cloud services, extending this way the currently developed In.Grid platform from COMAU.

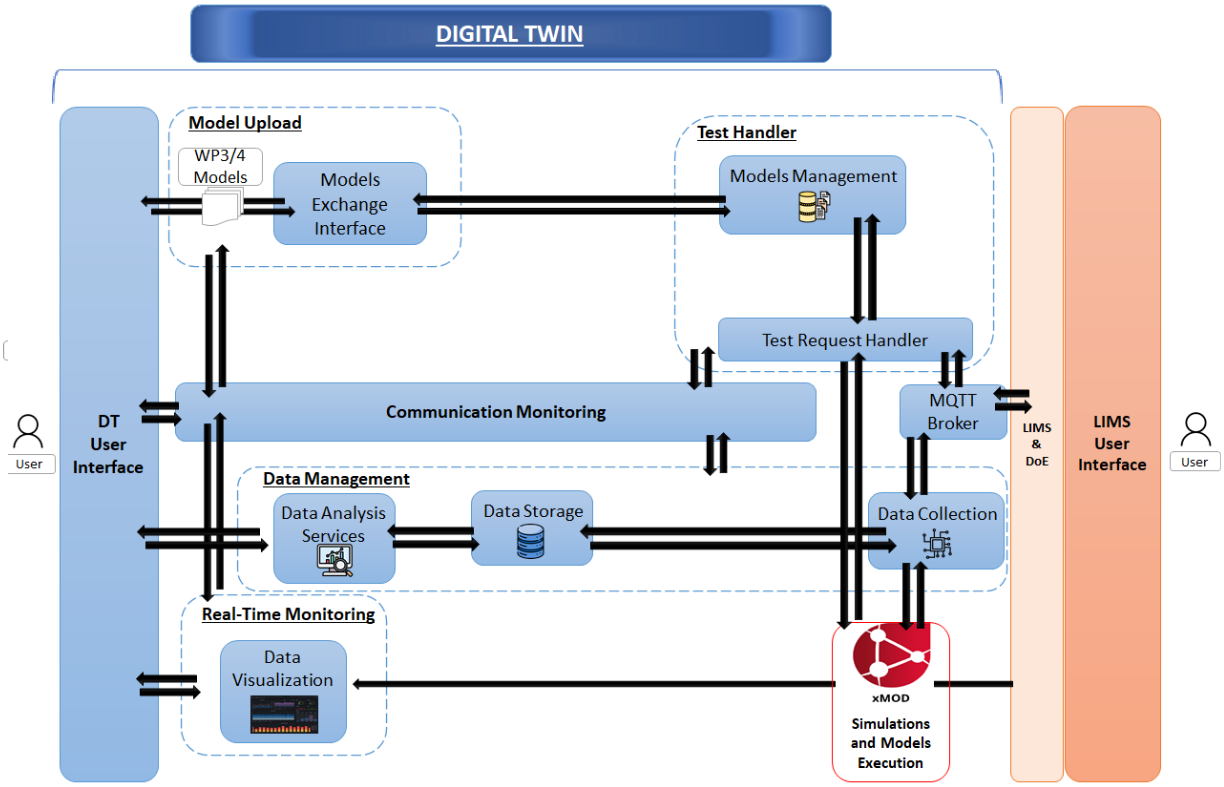

Figure 3 - Digital Twins internal architecture and core modules

As illustrated in Figure 3 the Digital Twins can be divided into the following function blocks:

- **DT User Interface**: This interface allows users to access, visualize and take actions on Digital Twin data remotely, and for most of the features on a graphical and user-friendly manner. The interface is transversal to all the remaining Digital Twin modules and features a range of pages and dashboards for data visualization, information on models, simulation sequences, historical test data, performed model simulations, and the outcomes of model simulations over time. It also includes functionalities for model upload and updates.

- **MQTT Broker:** acts as a central hub of data internally in the Digital Twin. This component coordinates messages between all client components, based on a publish/subscribe messaging model. It will be important to exchange data, at least, between the "Test Request Handler" and "Data Collector" modules, but during the project more modules may be needed to connect to the MQTT Broker as client.

- **Model Exchange Interface**: This component functions as a repository for storing uploaded models from WP3 and WP4. It has implemented its own backup strategy and is equipped with internal version control and file management systems.

- **Models Management**: This component plays an integral role in the Digital Twin framework. It receives test requests data from LIMS and determines which simulations, models and test procedures should run.

- **Test Request Handler**: Acting as a conduit between LIMS/DoE and the Models Management module, this handler facilitates the exchange of data between the two handling all requests that comes from LIMS, sends simulation and test procedures specifications for DoE to calculate the optimal procedures, receives optimal procedures from DoE and sends message to XMOD to start executing tests.
  It also sends messages with data related to the tests to be stored permanently in the Data Storage module via the internal MQTT Broker.

- **XMOD**: Although not a specific development of the FASTEST Digital Twin, XMOD is a co-simulation platform provided from FEV, vital for running the simulations.

- **Data Visualization**: This module processes the outputs from ongoing simulations and delivers them to the user interface for real-time display.

- **Data Collection**: Responsible for receiving data from XMOD or from the internal MQTT Broker. This module prepares the data for storage in the database.

- **Data Storage**: Serving as the database and considering multiple database types, this module stores data permanently from running tests and simulations and makes it available for further usage.

- **Data Analysis Services**: This module conducts analysis over stored data, either periodically or upon user request.

5. FUNCTIONAL REQUIREMENTS

Functional requirements outline the specific tasks, actions, and operations that the Digital Twin must perform to fulfill its intended purpose. These requirements are enumerated below and organized into four categories to provide a clear overview of the functionalities that collectively contribute to the system's goal.

5.1 Model Upload

The initial functional requirement centers around the "Model Upload" block, as illustrated in Figure 3. This block, as its name suggests, serves as the hub for uploading, updating, and securely storing various types of simulations and models developed within the scope of WP3 and 4 of the FASTEST project. In order to successfully fulfil the purpose of this feature and meet the requirement to manually upload and manage simulation and model files, the "Models Exchange Interface" repository will be made available to the user via the Digital Twin user interface, where all files related can be uploaded and stored.

Internally, as seen in Figure 4 a backup folder will be available, where a python script will run regularly to store backups attaching the date and time of each execution to the name of the file in order to generate the backup.

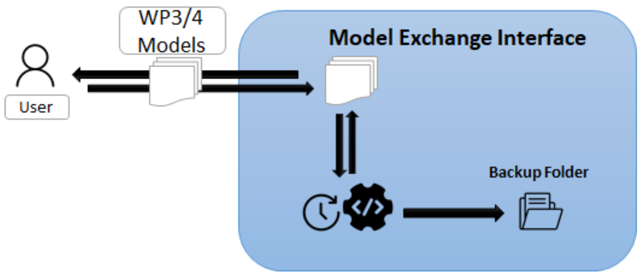

Figure 4 - Model Exchange Interface and Backup System

5.2 Test Handler

This module of the Digital Twin will handle all steps in between the tests execution, except for running the Simulations or Models, starting from receiving an order to execute a specific test, to specify which simulations and test procedures for the Unit Under Test should run, to send Simulation/Model files to XMOD run, among others.

To start, the Digital Twin receives a test request specifying the test type, test inputs, and a unique identification for the Unit Under Test (UUT). Using these parameters as inputs, the "Models Management" component determines which simulation type to use (either WP3 or WP4) and defines the testing procedures based on the specified test type.

Once these are specified, the testing procedures are sent to the Design of Experiments (DoE) module via "Test Request Handler" component. The DoE module then figures out the most suitable procedures for the given test and communicates back to the "Test Request Handler" of the Digital Twin. After defining the simulations to run and the optimal testing procedures, all necessary information is sent to the XMOD Platform. Additionally, the "Test Requests Handler" sends a message to XMOD to start the simulations and specifies their order of execution.

During the simulations, the data generated will follow three different and parallel routes. First, it is transmitted to the "Data Collector" function block for storage in the "Data Storage" component, which is discussed with more detail in the next section. Second, the data is sent to a Real-Time Visualization module where it should be possible to monitor the simulations/models executions, and last, the data is sent to the LIMS platform, where it is stored to show its results after test run.

5.3 Data management

As previously mentioned, the Digital Twin will have the capability to collect and store the output data generated by running simulations, as well as handle this data and store it in adequate structures for further analysis. When a simulation produces output data, XMOD transmits it according to the sampling time of the simulations to the "Data Collector" function block, which in turn stores permanently the received data in the "Data Storage" module. This should be an enabler for data analysis and insights using historical data from tests, simulations, testing procedures, and more. The primary objective of the "Data Analysis Services" function block is to explore Data Analytics techniques and strategies, such as to identify trends over time, detect outliers, find clusters in data based on the type of tests or physical test benches, etc.

5.4 Real-Time Monitoring

This module aims to deliver real-time information about ongoing simulations via its outputs. Depending on the sampling time of the simulations and volume of data generated, two methods for achieving this will be considered:

1. Direct Data Transfer: Data from the simulation, generated by XMOD, is sent directly to visualization, where all data received is immediately handled by the browser and the user interface components.
2. Database First: Alternatively, data is first stored in a database, from which this module then retrieves the relevant information by querying it.

Both approaches will be further analyzed to ensure that up-to-date information about the simulations can be displayed in a visualization page.
For visualization, dashboards will be built either as a fully customized and developed web page, or over a visualization tool, such as Grafana [2], that already provides built widgets and components like charts, heatmaps or scorecards, with specific configuration modules to connect to several possible data source types.

5.5 Communication Monitoring

In the Digital Twin ecosystem, the "Communication Monitoring" feature intend to ensure that all Digital Twin resources remain operational overtime. Its primary role is to continuously check and confirm the status of various system components to maintain smooth communication and functionality, and detect potential downsides or issues as soon as possible.

This module actively monitors the health and availability of Digital Twin resources, such as servers, interfaces and data exchange tools. It regularly assesses their operational status, responsiveness, and connectivity. When an issue or downtime occurs, the module detects it and takes prompt actions for resolution.

To achieve this, the module uses diagnostic checks and communication protocols to examine data pathways, interface responsiveness, and communication channel stability.

In case of a resource disruption or failure, the module sends alerts and notifications to inform relevant stakeholders, enabling swift response and remediation. This proactive approach minimizes downtime, improves system reliability, and ensures uninterrupted data flow within the Digital Twin environment.

The "Communication Monitoring" feature plays a vital role in maintaining the Digital Twin infrastructure's robustness and reliability, supporting the core objectives of the FASTEST project. It ensures seamless communication, safeguards data exchange and helps the Digital Twin effectively fulfill its mission.

6. NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements will define the characteristics, attributes, or constraints that describe how the Digital Twin should perform its functions, rather than what functions it should perform. These requirements address qualities such as performance, reliability, usability, security, and other aspects that impact the overall behavior and performance of the system. Non-functional requirements are equally important as functional requirements to determine the success of the Digital Twin in FASTEST project. These are listed below, to create a robust and efficient platform, being categorized for a better understanding of the impact of each one in the system.

6.1 Performance

- **Response Time**: all Digital Twin modules must respond to requests sent by users or other systems in a way that the tests are executed successfully and that a potential unreliability of the data is not caused by excessive delays in response or data loss during execution. There aren't yet reference values to define as a concrete requirement for this metric, so this should be specified once tests start and a baseline is defined.

- **Throughput**: as a metric that defines the number of transactions or operations the system should be able to handle in a given time period. It is related with the Response Time and is something to consider when tests start. It has not reference values defined at this stage of the project.

- **Scalability**: All Digital Twin modules should be scalable to handle enough complexity, size and volume to run virtual tests through the simulations and models developed in WP3 and 4, which can be of varying characteristics. It should be able, as well, to accommodate the growth of data and computational requirements according to the project needs and allowing the project's objectives to be achieved.

- **Availability and Disaster Recovery:** downtime periods for maintenance of the Digital Twin modules should be scheduled at appropriate times so it does not affect the tests execution and the collection of the respective data. Any other period of downtime must be analyzed in detail and its cause tracked, recorded and resolved, if possible in an isolated manner, affecting as little as possible on the remaining components of Digital Twin. DT Modules will be designed for high availability by distributing components across multiple availability zones and regions to minimize downtime. Backup and disaster recovery strategies will be implemented to ensure data resilience and quick recovery in case of failures.

6.2 Interoperability

All Digital Twin modules should be able to interact and exchange data with other systems without putting at risk the test response time and requirements, or the smooth running of the adjacent systems (LIMS, DoE). Interoperability standards and protocols are described in subsection 6.4 and should be followed to ensure compatibility with the other tools and platforms.

6.3 Security

Considering that Information Security has always been a complex subject, and it evolves quickly due to creative ideas and implementations of attackers and security researchers, this should be a relevant aspect to take into account for the Digital Twin development of FASTEST. Therefore, since the DT infrastructure and services will reside on Microsoft Azure Cloud Platform, its own security guides and best practices [3] shall be considered for this implementation. From the design of the framework itself, to specific cloud services to monitor Threat Indicators, to manage trustworthy devices or to manage risk.

The following are three relevant topics under Security aspect, but also others shall be considered.

- **Access Control:** The Digital Twin will consider several different profiles to show different levels of information. It will be implemented following the Least Privilege Principle, which reduces potential attack surfaces by granting only the minimal access to resources and permissions depending on the roles and responsibilities of users.
- **Data Encryption**: Certain data might be stored using and encryption algorithm rather than as plain text, as well as techniques to encrypt during transmission of data will be considered, such as HTTPS.
- **Authentication and Authorization**: The Digital Twin modules will be password protected, with mechanisms defined for Two-Factor Authentication, and should only be accessible by project members.

6.4 Communication/Interfaces

All Digital Twin modules must be able to interact and exchange data with other systems. Some interfaces and communication protocols, listed below, are already defined, nevertheless due to the innovative nature of the project, some changes or adjustments may be needed in order to adapt to future needs, as well as other interfaces with different specifications may arise:

- **WP3 and 4 Models Upload**: the repository (Models Exchange Interface) to manually upload the models will be served through an **HTTPS** URL.

- **WP3 and 4 Models from the Models Exchange Interface to XMOD** via File Transfer Protocol SSL(**FTPS**)

- **WP3 and 4 Models**, developed in MATLAB, will be integrated in XMOD Platform through the usage of **FMI [4] and FMU standards**

- **Test Handler** will provide interface with other systems, such as LIMS and/or DoE, through **REST API HTTP** and **MQTT**

- **DT communication between internal modules** should be performed using **MQTT**, taking advantage of DT internal MQTT Broker to handle all data forwarding, and in some cases using **REST API HTTP**

- **Test Handler, XMOD and Data Collector** communication should be performed using **MQTT** and **REST API HTTP**

6.5 Data Storage

Each test/simulation will provide outputs that must be stored persistently in the Digital Twin, which should be done using Database Systems. Multiple Database types, available on Microsoft Azure cloud platform [5], must be considered, due to characteristics of data itself, data acquisition procedures or even data analysis techniques that can be used to query data and extract the most valuable information, mainly when the volume of data becomes large. ETL processes to transform data after storage may be considered for better performance and quicker results on analyzing the acquired data.

The main database types, designs and services to be considered are:

- **SQL** - Relational database service that offers high-performance, scalable storage for structured data. It supports SQL queries, transactions, and analytics capabilities, making it suitable for analytical workloads requiring relational data storage.

- **NoSQL** – Unstructured storage that should support multiple data models including document, key-value, graph, and column-family.

- **Table-Storage** - Scalable and flexible NoSQL data store designed for storing semi-structured data such as key-value pairs or entities with a schema-less approach. It offers high availability, automatic scaling, and low-cost storage, making it suitable for applications that require fast and efficient access to large volumes of structured data.

- **Data Warehouse** - service to handle large-scale analytical workloads, it integrates various data sources and supports both relational and non-relational data.

- **Mixed Data Lake** - Scalable and secure service that allows storage and analysis of large volumes of structured and unstructured data. It can integrate with analytics services as well.

6.6 User Interface

As the Digital Twin modules will be served to the user via a web-based platform that runs in the browser, the developments will be made using Angular language and all implementation properties, syntax and web-components used must be compatible with the most modern web-browsers.

Add-on tools might be considered for specific parts of the Digital Twin, such as Grafana [2] for Real-Time Monitoring module, which provides a customizable monitorization tool with several widgets/components. This helps reducing implementation effort and allows the implementation team to be focused more time on specifics of battery testing itself.

7. SOFTWARE DEVELOPMENT REQUIREMENTS

Following good practices is fundamental for a project that involves software development. Within this part, some good principles compliant with the IEC61508 [6] standard will be used:

- Abstraction, which helps reducing complexities, hides implementation details and enhances code maintainability
- Modularity, which involves designing and organizing a system as a set of independent, interchangeable, and loosely coupled modules, therefore includes concepts such as encapsulation, reusability and code scalability
- Clear Information Flow, which involves defining understandable and manageable elements, such as variables, functions, classes to write effective and maintainable code.
- Single Responsibility Principle (SRP), where each class or function should have only one reason to change. It should have a single responsibility, making the code more focused and less prone to bugs
- Don't Repeat Yourself (DRY), to avoid duplicating code by encapsulating common functionalities in functions or classes and reusing them throughout the codebase.
- Keep It Simple (KISS), to strive for simplicity in the code, avoiding unnecessary complexity and over-engineering. Simple code is often easier to understand, maintain, and debug.
- Exception handling, which is a mechanism in software development that deals with runtime errors or exceptional situations that may occur during the execution the program and involves concepts such as try-catch blocks, throw, finally or custom exceptions to help prevent unexpected crashes
- Comments, to help understand, read and maintain each code part overtime

8. DATA FLOW SPECIFICATIONS

Dataflows represent the pathways of data movement and transformation. They are crucial for ensuring operational efficiency, maintaining data integrity, and adhering to compliance and security standards. Properly defined dataflows facilitate the effective and secure handling of data, essential for informed decision-making and organizational trust. Below, dataflows will be defined for each functional requirement.

8.1 Model Upload

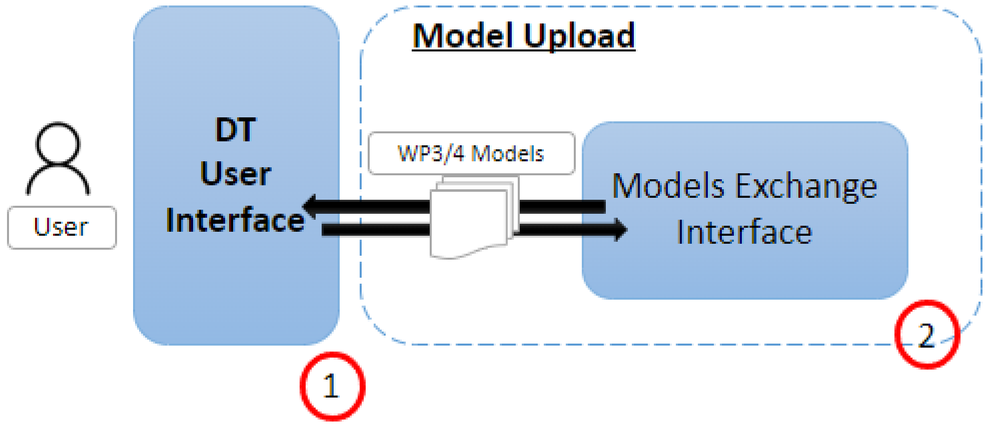

Figure 5 – Data Flow for Model Upload

Figure 5 illustrates the dataflow process associated with uploading, updating, and storing simulation files and models. Initially, the user interacts with the system at **1**. This interaction can occur through two communication protocols: HTTPS or FTPS. Both methods facilitate the drag-and-drop functionality for model upload/updates. It's noteworthy that the models follow the FMI/FMU standards. Once uploaded, the model is then transmitted to **2**, where it is securely stored.

8.2 Test Handler

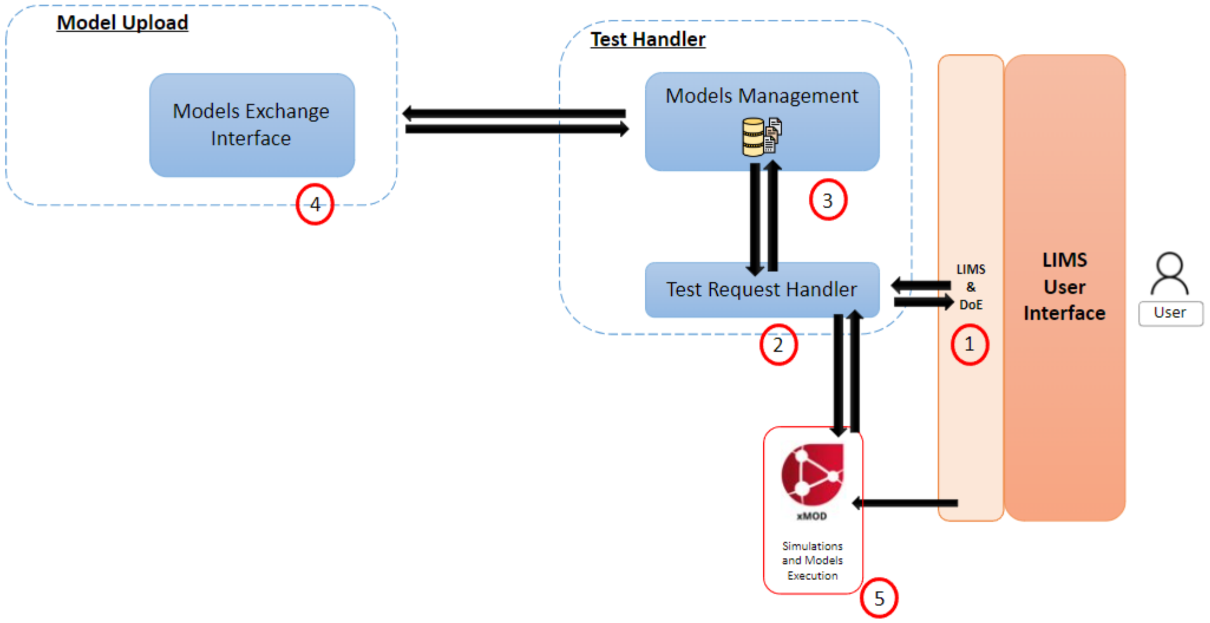

Figure 6 - Data Flow for Test Handler

Figure 6 delineates the sequence of operations involved in initiating a test, leading up to the beginning of model execution. It's important to clarify that the test scheduling process is not covered in this figure, as it falls outside the scope of the Digital Twin, being this feature covered by the LIMS.
The process begins once the user pre-schedules the test. This pre-scheduled test information is transmitted from **1** to **2** using a REST API/MQTT. At **2**, the message is forwarded to **3** where a selection is made for the appropriate models, test procedures, and Unit Under Test. This compiled information is then sent back from **3** to **2** and back to **1**, again via REST API/MQTT. Here, the DoE will use this data to determine the most efficient testing procedures, which are subsequently relayed back to **2** and back to **3**.
Armed with these optimized testing procedures, **3** then retrieves the model files from **4**. These files are conveyed back to **2** which will send the files to **5** using the FTPS communication protocol. Additionally, **2** sends the selected testing procedures to **5**, along with a command to initiate the running of the models. This comprehensive flow ensures a seamless and efficient process from test initiation to the execution of the models.

8.3 Data Management and Real-Time Monitoring

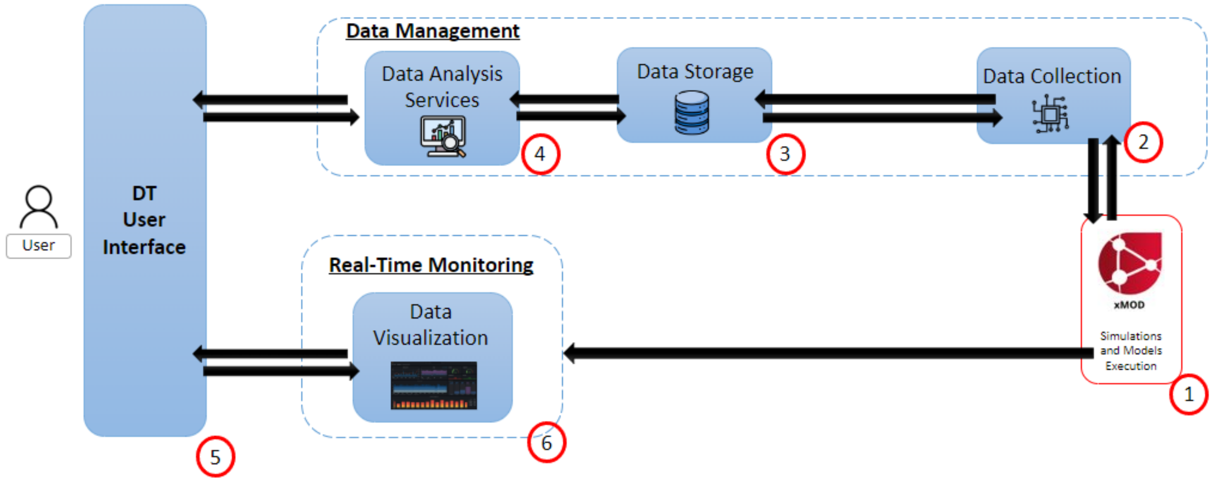

Figure 7 - Data Flow for Data Management and Real-Time Monitoring

Figure 7 demonstrates a sequence of events for managing the model's output data. Initially, as the model operates in **1**, produces data according to each specific sampling time, which needs permanent storage. This data is then routed through REST API/MQTT to **2** for handling. At this stage, the data is interpreted, normalized when needed and arranged, making it ready for storage in a database system **3**. In parallel, the generated data from **1** is also transmitted to **6**. Thus, when the user accesses the interface at **5**, can observe the progression and status of the simulation in real time.
Concurrently, there's a separate mechanism **4** that taps into the database system. It's used for continuous historical data analysis or for specific analyses requested by users. Once these analyses are completed, the results are displayed in interface **5**, providing insights and findings.

9. TESTING AND VALIDATION REQUIREMENTS

In the development of the Digital Twin for battery testing, the followed approach to define specifications focuses on a robust validation and testing framework, with a strong emphasis on accuracy to ensure the Digital Twin mirrors the complex dynamics of physical battery systems. This commitment to accuracy encompasses not only the direct representation of battery cells, modules, and packs but also aligns with the IEC 61508 standards, integrating safety and reliability principles into the core design and functionalities of the simulations, within the Digital Twin.

The testing strategy, considering IEC 61508 norms, places a significant focus on assessing the Digital Twin's reliability and robustness across a spectrum of operational conditions. This encompasses not only routine scenarios but also extreme and unforeseen circumstances, thereby ensuring the Digital Twin's resilience and consistent performance.

One critical aspect of the validation process involves testing to ensure that the behaviour observed in physical tests precisely matches the outcomes of virtual tests conducted under identical conditions, inputs, and test procedures. This verification process, conducted by comparing the results of the physical test with the results of the same virtual test, ensures that the Digital Twin faithfully replicates real-world battery behaviour, reinforcing its accuracy and reliability.

Additionally, tests to guarantee seamless communication within the Digital Twin ecosystem should be conducted, minimizing data loss due to latency or other communication-related issues. These tests are crucial for ensuring that data flows smoothly between different components of the DT and that latency does not compromise the accuracy and timeliness of information exchange.

In parallel, safety protocols that mirror the principles of IEC 61508 are designed and tested within simulations and models from WP3 and 4, placing a strong emphasis on risk management and hazard prevention. This approach equips the Digital Twin to predict and handle potential safety issues within battery systems, making it a reliable tool for managing the inherent risks associated with battery technologies.

10. CONCLUSIONS

This document sets the list of requirements and specifications for the Digital Twins developments so far. Being the Digital Twin an essential part of the FASTEST project that contributes for the acceleration of research and development (R&D) on battery testing systems, it is relevant for the project that these requirements and specifications are identified and detailed as thorough and comprehensive as possible. At first, to provide a visual flow of the information provided by actions targeting battery testing for three use cases, a Digital Flow was designed. This helps understand what connects each part on a digital level, and where data goes among the system. With this illustration, it is understandable that The core of the project involves executing specific tests on either physical or virtual units under test (UUTs), targeting various levels (e.g., cell or module) and test categories (e.g., electrical, thermal, cycling). Simulation models developed by different work packages (WP3 or WP4) process the test specifications, and their outputs are presented in real-time on a dedicated component with a customized user interface. The database system stores the simulation outputs for long-term storage, organized into three main parts: Test Categorization, Parameter Categorization, and Module Categorization. This structured data management facilitates the creation of discrete twins based on the lowest level of asset representation (cells) and composite twins representing modules.

Next, a detailed composition of all active sub-components in the Digital Twin is provided with a well-defined internal architecture to integrate smoothly with neighboring elements such as LIMS, DoE, etc. Function blocks of the Digital Twins are specified, such as the DT User Interface for remote access of data, the Model Exchange Interface for storing uploaded models, Models Management for receiving test data and determining simulations, Test Request Handler facilitating data exchange, XMOD for running simulations, Real-Time Monitoring, Data Collection, Data Storage and Data Analysis Services. This comprehensive structure ensures efficient operation and management of the Digital Twin framework.

Functional requirements for the Digital Twin in the FASTEST project are identified and organized into four categories to outline specific tasks and operations crucial for the system's functionality:Model Upload, Test Handler, Data Management, Real-Time Monitoring and Communication Monitoring. These, are essential for the Digital Twin to fulfill its purpose in revolutionizing battery testing through digitalization within this project.

Non-functional requirements are listed as well, including aspects related to performance (response time, throughput, scalability and availability), interoperability, security, communication interfaces, data storage and user interface.

Following, good software development practices with compliance with IEC61508 standard are identified as well, including abstraction, modularity, clear information flow, single responsibility principle, don’t repeat yourself, among others. Specific dataflows are also described within this document to provide a clear view of data direction for each specific functional requirement.

Lastly, Testing and Validation requirements are specified, aligning with IEC61508 standards and integrating safety and reliability principles into the core design. Testing strategies focus on assessing reliability and robustness across various operational conditions, including routine, extreme, and unforeseen circumstances. A critical aspect involves validating that virtual tests precisely mirror physical test outcomes, ensuring faithful replication of real-world battery behavior. Testing also ensures seamless communication within the Digital Twin ecosystem to minimize data loss and latency-related issues. Safety protocols, aligned with IEC 61508, are designed and tested within simulations, emphasizing risk management and hazard prevention to address potential safety issues within battery systems.

One of the KPIs of the FASTEST project is to instantiate three Digital Twins, one per each use case demonstrator. That should be achieved by deploying three generically similar architectures on Microsoft Azure cloud platform, nevertheless nuances and specific adaptations may be needed to adjust technologies for each use case context and needs, so in the end each use case should have its own Digital Twin architecture instantiadedand deployed, quite similar between each other, but with the appropriate adjustments and fine-tunes.

It is relevant to mention that due to the innovative nature of the FASTEST project, the requirements for the Digital Twins may continue to change or be subject to future fine-tunes/adjustments to meet the project goals.

11. REFERENCES

[1] https://www.ibm.com/topics/what-is-a-digital-twin

[2] https://grafana.com/

[3] https://learn.microsoft.com/en-us/azure/architecture/guide/security/security-start-here

[4] https://fmi-standard.org/

[5] https://azure.microsoft.com/en-us/products/category/databases/

[6] Bell, R. (2006, April). Introduction to IEC 61508. In Acm international conference proceeding series (Vol. 162, pp. 3-12).