DigInTraCE

D2.9 Specifications definition and architecture of digital tracing v1

D2.9 Specifications definition and architecture of digital tracing v1

A Digital value chain Integration Traceability framework for process industries for Circularity and low Emissions by waste reduction and use of secondary raw materials

DigInTraCE

Dissemination level	Public (PU)
Type of deliverable	Document
Work package	WP2 – Holistic Framework for digitalisation of circular value chains
Deliverable number	2.9 - Specifications definition and architecture of digital tracing v1
Status - version, date	V1, 04/6/2024
Deliverable leader	UST
Contractual date of delivery	30/6/2024
Keywords	Architecture, system, sub-units, financial, toolkit, mock-ups, web, technology

Quality Control

	Name	Organisation	Date
Peer review 1	Alexandros Chatzipavlidis	ICCS	24/06/2024
Peer review 2	Victor Ferandez, Francisco Javier Real Salas	CIRCE	17/06/2024

Version History

Version	Date	Author	Summary of changes
01	04/6/2024	UST HTECH ICCS DGS	Initial version.
02	14/6/2024	CIRCE	Added LCA component
03	14/06/2024	UST	Format adapted to Digintrace Template and MS Word editor revision passeed
04	24/06/2024	UST	Add disclaimer warranties. Change figure CLSC section.

DigInTraCE

Legal Disclaimer

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them. The information in this document is provided “as is”, and no guarantee or warranty is given that it is fit for any specific purpose. The DigInTraCE project Consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

DigInTraCE

Table of Contents

DigInTraCE

DigInTraCE

List of Figures

List of tables

DigInTraCE

DigInTraCE

1 Executive Summary

D2.9 Specifications definition and architecture of digital tracing v1 (M18) is the result of the work carried out during the execution of task 2.4 (M8-M28) whose objective was to present produced technical specifications of the Decentralized Traceability platform architecture and the modules to be developed, including access rights, confidentiality issues, and user roles. This v1 will evolve into the final version of document (D2.10) with a delivery date in M28.

The current version (v1) details the requirements, technical specifications, and functionalities offered by the digital tools, which together make up the DigInTraCE platform, considering the requirements and needs identified up to the moment of delivery of this deliverable in M18 and which are reflected in the Deliverables *Deliverable D2.1 – Methodology of Traceability v1, D2.4 – Digital Product Passport Concept v1 and D2.6 – Real-time information and secure data exchange requirements v1*.

In chapter 2, the specifications and architecture for the tools: Financial Toolkit, Blockchain Module, CLSC, MR Enabler and LCA tool are found. Each of these tools has been defined with a level of detail and depth in accordance with both the complexity of the module as well as the known requirements for the specific tool, although, as mentioned, the final version of this document may expand the details and include new features and information.

Subchapter 2.2 is dedicated to the Financial Toolkit. The Financial Toolkit will be a module in the DigInTraCE platform designed to enhance decision-making and risk management for stakeholders in the secondary raw materials market. The Financial Toolkit, dedicated to enriching the suite of financial services for stakeholders in the secondary raw materials sector, aims at providing an all-encompassing resource to support these stakeholders. Its goals encompass:

- Crafting strategies for mitigating price volatility risks associated with secondary raw materials.
- Integrating considerations for currency exchange in transactions involving secondary raw materials across different currencies.
- Utilizing technical analysis to spotlight significant pricing trends and forecast near-term price shifts in secondary raw materials.
- Mapping out and presenting the correlations between secondary raw materials and other globally traded commodities, along with major stock indices and currency pairs.
- Defining and pricing financial derivatives centered on secondary raw materials.

Version 1 (v1) of this toolkit introduces functionalities such as:

- Visualization charts for secondary raw materials.
- Technical indicators for identifying crucial price trends and forecasting imminent price changes.
- Strategies derived from technical indicators.
- A dynamic correlation matrix to map and showcase the links between secondary raw materials and other globally traded assets, including major stock indices and currency pairs.

DigInTraCE

- A forecasting feature for currency pairs to integrate currency exchange considerations in cross-currency dealings of secondary raw materials.

The toolkit's final version will feature:

- The creation of financial derivatives based on secondary raw materials.
- Pricing models for these financial derivatives.
- Hedging strategies employing financial derivatives to counter the risk of price fluctuations in secondary raw materials.

This section outlines the system architecture, sub-units, and web application requirements, including both functional and non-functional aspects, alongside mockups, use cases, and a technology study to guide development and implementation.

Subchapter 2.3 tackles Blockchain Module provided by UST. The Blockchain technology employed in the DigInTraCE project aims to provide a distributed, decentralized, transparent, immutable, and secure data structure for Digital Product Passports, track-and-trace activities across the value chain, and the potential to link traceable secondary raw materials to legal contract-based rights represented by tokens. This section details the different components that make up the so-called Blockchain Module and that go beyond just the use of the protocol or specific Blockchain technology, specifying on the one hand the requirements that a solution for DPP based on blockchain should contemplate and on the other hand another defining an integral solution modeled in different layers; from the specification of a Blockchain protocol, in this case Fabric, to the different components that allow systems to interact and integrate with the Blockchain (API Services), Smart Contracts, Digital Identity services and block explorers.

In Subchapter 2.4 we find a specific Closed Loop Supply Chain Management Tool, provided by DGS, which allows users to forecast, plan, schedule, and control production and materials of the whole business supply chain both in a simulative manner and in a unique confirmed scenario. Within DigInTraCE, the CLSC Tool is being analyzed as a useful tool with a particular focus on its Scheduling and MES Modules to find the right process receipt to optimize production. A functional description of how CLSC works is provided and a technical schema concerning the architecture representing the manner in which the system interacts with Microsoft Azure services, and it explains also the grade of communication it has with the Algorithm layer.

In subchapter 2.5 introduces the novelty Mix Reality Enabler Tool (MR) developer by ICCS. MR is composed of two applications; the first is suitable for Mix Reality headsets, such as Microsoft HoloLens, and the second is suitable for mobile devices. Both applications harness Mixed/Augmented Reality technology, which augments virtual content in the real world and provide end-users with the ability to retrieve information from the DPP component of the Decentralized Traceability Platform. The interaction between the end-user and the devices is described, as well as the technologies used to build mix reality solutions (Unity3D[i], OpenXR[ii]).

For last, subchapter 2.6 introduce the Life Cycle Tool (LCA) developed by CIRCE. The platform is transparent and interoperable, with the primary objective of enabling the traceability of various industrial processes and the calculation of different environmental and circularity indicators. This is achieved using a user-friendly

interface (UI) that facilitates the recording of stages, input/output data and quantities within industrial processes.

DigInTraCE

1. Introduction

1.1. Purpose of the deliverable

D2.9 Specifications definition and architecture of digital tracing v1 (M18) outlines the essential software requirements and architecture definition for the Digital Tools that are part of DigInTraCE platform. It serves as a guiding to ensure the development of a robust, scalable, and efficient software solution that meets the needs DigInTraCE objectives.

The definition of requirements, functionalities, and a reference architecture, for each of the components, must be a reference for the technical implementation that will be carried out in other work packages (mainly WP3 and WP4).

1.2. Intended audience

This technical document is specifically designed for software developers, system architects, and technical project managers who are directly involved in the development and deployment of the DigInTraCE platform.

Additionally, each of the modules can attract the interest of specific actors, such as financial analysts and strategists who will utilize the Financial Toolkit module, providing them with a detailed overview of its capabilities, functionalities, and integration into the larger platform.

Furthermore, business executives and decision-makers within organizations dealing with secondary raw materials will benefit from the document's insights into the platform's potential to enhance operational efficiencies and market competitiveness.

1.3. Structure of the deliverable its relationship with other work packages/deliverables

This deliverable has tried to follow a classic structure of software specifications and architecture document, defining a series of common sections to be completed by each of the digital tools but at the same time allowing each digital to include the level of detail necessary for its context.

The deliverable defines a complete and exclusive chapter for the digital Tools (Chapter 2), with exclusive subchapters for each of the tools:

- Subchapter 2.1 Financial Toolkit
- Subchapter 2.2 Blockchain Module
- Subchapter 2.3 CLSC Tool
- Subchapter 2.4 MR Enabler Tool

DigInTraCE

Regarding the relationship with other WP or deliverables, as inputs deliverables have been used the next deliverables:

- D2.1 – Methodology of Traceability v1.
- D2.4 – Digital Product Passport Concept v1.
- D2.6 – Real-time information and secure data exchange requirements v1.

This deliverable will be used as input in WP3 - Sensing, Sorting Technologies, and Process Optimization Algorithms, in the tasks related to build Digital Tools, and WP4 - Digital tools and platform development.

DigInTraCE

2. DigInTraCE Technical specifications and platform architecture

2.1. Financial Toolkit module

The Financial Toolkit is crafted to provide a wide range of financial services tailored for participants in the secondary raw materials market. Its objectives are multifaceted:

- To create strategies aimed at handling the volatility in secondary raw materials prices.
- To embed foreign exchange considerations within the trading mechanisms of secondary raw materials across various currencies.
- To apply technical analysis methodologies for identifying critical pricing trends and anticipating potential price movements in the near future for secondary raw materials.
- To elucidate and illustrate the correlations between secondary raw materials and other globally traded assets, as well as significant stock indices and currency pairs.
- To conceive and detail financial derivatives linked to secondary raw materials.

The current release of this toolkit brings to the forefront several functionalities:

- **Financial charting tools** for a graphical representation of secondary raw materials: Financial charts are indispensable in the analysis of financial assets, providing a visual representation of market trends and aiding in the decision-making process. These charts simplify the identification of patterns, support and resistance levels, and allow for the comparison of different financial assets, assisting in portfolio diversification and risk management.
- **Technical analysis indicators** to identify pivotal pricing trends and predict imminent price variations: Technical analysis indicators serve as essential instruments for evaluating financial assets, offering insights into trends, market momentum, and potential turning points. These indicators, through various mathematical computations, assist in making estimations about future price actions, thereby refining the investment and trading decisions.
- **Strategies** built upon these technical indicators: Strategies that leverage technical indicators furnish a structured approach for market analysis and decision-making. These strategies, by harnessing past price data, enable the identification of market opportunities, facilitating risk reduction and the enhancement of investment returns.
- A dynamic **correlation matrix** for elucidating the interrelations between secondary raw materials and other globally traded assets: The correlation matrix is a vital analytical tool that quantifies the relationships between different market segments, aiding in portfolio diversification and risk management. By understanding these correlations, investors can make informed asset allocation decisions to optimize portfolio performance.
- A mechanism for **forecasting currency pairs** to facilitate foreign exchange considerations in transactions involving secondary raw materials across different currencies: The currency pairs prediction tool is crucial for navigating the complexities of cross-currency trading, allowing for informed decision-making regarding transaction timing and currency risk

DigInTraCE

management, thereby enhancing the profitability and efficiency of transactions in the secondary raw materials market.

Below, we delve into the system architecture and design of the Financial Toolkit. This comprehensive overview will detail the foundational structure, operational mechanisms, and design principles that underpin this sophisticated financial platform. Designed with precision to cater to the nuanced demands of the secondary raw materials market, this section aims to elucidate how the toolkit is designed within its framework. From the underlying architecture that supports robust data processing and analysis to the intuitive design enabling user-friendly navigation, each aspect is meticulously crafted to enhance financial decision-making and operational efficiency for its users.

2.1.1. System Architecture

2.1.1.1. Architectural diagram – system units

The following diagram (Figure 1) illustrates the initial architecture of the Financial Toolkit module of the DigInTraCE platform.

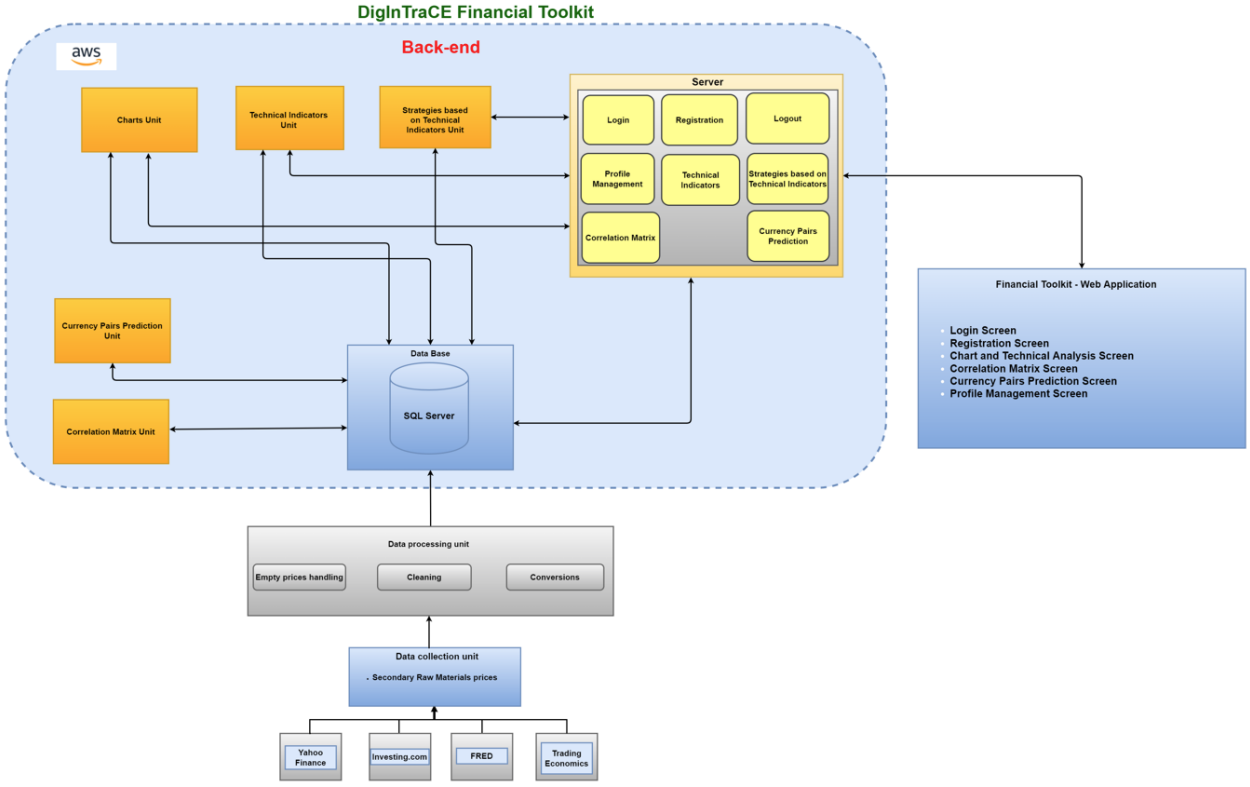

Figure 1 Financial Toolkit Architecture

The Financial Toolkit system, as depicted in the architecture diagram, consists of three subsystems, which are as follows:

- **Data Collection Subsystem**
- **Back-end Subsystem**
- **Web-based Application**

Each subsystem plays a crucial role in the overall functionality of the Financial Toolkit system, integrating seamlessly to process and analyze data effectively for currency exchange forecasting and related applications.

DigInTraCE

2.1.1.2. Brief description of the system units & communications

The subsystems and units of the Financial Toolkit system are presented in the Table below(Table 1):

Unit name	Subsystem	Functionality
Data Collection	Data Collection Subsystem	It collects the necessary data for the Financial Toolkit
Data Processing	Data Collection Subsystem	It provides the necessary actions for certain categories of data that require special handling before entering the Database
Database	Back-end subsystem	It stores and retrieves all the data used by the Financial Toolkit web application and the other units
Charts unit	Back-end subsystem	Provides the charting functionality of the Secondary Raw Materials
Currency Pairs Prediction Unit	Back-end subsystem	Provides the predictions of the major currency pairs
Correlation Matrix Unit	Back-end subsystem	Provides the calculation of the Correlation Matrix that contains Secondary Raw Materials, major stock indices, selected commodities, and currency pairs
Technical Indicators Unit	Back-end subsystem	Provides the calculations of the Technical Indicators
Strategies based on Technical Analysis Unit	Back-end subsystem	Provides the signals of the strategies based on the Technical Indicators
Server	Back-end subsystem	It serves the online web application of the Financial Toolkit system. It receives requests, communicates with the database, generates, and sends the responses.
Financial Toolkit Web Application	Web-based Application	A web application where each user can use all the functionalities of the Financial Toolkit

Table 1 Subsystems and units of the Financial Toolkit

2.1.2. Detailed description of the system units

2.1.2.1. Data collection unit

Purpose

The purpose of this unit is to collect data from various sources and send the data to other processing units and the Database. The data refer to market values of

DigInTraCE

Secondary Raw Materials, major stock indices, selected commodities, and currency pairs.

Functionality

This module is responsible for collecting data related to market prices used for the Financial Toolkit. In this version, data is collected manually. The sources that we use are:

- Yahoo Finance a widely utilized as a data source for market prices, offering extensive historical market data and real-time price updates.
- Investing.com is a comprehensive resource for market prices, providing real-time data, historical charts, and financial news.
- FRED (Federal Reserve Economic Data) is an invaluable data source for economic research, offering a vast array of economic data series, including market prices.
- Trading Economics which provides extensive data on market prices, including current Figures, forecasts, and historical data.

Data input/output

The data collection unit received the historical data in Excel format regarding the market prices of the required assets. Some of the Excel files were imported into the data collection unit and then sent to the Data Processing Unit, while others that did not require processing were directly imported into the Database.

Information flow

There is an information flow between the Data Collection Unit, the Data Processing Unit, and the Database. Specifically, data is transmitted from the Data Collection Unit to the Data Processing Unit and the Database.

2.1.2.2. Data processing unit

Purpose

The purpose of this module is to clean and transform the aforementioned stored data. Essentially, a clean final dataset must be constructed in a suitable form to be used by the Financial Toolkit.

Functionality

This specific module is responsible for the pre-processing of data that must be applied before the dataset is finalized and ready to be used by the Financial Toolkit. As a first step, the various data stored in Excel files must be integrated into a complete data frame. To accomplish this, we will merge the different data frames. Additionally, since some features are not reported on a daily basis, the forward-filling method will be used to fill in the intermediate values using the same observation until a new one becomes available. This unit will also implement the necessary normalizations to bring the data into the required format. Furthermore, even for the variables that are reported daily, there are some observations that are missing/not available (NA) and need to be handled as well.


DigInTraCE

DigInTraCE

Data input/output

The data inputs are multiple Excel files that contain the prices of all the required financial assets: Secondary Raw Materials, Commodities, Major Stock Indices and Currency Pairs.

The data output is a complete, continuous, and aligned data set with all the processes and the relevant transformations that have been applied.

Information flow

The flow of information occurs between the Data Collection Module, the Data Processing Unit, and the system's Database. Specifically, information is transmitted from the Data Collection Unit to the Data Processing Unit. Subsequently, the data preprocessing part is carried out, and then the data are forwarded to the system's Database. These data will be used by all the features and functionalities of the Financial Toolkit.

2.1.2.3. Database unit

Purpose

The Database of the Financial Toolkit serves as the central repository for all processed and raw data necessary for the system's features. Its primary purpose is to store meticulously cleaned and transformed data, which includes historical market prices, indicators, and derived variables created through processes such as normalization and technical analysis computations. This organized and accessible data storage allows for efficient retrieval of information, which is crucial for supporting real-time decision-making and the operational use of the Financial Toolkit. The Database is designed to handle a high volume of data inflows and query requests efficiently, ensuring data integrity and availability are maintained at all times. This infrastructure supports the Toolkit's overall aim to provide comprehensive, reliable financial predictions and insights, aiding in risk management, investment decision-making, and strategic planning.

Functionality

The functionality of the Database within the Financial Toolkit is multifaceted, primarily focusing on data integrity, security, and accessibility. Its core roles include:

- **Data Aggregation and Integration:** The Database collects data from various sources, including direct feeds from financial markets and economic data sources. It integrates these diverse data sets into a unified format, facilitating efficient data retrieval and consistency across the system.
- **Data Management:** It ensures the robust management of data through systematic storage, updates, and management of historical and real-time data. This management includes the application of schemas to organize data fields effectively, indexing for fast retrieval, and comprehensive backup systems to prevent data loss.
- **Support for the Financial Toolkit:** The Database is optimized to support Financial Toolkit's operations by providing processed data in formats that are readily consumable by various algorithms.
- **Query Handling and Data Retrieval:** It offers sophisticated query capabilities, allowing users to perform complex data retrievals. This functionality is vital for

analysts and financial professionals who need to extract specific datasets for detailed analysis.

- **Security Measures:** The Database implements stringent security measures to protect sensitive financial information. These measures include encryption, access controls, and audit logs, which safeguard data against unauthorized access and breaches.
- **Scalability and Performance Optimization:** It is designed to scale effectively as data volume grows. Performance optimizations such as the use of efficient querying algorithms and caching mechanisms ensure that the system remains fast and responsive under heavy loads.

Overall, the Database is a critical component of the Financial Toolkit, designed to handle the extensive demands of financial data processing and analysis, providing a reliable foundation for making informed financial decisions and predictions.

Data input/output

The Data Input/Output (I/O) functionality of the Financial Toolkit Database is crucial for ensuring that all features of the Toolkit operate efficiently and effectively. Data is primarily loaded manually from Excel files, which contain comprehensive details on financial products and market prices. Upon loading, these Excel files undergo a series of pre-processing steps designed to standardize and clean the data. This includes resolving inconsistencies, handling missing values, and converting data into a format that is uniform and easily accessible within the database.

The processed data is made accessible to all features of the Financial Toolkit. This includes various modules for data analysis all of which rely on the quality and availability of data from the Database. Efficient data retrieval mechanisms ensure that any feature of the Toolkit can access required data in real-time, enabling dynamic and responsive financial analysis.

Information flow

The information flow within the Database of the Financial Toolkit is strategically organized to ensure optimal data management, accessibility, and utility across all functions of the Toolkit.

Users manually upload data through Excel files. This is typically done for datasets that include market prices for all the required financial assets.

The information flow from server calls to the Database in the Financial Toolkit's web application is designed to handle and respond to user requests efficiently, ensuring timely and accurate data retrieval and processing. When a user interacts with the web application the server receives these requests and initiates a call to the Database. This process begins with the server parsing the request to determine the necessary data operations, such as SQL queries for fetching data.

These SQL queries are then executed against the Database, which is optimized for high performance with tables for rapid data access and complex query handling capabilities. The Database processes these queries, retrieves the required data, and performs any additional calculations or transformations as specified by the server's request. Once the data is prepared, it is sent back to the server, which then formats the data appropriately to respond to the client side of the web application.

DigInTraCE

The Database within the Financial Toolkit plays a pivotal role in facilitating the information flow to various analytical units such as charts, technical analysis indicators, strategies based on technical indicators, correlation matrices, and currency pairs prediction models. Once retrieved, this data flows into the respective units for further analysis.
For **correlation matrices**, the system computes the statistical correlation coefficients between various pairs of securities, essential for portfolio management and risk assessment.
For the **currency pairs prediction** models utilize this data to forecast future movements based on historical trends and quantitative indicators. These predictions are typically generated through machine learning models that have been trained on historical data stored in the Database.
The integration between the Database and these units is highly automated, ensuring that data flows smoothly and efficiently, thereby minimizing latency and maximizing the reliability of the outputs provided to the user.

2.1.2.4. Charts unit

Purpose

The Charts unit in the Financial Toolkit serves a critical role in visualizing data, which aids users in interpreting complex financial information swiftly and effectively. This unit's primary purpose is to provide a clear graphical representation of financial data, making it accessible and understandable for all users, whether they are seasoned analysts or novice investors.

Functionality

The chart functionality within the Financial Toolkit is designed to provide users with powerful and flexible visualizations that transform raw data into intelligible and actionable insights. This functionality encompasses the creation of a variety of chart types, such as line charts, bar charts, and candlestick charts, each suited for different kinds of data analysis and presentation needs. Users can visualize historical data trends, compare multiple data sets, and apply various technical indicators like Moving Averages, RSI, or MACD directly onto the charts. These capabilities are enhanced by interactive features that allow users to zoom in on specific time frames, pan across data points, and explore detailed information through tooltips and clickable elements. This comprehensive charting functionality is indispensable for conducting thorough market analysis, making informed investment decisions, and presenting data in an accessible format.

Data input/output

Data Input for the Financial Toolkit involves a user-initiated request for chart visualization. When a user selects a specific asset for charting, this request is sent to the server which then calls the Charts Unit. The primary steps involved are:

- User Request: Users access the Financial Toolkit web application and select an asset from the drop-down list.
- Server Processing: Upon receiving the user's request, the server forwards this request to the Charts Unit, which is responsible for gathering and processing the necessary data.
- Fetching Data: The Charts Unit retrieves daily data for the specified asset. This data includes various price points such as open, close, high, and low prices.

Data Output occurs after the processing stage, where the prepared data is visually represented and displayed to the user:

- Chart Generation: Using the processed data, the Charts Unit generates the chart based on the user's specifications. This involves plotting the data points correctly and applying any visual styles or interactive elements.
- Display to User: The final chart is then sent back to the user's interface where it is displayed as part of the Financial Toolkit web application. This chart is interactive, allowing users to zoom, scroll, and hover over data points for more detailed information.

This data input/output mechanism ensures that users receive a responsive and accurate visual analysis of financial data, tailored to their specific needs and preferences, facilitating informed decision-making based on up-to-date market conditions.

Information flow

The information flow within the Financial Toolkit for generating and displaying charts is a streamlined process designed to provide users with real-time data visualizations that are both accurate and interactive. Here's how the information flow operates from the initial user request to the final data output:

User Chart Request:

- **Initiation**: Users specify their chart preferences through the graphical user interface. This includes selecting the asset from a drop-down list.
- **Submission**: The chart request is submitted to the server, which acts as the mediator between the user interface and the backend processing units.

Server Processing:

- **Request Handling**: The server receives the request and routes it to the appropriate unit, in this case, the Charts Unit, which is tasked with chart generation.
- **Data Retrieval Call**: The server calls the Charts Unit, providing it with the necessary parameters to fetch and prepare the data.

Charts Unit Operation:

- **Data Fetching**: The Charts Unit retrieves the necessary daily data for the selected asset from the external data sources or real-time data feeds. This data typically includes prices like open, close, high, and low.
- **Data Processing**: Depending on the user's request, this data may undergo various transformations such as normalization or the calculation of technical indicators like moving averages or RSI, which are essential for certain types of charts.

Chart Construction:

- **Visualization**: The Charts Unit constructs the chart based on the processed data. It ensures that the chart reflects all user specifications regarding type, indicators, and styling.
- **Interactivity Integration**: The chart is made interactive, allowing users to manipulate the view (zoom in/out, scroll through time), and explore data points (hover for details, click for historical data).

Output Delivery:

- **Response Transmission**: The fully prepared chart is sent back from the Charts Unit to the server.

- **User Display**: The server then delivers the chart to the user's interface where it is displayed as part of the Financial Toolkit. The user can interact with the chart, receiving immediate feedback and detailed information on demand.

This information flow not only ensures efficiency and speed in handling user requests but also maintains high accuracy and reliability of the data presented, enhancing the user's analytical capabilities and decision-making processes in financial contexts.

2.1.2.5. Currency Pairs Prediction unit

Purpose

The purpose of this unit is to predict foreign exchange rates. After producing data in the appropriate format and applying feature selection models, multiple machine learning models will be trained and adapted with the goal of classifying the direction of future foreign exchange rates. The machine learning models will be of two types: numerical prediction (regression) and binary classification. Through numerical prediction, the machine learning models will predict the numerical value of the exchange rates for a given future time window. Through binary classification, the appropriate machine learning algorithms will predict, in a given future time window, the direction of the exchange rate (up or down), based on its price on the reference date.

The Currency Pairs Prediction Unit plays a pivotal role in analyzing Secondary Raw Materials by offering insights into currency fluctuations and their impact on market dynamics. Understanding the interplay between currency pairs and the prices of secondary raw materials is crucial for effective risk management and decision-making in international trade. By accurately predicting currency movements, this functionality enables users to anticipate changes in the cost of acquiring or selling secondary raw materials, thus optimizing procurement strategies, pricing models, and profit margins. Moreover, it facilitates the identification of arbitrage opportunities and helps mitigate currency-related risks, enhancing overall operational efficiency and competitiveness in the secondary raw materials market.

Functionality

This unit is responsible for executing feature selection techniques, running numerical prediction and binary classification ML algorithms, and generating predictions. Initially, the appropriate data are called from the Database, based on each time window and each exchange rate. The machine learning models that will be trained and deployed in the machine learning models unit fall into two categories: numerical prediction models that will perform numerical forecasts of exchange rates, and binary classification models that will predict the future direction of the price of exchange rates as either up or down. Once the predictions are made (either numerical or directional), some mathematical calculations will be performed, and the range calculation unit will be called. The results of the range unit, along with the predictions from the machine learning models unit, will be entered into the Database. The process described will run once a day at a predetermined time and will produce predictions for all requested timeframes, from one week to six months. The models will use the updated prices of all currency pairs through the daily data refresh unit, to produce the updated predictions.

The Currency Pairs Prediction Unit offers a comprehensive functionality that equips users with invaluable tools for informed decision-making in currency trading. Through a user-friendly interface, users can access a table displaying price ranges along with associated probabilities and cumulative probabilities for both upward and downward movements. This granular insight enables traders to gauge the likelihood of various price scenarios and adjust their strategies accordingly. By integrating statistical analysis and visualization, this functionality empowers users to navigate the complexities of currency trading with confidence, enabling them to capitalize on opportunities and mitigate risks effectively.

Data input/output

This unit is responsible for fetching the full set of data, which contains all relevant features that will be fed into the machine learning models for either numerical prediction or binary classification of currency pairs. The machine learning models will utilize specific forms and types of data (calculations, normalizations, aggregations, transformations); therefore, corresponding calls will be made to the Database for all required features of each currency pair.

The data input/output process of this module involves seamless interaction with the Database to retrieve historical data, generate predictions, and store the forecasted results back into the Database for future reference. Initially, the module retrieves historical data from the Database, including information such as past prices and other relevant metrics for the selected currency pair. This historical dataset serves as the basis for generating predictions.

Once the historical data is retrieved, the module applies predictive algorithms to analyze patterns and trends in the data. After the predictions are generated, the module stores the forecasted results back into the Database. This includes storing the predicted price ranges, probabilities, and cumulative probabilities for upward and downward movements, and any additional relevant information.

Overall, this data input/output process enables seamless integration between the Database and the predictive module, allowing for efficient retrieval, analysis, and storage of historical data and forecasted results.

Information flow

The Currency Pairs Prediction Unit is automatically called once a day at a predetermined time to generate predictions. The call is made after the data refresh through the daily data refresh unit has been completed. The data is fetched from the Database, for all the required features by the machine learning models. The machine learning models use this data to make new predictions, either numerical or directional. The results of the predictions are stored in the database and channeled to the range calculation unit to produce price ranges with corresponding probabilities, which are in turn stored in the database.

2.1.2.6. Correlation Matrix Unit

Purpose

The Correlation Matrix Unit in the Financial Toolkit primarily functions to analyze and elucidate the relationships between secondary raw materials and other financial entities such as commodities, major stock indices, and currency pairs. By

calculating the correlation coefficients, this unit provides invaluable insights into how price movements in one domain can influence or reflect changes in another. This analysis is crucial for risk management, enabling businesses and investors to strategize their portfolio diversifications effectively. Moreover, it aids in predicting market trends by identifying patterns and dependencies across different markets, which is essential for making informed investment decisions in the volatile arena of secondary raw materials and associated financial markets.

Functionality

The Correlation Matrix Unit operates by employing advanced statistical methodologies to calculate Pearson correlation coefficients, which quantitatively assess the linear relationships between secondary raw materials, various commodities, major stock indices, and currency pairs. This unit harnesses daily data integration, swiftly adapting to market fluctuations by updating correlation values continuously. Technically, the unit can handle non-stationary data through techniques such as differentiating and transformations to stabilize variance, making the correlations meaningful even under volatile market conditions. Users can conFigure the correlation window, by providing periods (days) and customizing the analysis to suit specific strategic goals.

Data input/output

Each day, data concerning prices from secondary raw materials, commodities, stock indices, and currencies are fed into the system. This integration ensures that the analysis reflects the most current market conditions. An automated service runs once a day to recalibrate and update the correlation coefficients between these assets based on the new data. The output is a dynamically updated correlation matrix that provides daily insights into the interdependencies and potential influences among the tracked financial entities. This continuous refresh cycle allows users to rely on the latest data for making informed decisions, crucial for navigating the fast-paced financial markets effectively.

Information flow

The information flow within the Correlation Matrix Unit is streamlined and efficient, facilitating quick processing and accessibility of data. Initially, raw data from various financial markets, including secondary raw materials, commodities, stock indices, and currency pairs, are collected from multiple trusted sources. This data undergoes preprocessing to clean, normalize, and prepare it for analysis, ensuring accuracy and relevance. Once a day, the data is inputted into the correlation calculation module where Pearson correlation coefficients are computed. The results are then formatted into a comprehensive correlation matrix, which saved to the Database. These outputs are accessible to users via a web-based interface or integrated financial analysis platforms, allowing for easy interpretation and strategic application in real-time decision-making processes. This efficient flow from data collection to user interface ensures that stakeholders have access to up-to-date and actionable financial insights.

2.1.2.7. Technical Indicators Unit

Purpose

DigInTraCE

The Technical Indicators Unit is a critical component of the financial toolkit designed to provide traders and investors with quantitative data used to predict future market trends. This unit helps in the identification of potential buy and sell signals and the overall market direction. The functionality of this unit is essential for technical analysis, where traders evaluate the secondary raw materials through statistics generated by market activity, such as past prices. Technical indicators such as moving averages, relative strength index (RSI), MACD, and others are vital for making well-informed trading decisions, especially in volatile markets such as those for secondary raw materials.

Functionality

The Technical Indicators Unit operates by calculating various technical indicators that help in analyzing market trends and forecasting future movements. These calculations include trend indicators like moving averages, momentum indicators like the RSI, and volatility indicators like Bollinger Bands. Each of these indicators provides different insights, for example, moving averages can help identify a trend, RSI can indicate overbought or oversold conditions, and Bollinger Bands provide a measure of market volatility. The unit integrates these indicators to perform complex predictive analyses and generate signals that can guide users in their trading decisions.

Below, we explore the functionalities and key attributes of several critical indicators used within the unit: MACD, RSI, EMAs, Bollinger Bands, and DMI (Directional Movement Index):

MACD (Moving Average Convergence Divergence)

- **Purpose**: The MACD is used to identify changes in the strength, direction, momentum, and duration of a trend in a stock's price.
- **Calculations**:
  - The MACD line is calculated by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA.
  - The signal line is the 9-period EMA of the MACD line.
  - The histogram represents the difference between the MACD line and its signal line.
- **Key Signals**:
  - **Bullish Signal**: MACD line crosses above the signal line; potential buy signal.
  - **Bearish Signal**: MACD line crosses below the signal line; potential sell signal.
  - **Over Zero**: Bullish trend strength (MACD line above zero).
  - **Below Zero**: Bearish trend strength (MACD line below zero).

RSI (Relative Strength Index)

- **Purpose**: RSI measures the speed and change of price movements to indicate overbought or oversold conditions in the price of a stock.
- **Calculation**:
  - RSI is calculated based on average gain and average loss over a specified period, typically 14 days (about 2 weeks).
  - RSI = 100 - (100 / (1 + RS)), where RS is the average gain divided by the average loss.
- **Key Signals**:

- **Overbought**: RSI above 70 indicates a potential sell opportunity as the asset may be becoming overpriced.
- **Oversold**: RSI below 30 suggests a potential buy opportunity as the asset may be undervalued.
- **Mid-line Cross**: RSI crossing above 50 can signal a potential bullish momentum, below 50 indicates bearish momentum.

EMAs (Exponential Moving Averages)

- **Purpose**: EMAs are used to identify the predominant trend direction and can act as support and resistance levels.
- **Calculation**:
  - The EMA applies more weight to the most recent data points, making it more responsive to added information.
  - Calculated continuously for different periods (6, 12, 20 days).
- **Key Signals**:
  - **Crossover**: A shorter period EMA crossing above a longer period EMA is a bullish signal, while the opposite is a bearish signal.
  - **Price Relation**: Prices above the EMA suggest a bullish trend, and prices below the EMA suggest a bearish trend.

Bollinger Bands

- **Purpose**: Bollinger Bands measure market volatility and provide a relative definition of high and low prices.
- **Calculation**:
  - The bands are formed by adding and subtracting a standard deviation calculation, which is a measure of volatility, from a moving average (typically the 20-day MA).
  - The bands expand during periods of increased volatility and contract during diminished volatility.
- **Key Signals**:
  - **Price Touching the Upper Band**: Potentially overbought market.
  - **Price Touching the Lower Band**: Potentially oversold market.
  - **Band Squeeze**: Indicates a period of low volatility and often precedes a significant price move.

DMI (Directional Movement Index)

- **Purpose: The DMI determines the strength of price movement in positive and negative directions and the overall market trend.**
- **Calculation**:
  - Includes the +DI and -DI components which measure the positive and negative movement, respectively.
  - The ADX (Average Directional Index) is derived from these and measures the overall strength of the direction (regardless of positivity or negativity).
- **Key Signals**:
  - **+DI crossing above -DI**: Indicates that positive price movement is stronger; potential buy signal.
  - **-DI crossing above +DI**: Indicates stronger negative price movement; potential sell signal.
  - **ADX Value**: A value above 25 suggests a strong trend presence, while below 20 often indicates a weak or absent trend.

DigInTraCE

Data input/output

Data Input for the Technical Indicators Unit in the Financial Toolkit involves a user-initiated request for the calculation and visualization of technical indicators. When a user selects a specific asset and sets parameters for technical analysis, this request is channeled to the server which then activates the Technical Indicators Unit. The key steps involved are:

- User Request: Users navigate the Financial Toolkit web application and choose an asset along with specific indicators (e.g., MACD, RSI) they wish to analyze from the drop-down list.
- Server Processing: Upon receiving the user's request, the server forwards this to the Technical Indicators Unit, tasked with fetching the necessary data and computing the indicators.
- Fetching Data: The Technical Indicators Unit retrieves daily data for the selected asset. This includes price points such as open, close, high, and low prices, essential for indicator calculations.

Data Output follows after processing, where the calculated indicators are visualized and presented to the user:

- Indicator Calculation: Utilizing the retrieved data, the Technical Indicators Unit calculates the requested technical indicators according to the specified parameters.
- Chart Generation: The processed indicators are then plotted on a chart, which is conFigured based on the user's preferences for better visualization and analysis.
- Display to User: The complete chart with technical indicators is sent back to the user's interface, displayed as part of the Financial Toolkit web application. This chart is interactive, allowing users to zoom, scroll, and hover over data points for more intricate details.

This data input/output mechanism ensures that users receive precise and responsive visual analyses of technical indicators, tailored to their specific analytical needs, thereby facilitating informed financial decisions based on robust, up-to-date data.

Information flow

The information flow for the Technical Indicators Unit is meticulously designed to provide real-time, accurate visualizations of technical indicators that enhance user experience and decision-making:

User Chart Request

- Initiation: Users define their technical analysis preferences via the graphical user interface, including selecting the asset and technical indicators from a drop-down list.
- Submission: This chart request is submitted to the server, serving as the intermediary between the user interface and the backend processing units.

Server Processing

- Request Handling: The server receives the request and directs it to the Technical Indicators Unit, designated for generating technical analyses.

- Data Retrieval Call: The server calls the Technical Indicators Unit, supplying it with the necessary parameters to fetch the required data and compute the indicators.

Technical Indicators Unit Operation

- Data Fetching: The Technical Indicators Unit acquires the essential daily data for the chosen asset from external data sources or real-time data feeds, including critical price points.
- Data Processing: The data is then used to calculate the specified technical indicators, such as moving averages, RSI, or MACD, depending on user selection.

Chart Construction

- Visualization: The Technical Indicators Unit constructs the chart integrating the calculated indicators. The design of the chart is reflective of all user-specified requirements regarding type, indicators, and aesthetic preferences.
- Interactivity Integration: The chart is enhanced to be interactive, allowing users to modify the view (zoom in/out), scroll through different periods, and investigate data points (details on hover, historical data on click).

Output Delivery

- Response Transmission: The fully prepared chart, featuring the technical indicators, is transmitted from the Technical Indicators Unit back to the server.
- User Display: Subsequently, the server presents the chart on the user's interface within the Financial Toolkit. Users interact with this dynamic chart, gaining instant feedback and detailed insights as required.

This structured information flow ensures not just the efficiency and rapid handling of user requests but also upholds the accuracy and reliability of the data presented. It significantly boosts the user's analytical capabilities and aids in making knowledgeable financial decisions.

2.1.2.8. Strategies based on Technical Indicators Unit

Purpose

The purpose of the Strategies Based on the Technical Indicators Unit is to operationalize the data provided by the Technical Indicators Unit by automatically generating actionable trading signals. This unit is designed to aid traders and investors by simplifying the decision-making process through the use of pre-defined and custom trading strategies that interpret technical indicators to predict market movements and identify optimal trading opportunities. The strategies encompass a variety of trading styles and preferences, focusing on maximizing returns and minimizing risks through precise buy and sell signals.

Functionality

This unit automates the application of various trading strategies which are based on technical indicators. Each strategy incorporates a set of rules that apply to

specific indicators like moving averages, RSI, MACD, etc., to determine market entry and exit points. The Strategies Unit not only computes these signals but also timestamps them to track the exact date and time each signal occurs, thus providing traders with precise and timely trading directives. The strategies that have been developed are:

MACD Strategy (Cross Over/Below 0)
Purpose: This strategy uses the Moving Average Convergence Divergence (MACD) to determine the momentum and trend direction. A buy signal is generated when the MACD line crosses above zero, indicating a bullish momentum. Conversely, a sell signal is issued when the MACD line crosses below zero, suggesting a bearish momentum.
Signal Generation:

- **Buy Signal**: MACD crosses from below to above zero.
- **Sell Signal**: MACD crosses from above to below zero.

RSI Strategy (Cross Over/Below 50)
Purpose: The Relative Strength Index (RSI) strategy uses the RSI indicator to identify potential reversals in price movement. The key threshold is 50; an RSI reading above 50 indicates potential bullish momentum, while below 50 suggests bearish momentum.
Signal Generation:

- **Buy Signal**: RSI crosses above 50.
- **Sell Signal**: RSI crosses below 50.

ADX Strategy (Cross Over/Below +DI with -DI)
Purpose: This strategy utilizes the Average Directional Index (ADX) along with the positive (+DI) and negative directional indicators (-DI) to identify the strength of a trend and potential entry points. A buy signal is typically generated when +DI crosses above -DI, reinforced by an ADX value above 20 (indicating a strong trend).
Signal Generation:

- **Buy Signal**: +DI crosses above -DI and ADX > 20.
- **Sell Signal**: -DI crosses above +DI and ADX > 20.

EMA Strategy (Cross Over/Below of EMAs)
Purpose: The Exponential Moving Average (EMA) strategy looks for crossovers between short-term and long-term EMAs to signal changes in market trends. A crossover of a shorter EMA above a longer EMA indicates a potential bullish turn, whereas the opposite suggests a bearish turn.
Signal Generation:

- **Buy Signal**: Short-term EMA crosses above long-term EMA.
- **Sell Signal**: Short-term EMA crosses below long-term EMA.

MACD/RSI Strategy
Purpose: This strategy combines MACD and RSI indicators to filter signals more effectively. The strategy calls for a MACD cross over or below zero combined with an RSI that is not in the overbought (>70) or oversold (<30) condition to reduce false positives.
Signal Generation:

- **Buy Signal**: MACD crosses above zero and RSI is not overbought.
- **Sell Signal**: MACD crosses below zero and RSI is not oversold.

Bollinger Bands Strategy
Purpose: Utilizing Bollinger Bands, this strategy issues buy or sell signals based on the price closing outside the bands. A close below the lower Bollinger Band is seen as an oversold condition (buy signal), while a close above the upper band suggests an overbought condition (sell signal).
Signal Generation:

- **Buy Signal**: Price closes below the lower Bollinger Band.
- **Sell Signal**: Price closes above the upper Bollinger Band.

MACD/ADX Strategy
Purpose: This strategy integrates the MACD and ADX indicators to pinpoint stronger trend-based movements. A MACD crossover of zero confirms a potential start/end of a trend, which is considered reliable when accompanied by an ADX reading above 20, indicating a strong trend.
Signal Generation:

- **Buy Signal**: MACD crosses above zero and ADX > 20.
- **Sell Signal**: MACD crosses below zero and ADX > 20.

Each strategy outlined above provides specific criteria based on technical indicators for generating buy and sell signals. These strategies are integral to the Strategies Based on the Technical Indicators Unit, helping traders and investors make informed decisions by pinpointing potential entry points in the market. The automation of these strategies ensures timely and accurate signal generation, which is crucial for capitalizing on market opportunities and managing risks effectively.

Data input/output

Data Input for the Strategies Based on Technical Indicators Unit involves a user-driven request to execute specific trading strategies based on predetermined technical indicators. When a user selects an asset, the predefined strategies such as crossover signals for moving averages or conditions for RSI levels, this information is conveyed to the server, which then engages the Strategies Unit. The detailed steps involved are:

- User Request: Users access the Financial Toolkit via the web application and choose an asset along with the specific strategies. These strategies are predefined and include setups like "MACD Signal Crosses" or "RSI Threshold Breaches."
- Server Processing: Upon receiving the user's request, the server forwards it to the Strategies Based on Technical Indicators Unit, which is tasked with the necessary data retrieval and computation of the strategies.
- Fetching Data: This unit retrieves the necessary daily data for the asset selected. This data typically encompasses price points such as open, close, high, and low, alongside volume statistics, which are crucial for the accurate calculation of the technical indicators that underpin these strategies.

Data Output follows the computation phase, where the results from the strategy calculations are presented to the user:

- Strategy Computation: Utilizing the fetched data, the unit computes the requested strategies. This might involve identifying crossover points for moving averages, checking RSI levels against user-defined thresholds, or other algorithmic conditions specified by the user.

- Signal Generation: Based on the computations, trading signals (buy, sell) are generated whenever the set conditions are met. These signals are plotted alongside the asset's price chart for clear visualization.
- Display to User: The resultant signals and the corresponding dates are then relayed back to the user's interface within the Financial Toolkit web application.

This input/output mechanism ensures that users receive precise and actionable trading signals based on technical indicators, customized to their strategic preferences, facilitating informed trading decisions based on dynamic, real-time data.

Information flow

The information flow for the Strategies Based on Technical Indicators Unit is designed to efficiently process user requests for trading signals derived from technical analyses, providing both accuracy and interactivity:

User Strategy Request

- Initiation: Users specify their strategy preferences through the graphical user interface. This includes selecting the asset and the desired trading strategies which are based on technical indicators.
- Submission: The strategy request is submitted to the server, which acts as the conduit between the user interface and the backend processing units.

Server Processing

- Request Handling: The server receives the request and routes it to the Strategies Based on Technical Indicators Unit, which is tasked with strategy generation.
- Data Retrieval Call: The server communicates with the unit, providing it with the parameters necessary to fetch the required data and execute the strategies.

Strategies Unit Operation

- Data Fetching: The unit retrieves essential daily data for the selected asset the Database.
- Strategy Calculation: With the data in hand, the unit calculates the strategies as per the user-defined settings. This involves complex algorithms to apply technical indicators and identify conditions for trading signals.

Strategy Visualization and Output

- Signal Visualization: The Strategies Unit constructs a table that incorporates the generated signals and the corresponding dates.

This streamlined information flow not only ensures rapid and efficient handling of requests but also maintains the reliability and precision of the data and signals provided, enhancing the user's capabilities for making well-informed trading decisions based on sophisticated technical strategies.

2.1.2.9. Server

DigInTraCE

Purpose

The Server is the unit of the Financial Toolkit that aims to handle the requests of the users of the web application.

Functionality

The functionalities that the Server will support through the requests of the Financial Toolkit users are as follows:

- **User Registration** – Users will be able to register for the service by providing the required information.
- **User Login** – Users will be able to log into the pilot application by providing their username and password.
- **Account Management** – Users will be able to change all personal details of their account, including the password, investment profile, and base currency.
- **Charts** - This functionality will provide users with the ability to see interactive charts with secondary raw materials as time series under various visualizations.
- **Technical Indicators** – This functionality will provide users with the ability to utilize various technical indicators which are fundamental tools in financial market analysis. These indicators help in predicting future market behavior based on past and current data.
- **Strategies Based on Technical Indicators** – Users will be able to see the signals of different trading strategies that are based on technical indicators. This includes strategies like moving average crossovers, RSI levels, and MACD signals among others, which help in making informed trading decisions.
- **Correlation Matrix** – This feature will allow users to view a matrix that displays the correlation coefficients between secondary raw materials and other assets. This is crucial for managing portfolios and diversifying investments to minimize risk.
- **Currency Exchange Rate Prediction** – Users will be able to request predictions of currency exchange rates, providing the exchange rate pair and the date for which the prediction is desired.

Each of these functionalities is geared towards enhancing the trading capabilities of users by providing them with sophisticated tools and data-driven insights. The server acts as a robust backend system designed to process and respond to various types of financial data requests efficiently.

Data input/output

The data input and output for the Server for each request and response are presented in the following Table 2:

DigInTraCE

Functionality	Request (Input)	Response (Output)	Error Type (if applicable)
User Registration	Required details: Name, Email, Username, Password etc.	Confirmation of registration, User ID	User already exists, Data validation error
User Login	Email, Password	Login Success/Failure, Session Token	Invalid credentials, User not found
User Logout	Session Token	Logout Success/Failure	Invalid or expired token
Account Management	Modifications: Password, Investment Profile, Base Currency etc.	Confirmation of changes, Updated Account Details	Unauthorized access, Invalid details
Charts	Secondary Raw Material and Type of chart (e.g., Solid, Area)	Chart data	Incorrect parameters, Data not found
Technical Indicators	Type of indicator requested (e.g., MACD, RSI), parameters (e.g., time period)	Indicator data	Incorrect parameters, Data not found
Strategies based on Technical Indicators	Strategy selection (e.g., MACD crossover), relevant parameters	Strategy outcome (buy/sell signals), signal timestamp	Strategy not found, Parameter error
Correlation Matrix	Request for correlation data among selected assets	Correlation coefficients, graphical matrix representation	Data unavailable
Currency Pairs Prediction	Currency pair (e.g., EUR/USD), prediction date	Predicted exchange range, Probabilities associated with ranges	Prediction error, Insufficient data

Table 2 Data input/output for the Server

This table systematically maps out how users interact with the Server via specific requests and what output they receive in response, providing a clear understanding of the data flow within the system. Each functionality utilizes different sets of data inputs to deliver precise outputs aimed at assisting users in making informed decisions based on accurate, real-time financial data.

Information flow

DigInTraCE

Here's the information flow for each functionality provided by the Server, ensuring a comprehensive understanding of how data is managed from the initial request through to the final response:

User Registration

- **Input**: The user submits registration details such as name, email, username, and password.
- **Process**: The server validates the data for completeness and uniqueness, especially checking if the username or email already exists.
- **Output**: If successful, the server creates a new user account and generates a unique user ID which is returned to the user along with a confirmation message.
- **Error Handling**: Errors such as 'User already exists' or 'Data validation error' are caught and an appropriate error message is returned.

User Login

- **Input**: The user provides their email and password.
- **Process**: The server checks these credentials against its database.
- **Output**: If the credentials are correct, the server issues a session token for maintaining the logged-in state.
- **Error Handling**: If the credentials are incorrect or the user does not exist, an error message like 'Invalid credentials' or 'User not found' is returned.

User Logout

- **Input**: The user sends a request to log out along with the session token.
- **Process**: The server validates the session token.
- **Output**: Upon successful validation, the server terminates the session and confirms the logout process.
- **Error Handling**: If the session token is invalid or expired, an error message 'Invalid or expired token' is returned.

Account Management

- **Input**: Users submit changes they want to make to their account details such as password, personal details, and base currency.
- **Process**: The server first authenticates the user via their session token, then validates and applies the changes.
- **Output**: The server confirms the changes and updates the account details accordingly.
- **Error Handling**: Errors like 'Unauthorized access', or 'Invalid details' are managed, and relevant error messages are returned.

Charts

- **Input**: The request specifies the Secondary Raw Material and the type of chart
- **Process**: The server retrieves the requested data from the system's Database.
- **Output**: The calculated chart and a graph or visual representation are returned.
- **Error Handling**: The server handles errors such as 'Incorrect parameters' or 'Data not found' by returning an appropriate error message.

Technical Indicators

- **Input**: The request specifies the type of technical indicator needed (e.g., MACD, RSI) and the parameters (e.g., time periods).
- **Process**: The server retrieves the requested indicator from the system's Database.
- **Output**: The calculated indicator values and a graph or visual representation are returned.
- **Error Handling**: The server handles errors such as 'Incorrect parameters' or 'Data not found' by returning an appropriate error message.

Strategies Based on Technical Indicators

- **Input**: Selection of a trading strategy and its relevant parameters.
- **Process**: The server applies the selected strategy using the specified indicators to generate trading signals.
- **Output**: Buy/sell signals, their strength, and the timestamps are returned.
- **Error Handling**: Errors such as 'Strategy not found' or 'Parameter error' are returned as appropriate.

Correlation Matrix

- **Input**: Request for correlation data among selected financial assets.
- **Process**: The server retrieves the correlation coefficients from the system's Database.
- **Output**: A matrix showing correlation coefficients and a graphical representation.
- **Error Handling**: The server handles 'Data unavailable' or 'Calculation error' by providing relevant error messages.

Currency Exchange Rate Prediction

- **Input**: The currency pair and the date for which the prediction is needed.
- **Process**: The server retrieves the predictions from the system's Database
- **Output**: The predicted exchange rate with price ranges and associated probabilities are returned.
- **Error Handling**: 'Service not available' or 'Insufficient data' messages are returned if the prediction cannot be made accurately.

Each functionality's flow from input through processing to output is designed to ensure the system is robust, secure, and highly responsive, providing users with reliable and actionable financial data and insights. Object of value range with corresponding probabilities

2.1.3. Financial Toolkit web application

The Financial Toolkit application will be a web application developed in React. It will offer functionalities centered around the Financial Toolkit for managing and analyzing financial data, specifically focusing on secondary raw materials and their related financial entities. The functionalities include:

- **Charts** – This tool allows users to see the Secondary Raw Materials in charts.
- **Technical Indicators** - This tool allows users to calculate various technical indicators which are used to analyze historical price action and volume to predict future movements in the financial markets.
- **Strategies Based on Technical Indicators** - Users can employ strategies that are based on the technical indicators to automate trading signals (buy/sell).

DigInTraCE

- **Correlation Matrix** - Users can request a correlation matrix that displays the correlation coefficients between selected secondary raw materials, commodities, stocks, or indices to analyze how closely they are related to one another.
- **Currency Exchange Rate Prediction** - This involves predicting future currency exchange rates using historical data, which can help in managing risks and investments more efficiently.
- **User Registration and Login: Users must register and log into the application to access personalized data and settings.**

In this document, we are making an initial recording of the requirements for the Financial Toolkit web application, following the logic of initially capturing high-level requirements (what the system does) and progressing to more detailed requirements (how the system does something), and completing the analysis with requirements for the User Interface. For each requirement, the requirement code number, the title of the requirement, the description of the requirement, and the degree of criticality of the requirement are recorded. The degree of criticality for a requirement is specified in the Table below (Table 3).

Requirement criticality	Description
Critical	This requirement is critical for the project's success. The project will not be feasible without fulfilling this requirement.
High	This requirement is of high priority, but the project can be minimally implemented without fulfilling this requirement.
Medium	This requirement is important as it provides some value, but the project can proceed without it.
Low	This is a low priority requirement or a "nice to have" feature, if time and cost permit.
Future	This requirement is outside the scope of this project and is included here for potential inclusion in a future release.

Table 3 Requirements degree of criticality

These criticality levels help prioritize development efforts and resource allocation to ensure that critical and high-priority requirements are addressed first, ensuring the system's effectiveness and efficiency right from its initial release. The categories also allow for strategic planning regarding features that can be developed if additional resources and time become available, such as those classified under "Low" and "Future" criticalities. This structured approach to requirement criticality ensures that stakeholders are aware of what functionalities are pivotal to the project's success and which ones can be deferred to later stages or updates.

2.1.3.1. System requirements

This section presents the requirements of the Financial Toolkit web-based application, divided into Functional (Table 4) and Non-functional requirements (Table 5).

DigInTraCE

ID#	Requirement Title	Requirement Description	Criticality Level
HLR1	User Registration	Enable new users to register for the application by providing necessary personal details such as name, email, and a secure password. This functionality is fundamental to creating a personalized and secure environment for managing financial activities.	Critical
HLR2	User Login	Allow users to log into their accounts using their username and password. This function must ensure security during the login process and provide error feedback on unsuccessful login attempts.	Critical
HLR3	Currency Exchange Rate Prediction	Prediction of currency exchange rate movements across multiple time horizons using big data analysis and advanced machine learning algorithms that examine cross-correlations between exchange rates and financial variables to identify trends and patterns with strong predictive power.	High
HLR4	Charts	Visual representation of the Secondary Raw Materials as time series	Critical
HLR5	Technical Indicator Calculations	Calculation of various technical indicators (e.g., MACD, RSI, Bollinger Bands) to analyze market conditions and assist in predictive assessments. These indicators will serve as the basis for generating actionable trading signals.	Critical
HLR6	Strategy Implementation Based on Indicators	Implementation of trading strategies based on technical indicators which automate the generation of buy/sell signals. Strategies will involve criteria established through historical data analysis to predict future movements accurately.	High
HLR7	Correlation Matrix Analysis	Generation of a correlation matrix that displays the correlation coefficients between different financial entities such as commodities, stocks, and currency pairs. This matrix helps in understanding the interdependencies among these variables and aids in risk management and portfolio diversification strategies.	High

Table 4 Critical and high criticality requirements

DigInTraCE

ID#	Requirement Title	Requirement Description	Criticality Level
MLR1	Real-time Data Feeds	Integrate real-time data feeds to provide up-to-the-minute financial data, which is vital for high-frequency trading strategies and day traders who require immediate information to make quick decisions.	Medium
MLR2	Alerts and Notifications	Implement customizable alerts for various market conditions and indicators to notify users of significant market events or when certain conditions are met	Medium
MLR3	Scalability Improvements	Enhancements to the system architecture to support scalability, allowing for an increase in user base and data volume without performance losses. This includes upgrading infrastructure and optimizing databases and application code to handle larger volumes of data.	Low
MLR4	Multi-language Support	Provide multi-language support to cater to users from different regions, enhancing accessibility and usability of the application by non-English speakers. This includes user interface translation.	Low
FR1	Customizable Dashboard	Offer a customizable dashboard where users can personalize their view with widgets or modules based on their preferences and priorities, such as specific currency pairs, stocks, commodities, or technical indicators they want to monitor closely.	Future
FR2	Historical Data Analysis	Enable extensive historical data analysis for back-testing trading strategies over various time periods, which helps in refining these strategies and understanding market patterns over long durations.	Future
FR3	Mobile Optimization	Ensure the web application is fully optimized for mobile devices, providing a seamless and responsive user experience on smartphones and tablets. This includes touch-optimized controls and adaptive design for different screen sizes.	Future
FR4	API Access for Third-party Integrations	Provide API access for integration with third-party applications such as personal finance management tools, advanced analytics platforms, or custom user-developed applications, allowing for extended functionality and personalization.	Future
FR5	Enhanced Security Protocols	Implement advanced security protocols including two-factor authentication, end-to-end encryption, and continuous security	Future

DigInTraCE

		audits to enhance the protection of user data and transactions, especially considering the sensitivity of financial information.	

Table 5 Medium, low and future criticality requirements

2.1.3.2. Functional requirements

The functional requirements document the intended behavior of the Financial Toolkit web-based application – precisely what the system is supposed to do. This behavior can be expressed as services, tasks, or functionalities that the system must perform.

Functional requirements should define the fundamental actions that must be performed by the system of the Financial Toolkit web-based application for the acceptance and processing of inputs and the processing and production of the corresponding outputs/results.

For functional requirements, the complete list is organized into the following groups:

- **General Features**
- **Output Format**
- **Visualization – Data**
- **Visualization – Forms**

Functional requirements are closely related to the use cases. All functionalities of a system are always developed to meet the needs of the users. This correlation will become apparent during the implementation of the web application, where for each use case, the verification of every functional requirement related to the scenario will be performed.

2.1.3.2.1. General features

ID	Name	Description
1.1.	General Features	Define the requirements related to the general functions that the web application must support.
1.1.1	Registration	The web application must ensure simple, easy, and secure registration for each user.
1.1.2	Login Permissions	The web application must allow users to log into their account by entering their email and password. Only registered users who are logged in should be able to have access to the application's functionalities.
1.1.3	Data Security	The web application must not allow the manipulation/input/update/deletion of data by external users for security reasons.
1.1.4	Email Support	The pilot application should be able to send emails for validating user registration.

Table 6 Functional requirements - General features

2.1.3.2.2. Output format

ID	Name	Description
1.2.	Application Format	Defines the requirements related to the application format.
1.2.1	Web Application	The web application should be accessible online as a compact set of web pages.

Table 7 Functional requirements - output format

DigInTraCE

2.1.3.2.3. Visualization – data

ID	Name	Description
1.3.	Data Visualization	Defines the requirements related to the extent to which the web application must display the data it records and generates.
1.3.1	Asset Prices	The web application should be able to display prices for all the assets of the Financial Toolkit: secondary raw materials, commodities, stock indices and currency pairs
1.3.2	Range of Currency Pairs Prediction Prices	The web application should be able to show a Table of predictions with a range of prices and corresponding probabilities for each user-selected date and for each currency pair.
1.3.3	Consolidated Prediction	The web application should be able to display the consolidated prediction (up/down) and the corresponding probability for each user-selected date and for each currency pair.
1.3.4	Price Charts	The web application should be able to display price charts of secondary raw materials, graphically illustrating historical prices.
1.3.5	Technical Indicators	The web application should provide various technical indicators such as Moving Averages, RSI, MACD, etc., which are used to analyze and predict market behavior.
1.3.6	Strategies Based on Technical Indicators	The web application should display the results and effectiveness of trading strategies that utilize technical indicators. This includes buy/sell signals and the timestamp.
1.3.7	Correlation Matrix	The web application should be able to display a correlation matrix that shows the correlation coefficients between different traded assets, helping users identify relationships that might affect their trading strategies.

Table 8 Functional requirements – Visualization – data

2.1.3.2.4. Visualization – forms

ID	Name	Description
1.4	Visualization - Forms	Defines the requirements related to the extent to which the web application must use alternative forms of data visualization.
1.4.1	Continuous Lines	The application should be able to display time series data as continuous lines or bar charts.
1.4.2	Tables	The application should be able to display values and predictions in table format.

Table 9 Functional requirements - Visualization – forms

DigInTraCE

2.1.3.3. Non-Functional requirements

The web application Financial Toolkit must meet all the non-functional requirements presented below:

Ease of Understanding: Understanding refers to the likelihood that a system can be easily comprehended. The more understandable a system is, the easier it will be for engineers to make changes in a predictable and safe manner. The requirement for the degree of understanding will be based on the following criteria:

- **Completeness** - The system should be provided with complete and understandable source code along with corresponding comments and supporting documents to cover all information.
- **Content / Conciseness** - The structure of the system must use the required use of abstract concepts (abstraction) and the required separation so that each task is not confused with others and any specific change affects the rest of the system to the least possible extent.
- **Clarity** - The system's code must be such that there is consistency, clear rules, good structure, comments, and underlining to greatly facilitate the reader – software engineer.

Documentation and User Instructions: The web application must be accompanied by documents for the technical and functional description of the system and the detailed installation method.

Easy Installation: The system must be able to be installed easily in a specific and selected environment that meets the technical requirements. The installation of the system should be a straightforward process for trained technicians.

Ease of Learning: The system must be developed based on the users' ability to learn easily, without misunderstandings and difficulties with navigation.

Ease of Testing: The web application must be designed in such a way that facilitates good operational testing, with a structured approach that will cover all the functionalities of the system.

Portability: The web application must be designed to be compatible with different environments and different platforms and technologies.

Ease of Changes: In applications like the Financial Toolkit web application, it is necessary to ensure that any required future changes can be implemented easily and quickly based on proper programming and required security.

Interoperability: The web application must be implemented so that the interfaces and screens are understandable to all users and follow common standards and be compatible with similar systems and applications.

Configurability: The web application must be implemented in such a way that certain changes in the system can be made based on changes in specific parameters and with minimal interventions in the heart of the system and in the source code.

DigInTraCE

Performance: The web application must have acceptable response times to the users, for all requests.

Reliability: Under certain conditions, with specific tests and within a specific time, the likelihood of there being the wrong response and the appropriate result during the execution of the use scenarios should be close to zero.

Scalability: The application must meet all the requirements to be able to accept an increased workload of users and tasks with the simple addition of resources to the system.

These non-functional requirements are crucial for ensuring that the Financial Toolkit web application not only functions efficiently but also integrate seamlessly into various systems and meets user expectations in terms of usability, performance, and reliability. These criteria help lay down the foundational principles for system design and user interaction, making the application robust and versatile for financial analysis and trading activities.

2.1.4. Financial Toolkit web application mock-ups

The following section presents a detailed overview of the mock-up designs for the Financial Toolkit's web application. These mock-ups serve as visual guides and prototypes for the various screens within the application, providing a clear and tangible representation of the user interface before actual development begins. Each mock-up has been carefully designed to meet the functional and aesthetic requirements of the toolkit, ensuring that users will find the interface intuitive and user-friendly. The designs cover critical aspects of the application, including login, registration, and visualization/tools screens. Through these visual prototypes, stakeholders can better understand the layout, navigation, and interaction dynamics proposed for the toolkit, facilitating feedback and iterative improvements to enhance usability and functionality. This section aims to bridge the gap between conceptual development and real-world application, setting the stage for a seamless transition into the development phase.

2.1.4.1. Application Login screen

The Application Login Screen mock-up presented below (Figure 2) illustrates the initial user interaction interface, designed for simplicity and security, guiding users through a straightforward login process to access the Financial Toolkit's features.

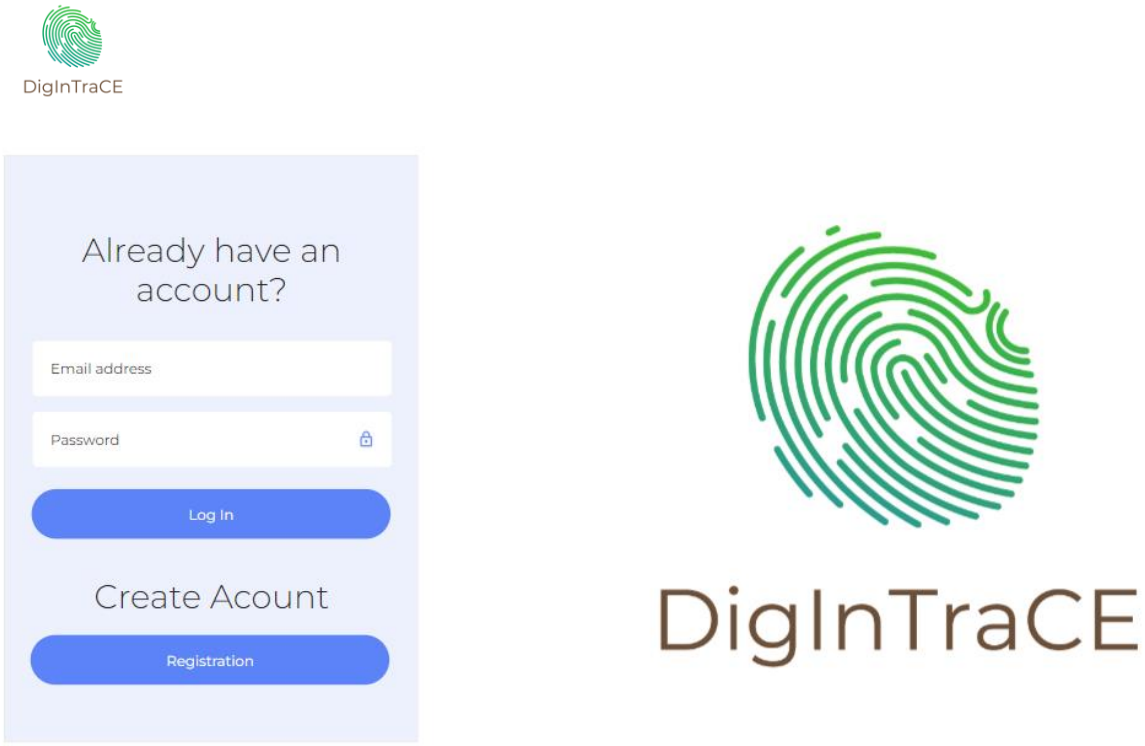

Figure 2 Web application login screen.

2.1.4.2. Registration screen

The Registration Screen mock-up displayed below (Figure 3) showcases the user-friendly interface designed to facilitate a quick and secure registration process, allowing new users to easily create accounts and access the Financial Toolkit's comprehensive features.

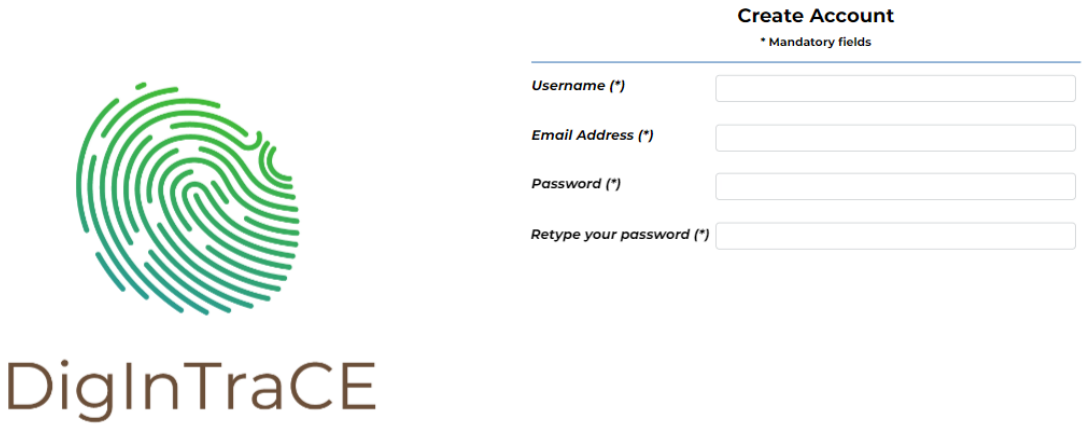

Figure 3 Web application registration screen.

2.1.4.3. Main Page screen

The Main Page mock-up, outlined below (Figure 4), serves as the hub of the Financial Toolkit, providing streamlined access to all critical functionalities, including:

DigInTraCE

1. **Hedging** - This section, to be detailed in the final version of the deliverable, will offer sophisticated tools designed for risk management and hedging strategies tailored to the specific needs of users in the secondary raw materials market.
2. **Visualization** - Featuring comprehensive screens for Charts & Technical Analysis, Correlation Matrix, and Currency Pairs Prediction facilitates in-depth financial analysis and visualization, helping users interpret complex data and market trends effectively.
3. **My Profile** - This section allows users to manage their personal information, preferences, and security settings, ensuring a personalized and secure user experience.
4. **Log Out** - Easily accessible, this function ensures users can securely exit their accounts.

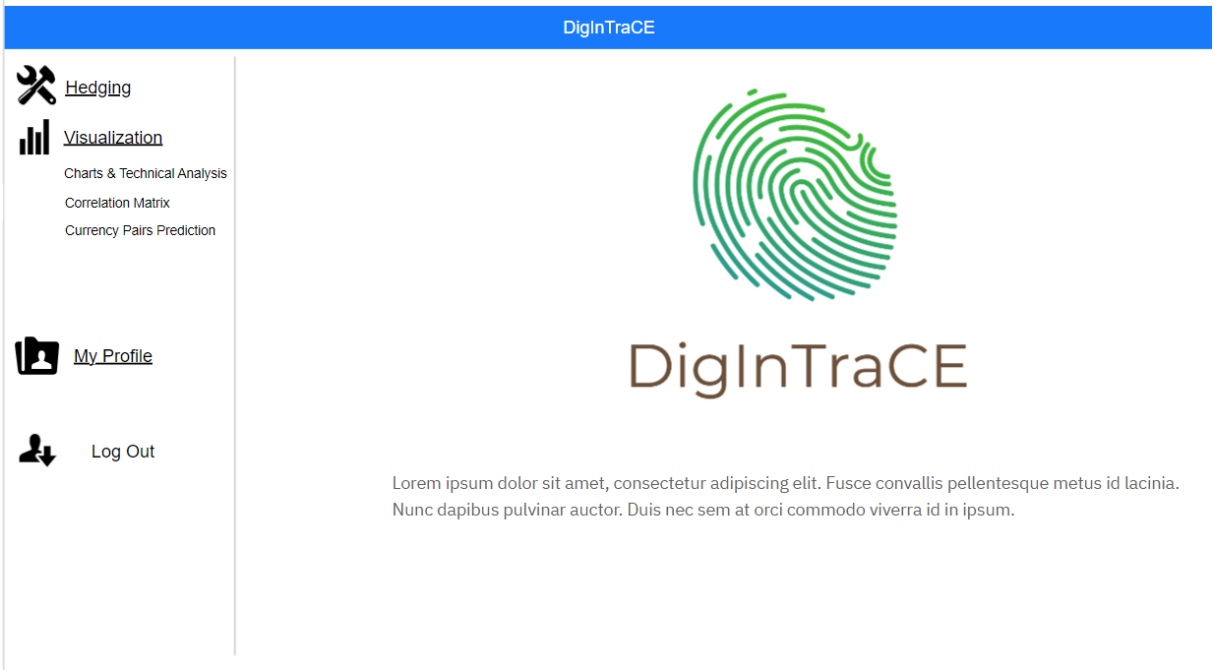

Figure 4 Web application main page screen

2.1.4.4. Visualization screen

The Visualization entity mock-ups, detailed below (Figure 5), encompass three key screens: Charts & Technical Analysis, Correlation Matrix, and Currency Pairs Prediction. Each screen is tailored to provide users with intuitive and effective tools for conducting in-depth financial analysis.

The **Charts & Technical Analysis** screen mock-up, illustrated below, is meticulously designed to empower users with comprehensive tools for financial analysis, specifically tailored for secondary raw materials. Key features of this screen include:

- **Drop-Down List for Selection of the Secondary Raw Material** - This feature allows users to easily select the specific raw material they wish to analyze, ensuring a focused and relevant analysis.
- **Chart** - A central visual display provides representation of the selected raw material, offering insights into market trends and price movements.

DigInTraCE

- **Technical Indicators** - Users can apply various technical indicators, such as MACD, RSI, or Bollinger Bands, directly on the chart to analyze market behaviors and identify trading opportunities.
- **Chart Style** - Options to customize the appearance of the chart, such as area, bar, or solid styles, cater to user preferences and enhance readability.
- **Strategies** - This section includes a detailed list of trading strategies applied to the selected secondary raw material, showing strategy name, the date that a signal occurred, and the type of signal (buy or sell), assisting users in making informed decisions based on proven analytical methods.

Figure 5 Web application Charts & Technical Analysis screen

The **Correlation Matrix** screen mock-up, displayed below (Figure6), is strategically crafted to provide users with a powerful tool for assessing relationships between various financial entities, including secondary raw materials, stock indices, commodities, and currency pairs. Key features of this screen include:

- **Selections of Secondary Raw Materials, Stock Indices, Commodities, and Currency Pairs** - This comprehensive selection tool allows users to choose multiple financial instruments for correlation analysis. Users can mix and match various categories, tailoring the analysis to their specific market focus or investment strategy.
- **Selection of Number of Periods** - Users can specify the number of periods over which the correlations are calculated, offering flexibility in assessing short-term versus long-term relationships. This feature enables users to adjust the temporal scope of the analysis to suit various trading or investment horizons, providing insights that are most relevant to their strategic needs.

DigInTraCE

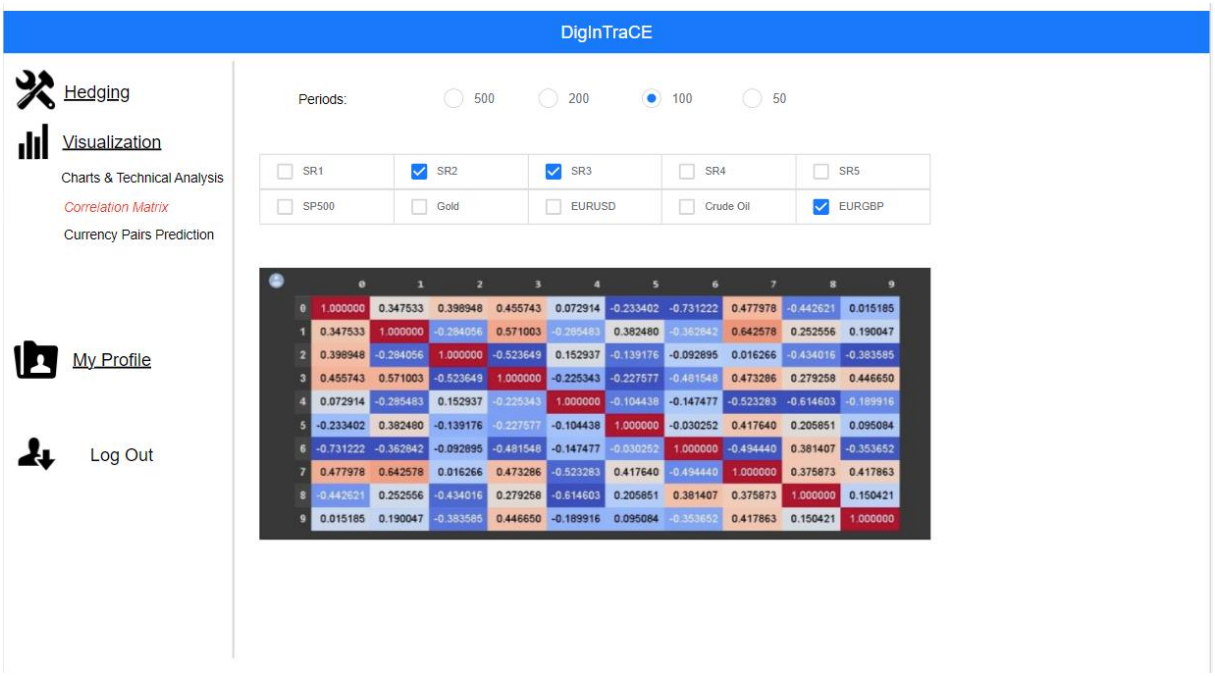

Figure 6 Web application Correlation Matrix screen

The **Currency Pairs Prediction** screen mock-up (Figure7), carefully designed within the Financial Toolkit, focuses on predictive analytics for currency trading. This screen allows users to assess future movements of currency pairs based on sophisticated modeling techniques. The main features of this screen include:

1. **Currency Pair Selection**: Users begin their analysis by selecting a currency pair from a dropdown list. This selection determines which currency exchange rates will be analyzed for potential future movements.
2. **Prediction Date Selection**: Users specify the date for which they want the predictions to apply, allowing them to focus on short-term trades or longer-term outlooks depending on their trading strategies. Allowed dates selections are from one week to six months.

Upon making these selections, the following outputs are generated:

- **Prediction Table**: This table displays a range of predicted prices for the selected currency pair at the chosen future date. Each price range is accompanied by a probability, indicating the likelihood of the currency pair reaching within that range.
- **Cumulative Probabilities**: Alongside individual probabilities for specific price ranges, the table also includes cumulative probabilities for the currency's price moving up or down from the current rate. This provides a clearer risk assessment and helps in making hedging decisions.
- **Interactive Chart**: Complementing the predictive data table , the toolkit will provide an interactive chart for the selected currency pair, allowing users to visualize potential price movements over time. This chart enhances user engagement by enabling zoom, scroll, and hover functions to examine detailed data points and trends.

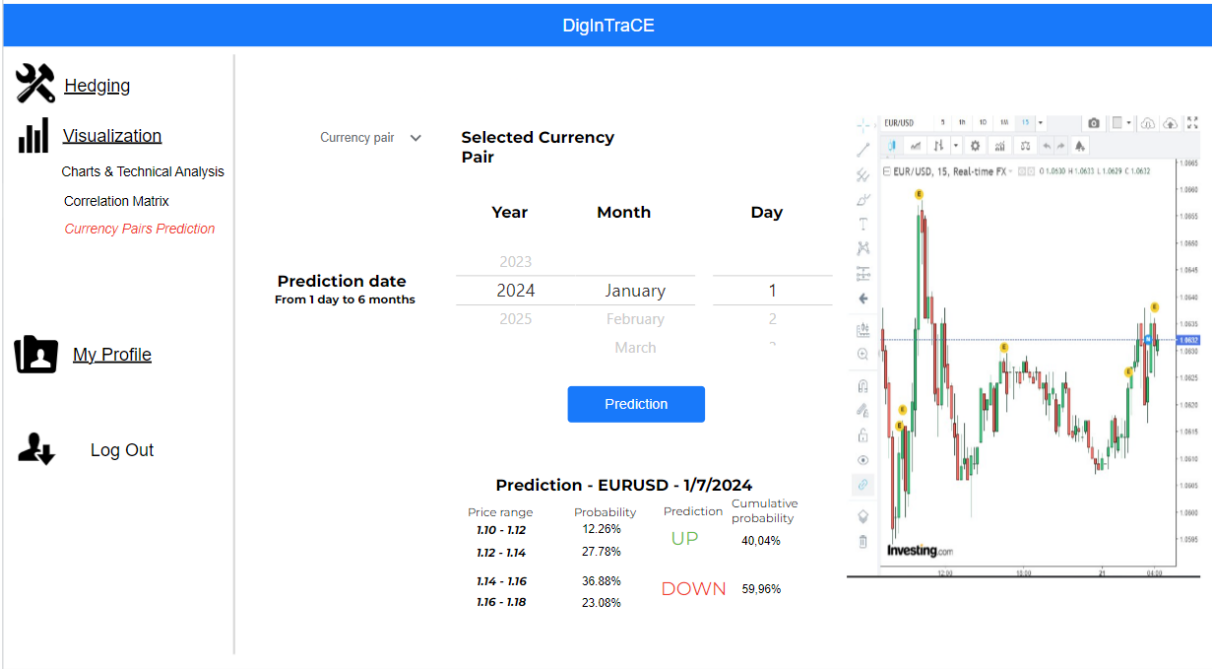

Figure 7 Web application Currency Pairs Prediction screen.

2.1.5. Technology study

The development of the Financial Toolkit web application necessitates a comprehensive technology study to ensure the selection of optimal technologies that align with the project's requirements for efficiency, scalability, and security. This technology study aims to evaluate various software, frameworks, and infrastructure solutions to construct a robust foundation for the application. The focus areas include cloud infrastructure services, programming languages, development frameworks, and associated technologies that will collectively support the functionalities planned for this financial application.

Objectives of the Technology Study

The primary objectives of this technology study are to:

1. **Identify Suitable Technologies**: Assess and identify the most suitable technologies, including backend and frontend frameworks, databases, and cloud platforms that meet the specific needs of the Financial Toolkit.
2. **Ensure Scalability and Performance**: Choose technologies that not only meet the current needs but also accommodate future expansion with minimal adjustments required, thereby supporting scalability and high performance.
3. **Promote Security and Reliability**: Prioritize technologies known for robust security features and reliability, ensuring the protection of sensitive financial data and the consistent availability of the application.
4. **Facilitate Ease of Maintenance**: Select technologies that are supported by extensive documentation and community support, which facilitate ease of maintenance and efficient problem resolution.
5. **Optimize Cost**: Consider cost-effective technologies that provide the best value for investment without compromising on the performance and capabilities required for a financial environment.

DigInTraCE

Scope of the Study

The technology study will cover the following areas:

- **Cloud Infrastructure**: Evaluate leading cloud service providers and their offerings to determine the best platform for hosting the application. The analysis will focus on comparing services such as AWS Lightsail, Google Compute Engine, Azure Virtual Machines, and DigitalOcean Droplets based on performance, scalability, cost, and ease of use.
- **Backend Programming Languages**: Examine languages that will be assessed for their performance in server-side logic, integration capabilities with databases and other services, and their ability to handle backend processes efficiently.
- **Frontend Programming Languages**: Analyse languages which are pivotal in developing interactive and dynamic user interfaces.
- **Database Management**: Compare SQL databases, considering factors such as data structure flexibility, speed, scalability, and the complexity of queries necessary for financial computations.

This study will provide a detailed analysis of each technological area, including potential benefits and limitations, to support informed decision-making. The outcome will be a well-rounded technological framework that supports the complex needs of the Financial Toolkit, ensuring it is built on a reliable, secure, efficient, and scalable architecture.

2.1.5.1. Cloud technology study

As part of the foundational infrastructure for the Financial Toolkit web application, we explored various cloud technologies to determine the most suitable platform based on reliability, scalability, ease of management, and cost-efficiency. Our selection is AWS Lightsail for hosting the application. Below (Table 10), we detail the rationale for this choice and compare it with other popular cloud services.

Overview of Cloud Service Providers

Service Provider	Core Features	Pros	Cons
AWS Lightsail	Virtual servers, managed databases, scalable storage, networking capabilities	Easy to manage, predictable pricing, integrates with AWS ecosystem	Limited configurations compared to EC2
Azure Virtual Machines	Scalable virtual machines, support for Linux and Windows	Extensive integration with Microsoft products, high scalability	Can be more expensive, steeper learning curve
Google Compute Engine	Custom and predefined machine types, per-second billing	High customizability, live migration of VM instances	Complex pricing structure

DigitalOcean Droplets	Simple virtual machines, SSD based storage, flat pricing	Straightforward to use, predictable cost, good for beginners	Less extensive services compared to others

Table 10 Overview of Cloud Service Providers

Selection Criteria for AWS Lightsail

AWS Lightsail offers several advantages that align closely with the needs of the Financial Toolkit application:

- Simplicity and Predictability: Lightsail provides a straightforward and simple pricing model that includes everything you need to start a small to medium project – compute power, memory, and storage. This predictability in cost helps in budgeting, especially for a pilot project.
- Scalability: While initially intended for simpler applications, Lightsail can scale as needed. You can increase the capacity of your instances or connect to more sophisticated AWS services as your project grows.
- Integrated Ecosystem: Lightsail is seamlessly integrated with other AWS services, providing a comprehensive environment that can expand to use additional AWS services like S3 for storage, RDS for managed databases, and more.
- Management and Maintenance: Lightsail automates some of the more routine tasks like backups (snapshots), which are essential for maintenance and disaster recovery.
- Security and Reliability: Comes with built-in networking capabilities that allow you to isolate your application using a virtual private cloud (VPC) and connect securely using AWS's extensive network.

AWS Lightsail Features for the Financial Toolkit (Table 11):

Feature	Description
Virtual Private Server	Provides a preconFigured server with SSD-based storage, DNS management, and a static IP.
Managed Databases	Supports SQL Server, MySQL, PostgreSQL, and other databases which are fully managed, relieving the burden of database administration tasks.
Automatic Snapshots	Automates the backup process without needing manual intervention, ensuring data integrity, and facilitating easy recovery from failures.
API Access	Offers API access for automation and integration with other systems or applications, enhancing functionality and user experience.
Scalable Instances	Allows upgrading or downgrading the compute, storage, and networking capacity to align with the application's requirements as it scales.

Table 11 AWS Lightsail Features

DigInTraCE

AWS Lightsail stands out as an optimal solution for the Financial Toolkit web application due to its simplicity, cost efficiency, and robust capabilities that cater well to startups and SMBs looking to leverage cloud computing without the complexity typically associated with more granular products like AWS EC2 or Google Compute Engine. Its straightforward setup, coupled with the power of the AWS ecosystem, provides a balanced platform that meets the technical and budgetary requirements of the Financial Toolkit project, ensuring a scalable, secure, and cost-effective deployment solution.

2.1.5.2. Programming languages and technologies (backend)

For the backend development of the Financial Toolkit, selecting the right programming languages is crucial to achieve optimal performance, maintainability, and scalability. After evaluating diverse options, we have chosen to use **C# as part of the .NET Core framework** for the general backend services and **Python** for all computational and algorithmic services involved in the application. Below we detail the rationale behind these choices and the specific roles each programming language will fulfill.

C# with .NET Core Framework

Selection Rationale:

- **Performance and Efficiency**: C# on .NET Core is renowned for its exceptional performance and efficiency. It is optimized for modern cloud-based applications, offering fast execution and lower resource consumption.
- **Cross-Platform Compatibility**: .NET Core provides a flexible and modular framework that is designed to run across multiple platforms including Windows, Linux, and macOS. This cross-platform support ensures that the application can be deployed on various infrastructures, thereby enhancing its accessibility and reach.
- **Robust Ecosystem**: The .NET Core ecosystem is supported by Microsoft, which ensures it is continually updated with the latest features and security enhancements. It comes with a comprehensive set of libraries and tools that simplify many programming tasks.
- **Scalability**: .NET Core's lightweight nature and stateless design make it ideal for applications that require horizontal scalability, which is a necessity for handling varying loads efficiently.
- **Community and Support**: C# and .NET Core have a large community of developers and a wealth of documentation, tutorials, and third-party resources that provide robust support for developers.

Primary Uses in Financial Toolkit:

- **Web API Development**: C# will be primarily used to build RESTful APIs that handle client-server interactions securely and efficiently.
- **Middleware Services**: Handling business logic, authentication, and data integration services that connect the frontend with various computational modules and external APIs.
- **Database Interactions**: Integrating with relational databases for CRUD (Create, Read, Update, Delete) operations, using Entity Framework for object-relational mapping which simplifies data manipulation and reduces boilerplate code.

Python

Selection Rationale:

- **Scientific Computing**: Python is a leader in scientific computing with extensive libraries for data analysis and manipulation (e.g., NumPy, pandas), which are essential for dealing with financial data.
- **Machine Learning and AI**: Python's ecosystem has the best support for machine learning and artificial intelligence with libraries like Scikit-learn, TensorFlow, and PyTorch. This is crucial for developing predictive models and algorithmic trading strategies.
- **Flexibility and Readability**: Python code is easy to write and read, making it ideal for teams that require fast development cycles and collaborative coding. Its dynamic nature allows for rapid testing and iteration which is advantageous in a fast-paced financial technology environment.
- **Integration Capabilities**: Python integrates seamlessly with other parts of the technology stack and is compatible with numerous APIs and data sources, which facilitates a comprehensive system architecture that can evolve with changing needs.

Primary Uses in Financial Toolkit:

- **Algorithmic Services**: Python will handle complex computations, statistical analysis, and processing of financial models and predictions which are critical for the toolkit's functionality.
- **Data Processing Tasks**: Utilizing Python for data cleansing, transformation, and preparation, making the data suitable for analysis or machine learning models.
- **Machine Learning Pipelines**: Developing and deploying machine learning models.

The combination of C# with .NET Core and Python offers a balanced approach, aligning with the project's needs for robust performance, ease of maintenance, and advanced computational capabilities. C# and .NET Core will provide the secure and scalable infrastructure necessary for handling web services and business logic, while Python will drive the computational and algorithmic backbone of the Financial Toolkit, ensuring high precision and efficiency in data handling and processing. This strategic selection of programming languages positions the Financial Toolkit to leverage the best of both ecosystems, facilitating future growth and technological adaptability.

2.1.5.3. Programming languages and technologies (frontend)

For the front-end development of the Financial Toolkit web application, the selection of effective programming languages and technologies is critical to ensure an intuitive user interface, responsiveness, and interactive user experience. **React** has been chosen as the primary technology due to its efficiency and robust ecosystem. This choice is bolstered by its widespread use, comprehensive support, and flexibility, making it ideal for modern web application development.

React for Frontend Development

Selection Rationale:

- **Component-Based Architecture**: React's component-based structure allows for the modular building of user interfaces, which promotes reusable code, easier maintenance, and a cohesive development process. This

architecture facilitates the development of highly scalable and sophisticated applications.

- **High Performance**: Utilizing a virtual DOM (Document Object Model), React efficiently updates user interfaces by re-rendering only the components that change, rather than reloading entire pages. This leads to impressive performance enhancements, particularly in dynamic and data-intensive applications.
- **Strong Community and Ecosystem**: Backed by Meta and a vast community of developers, React is continuously updated with the latest features and optimizations. It boasts a rich ecosystem of libraries and tools that extend its capabilities, including state management tools (Redux, MobX), routing solutions (React Router), and more.
- **JSX and Reactive Updates**: React uses JSX for templating instead of regular JavaScript. It is easier to develop and maintain complex applications with JSX due to its HTML-like syntax which makes code more readable and writing components simpler. React's reactive update mechanism ensures that the UI is consistent with the application state by managing updates in a predictable fashion.

Primary Uses in Financial Toolkit:

- **Interactive UIs**: Building dynamic and responsive user interfaces that provide real-time financial data updates, interactive charts, and responsive forms to enhance user engagement and satisfaction.
- **State Management**: Managing application state with complex interactions across multiple components using state management libraries like Redux, which integrate seamlessly with React.
- **Routing and Navigation**: Implementing smooth and intuitive navigation across the application's various features and views with the help of React Router.

Integration with Backend Services

React will integrate with the backend services developed using C# on .NET Core and Python through RESTful APIs. These APIs will facilitate the interaction between the frontend and the backend, allowing data to flow seamlessly from the server to the client-side React application and back. This integration is crucial for performing real-time data updates, processing user inputs, and managing application state effectively.

Development Tools and Additional Technologies

- **Create React App**: Utilized for bootstrapping the project, providing a robust setup for React along with a standard development environment which includes built-in Babel and Webpack configurations.
- **TypeScript**: Enhancing code quality and reliability by using TypeScript with React for static type checking, reducing runtime errors and promoting more maintainable code.
- **Styled Components**: Employing styled components to write actual CSS code to style components. This approach helps in managing styles in a component-specific manner, reducing the complexity, and enhancing maintainability.

- **Testing Frameworks**: Utilizing Jest alongside React Testing Library to write and manage unit and integration tests which ensure that components function correctly individually and in combination.

The selection of React for front-end development in the Financial Toolkit project is justified by its scalable architecture, efficient data handling capabilities, and extensive community support. Combined with C# on .NET Core and Python for backend services, React will help in building a powerful, interactive, and user-friendly web application. This integration promises a robust, maintainable, and high-performance solution that meets the complex demands of financial applications, ensuring a superior user experience and streamlined development process. This strategic choice positions the Financial Toolkit to leverage innovative technology effectively, ensuring future growth and adaptability.

2.1.5.4. Database technology study – design and implementation of the database

2.1.5.4.1. Database technologies and selection

For the Financial Toolkit web application, the choice of database technology is critical due to the need for high performance, reliability, security, and scalability. After evaluating several database systems, **SQL Server Express** has been selected as the primary database technology. This decision is supported by SQL Server Express's robust feature set, which is particularly well-suited to handling complex queries and extensive datasets typically involved in financial applications.

Overview of Database Technologies Considered (Table 12)

Database	Core Features	Pros	Cons
SQL Server Express	High performance, reliable, integrated with .NET, cost-effective	Powerful data handling, widely used, excellent integration with .NET	Scalability limits with Express edition
MySQL	Open-source, reliable, widely used	Cost-effective, flexible, strong community support	Performance can vary with larger databases
PostgreSQL	Open-source, advanced features	Supports complex queries, extensible	Less integration with .NET compared to SQL Server
MongoDB	Schema-less, NoSQL, horizontal scaling	Flexibility, high performance for unstructured data	Not ideal for complex transactions typical in financial systems

Table 12 Overview of Database Technologies

Selection Criteria for SQL Server Express

SQL Server Express was chosen based on several criteria that align closely with the operational needs and objectives of the Financial Toolkit:

DigInTraCE

1. **Cost-Effectiveness**: SQL Server Express is a free edition of Microsoft's SQL Server, which offers a substantial amount of its full capabilities without any licensing costs. This makes it an economical choice for a startup or midsize company while still providing robust database solutions.
2. **Performance and Reliability**: Known for its high performance, SQL Server Express can handle demanding database applications, which is essential for the data-intensive operations of the Financial Toolkit. It ensures data integrity and transactional reliability.
3. **Seamless Integration with .NET**: Given that the backend of the Financial Toolkit is developed using the .NET Core framework, SQL Server Express offers native compatibility and optimal integration, facilitating easier and more efficient data transactions.
4. **Scalability and Maintenance**: While SQL Server Express has limitations in terms of database size (up to 10 GB of data) and the number of cores (utilizes the lesser of 1 socket or 4 cores), it remains a scalable option for small to medium-sized applications and is typically sufficient for the initial phases of web applications.
5. **Security Features**: It includes robust security features such as backup log shipping, which are critical for financial data handling and meet compliance regulations typically required in financial environments.

SQL Server Express Features for the Financial Toolkit (Table 13)

Feature	Description
Database Size Limit	Supports databases up to 10 GB, which is adequate for the Financial Toolkit's initial data requirements and expected data growth in the preliminary stages.
Performance	Optimized for Windows and .NET environments, offering fast processing speeds for complex financial transactions and queries.
Security	Includes robust security measures such as encryption and comprehensive data protection capabilities to secure sensitive financial data.
Maintenance Tools	Comes equipped with automated tools for database maintenance, including performance tuning and monitoring functionalities.
Backup and restore	Features straightforward backup and restore functions essential for data integrity and recovery, ensuring operational continuity.

Table 13 SQL Server Express Features

SQL Server Express is an excellent choice for the Financial Toolkit owing to its powerful processing capabilities, cost efficiency, and strong integration with the .NET framework. Its features cater well to the needs of financial data management by ensuring security, reliability, and effective performance. While it presents some scalability limitations, these constraints are manageable within the projected scope and growth of the application. This database technology, combined with the application's backend and frontend technologies, will provide a comprehensive, robust, and efficient solution tailored to the complex demands of financial applications. This strategic selection ensures a scalable, secure, and cost-effective

database solution that enhances the overall performance and reliability of the Financial Toolkit.

2.1.5.4.2. Design and implementation of the database schema

The database schema for the Financial Toolkit is crafted to efficiently handle, store, and retrieve the extensive financial data necessary for the application's operations and functionalities. This schema is structured around several critical tables, each designed to cater to specific aspects and relationships inherent in financial systems. This setup within the SQL Server Express framework ensures high performance, security, and scalability. Below is the detailed design and implementation of the main tables within the Financial Toolkit's database.

Overview of Main Database Tables

The Financial Toolkit's database incorporates the following tables:

1. **AspNetUsers**
2. **Correlation**
3. **ModelsDailyData**
4. **Predictions**
5. **Currencies**

Each of these tables has been designed to serve distinct functionalities critical to the application's requirements and user interactions.

Additional System Tables for Frontend Authentication and Authorization

These tables are part of the default schema created by ASP.NET Identity for handling user authentication and authorization. They are not directly utilized in the current version of the Financial Toolkit for financial calculations but are crucial for managing access and roles:

1. **AspNetRoleClaims**
    - Stores claims related to roles. Claims are a form of user permissions or capabilities.
1. **AspNetRoles**
    - Manages roles within the application. Roles are used to group permissions and can be assigned to users.
1. **AspNetUserClaims**
    - Contains claims granted to users. Like role claims, these are used to verify user permissions.
1. **AspNetUserLogins**
    - Holds information about the user logins. This includes details of login providers (e.g., Google, Facebook).
1. **AspNetUserRoles**
    - Links users to roles, indicating which roles apply to which users.
1. **AspNetUserTokens**
    - Manages tokens for user authentication. These are used in processes such as password resets and email confirmations.

These tables collectively ensure the database supports not only the financial data needs of the Financial Toolkit but also the comprehensive security, authentication, and user management requirements essential for modern financial applications.

Each table plays a pivotal role in maintaining the integrity, security, and efficiency of the system.
In the current version of the Financial Toolkit, the tables specifically created for front-end authentication and authorization (AspNetRoleClaims, AspNetRoles, AspNetUserClaims, AspNetUserLogins, AspNetUserRoles, AspNetUserTokens) are not utilized. These tables are integral to ASP.NET Identity and are designed to support complex security models involving roles and claims-based authentication mechanisms. However, they remain part of the Database schema to facilitate potential future enhancements and integration with advanced user management and security features. Their inclusion ensures that the application infrastructure can seamlessly evolve to incorporate comprehensive authentication and authorization capabilities as needed.

Detailed Main Table Descriptions

AspNetUsers
Purpose: Manages user information for authentication and user profile settings. This table is integral for handling user logins, registrations, and storing essential user-specific settings and preferences.

Correlation
Purpose: Stores correlation data between different financial instruments. This table is crucial for risk management and investment strategies, as it helps in understanding the relationships and dependencies between various assets.

ModelsDailyData
Purpose: Contains all historical data for all assets tracked by the application. This comprehensive data repository is used for technical indicators, trading strategies, calculating statistical measures, and generating predictive models.

Predictions
Purpose: Holds predictions specifically for currency pairs. This table is central to the forecasting functionality of the Financial Toolkit, where predictions generated by the system's models are stored.

Currencies
Purpose: Maintains a list of all available currencies that users can select as their base currency. This table is crucial for supporting internationalization and providing users with personalized data presentation based on their preferred currency.

Schema Implementation Strategies

The implementation of the database schema in SQL Server Express involves several strategic considerations to optimize performance and scalability:

- **Normalization**: Each table is normalized to reduce data redundancy and improve data integrity. This ensures that updates, deletions, and inserts are less costly, and that the database remains consistent.
- **Indexes**: Appropriate indexing strategies are employed, especially on frequently queried fields such as dates in ModelsDailyData and Predictions, and asset identifiers in Correlation. Indexes speed up data retrieval processes significantly.

- **Data Types**: Choosing the correct data types for each field to minimize space usage and optimize query performance. For instance, using integers for user IDs, datetime types for timestamps, and appropriately sized varchar for textual data.
- **Concurrency Management**: Implementing concurrency controls to handle simultaneous operations by multiple users without data conflicts or performance degradation.

Security Considerations

- **Data Encryption**: Sensitive data, particularly in AspNetUsers, is encrypted using SQL Server's built-in cryptographic features to protect personal information and comply with privacy regulations.
- **Access Controls**: Proper access controls are conFigured to ensure that users can only access data relevant to their permissions. This is critical for maintaining data security and user privacy.

The database schema for the Financial Toolkit is designed to be robust, secure, and efficient, capable of handling complex financial data and operations. With SQL Server Express as the backbone, the schema supports extensive data analysis, secure data storage, and effective data retrieval mechanisms, which are pivotal for the success of any financial application. The careful design and implementation reflect the application's need for precision, reliability, and scalability in processing financial information. This database structure not only supports the current functionalities but is also adaptable for future expansions and enhancements.

Following are all the tables with their fields and their corresponding descriptions.

AspNetUsers Table Description

The AspNetUsers table is crucial for managing user data within the Financial Toolkit. It includes comprehensive details ranging from basic authentication to personal preferences and security settings. Below is the detailed description of each field in the AspNetUsers table:

Fields Overview

- **Id** *(string)*: The primary key for the AspNetUsers Table, uniquely identifying each user that is auto incremented with each new entry.
- **UserName** *(string)*: The username chosen by the user; typically used for displaying user identification within the application.
- **NormalizedUserName** *(string)*: A normalized (usually uppercased) version of the UserName, used to enforce case-insensitivity on the user's name.
- **Email** *(string)*: The user's email address.
- **NormalizedEmail** *(string)*: A normalized (uppercased) version of the Email, used for case-insensitive searches.
- **EmailConfirmed** *(bool)*: A boolean flag indicating whether the user's email address has been verified.
- **PasswordHash** *(string)*: A hash of the user's password.
- **SecurityStamp** *(string)*: A security stamp that changes whenever a user changes their password, helping to invalidate old tokens.
- **ConcurrencyStamp** *(string)*: A stamp used to handle record concurrency, ensuring data is correctly handled in asynchronous environments.
- **PhoneNumber** *(string)*: The user's phone number.

- **PhoneNumberConfirmed** *(bool)*: Indicates whether the user's phone number has been verified.
- **TwoFactorEnabled** *(bool)*: Specifies if two-factor authentication is enabled for the user, providing an additional layer of security.
- **LockoutEnd** *(DateTimeOffset?)*: The end time of when the user is locked out of their account (if applicable).
- **LockoutEnabled** *(bool)*: Indicates whether the account can be locked out after multiple failed login attempts.
- **AccessFailedCount** *(int)*: The count of failed login attempts; used for implementing lockout policies.
- **Address** *(string)*: The user's street address.
- **Address_Code** *(string)*: The postal code associated with the user's address.
- **BaseCurrency** *(string)*: The base currency used by the user for financial transactions and calculations within the application.
- **City** *(string)*: The city part of the user's address.
- **Country** *(string)*: The country part of the user's address.
- **FirstName** *(string)*: The user's first name.
- **LastName** *(string)*: The user's last name.
- **Telephone_Number** *(string)*: An additional phone number (e.g., landline) associated with the user.
- **has_LoggedIn** *(bool)*: A flag indicating whether the user has successfully logged in at least once.

Correlation Table Description

Here is the detailed breakdown of each field within the Correlation Table , designed to store the correlation coefficients between pairs of financial instruments over specific periods:

- **Id** *(int)*: A unique identifier for each record in the Correlation Table . This is the primary key that is auto incremented with each new entry.
- **Symbol1** *(string)*: The trading symbol of the first asset in the correlation pair, such as ‘Plywood'.
- **Symbol2** *(string)*: The trading symbol of the second asset in the correlation pair, such as 'GOLD’.
- **Period** *(DateTime)*: The date representing the end of the period over which the correlation was calculated. This helps in identifying when the correlation calculation was relevant and is used for historical comparison. The periods of the system are 50, 100, 200 and 500 days, since all the data are daily.
- **Correlation** *(decimal)*: The calculated correlation coefficient between the two assets. This value ranges from -1.0 to 1.0, where 1.0 indicates a perfect positive correlation, -1.0 a perfect negative correlation, and 0 no correlation at all.

ModelsDailyData Table Description

The ModelsDailyData Table plays a critical role in the Financial Toolkit by storing daily market data for various assets. This data includes open, close, high, and low prices, which are essential for performing technical analysis, back-testing strategies, and generating predictive models. Below is a detailed description of each field in the ModelsDailyData Table:

- **Asset** *(string)*: The trading symbol of the asset, such as Plywood'. This symbol is used to identify the asset for which the data is recorded.

- **Date** *(DateTime)*: The specific date for which the data is recorded. This field helps in organizing data chronologically and is crucial for time-series analysis.
- **OpenPrice** *(decimal)*: The price of the asset at the opening of the trading day. This price is critical for calculating daily price movements and technical indicators.
- **ClosePrice** *(decimal)*: The price of the asset at the closing of the trading day. Close prices are commonly used in financial analyses and are considered the standard benchmark for daily price evaluations.
- **HighPrice** *(decimal)*: The highest price at which the asset traded during the day. This data is used to assess volatility and market sentiment.
- **LowPrice** *(decimal)*: The lowest price at which the asset traded during the day. Low prices are similarly used to evaluate volatility.

Additional Table Considerations

- **Indexing**: Creating indexes on Asset and Date will enhance query performance, particularly for fetching historical trends and data slicing which are frequent operations in financial analysis applications.
- **Data Integrity**: Implementing constraints to ensure that prices (open, close, high, low) cannot be negative, reflecting valid trading data. Additionally, ensuring Date and Asset combinations are unique to prevent duplicate records.

Prediction Table Description

The Predictions Table is integral to the Financial Toolkit, specifically designed to store forecast data related to currency pairs. This table includes predictions about future price movements along with probability distributions for different price ranges, providing a comprehensive outlook used for trading decisions. Below is a detailed description of each field in the Predictions Table:

- **Id** *(int)*: The primary key for the table, uniquely identifying each prediction entry.
- **CurrencyPairId** *(string)*: Identifies the currency pair (e.g., 'EURUSD') for which the prediction is made.
- **CurrentDate** *(DateTime)*: The date on which the prediction is made.
- **PredictionDate** *(DateTime)*: The future date for which the prediction is applicable.
- **MeanValue** *(decimal)*: The predicted average (mean) price for the currency pair on the PredictionDate.
- **CurrentPrice** *(decimal)*: The current price of the currency pair at the time of prediction.

Price Range Fields

These fields represent various predicted price ranges for the currency pair:

- **PriceRangeUp1Low** *(decimal)*: The lower bound of the first upward price range.
- **PriceRangeUp1High** *(decimal)*: The upper bound of the first upward price range.
- ... (similar fields for other price ranges up to PriceRangeUp5High)
- **PriceRangeDown1Low** *(decimal)*: The lower bound of the first downward price range.

- **PriceRangeDown1High** *(decimal)*: The upper bound of the first downward price range.
- … (similar fields for other price ranges down to PriceRangeDown5Low)

Probability Fields

These fields store the probabilities associated with each predicted price range:

- **PriceRangeUp1Prob** *(float)*: The probability of the price reaching within the first upward price range.
- … (similar fields for probabilities of other price ranges up to PriceRangeUp5Prob)
- **PriceRangeDown1Prob** *(float)*: The probability of the price reaching within the first downward price range.
- … (similar fields for probabilities of other price ranges down to PriceRangeDown5Prob)

Additional Table Considerations

- **Indexing**: Creating indexes on CurrencyPairId, CurrentDate, and PredictionDate to enhance query performance, which is critical for fetching timely prediction data and analyzing historical accuracy.
- **Normalization**: Ensuring consistent formatting and case for CurrencyPairId to avoid discrepancies and facilitate reliable data aggregation and retrieval.
- **Data Integrity**: Implementing constraints to ensure all price fields are non-negative and probabilities are within the 0 to 1 range to maintain logical consistency and prevent data entry errors.

Currencies Table Description

The Currencies Table is essential for managing various currencies used within the Financial Toolkit, particularly for setting and referencing the base currency preferences of users. This table not only helps in standardizing currency data across the application but also supports internationalization and localization efforts. Below is a detailed description of each field in the Currencies Table:

- **Id** *(int)*: The primary key for the table, uniquely identifying each currency entry. This field is auto incremented for each new record.
- **Symbol** *(string)*: The international standard symbol for the currency, such as 'USD' for US Dollars or 'EUR' for Euros. This is typically a three-letter code defined by ISO 4217 which is used universally in financial and business transactions.

2.1.6. External dependencies

External dependencies are identified, pinpointing the necessary external systems and integration with other modules within the DigInTraCE platform. The Financial Toolkit requires seamless connectivity with various external systems to enhance its functionality and effectiveness. Key dependencies include data feeds for real-time market prices, historical price databases for secondary raw materials, and currency exchange rates. Additionally, integration with other modules within the DigInTraCE platform is crucial for comprehensive data analysis and decision-making. These dependencies are essential for providing stakeholders with accurate, timely, and

DigInTraCE

actionable insights, enabling them to make informed decisions and effectively manage risks.

2.1.7. Summary and conclusions

This deliverable has detailed the comprehensive architecture and functionalities of the DigInTrace Financial Toolkit module, outlining both its technical specifications and platform architecture as integral components of the broader system. Through the articulation of system units, sub-units, and the web application, this document has laid a foundational blueprint that not only aligns with the project's objectives but also enriches the user experience and operational efficiency. Below, we summarize the key points covered and conclude with reflective insights on the project's scope and future directions.

Key Points Covered

System Architecture:

- **Architectural Overview**: The system's architecture was thoroughly explained with the help of an architectural diagram that visually represents the interactions between various system units including data processing units and user interface components.
- **System Units Description**: Each system unit's purpose and functionality were detailed, emphasizing their roles in ensuring robust data handling, processing, and security within the Financial Toolkit.

System Sub-Units:

- **Detailed Functions**: The functionalities of various sub-units such as the Technical Indicators Unit and the Strategies Based on Technical Indicators Unit were explored. These descriptions included their roles in analyzing financial data through technical indicators and executing trading strategies, respectively.
- **Data Management**: Processes involving data input/output and information flow were elaborated, illustrating how data is meticulously handled to maintain integrity and relevance throughout user interactions.

Financial Toolkit Web Application:

- **Application Requirements and Mockups**: Functional and non-functional requirements were outlined along with detailed mockups of the login, registration, and visualization/tools screens, providing a clear preview of the user interface and expected functionalities.
- **Use Case Scenarios**: Will be delivered at the final version of the document.

Technology Studies:

- **Cloud and Database Technologies**: Studies on cloud solutions and Database technologies detailed the selection criteria, benefits, and implementations that support the toolkit's infrastructure, emphasizing scalability and performance.
- **Backend technologies**: The choice of programming languages for backend development was justified with an analysis of their performance, maintainability, and scalability.
- **Web Technologies**: The choice of web technologies was justified with an analysis of their compatibility and efficiency in supporting the toolkit's operational demands.

DigInTraCE

The DigInTrace Financial Toolkit exemplifies advanced software engineering designed specifically for the financial industry, including the niche sector of secondary raw materials. The strategic architectural decisions and meticulous development of its features showcase a powerful platform that delivers both high-performance and dependable financial analytics and trading tools. These tools are particularly advantageous for users dealing with secondary raw materials, providing them with precise market analysis, risk assessment capabilities, and strategic planning resources. By incorporating innovative, scalable technologies, this toolkit remains pertinent in today's tech landscape and is well-equipped for future enhancements. This ensures that it will continue to serve as a critical resource in navigating the complexities of the secondary raw materials market, facilitating more informed decision-making, and enhanced operational efficiency.

In the last version of this document, we will include comprehensive sections dedicated to the definition and modeling of financial derivatives tied to secondary raw materials. This section will aim to equip users with the tools and methodologies necessary for effective hedging strategies. It will cover diverse types of derivatives such as futures, options, and swaps that are commonly used in the financial markets to hedge against price volatility of raw materials.

This addition will not only enhance the Financial Toolkit's utility in risk management practices but also broaden its applicability to sectors heavily reliant on such materials. Users will gain insights into derivative pricing and in hedging methodologies. This will aid stakeholders from industries dealing with secondary raw materials to devise more robust financial strategies and protect against adverse price changes effectively, thereby ensuring economic stability and profitability in their operations. Through this expansion, the toolkit will provide comprehensive support for complex financial decision-making and strategy formulation, aligning with the latest financial modeling techniques and economic theories.

2.2. Blockchain Module

2.2.1. Introduction

The use of Blockchain technology in the realm of DigInTraCE project intends to provide a distributed, decentralized, transparent, immutable, and secure data structure that can be applied to both Digital Product Passports as track and trace activities related to value chain , and even to examine the potential of tying traceable secondary raw materials to rights enshrined in legal contracts (financial derivatives) represented by tokens (taking advantage the information stored into the DPP).

As previously stated in DigInTraCE deliverable *D2.4 Digital Product Passport Concept v1,* the information included in the Ecodesign for Sustainable Products Regulation (ESPR)[iii] serves as the foundation for the developments DPPs. Therefore, it is highly recommendable that the ESPR proposal is utilised as a primary reference point for both the content and IT system architecture development.

Chapter III[iv] and its articles 8-13 provide a comprehensive overview of the various aspects to be considered when implementing the Digital Product Passport under ESPR normative.

DigInTraCE

A summarized and adaptation of these requirements can be found in the next Table (Table 14):

ID#	Requirement Title	Requirement Description
espr1	Free access to data	The DPP should offer free access to data to actors along the entire value chain including customs authorities.
espr2	Open standards	This information must be based on open standards and inter-operable formats and be machine-readable, searchable, and structured.
espr3	Differentiated access	DPP will allow differentiated access depending on the type of information and typology of stakeholders.
espr4	Granularity	A DPP will be specific to the item, batch, or product model, depending on the complexity of the value chain, the size, and nature or impacts of the product.
espr5	Data carrier	DPP should be easily accessible by scanning a data carrier, such as a watermark or a QR code
espr6	Data carrier long life	The data carrier should be on the product itself to ensure the information remains accessible throughout its life cycle and comply with the ISO/IEC 15459:2015 standard
espr7	Decentralized data	DPP itself should be based on a decentralized data system set up and maintained by economic actors.
espr8	Authentication	Authentication, reliability, and integrity of the data shall be guaranteed.
espr9	Security	Product passports shall be designed and operated to ensure high security and privacy and prevent fraud.

Table 14 ESPR DPP requirements adaptation

These requirements have been considered as a reference for the design and build of the Blockchain Layer and the services that this component should provide in the context of DigInTraCE.

It is also worth noting that ESPR normative does not indicate a set of specific technologies nor an architecture to build DPP solutions. ESPR defines a set of capabilities that a DPP solution should offer, for instance, data security, access control, interoperability, etc.

The requirements can be implemented with a wide range of technical solutions and approaches, as can be clearly seen through the CIRPAS[v] project, where the deliverable *D3.1 Benchmark of existing DPP-oriented reference architectures*[vi] gathers more than 200 DPP solutions references.

These solutions can be found both in centralized and proprietary technology as well as in open and decentralized solutions using technologies such as Blockchain (both

public and private), Federated data spaces (such as Gaia-X[vii]) or technologies based on Decentralized Digital Identity (Verifiable Credentials [viii] /Decentralised Identifiers[ix]).

This plethora of approaches exemplifies the adaptability with which the requisites for implementing a DPP solution can be met. In the context of DigInTraCE, the utilisation of Blockchain technology is proposed due to its capacity to fulfil the requirements pertaining to decentralisation, data security, traceability, and even the potential for the creation of novel business models through the generation of digital assets, represented by tokens, which can be traded.

As is indicated in DigInTraCE deliverable *D2.6 Real-time information and secure data exchange requirements v1*, the term "Blockchain" encompasses a vast array of technologies, solutions, and even approaches or ways of understanding decentralized technologies (e.g., public blockchain vs. private blockchain).

This extensive ecosystem must be meticulously analysed to identify the optimal Blockchain technology that aligns with the proposed use case. For this reason, and with the inputs collected in the DigInTraCE deliverables D2.1, D2.4 and D2.6 and during the work executed under task 2.4, a set of core requirements that the Blockchain module should meet has been defined.

These requirements have been grouped into a series of topics, which are described in the next Table (Table 15):

ID#	Requirement Title	Requirement Description	Criticality Level
bc1	Verifiable identity	Each actor must have a unique and verifiable identity on the blockchain.	Critical
bc2	Secure authentication	Enable secure authentication of authorized users and devices accessing and participating in the network.	Critical
bc3	Roles and permissions	Establish roles and permissions to regulate access to information and transactions within the network.	Critical
bc4	Immutability	Data stored in the blockchain should be unalterable once registered	Critical
bc5	Transaction Recording	Each change or transaction must be recorded and accessible for authorized users.	Critical
bc6	Privacy and Confidentiality	Guarantee data privacy through encryption techniques and controlled access mechanisms.	Critical
bc7	Smart contracts visibility	Implement smart contracts that are visible only to relevant parties	High
bc8	Standards	Use recognized standards to facilitate integration with other systems	High
bc9	Interoperability	Ensure that the blockchain can communicate with legacy systems.	High
bc10	Scalability and Performance	Blockchain network should handle high transaction volumes without sacrificing performance.	High

DigInTraCE

bc11	Identity Management	Allow users to control their own identity.	High
bc12	Identity Authorization	Define roles and permissions for accessing specific functions.	High
bc13	Auditing and Compliance	Record all actions performed on the blockchain. Record all transactions transparently to facilitate auditing and compliance verification	Critical
bc14	Backup	Implement mechanisms to back up data across multiple nodes.	Critical
bc15	Recovery	Define procedures for data recovery in case of failures.	Critical
bc16	Costs and Sustainability	Optimize resource utilization to minimize operational costs and maximize sustainability	High

Table 15 General requirements to Blockchain Module.

These requirements allow for an assessment of the different Blockchain technologies that could be used to implement the requirements and decide on the platform to use, based on the degree of compliance with the criteria and needs detected for DigInTraCE.

Finally, after different work sessions, with Institute of Communication and Computer Systems (ICCS) and Imperial London Team, an initial list of functional requirements for DPP and traceability aspects have been defined.

The requirements represent a preliminary version of the capabilities that the Blockchain module should serve to DigInTraCE platform. As is typical of agile approaches[x], new requirements may be identified and incorporated in subsequent versions. The set of requirements is presented in the next Table (Table 16):

ID#	Requirement Title	Requirement Description	Criticality Level
dpp1	Data storage	The system shall independently store in the Blockchain the specific content of a DPP, and the information related to its traceability.	High
dpp2	Traceability reference	A reference shall be established in the content of a DPP to allow access to associated traceability information.	High
dpp3	Unique identifier for DPP and Traceability information	The content of a DPP and its traceability shall be retrievable independently using the same unique identifier.	High
dpp4	Unique Identifier	DPPs shall have unique identifiers established by an external agent.	High
dpp5	Sequential data creation	The content of the DPP shall be generated sequentially and incrementally during the	Critical

DigInTraCE

		manufacturing process, allowing data updates.	
dpp6	Public and private information	The content of the DPP shall contain both public information, accessible to any actor, and restricted access information.	High
dpp7	Access control	For restricted access content, a reference shall be established in the DPP content to allow authorized actors to access such information.	High
dpp8	Integrity of restricted access data	A reference shall be established in the DPP content to validate the integrity of restricted access data.	High
dpp9	DPP modular content	The content of the DPP shall consist of different sections (general information, circularity, sustainability, etc.) that shall be completed at various stages of the process.	Critical
dpp10	Data control	Each section of the DPP shall be completed only by authorized actors.	Critical
dpp11	DPP Transaction identification	Each time information is recorded, a transaction shall be generated whose identifier can be added to the DPP content.	Medium
dpp12	DPP Data format	A mechanism shall be established to retrieve and compose the complete information of a DPP stored in the Blockchain in a machine-readable and interoperable format.	Critical
dpp13	Traceability model	Information related to the traceability of DPP components shall be sent to the Blockchain in EPCIS 2.0 format using appropriate events in each situation.	High
dpp14	Traceability access	Traceability information shall be stored in the Blockchain in a manner that allows retrieval in a compatible EPCIS 2.0 format.	High
dpp15	Traceability Transaction identification	Each time information is recorded, a transaction shall be generated whose identifier can be added to the traceability content.	Medium
dpp16	Traceability Data format	A mechanism shall be established to retrieve and compose the complete traceability information stored in the Blockchain in a machine-readable and interoperable format.	Critical

Table 16 Functional requirements for DPP - Blockchain module

2.2.2. Architecture overview

In the context of the DigInTraCE project, a Blockchain Module is proposed to offer a distributed, decentralised, transparent, immutable, and secure data structure that allows for the offering of higher levels of security (in comparison to centralised approaches) and transparency on the information and data that will be managed through the implementation of DPP.

DigInTraCE

The design of the Blockchain Module must satisfy the requirements mentioned and offer functionalities to allow the complete operation of the DPP (creation, reading, modification and deletion) while maintaining the inherent properties of DLT technologies (security, immutability, transparency, etc.). Furthermore, the Blockchain Module must offer functionalities to manage the traceability of events that occur during the production chain and that will be associated with a DPP.

Note: Features and functionalities related to financial derivatives tokenization will be present in the last version of this deliverable (D2.10).

2.2.3. Components and internal relationships

As with the Open Systems Interconnection model (OSI)[xi], where each of the seven layers performs a specific function in the communication process, a Blockchain solution can be composed of different layers with defined and limited responsibilities as is represented in the Figure 8:

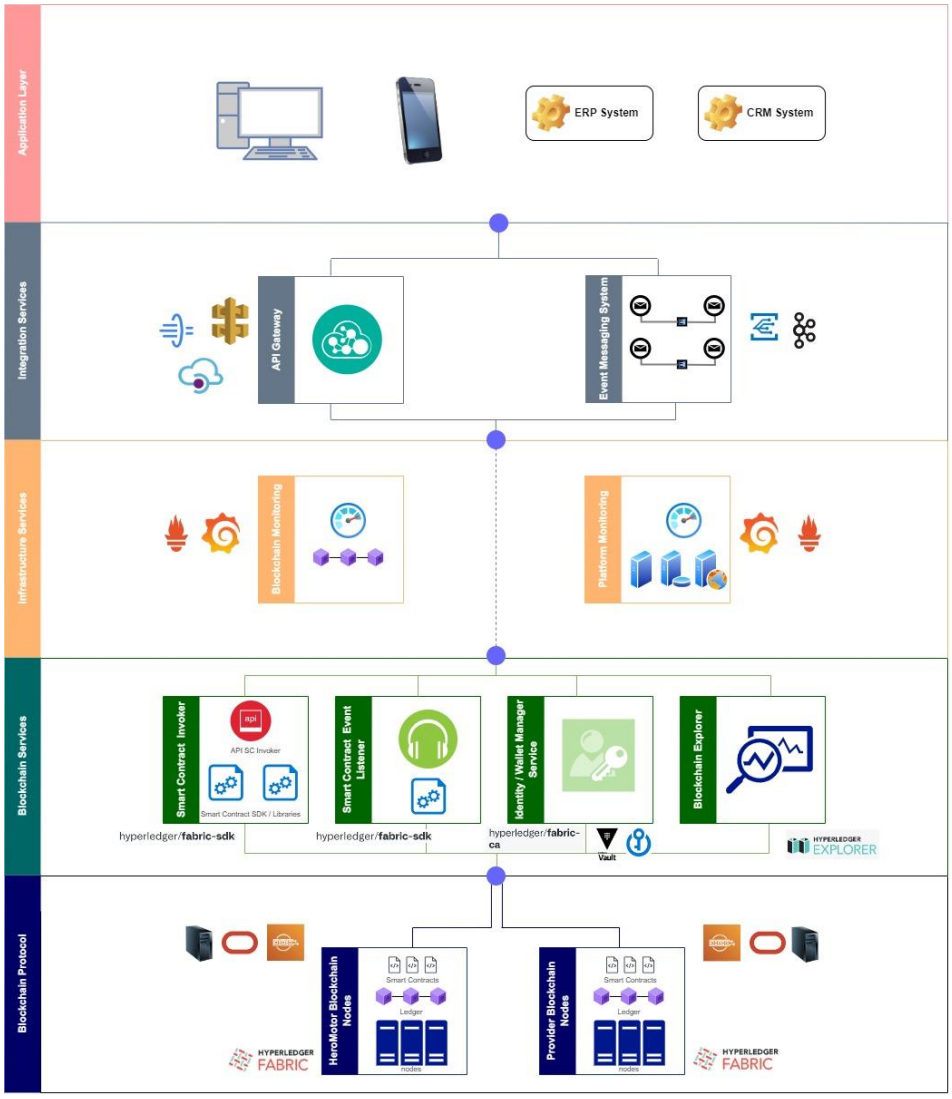

Figure 8 Blockchain Solution Layers

DigInTraCE

The definition of this layered model can be applied to both public and private protocols and allows to identify the components to be developed and adapt them to specific technologies.

In the model, the following layers are found:

Blockchain Protocol:

It refers to a specific blockchain technology protocol. This protocol could be based on public or permissioned blockchains, depending on the use case and other requirements such as privacy concerns, performance, etc.

Main components are:

- **Smart Contracts:** Smart contracts are self-executing computer programs that operate on a blockchain. These digital contracts are designed to automate, execute, and enforce agreements or transactions in a transparent and secure manner, without the need for intermediaries.
- **Ledger:** is an immutable and distributed record of all transactions made on a blockchain. It functions as a decentralized and transparent database that chronologically records each transaction, ensuring the integrity and reliability of the stored data.
- **Nodes:** are devices or computer systems that are part of a blockchain network. Each node holds a copy of the ledger and participates in the validation and verification of transactions.

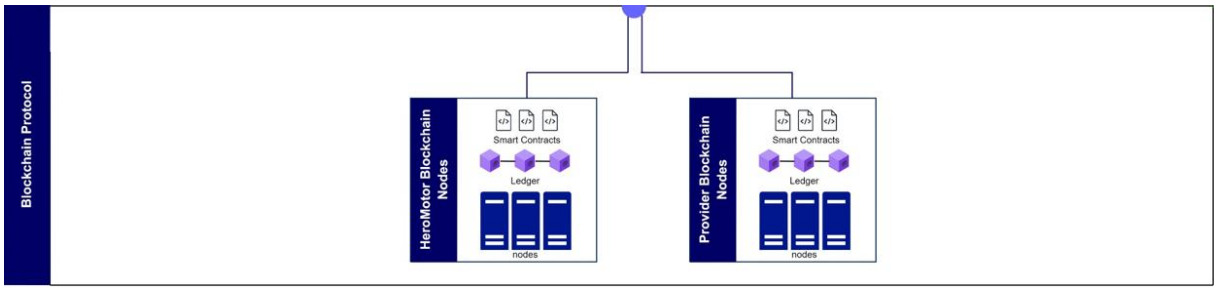

Figure 9 Blockchain Protocol

Blockchain Services:

It refers to a set of services necessary to interact with the blockchain protocol components:

- **Smart Contract Invoker:** are pieces of software that use SDKs (Software Development Kits) or libraries to execute Smart Contracts' functions from outside the blockchain. This allows other components to interact with the blockchain programmatically.
- **Smart Contract Event Listener**: Like Smart Contract Invoker, this software utilises SDK or libraries to establish a connection with the Blockchain Protocol and specific Smart Contracts. This is typically achieved using WebSocket technology, which enables it to listen to the events raised by a method of Smart Contract.

DigInTraCE

- **Identity/Wallet Manager**: is a software application or hardware device that allows users to securely store and manage the cryptomaterial needed for interacting with a Blockchain. Depending on Blockchain Protocol and technology, this cryptomaterial can be cryptographic keys (PKI[xii]), digital certificates (X.509), etc.

- **Blockchain Explorer**: is a web-based tool or application that allows users to browse and interact with data stored on a blockchain. It provides a user-friendly interface to explore the contents of individual blocks, transactions, and addresses within the blockchain network. Block explorers typically display information such as transaction history, block height, block timestamp, transaction confirmations, and details of the parties involved in each transaction.

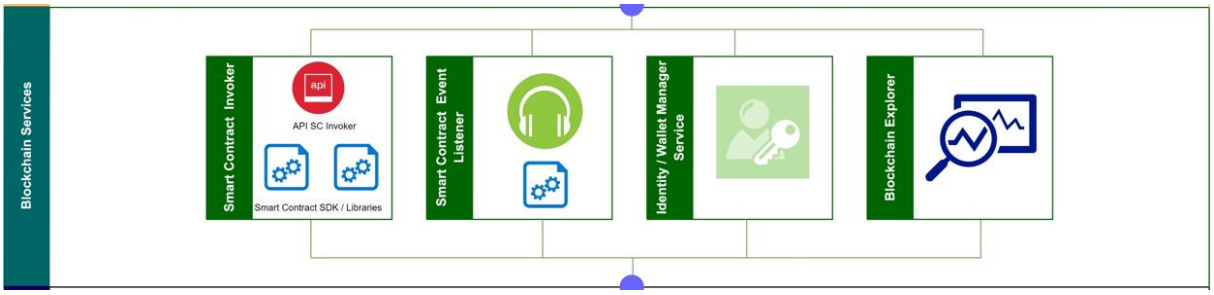

Figure 10 Blockchain Services

Infrastructures Services:

This refers to the services provided by the Infrastructure Layer that allow the status of components to be monitored and logged, traced, alerted, and metrics to be collected about all the running services in a system:

- **Blockchain monitoring:** this refers to monitor Blockchain Nodes status, block production, transaction performance, network latency, etc.

- **Platform monitoring**: this focusses to monitor components such as containers, application servers, etc.

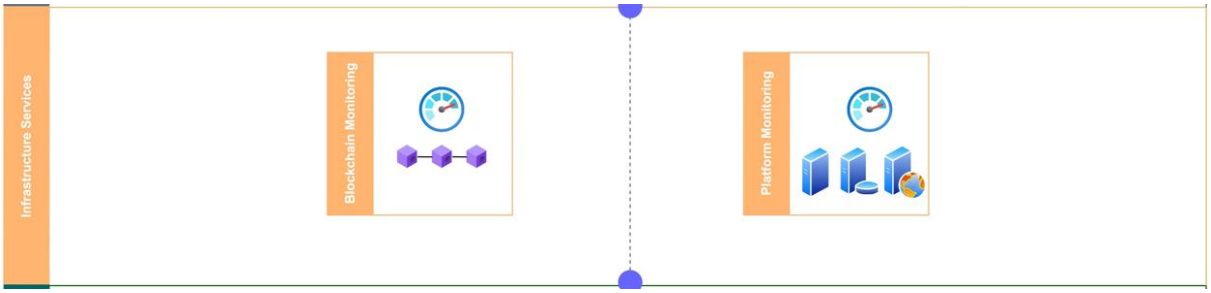

Figure 11 Infrastructure Services

Integration Services:

It refers to the services that allow to connect enterprise applications with the Blockchain ecosystem (Blockchain Services and Protocol) adding security functionalities, request routing, balancing, etc:

DigInTraCE

- **API Gateway**: is a centralized service that acts as an entry point for client requests to access backend services or APIs. It serves as a reverse proxy that receives incoming API calls, applies various policies and transformations, and forwards requests to the appropriate backend services. API gateways are responsible for managing authentication, authorization, rate limiting, caching, logging, monitoring, and other cross-cutting concerns, providing a unified interface for clients to interact with multiple services.
- **Event Messaging System**: allows applications to communicate asynchronously via events, facilitating decoupled, scalable, and resilient architectures.

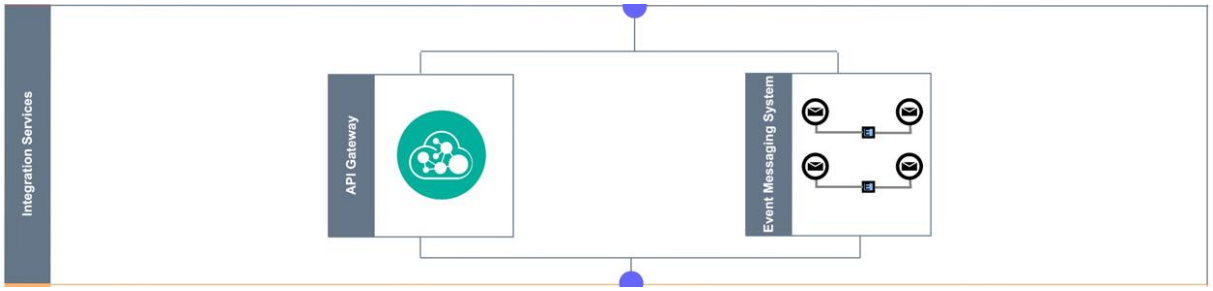

Figure 12 Integration Services

Application Layer:

It refers to final applications such as CRM, ERP, Web applications, or others that need to communicate and interact with the Blockchain to participate in a specific process executing operations, querying information, etc...

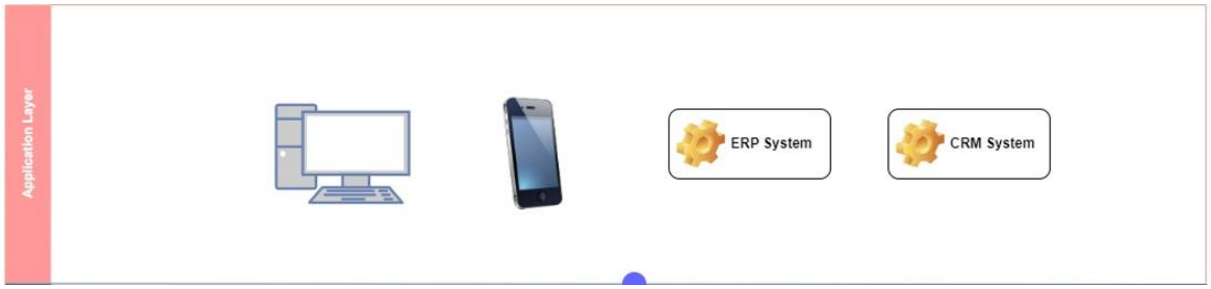

Figure 13 Application Layer

2.2.3.1. Components diagram

Using the C4[xiii] model for visualizing software architecture, the representation of the components and their relationships, the Blockchain module and the connections with other systems that are part of the DigInTrace Platform can be seen in Figure 14.

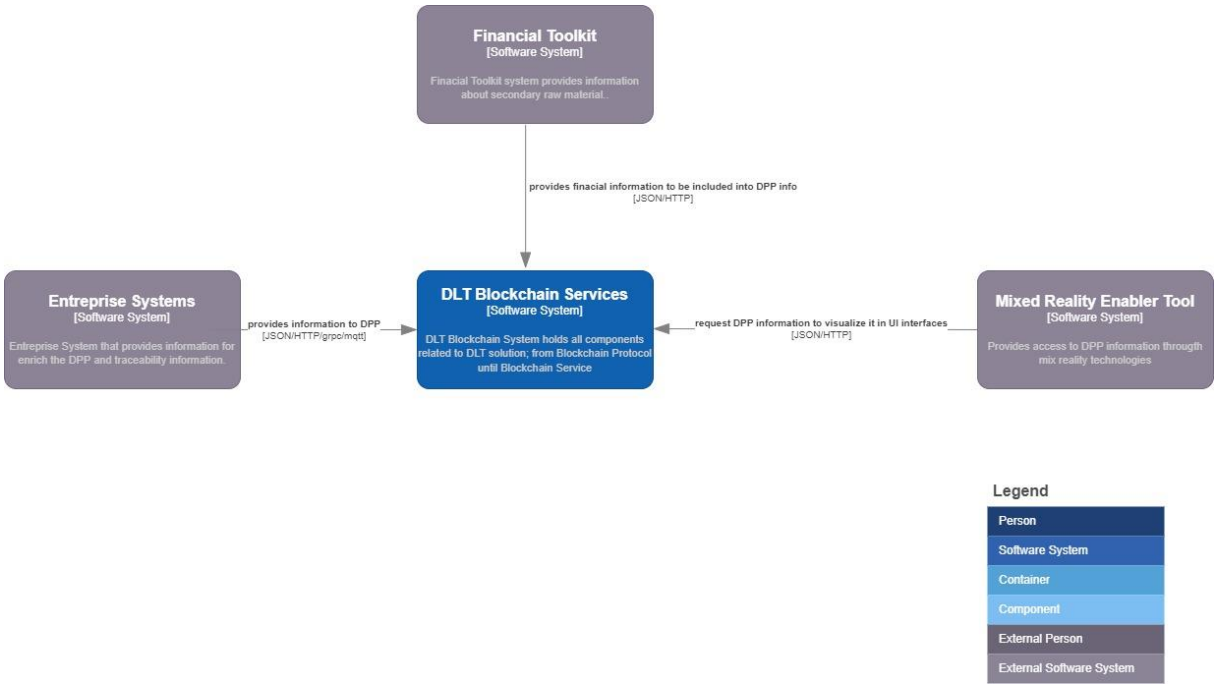

Figure 14 C4 System Diagram Context

A container diagram is used to zoom into the scope of the software system, showing the high-level technical building blocks, which are represented in Figure 15:

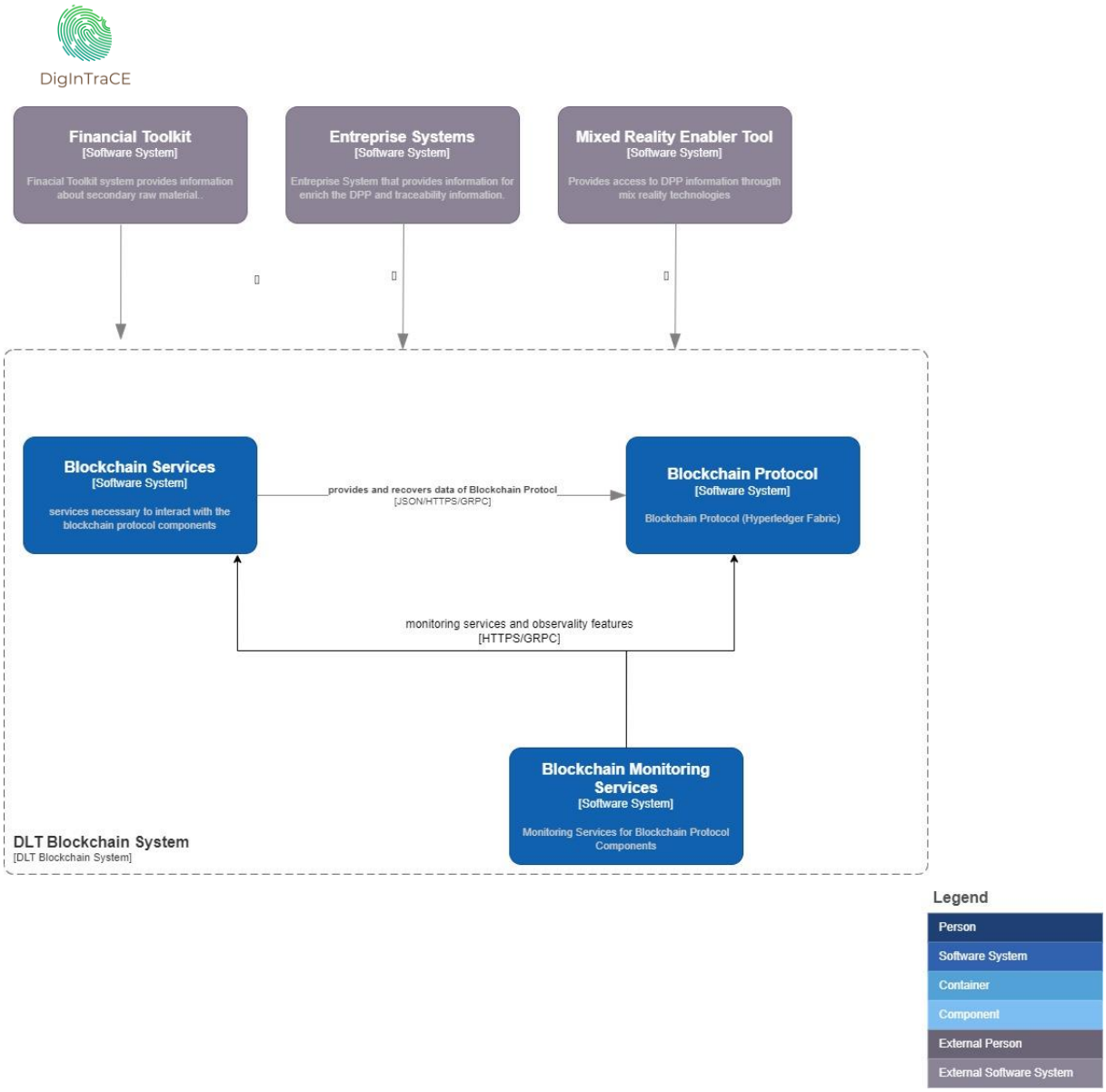

Figure 15 C4 System Diagram Container

Following an architectural style based on Microservice, instead of monolithic application, a set of components have been defined that interacting with the Blockchain Protocol selected for DigInTraCE platform. These components can be mapped into the Blockchain Layer Model mentioned in 2.3.3.

2.2.3.2. Blockchain Protocol

2.2.3.2.1. Hyperledger Fabric

Hyperledger Fabric[xiv] is an open-source enterprise-grade permissioned distributed ledger platform for developing modular blockchain applications and solutions with a high degree of confidentiality, resiliency, flexibility, and scalability.

It is designed to be a foundation for developing blockchain-based applications or solutions with a modular architecture, allowing components, such as consensus and membership services, to be plug-and-play.

Hyperledger Fabric leverages a permissioned network model, where participants must be explicitly granted access to the network, providing a higher degree of control over confidentiality and data privacy.

Hyperledger Fabric consists of several key components:

- **Peers**: They are nodes within the network that maintain a copy of the ledger and execute chaincode (smart contracts in Hyperledger terminology). There are two types of peers: endorsing peers, which simulate and endorse transactions, and committing peers, which validate and commit transactions to the ledger.
- **Ordering Service:** It is responsible for packaging transactions into blocks, maintaining the order of transactions, and delivering them to the peers for validation and commitment to the ledger. It ensures that all peers in the network have a consistent view of the transaction order.
- **Channels:** They are private subnetworks within the main blockchain network where specific subsets of network participants can conduct private and confidential transactions. Each channel has its own ledger, smart contracts, and access control policies, allowing for data segregation and privacy.
- **Fabric-CA (Certificate Authority):** It is a certificate authority service responsible for managing identities, authentication, and access control within the network. It issues X.509 certificates to network participants, such as peers, clients, and administrators, enabling secure communication and transaction authorization.

It also supports smart contracts (known as chaincode) written in various programming languages, that encapsulates the business logic and rules governing transactions within the blockchain network, allowing for decentralized execution and verification.

Once deployed to the network, a chaincode is immutable, meaning its code cannot be altered or tampered with. This ensures the integrity and security of transactions over time.

Chaincode specifies an endorsement policy that dictates which network peers must endorse a transaction before it can be validated and committed to the ledger. This policy ensures that transactions meet the required criteria for execution.

Lastly, a chaincode defines functions (or methods) that can be invoked by network participants to interact with the ledger, query data, and execute transactions. These functions are invoked using a standardized interface provided by the Hyperledger Fabric APIs.

Hyperledger Fabric has the capabilities to meet the requirements indicated in Table 15 (General requirements to Blockchain Module). The following section outlines how Hyperledger Fabric addresses each of the requirements (Table 17):

Requirement	Functionality
bc1 Each actor must have a unique and verifiable identity on the blockchain.	Hyperledger Fabric uses digital certificates to identify the actors involved in the network.

DigInTraCE

bc2 Enable secure authentication of authorized users and devices accessing and participating in the network.	Each actor generates its own certificate that it must use to operate against the network. Additionally,
bc3 Establish roles and permissions to regulate access to information and transactions within the network	ACL and other mechanisms can be established at different levels; network, channels, chaincode.
bc4 Data stored in the blockchain should be unalterable once registered.	Data is stored using Blockchain data structure and replicated into the peers that conform to the network.
bc5 Each change or transaction must be recorded and accessible for authorized users.	Every action in Hyperledger Fabric generates a transaction that is added to the ledger.
bc6 Guarantee data privacy through encryption techniques and controlled access mechanisms.	Privacy can be set at several levels; first, access to the network is restricted only to authorized members. Additionally, other mechanism can be used, like channel or private data collections.
bc7 Implement smart contracts that are visible only to relevant parties.	SmartContract are deployed in specific channels. The channels are formed with only authorized members.
bc8 Use recognized standards to facilitate integration with other systems.	The data are stored in the ledger using formats like JSON, so, can be used for third-party system.
bc9 Ensure that the blockchain can communicate with legacy systems.	Hyperledger Fabric provides SDK implemented in several languages (java, node, go) that can be used to connect with legacy systems.
bc10 Blockchain network should handle high transaction volumes without sacrificing performance	Hyperledger Fabric offers a high volume of transactions (TPS). Although it depends on configurations and the resources, more than 1000 tx/seg can be achieved.
bc11 Allow users to control their own identity.	Hyperledger Fabric CA can be used with own digital certificates emitted by the organization.

DigInTraCE

bc12 Define roles and permissions for accessing specific functions.	Hyperledger Fabric Administration is assigned to specific roles.
bc13 Record all actions performed on the blockchain. Record all transactions transparently to facilitate auditing and compliance verification.	All actions (not only chaincode execution) are stored in the ledger.
bc14 Implement mechanisms to back up data across multiple nodes.	Hyperledger Fabric is a decentralized platform; therefore, several peers should be involved. The backup process is a continuum activity.
bc15 Define procedures for data recovery in case of failures.	Hyperledger Fabric is a decentralized platform; therefore, several peers should be involved. The backup process is a continuum activity.
bc16 Optimize resource utilization to minimize operational costs and maximize sustainability.	Hyperledger Fabric doesn't use Proof of Work or other intensive consume energy mechanisms. The components can be deployed in several ways; from on-premises to cloud environments and optimize the resources.

Table 17 Hyperledger Fabric functionalities

Continuing with the representation using the C4 model, the components that make up the Blockchain Protocol can be seen in Figure 16:

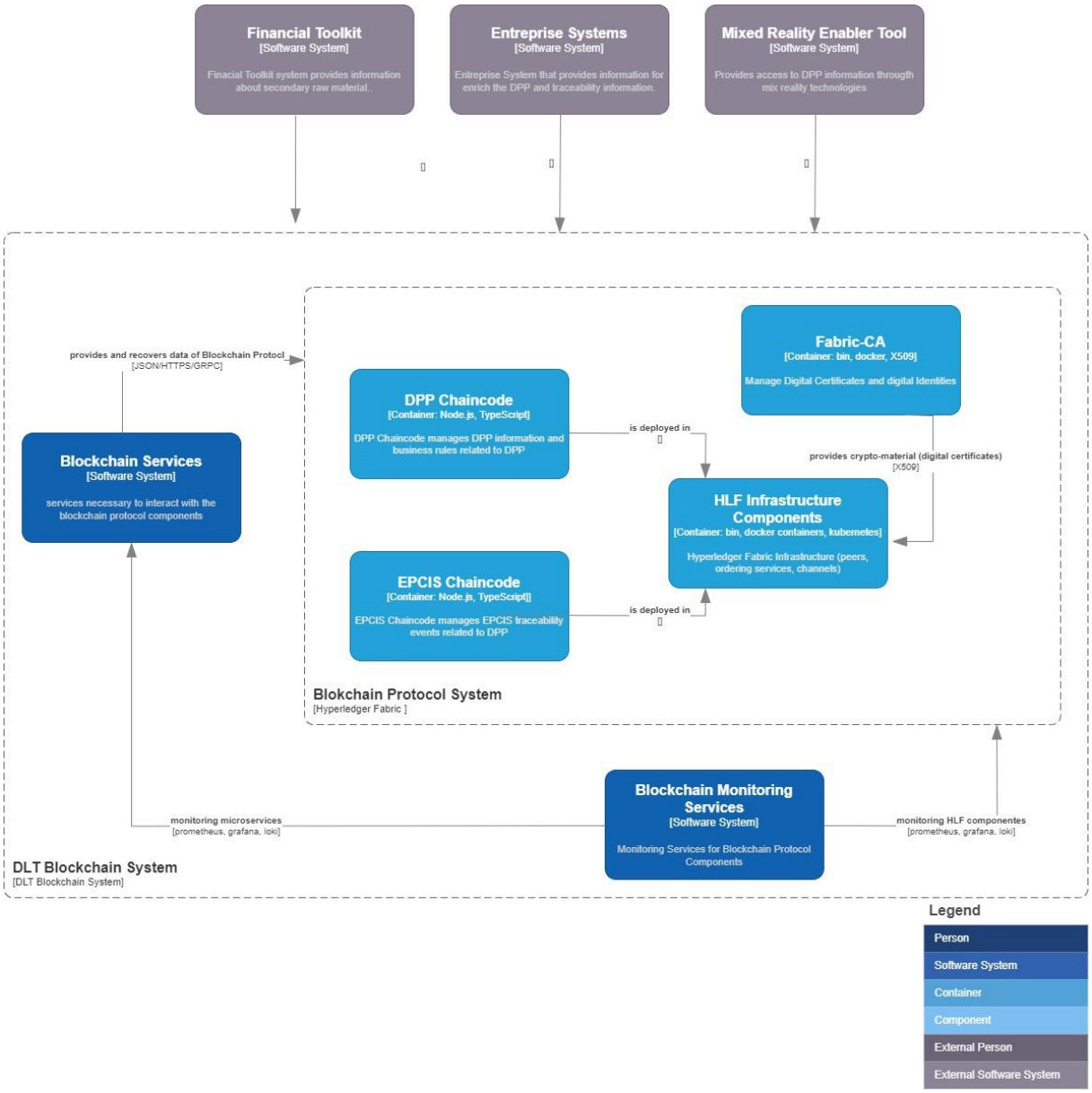

Figure 16 C4 Blockchain Protocol Diagram

2.2.3.2.2. DPP Chaincode

DPP Chaincode component is the Smart Contract in charge of manage all operations related to create and update Digital Product Passport instances into Blockchain Protocol. The operations (transactions) are stored in the Ledger and secured with the mechanics provide by Hyperledger Fabric.

In the context of DigInTraCE platform, DPP Chaincode provides a set of functionalities described in the next section (Table 18):

Id Function	Functionality

DigInTraCE

FN-DPP-01	Register the information associated with a specific DPP on the blockchain using a unique identifier that will be used as a key.
FN-DPP-02	For an existing DPP on the blockchain, allows updating the DPP information.
FN-DPP-03	Ensure that both the creation of a DPP and the modification of the data are carried out only by authorized users
FN-DPP-04	Perform the necessary data checks before persisting the information on the blockchain.
FN-DPP-05	Apply a set of pre-coded business rules based on the defined input parameters and the information accessible in the blockchain itself.
FN-DPP-06	Establish a relationship with the traceability information associated with the DPP and managed by the EPCIS Chaincode.

Table 18 Functionalities DPP Chaincode

DPP Chaincode is deployed on a specific Channel on Hyperledger Fabric with access restricted to the authorized participants for this Channel.

Specific details on the implementation can be found in Deliverable *D4.3 Blockchain and Distributed ledger V1*.

2.2.3.2.3. Traceability Chaincode

Traceability Chaincode is the Smart Contract in charge of manage all operations related to create and update traceability events associated to a Digital Product Passport instance into Blockchain Protocol. The operations (transactions) are stored in the Ledger and secured with the mechanics provide by Hyperledger Fabric.

Traceability Events are defined using EPCIS 2.0 Standard [xv] and specification. Traceability Chaincode manages the information directly in EPCIS JSON format.

In the context of DigInTraCE platform, EPCIS Chaincode provides a set of functionalities described in the next section (Table 19):

Id Function	Functionality
FN-EPC-01	Register traceability information, based on EPCIS event, and associated with one or several DPPs on the blockchain using a unique identifier that will be used as a key.
FN- EPC - 02	Ensure that traceability events are created only by authorized users.

DigInTraCE

FN- EPC - 03	Perform the necessary data checks before persisting the information on the blockchain.
FN- EPC - 04	Apply a set of pre-coded business rules based on the defined input parameters and the information accessible in the blockchain itself.
FN- EPC - 05	Check if DPP exists before to persist the traceability information into the blockchain

Table 19 Functionalities Traceability Chaincode

Traceability Chaincode is deployed on a specific Channel on Hyperledger Fabric with access restricted to the authorized participants for this Channel.

Specific details on the implementation can be found in Deliverable D4.3 Blockchain and Distributed ledger V1.

2.2.3.2.4. Fabric CA

These requirements

The Hyperledger Fabric CA is a Certificate Authority (CA) for Hyperledger Fabric. It provides features such as:

- **Identity Management:** Fabric-CA enables the issuance, revocation, and management of identities for blockchain network participants, including peers, clients, and administrators.
- **Secure Authentication:** It provides a secure mechanism for authenticating network users and nodes using X.509 certificates, guaranteeing the integrity and authenticity of communications.
- **Authorisation and Access Control:** Granular access policies can be established to control who has permission to perform specific operations within the network, such as invoking a chaincode or updating the ledger status.
- **Certificate lifecycle management** is also provided, with Fabric-CA managing the complete lifecycle of digital certificates, including the issuance, renewal, and revocation of certificates in an automated and secure manner. Fabric-CA is designed to integrate easily with existing security services, such as LDAP or Active Directory, to streamline identity management and facilitate integration with enterprise systems.
- **Audit Trail:** Fabric-CA records and audits all actions related to the issuance and revocation of certificates, providing a detailed record of identity management activities.

2.2.3.3. Blockchain Services

Blockchain Services refers to a set of services necessary to interact with the blockchain protocol components. These set of services allows from execute

operations against a chaincode (store or retrieve information), manage digital identities and certificates until use Block explorers to see the Blockchain activity (and thus, offer the inherent transparency feature in Blockchain solutions).

In the context of DigInTraCE, a representation of this components can be observed in the Figure17:

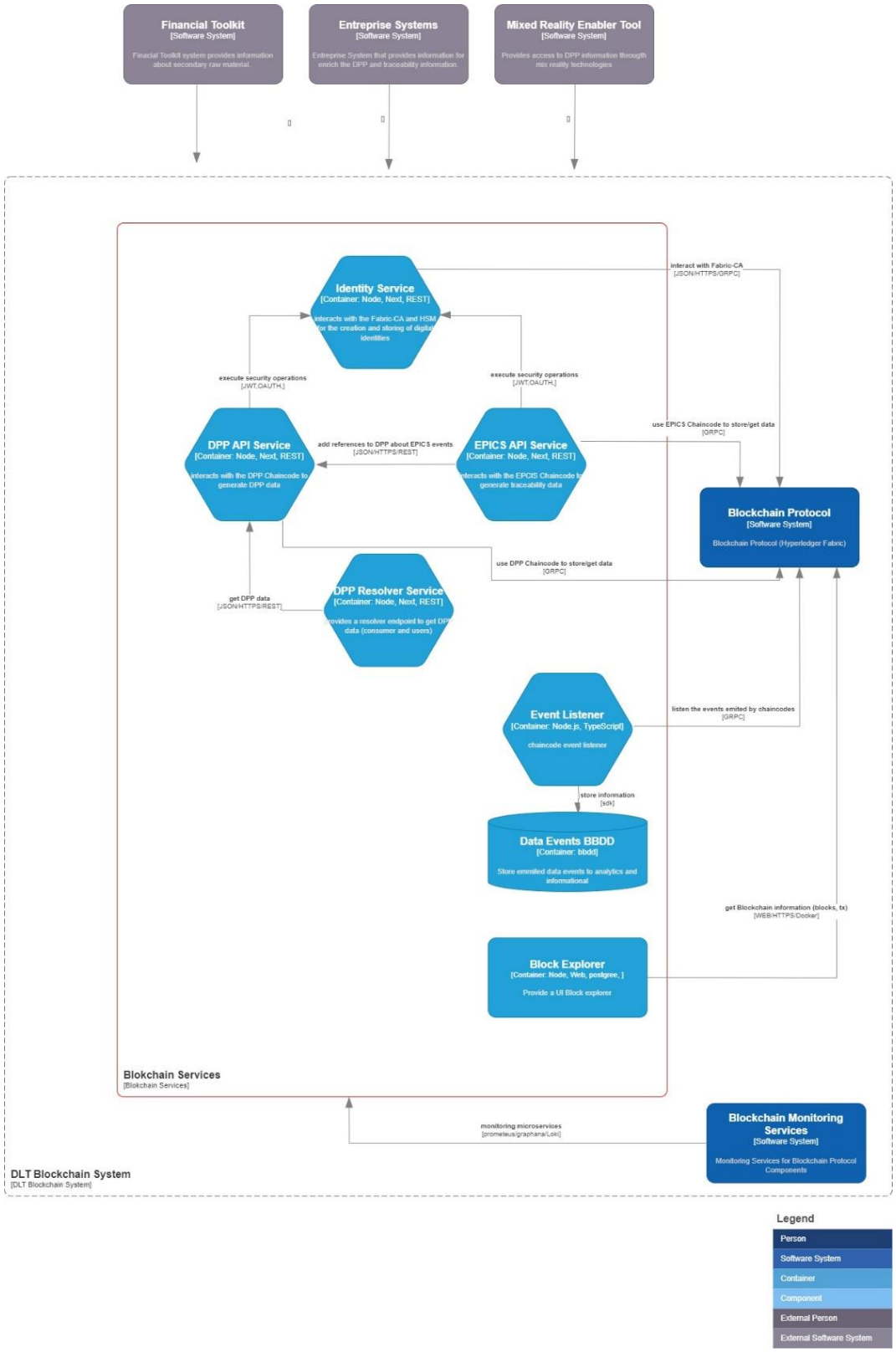

Figure 17 C4 Blockchain Services Diagram

These services offer the necessary functionalities to manage the “End2End” process of creating of DPPs (including traceability and additional features).

Each component is built as a microservice, with specific responsibilities, and low coupling to guarantee the highest degree of independence among different components.

2.2.3.3.1. Identity Service

The Identity service interacts with the Fabric-CA component offering services, in REST API mode, for the creation of digital identities that will be assigned to network participants.

The identity service has also capabilities to interacts with other security systems, such as Hashicorp Vault[xvi], for the management of cryptographic material and sensitive keys.

In the context of DigInTraCE platform, Identity Service provides a set of functionalities described in the next section (Table 20):

Id Function	Functionality
FN- IDS -01	Create users in the system to operate against the Blockchain network generating the necessary cryptomaterial (digital certificates, private keys) and executes the process related to registration and enrolment.
FN- IDS -02	Provide security mechanisms to prevent unauthorized access and consumption of other services.
FN- IDS -03	Retrieve and provide controlled access to the necessary cryptographic material to users.

Table 20 Functionalities Identity Service

Identity Services is built as microservice and deployed into a specific process.

Specific details on the implementation can be found in Deliverable D4.3 Blockchain and Distributed ledger V1.

2.2.3.3.2. DPP API Service

This microservice offers a set of API REST endpoints that expose the operations to create DPP instances and allows to client applications to submit transactions and query ledger state, using for this, Fabric Gateway library[xvii]. In addition, this API service offers security mechanisms in several levels:

- the use of API REST endpoints is restricted to specific user, therefore, is necessary to use authentication to consume the endpoints.
- To submit transactions is necessary to use the cryptographic material (digital certificates) associated with the users.

The Figure 18 shows the subcomponents that make up this service and their integration within the general architecture.

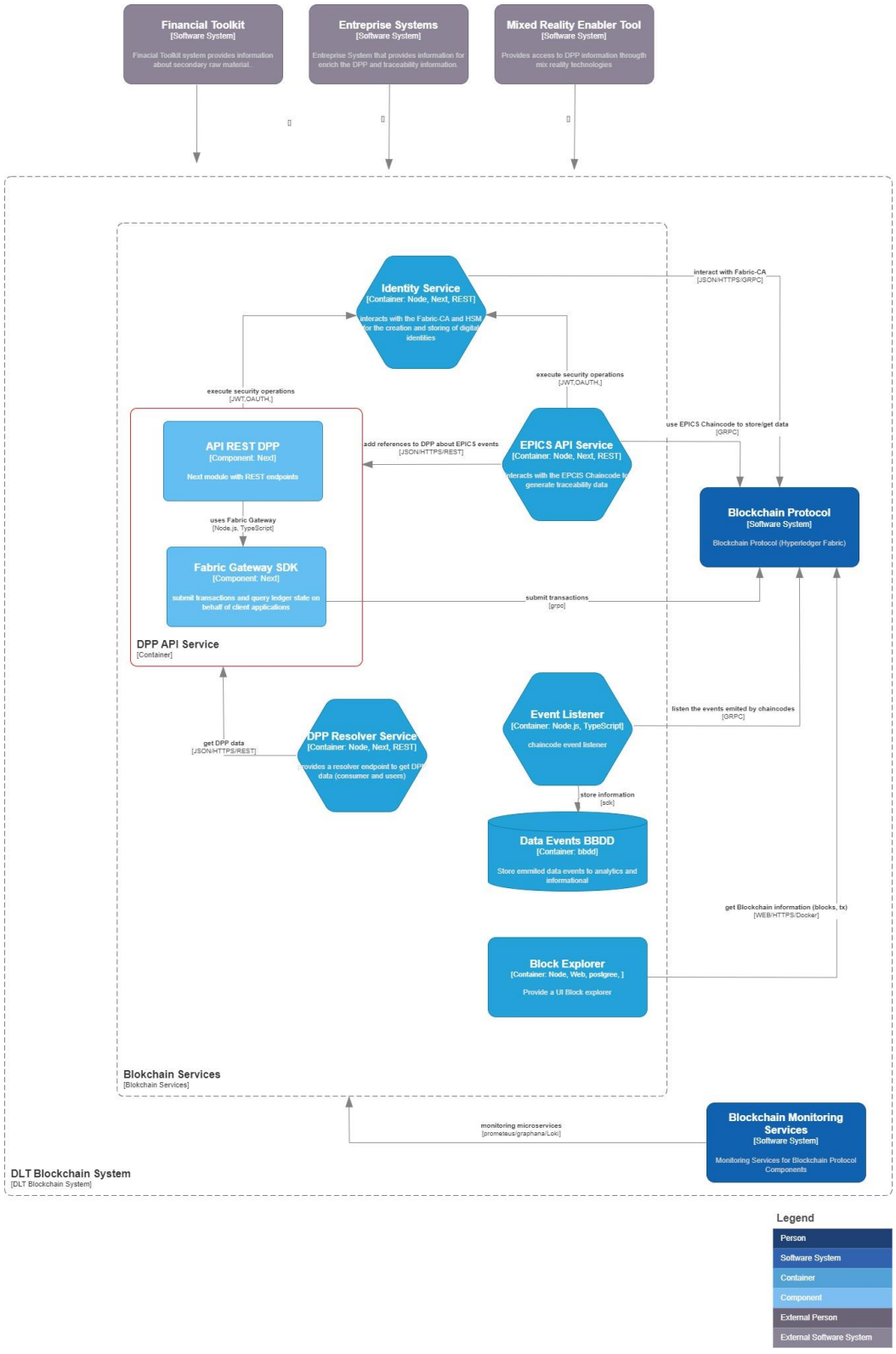

Figure 18 C4 Blockchain Component API

DigInTraCE

In the context of DigInTraCE platform, DPP API Service provides a set of functionalities described in the next section (Table 21):

Id Function	Functionality
FN- DAS -01	Expose operations in format of API REST endpoints to create, update and read DPP instances.
FN- DAS -02	Submit transactions to Blockchain Protocol to store the information into the Ledger.
FN- DAS -03	Provide security mechanisms to prevent unauthorized access and consumption of other services

Table 21 Functionalities DPP API Service

Specific details on the implementation can be found in Deliverable *D4.3 Blockchain and Distributed ledger V1*.

2.2.3.3.3. Traceability API Service

This microservice offers a set of API REST endpoints that expose the operations to create traceability events and allows to client applications to submit transactions and query ledger state, using for this, Fabric Gateway library. In addition, this API service offers security mechanisms in several levels:

- the use of API REST endpoints is restricted to specific user, therefore, is necessary to use authentication to consume the endpoints.
- To submit transactions is necessary to use the cryptographic material (digital certificates) associated with the users.

The Figure 19 shows the subcomponents that make up this service and their integration within the general architecture.

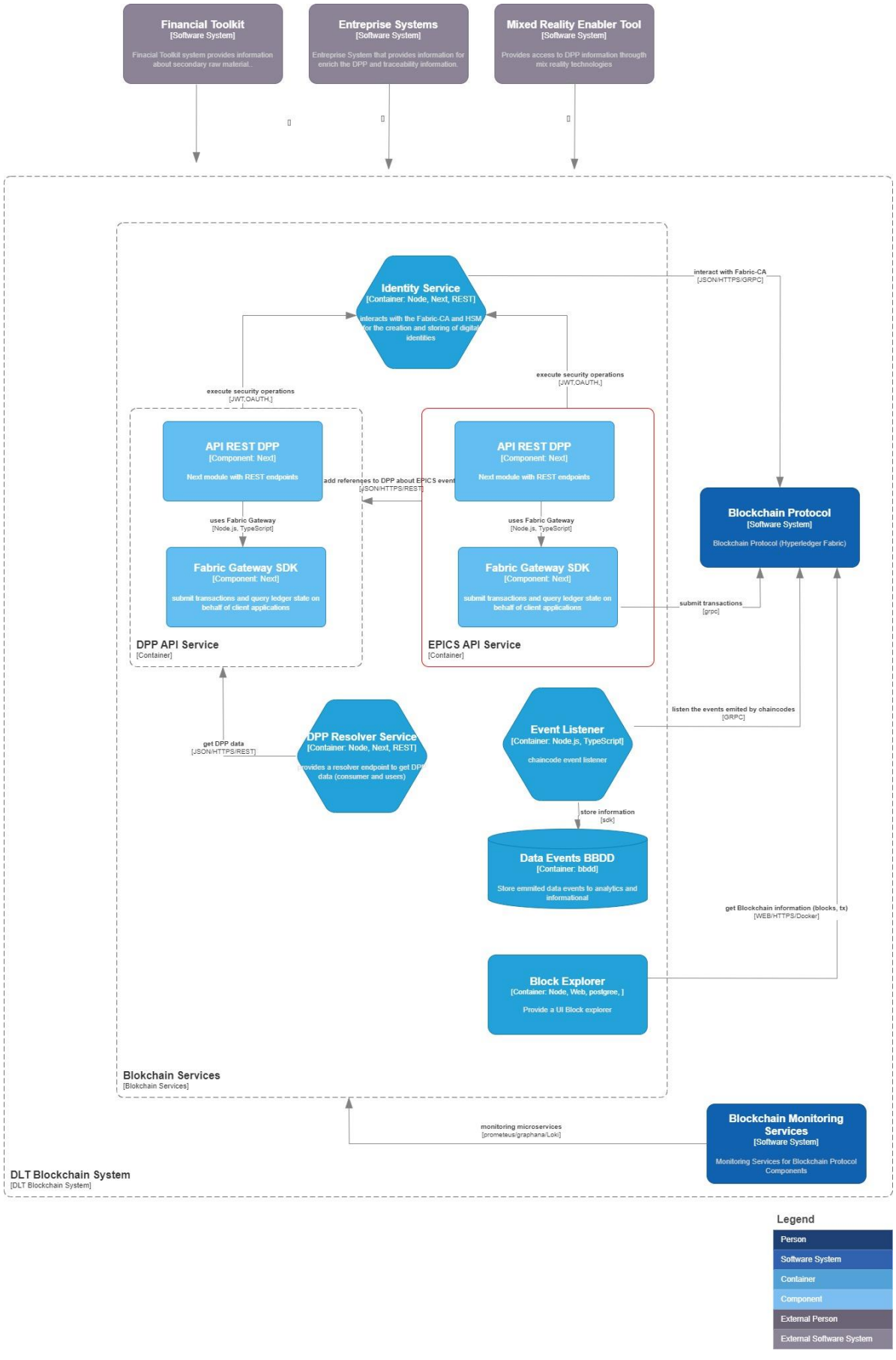

Figure 19 C4 Blockchain Component API EPCIS

In the context of DigInTraCE platform, EPCIS API Service provides a set of functionalities described in the next section (Table 22):

Id Function	Functionality
FN- EAS - 01	Expose operations in format of API REST endpoints to create, update and read traceability events related to specific DPP.
FN- EAS - 02	Submit transactions to Blockchain Protocol to store the information into the Ledger.
FN- EAS - 03	Provide security mechanisms to prevent unauthorized access and consumption of other services

Table 22 Functionalities Traceability API Service

Specific details on the implementation can be found in Deliverable *D4.3 Blockchain and Distributed ledger V1*.

2.2.3.3.4. DPP Resolver Service

This microservice is responsible for exposing the endpoints that provide the services necessary to retrieve the information associated with a specific DPP (and its traceability events) and provide it to consumers in an appropriate manner (machine-readable format).

This service could be consumed by a wider set of actors than those defined for the DPP API Service and the EPCIS API Service, as this service focuses solely and exclusively on querying the information of a DPP (not on creating or editing information) and could therefore expose public access services to specific DPP information or restricted access for certain user groups.

The Figure 20 shows the subcomponents that make up this service and their integration within the general architecture.

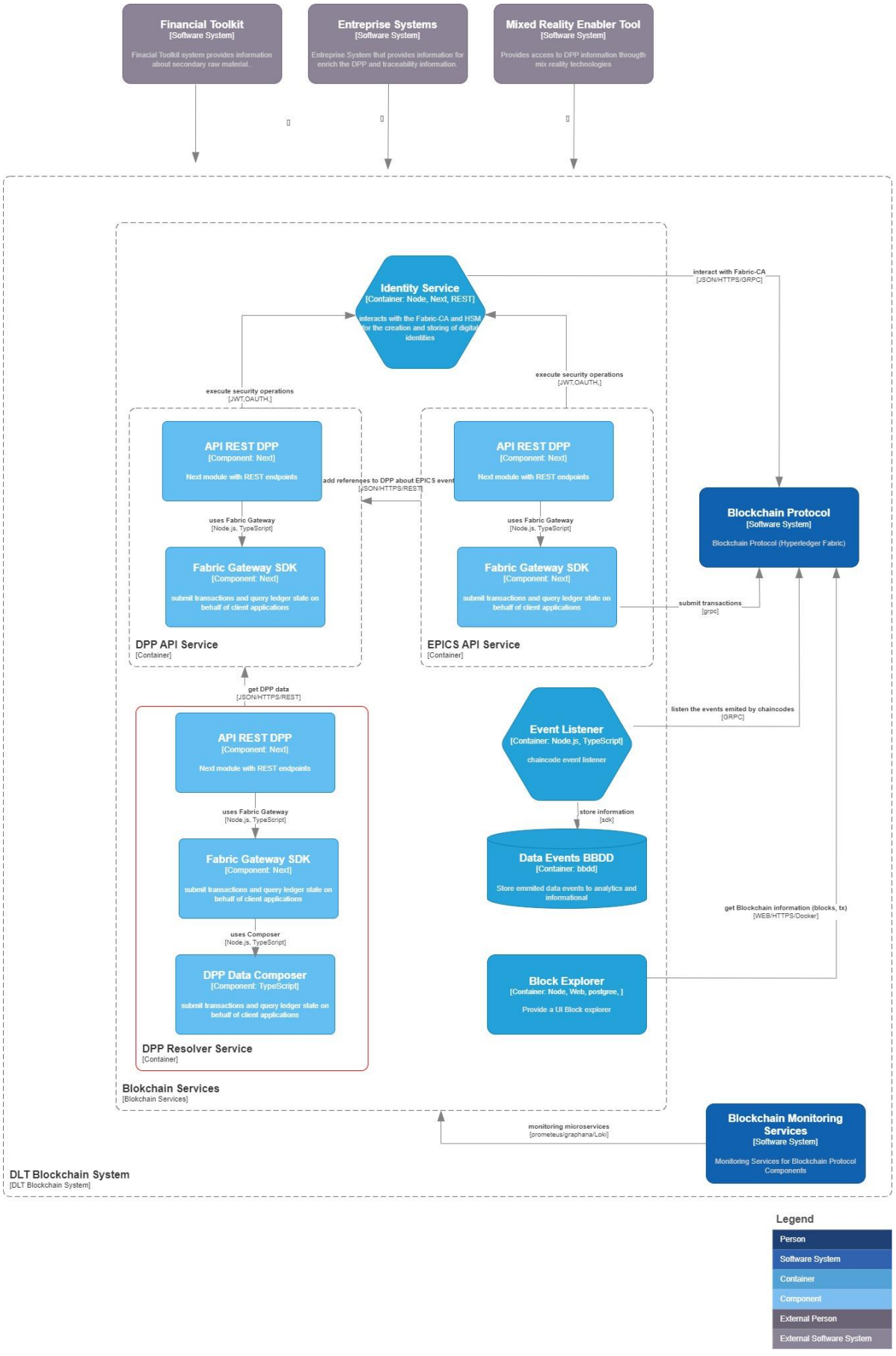

Figure 20 C4 Blockchain Component Resolver

DigInTraCE

In the context of DigInTraCE platform, DPP Resolver Service provides a set of functionalities described in the next section (Table 23):

Id Function	Functionality
FN- DRS -01	Expose publics operations in the format of API REST endpoints to get specific DPP data.
FN- DRS -02	Expose restricted access operations in the format of API REST endpoints to retrieve specific DPP data.
FN- DRS -03	Offers a Content resolution mechanism that "resolves" (finds the location of the document/DPP) using mechanism as URN, URL or DID.
FN- DRS -04	Querying to Blockchain Protocol submitting transactions (read transactions) to get the data from the ledger.

Table 23 Functionalities DPP Resolver Service

Specific details on the implementation can be found in *Deliverable D4.3 Blockchain and Distributed ledger V1*.

2.2.3.3.5. Event Listener Service

This microservice offers capabilities to listen and monitor the events produced in the DLT when specific chaincode methods are executed. The events contain personalized information that could be used, for instance, to analyse information in real-time, triggering external operations or store information in off-chain repository.

The Figure 21 shows the subcomponents that make up this service and their integration within the general architecture.

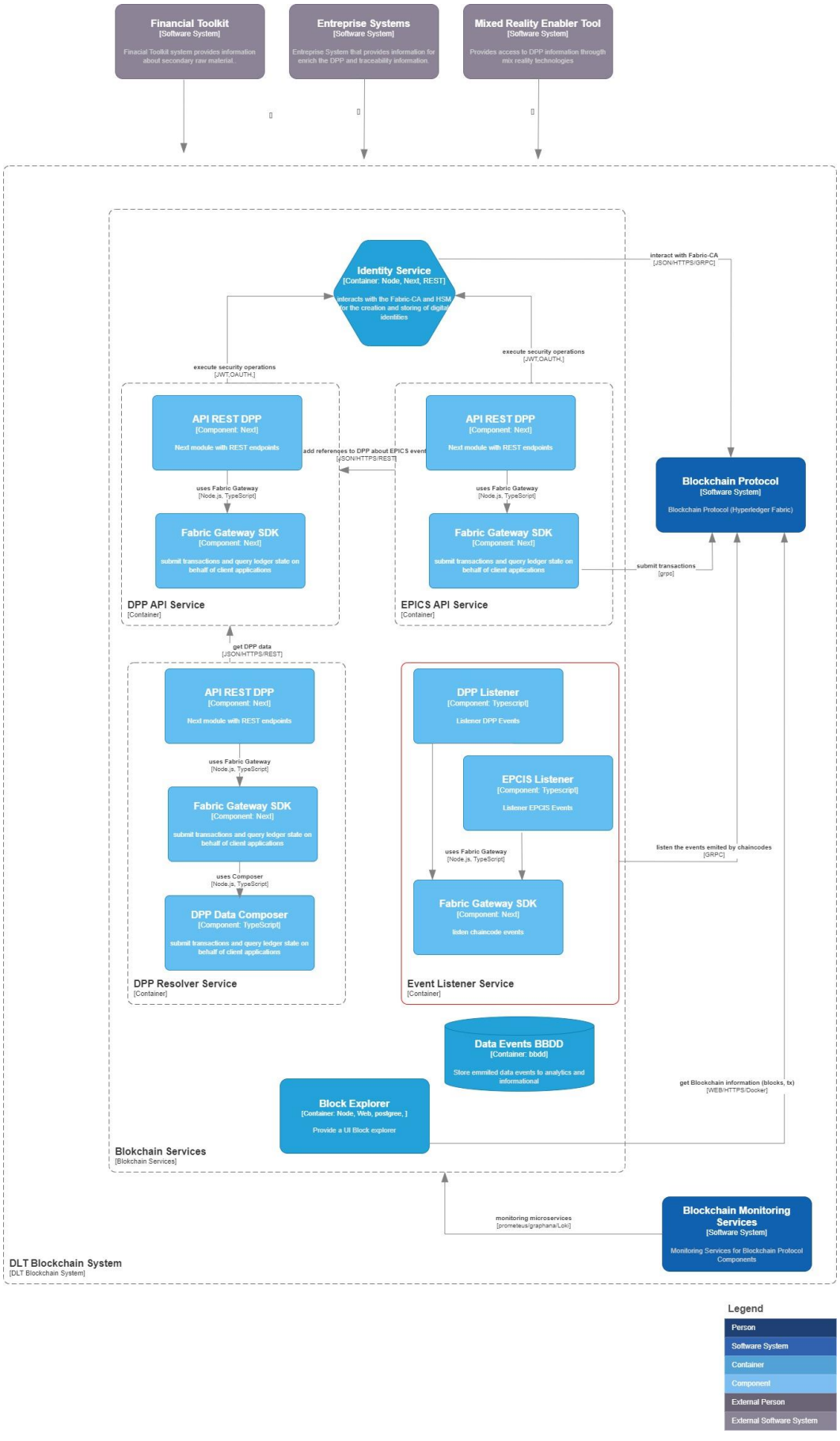

Figure 21 C4 Blockchain Component Listener

DigInTraCE

In the context of DigInTraCE platform, Event Listener Service provides a set of functionalities described in the next section (Table 24).

Id Function	Functionality
FN- ELS -01	Listen the events produced by DPP Chaincode.
FN- ELS -02	Listen the events produced by EPCIS Chaincode.
FN- ELS -03	Provide default configuration to store information (events listened) in off-chain repository (database).

Table 24 Functionalities Event Listener Service

Specific details on the implementation can be found in Deliverable *D4.3 Blockchain and Distributed ledger V1*.

2.2.3.3.6. Block Explorer Service

This service is a web-based tool that allows users to interact with and explore the data stored on the Fabric blockchain network. It provides functionalities to browse through blocks, transactions, and other relevant information such as chaincode deployed, participants (nodes) or channels deployed.

The Figure 22 shows the subcomponents that make up this service and their integration within the general architecture.

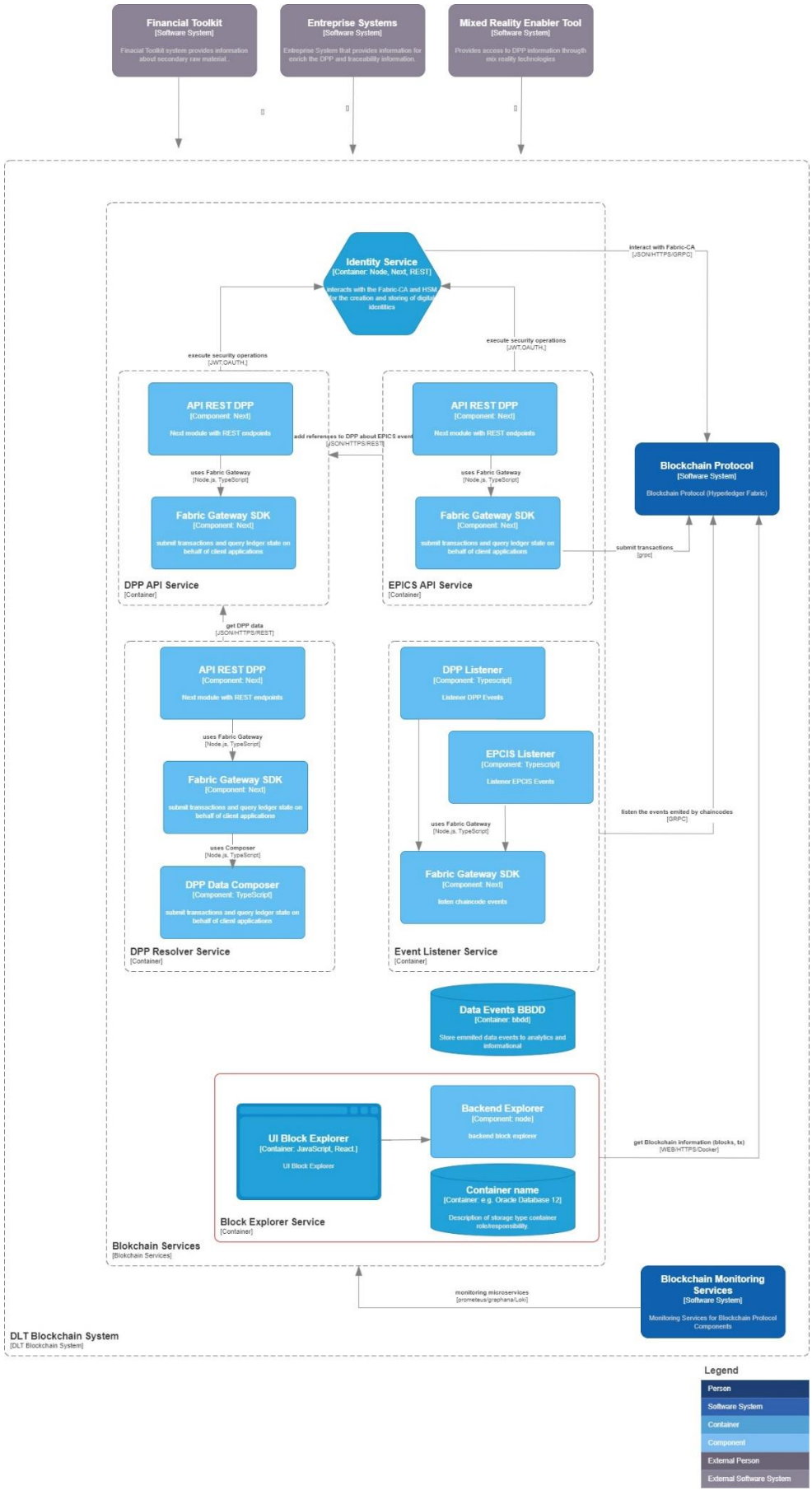

Figure 22 C4 Blockchain Component Explorer

DigInTraCE

In the context of DigInTraCE platform, Block Explorer Service provides a set of functionalities described in the next section (Table 25).

Id Function	Functionality
FN- BES -01	Visualize Blocks and transactions information.
FN- BES -02	Set access control (login) and role-based policies.
FN- BES -03	Search by specific transaction identifier and recovery information about the transaction.

Table 25 Functionalities Block Explorer Service

Specific details on the implementation can be found in Deliverable *D4.3 Blockchain and Distributed ledger V1*.

2.2.3.4. Requirement traceability matrix

The next Table 26 shows a Requirement traceability matrix where each DPP requirement identified in previous section is implemented by one or several functionalities.

Requirements	Functionalities involved
req-dpp-01	FN-DPP-01, FN-EPC-01, FN- EAS -01, FN- DAS -01, FN- ELS -01
req-dpp-02	FN-DPP-06, FN- EPC -05
req-dpp-03	FN-DPP-01, FN- EPC -05
req-dpp-04	FN-DPP-01, FN- EPC -05
req-dpp-05	FN-DPP-02
req-dpp-06	FN-DPP-03, FN- EPC -02, FN- IDS -01, FN- IDS -02, FN- IDS -03, FN- DAS -03, FN- EAS -03
req-dpp-07	FN- DAS -03, FN- EAS -03, FN- DRS -01, FN- DRS -02
req-dpp-08	FN-DPP-05, FN- EPC -04
req-dpp-09	FN-DPP-02
req-dpp-10	FN-DPP-03, FN- EPC -02
req-dpp-11	FN- DAS -02, FN- EAS -02
req-dpp-12	FN- DRS -01, FN- DRS -02, FN- DRS -03, FN- DRS -04
req-dpp-13	FN-EPC-01, FN- EAS -01
req-dpp-14	FN-EPC-02, FN- EAS -02
req-dpp-15	FN- EAS -01
req-dpp-16	FN- DRS -01, FN- EAS -01, FN- ELS -02

Table 26 Requirement traceability matrix

2.2.4. External dependencies

No dependencies have been identified for the correct functioning of the Blockchain module. This module will provide service to other modules of the platform, for example, to MR Enabler tool to obtain information stored in the Blockchain or it will feed on information provided by other components, such as CLSC tool, which in any case will interact through the API Services.

DigInTraCE

2.2.5. Technological decisions

Blockchain is an innovative and powerful technology with unique features around distributed and decentralized systems, cryptography, data security, etc. However, these characteristics are valuable when the use case where blockchain is applied makes sense.

To know if Blockchain technology is suitable, several questions should be planted in the early steps of the project. Literature and resources in the form of decision models exist on the Internet[xviii].

Some references are CompTIA 2019[xix], represented in Figure 23:

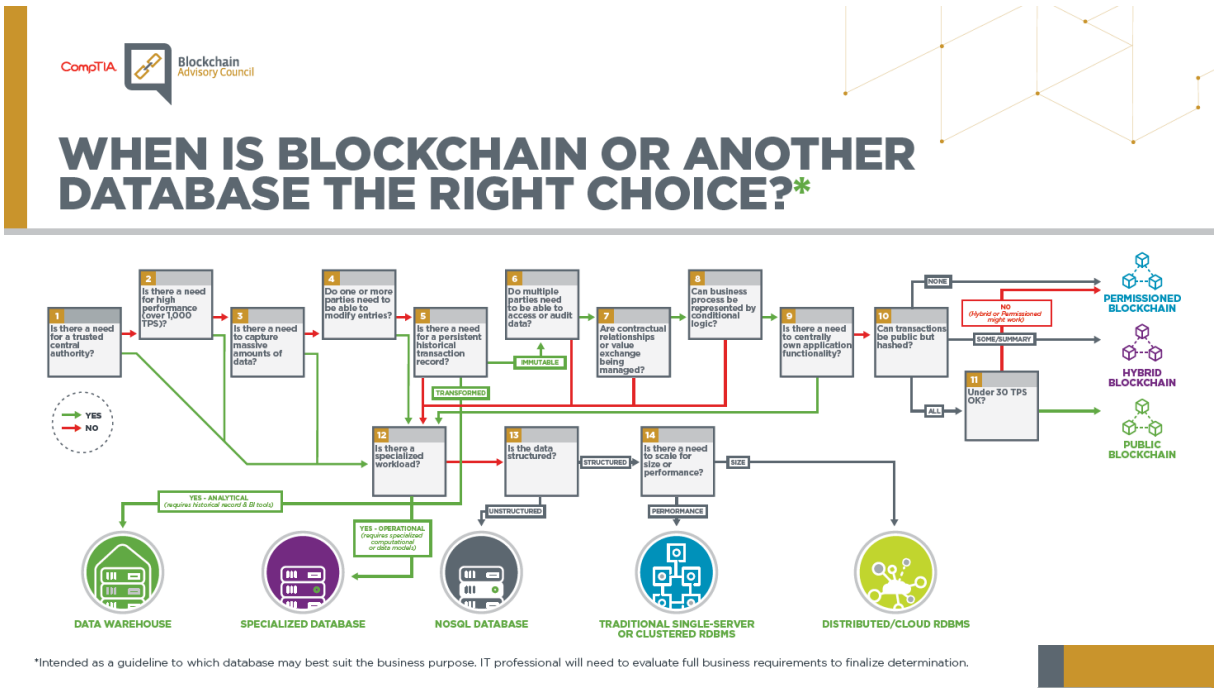

Figure 23 Blockchain-decision-tree

Or the resource published by World Economic Forum[xx] (Figure24):

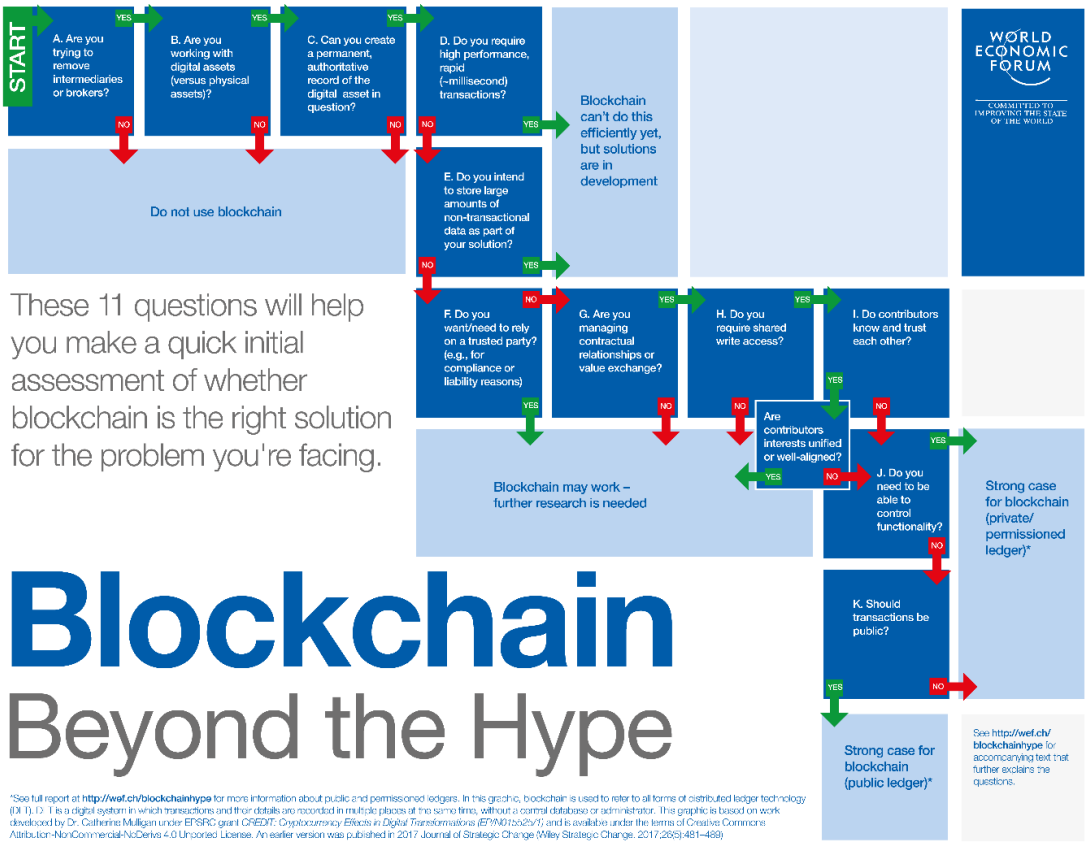

Figure 24 Questions-blockchain-toolkit-right-for-business.

In a very summarized way, it is possible to determine there are four key points to be considered:

1. **Data sharing.** DLT/Blockchain systems (and database systems) support the need to share data with a number of independent actors who do not need to trust each other. If there is no need for trusted data sharing, there is no need for DLT/Blockchain.

2. **Multiple writers**. DLT/ Blockchain systems have multiple writers. If there is only one writer, DLT/Blockchain is not necessary.

3. **Trusted parties**. DLT/Blockchain is suitable for eco-systems with parties who do not trust each other. If exists a centralized authority that holds the power of take decisions, there is no need for DLT/BC

4. **Immutability.** Immutable data cannot be changed or deleted. It can only be created, and then it exists permanently. Immutability is a property of

DigInTraCE

Blockchain technology, if this property is not valuable or necessary, DLT/Blockchain is not necessary.

There are mature Blockchain solutions in production related to traceability and supply chain and it could be considered one of the main use cases of Blockchain in the business field.

On the other hand, although DPPs are a more novel area of application, they also have several projects at both a commercial and academic level where Blockchain technology has been chosen for implementation.

In the next Table 27, it is possible to observe a resumed rational for the use of Blockchain in DPP use cases:

BC Property	Apply to DPP / DigInTraCE context?
Data sharing	**Yes**, DPP and traceability will be shared among different organizations.
Multiple writers.	**Yes**, several actors will contribute with their data into the value chain
Trusted parties	**Yes**, the actors must establish trust mechanism and not exists a central authority.
Immutability	**Yes,** DPP and traceability data must be immutable and provide guaranties about not data manipulation.

Table 27 DPP Blockchain rationale

Once the technology is deemed appropriate, it is necessary to determine if a public or private blockchain will be used.

In the context of DigInTraCE project, and as mentioned in previous sections, Hyperledger Fabric, an open source DLT/Blockchain technology, has been chosen to deploy a private Blockchain network.

The reasons to use this technology versus public networks as Ethereum[xxi], Polygon PoS[xxii] or other alternatives not based on EVM compatible protocols as Solana[xxiii] are based in the next points:

- **Security and control:** The use of a private blockchain like Hyperledger Fabric allows for greater control and security over the data and transactions of the Digital Product Passport. This is crucial to ensure the integrity and confidentiality of sensitive information.
- **Scalability and performance:** Public blockchains often have limitations in terms of scalability and performance, especially when handling large volumes of data. In contrast, a private blockchain like Hyperledger Fabric can offer better performance and higher processing capacity to meet the project's needs.
- **Governance and permissions:** A private blockchain enables the definition and control of access permissions and governance rules, ensuring that only authorized parties can interact with the Digital Product Passport. This is fundamental to comply with regulatory and security requirements.
- **Industry-specific interoperability:** The use of a private blockchain like Hyperledger Fabric allows for the leveraging of solutions and tools

developed specifically for the sector or industry, which facilitates integration and data exchange with other key stakeholders.

- **Regulatory compliance:** Private blockchains, such as Hyperledger Fabric, can be better adapted to the regulatory and compliance requirements applicable to the project, such as data protection and privacy.

2.2.6. Technical specifications

2.2.6.1. Operational

Operational architecture characteristics cover capabilities:

	Description
Availability	Hyperledger Fabric nodes (peers) are provided by organizations that support the Blockchain network. At least, 2 nodes by organization are provide running in 24/7 mode. Blockchain services (APIs, listener, Identity) are deployed in Kubernetes cluster with appropriate configuration to guarantee availability.
Continuity	Hyperledger Fabric nodes (peers) are redundant with at least 2 peers by organization.
Performance	Hyperledger Fabric has a high performance[xxiv] in TPS (transaction per second) being capable of servicing thousands of transactions per second.
Recoverability	Hyperledger Fabric can execute automatic recoverability process in case some node down fall. Synchronization process between nodes is executed and ledger and world state updated with latest status.
Robustness	Hyperledger Fabric and its components offers mechanism to avoid the execution of bad transaction (with incoherences in the state of the Blockchain)
Scalability	The number of nodes can be increased easily if is required.

Table 28 DLT Operational characteristics

2.2.6.2. Structural

Structural architecture characteristics cover capabilities:

	Description
Configurability	Configuration changes in some aspects of Hyperledger Fabric network can be modified with the agreement of organizations; endorsement policies, channels configurations, etc.
Extensibility	Chaincode functionality can be updated by deploying new versions without breaking previous states or information
Portability	Hyperledger Fabric and rest of component runs in containers deployed in Kubernetes with an approach totally cloud agnostic.
Maintainability	Configuration changes in some aspects of Hyperledger Fabric network can be modified with the agreement of

	organizations; endorsement policies, channels configurations, etc.
Installability	Hyperledger Fabric and rest of component runs in containers deployed in Kubernetes with an approach totally cloud agnostic

Table 29 DLT Structural characteristics

2.2.6.3. Cross-cutting

Cross-cutting architecture characteristics cover capabilities:

	Description
Accessibility	NA. Accessibility from other system, not final user.
Archivability	Not necessary. Information is secured stored into Ledger structure (blockchain).
Authentication	Several levels of authentication: API keys, JWT, digital certificates...
Authorization	Operations over the Hyperledger Network are restricted to authorized user granted by the own organizations. Chaincode operations can be restricted to specific users, roles, etc.
Privacy	Privacy data can be managed with specific features (confidential data channels) or specific channel with restricted use.
Security	Secure protocols are used: https[xxv], gRPCs [xxvi]
Usability/Achievability	Technical user documents are provided, and some items have their own self-contained documentation (example APIs REST)
Interoperability	System is interoperable with enterprise and legacy system through SDK (fabric gateway) and API services.

Table 30 DLT Cross-cutting characteristics

2.3. CLSC Tool

2.3.1. Architecture overview

The CLSC Tool stands for Closed Loop Supply Chain Tool, and it covers the overall Supply Chain Management System, an end-to-end solution which supports the entire supply chain from the demand planning to the scheduling and MES phases. It is modulable and adaptable to several contexts where it can be implemented and configurated. It is a cloud-based solution, using Microsoft Azure components, tools, and services.

Within DigInTraCE, it will be interesting to utilize the CLSC Tool with a particular focus on its Scheduling and MES Modules to find the right process receipt to optimize the production.

The CLSC Tool can receive as an input (.xlsx/csv) the following pieces of information:

- A project folder with metadata, data, and versioning concept (assembly, part, drawing)
    - Products' CAD file (mandatory)

  - raw materials, semi-finished products, finished products Masterdata (mandatory)
  - BoMs (mandatory)
  - Warehouses Masterdata (not mandatory, useful in case the Demo should involve a whole network or specific materials stock analysis may be required)
- Current Stock (not mandatory, useful in case the Demo should demonstrate also the coverage by stock)
- Resources Masterdata: machinery, availability, set up time, equipment (mandatory)
- Routings with a clear definition of the relationship among the Item produced, the resource where it is produced, and the time required to produce a single unit or a batch (mandatory)
  - Item 1, resource Alfa, 10min/unit
  - Item 1, resource Beta, 15min/unit
  - Item 2, resource Alfa, 5min/unit
- Calendar, which is useful to identify the capacity and it can be expressed either as shifts or hour capacity per day (mandatory)
- Production orders (not mandatory)
- Client Masterdata (not mandatory, useful in case the system should consider prioritization parameters or some customers' attributes should be shown)
- Customer Orders (mandatory)
- Forecast Orders (not mandatory)
- Supplier Masterdata (not mandatory)
- Purchase Orders (not mandatory)
- Optimization matrix such as setup/cost (not mandatory)

The scheduling engine will try to optimize the job sequences to deliver the products on time and minimize the set-up cost. It will be supported by the information provided by the AI algorithms which will guide production by collecting several real-time parameters from the working machine. The AI algorithms' aim will be to anticipate a production deviation to prevent defeats into the product. The algorithms will collect different production parameters as temperature, humidity… provided by the machine. Thanks to the data that will be available different thoughts will be taken into consideration, i.e., energy consumption and its reduction. Some of these KPIs could be introduced into the DPP.

DigInTraCE

Hereby it is provided a technical schema concerning the architecture (Figure 25). It represents the manner in which the system interacts with Microsoft Azure services, and it explains also the grade of communication it has with the Algorithm layer.

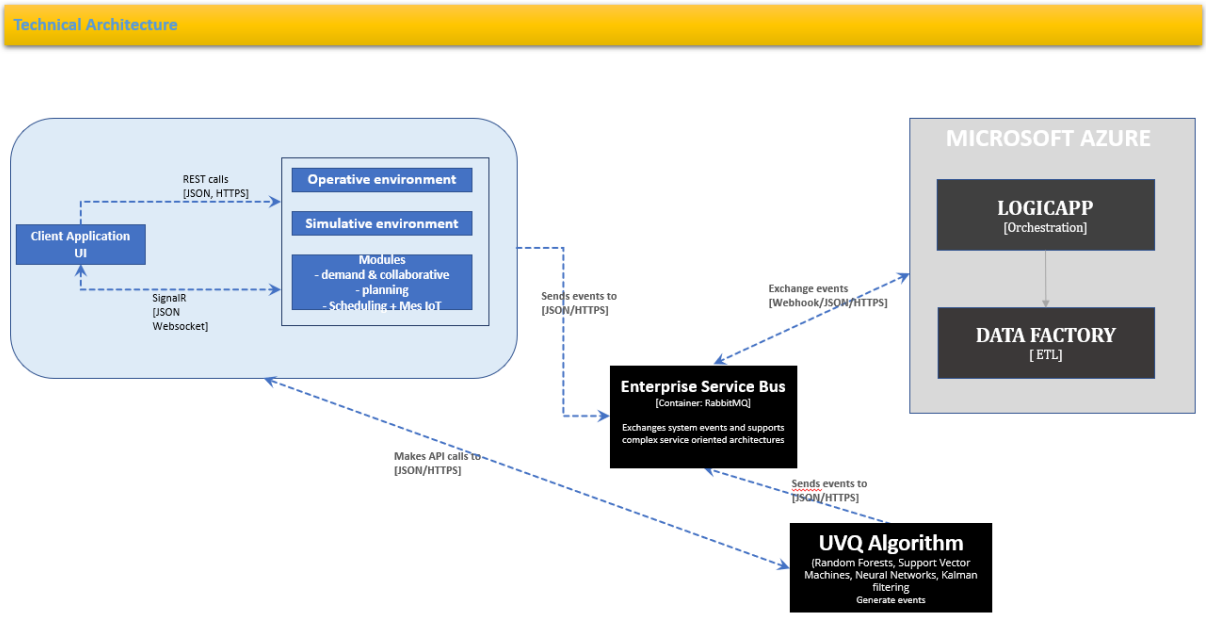

Figure 25 Technical architecture

It is possible to represent the functional flow following the schema below (Figure 26):

Functional Flow:

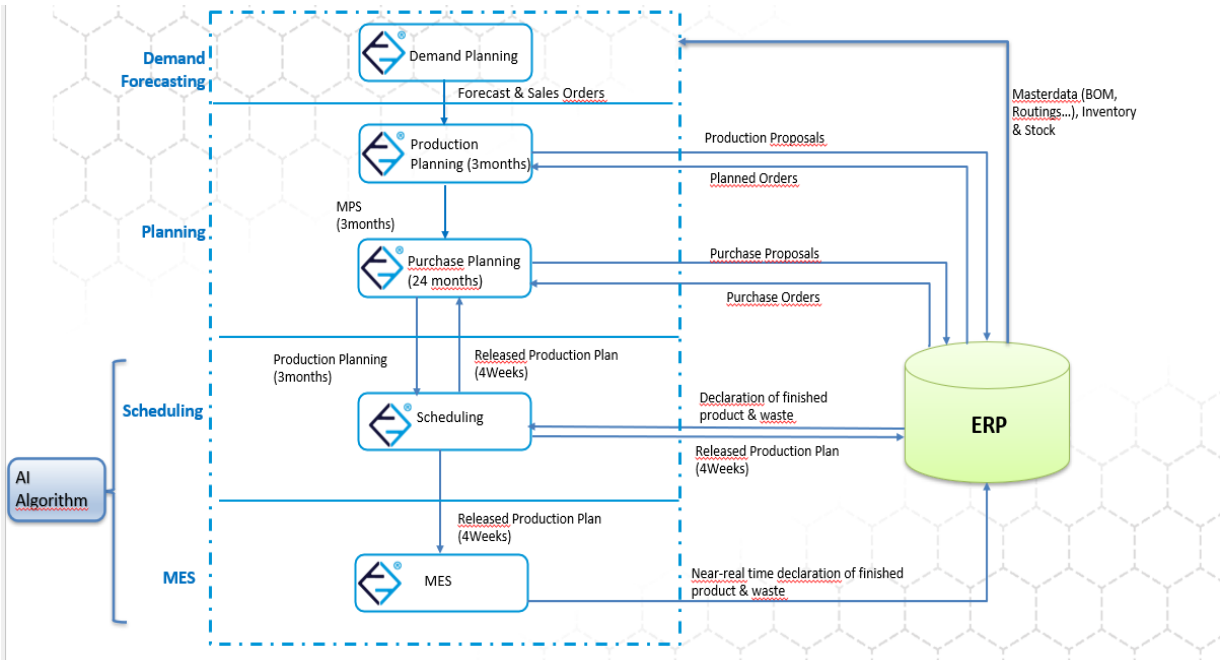

Figure 26 Functional workflow

DigInTraCE

Usually, a CLSC tool is integrated to an ERP with which it can bidirectionally communicate and exchange information. Nevertheless, it can be interfaced with multiple technologies and third parties. The Supply Chain Flow starts from the Demand Planning where it is possible to identify Sales Forecast and Sales Orders.

The Production Planning engine typically works with the Inventory Stock given by an ERP or a WMS. Its first step consists of calculating the MPS, which MPS (Master Production Schedule) stands for Master Production Schedule also known as the aggregate production plan. The MPS is a high-level production plan that evaluates the production of individual finished products. Its goal is to translate the demand for finished goods into production capacity requirements and compare them with available production capacity. The MPS contains relevant production information, including timing, such as production lead time. This production planning tool aligns production with demand, detailing products, quantities, and timing over a time horizon of several months or years. Its functions include optimizing inventory, forecast accuracy, and assessing capacity, labour, and costs. When it is interfaced with an ERP typically it sends Production Proposals, and the Planner transforms them into confirmed Planned Order. The Production Planning engine is useful also to the Purchase Function because it generates the Purchase Proposals as well. When a Purchase Proposal is confirmed into the ERP or wherever, it is returned as Purchase Order.

The Scheduling engine is useful to allow to a further optimization, its functioning its strictly linked to an optimization function and its horizon consists in few weeks despite several months that could be taken into the Production Planning. It helps the planner to have the best optimized sequence of batches to produce.

In addition, if the production scenario utilizes machineries, then it may result useful to also have the MES module, that typically gives information near-real time about the ongoing production.

In this project the focus will be on Scheduling and MES modules, because the data belonging to the other processes such us demand forecasting will be considered as an input to the system, out of scope of this research.

The simulative Scheduling engine will take into consideration the various parameters that have been given to the system as mandatory data and it will donate as output the optimized production sequence.

There is a matrix of optimization, and it can consider different drivers such as machinery energy consumption, set-up cost, number of different colors that can be produced in an interval of time.

The system, once a production deviation is forecasted by the AI algorithm, will be able to send a signal or a list of instructions directly to the machinery or it will be able to reallocate the production and to redefine a new production plan ensuring that it will be the most efficient one despite maintenance activities it should then take into consideration as well.

2.3.2. Components and internal relationships

The CLSC Tool allows users to forecast, plan, schedule, control production and materials of the whole business supply chain both in a simulative manner and in a unique confirmed scenario. It can be integrated to other Enterprise Software such

as ERP, CRM, WMS to receive data in input and to produce data in output which are able to be more effective and precise.

Different kind of users can access to it from Administrators to other Supply-chain actors such as external suppliers which can share their ongoing production plan including incidents or warnings of delay.

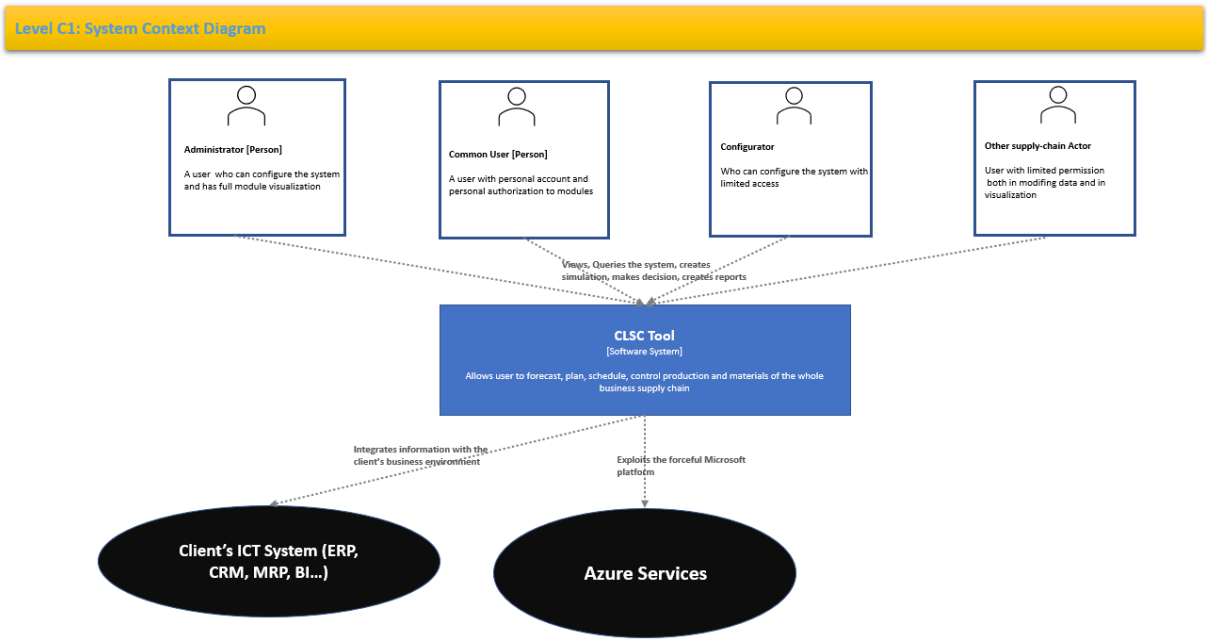

Figure 27 C4 System context Diagram

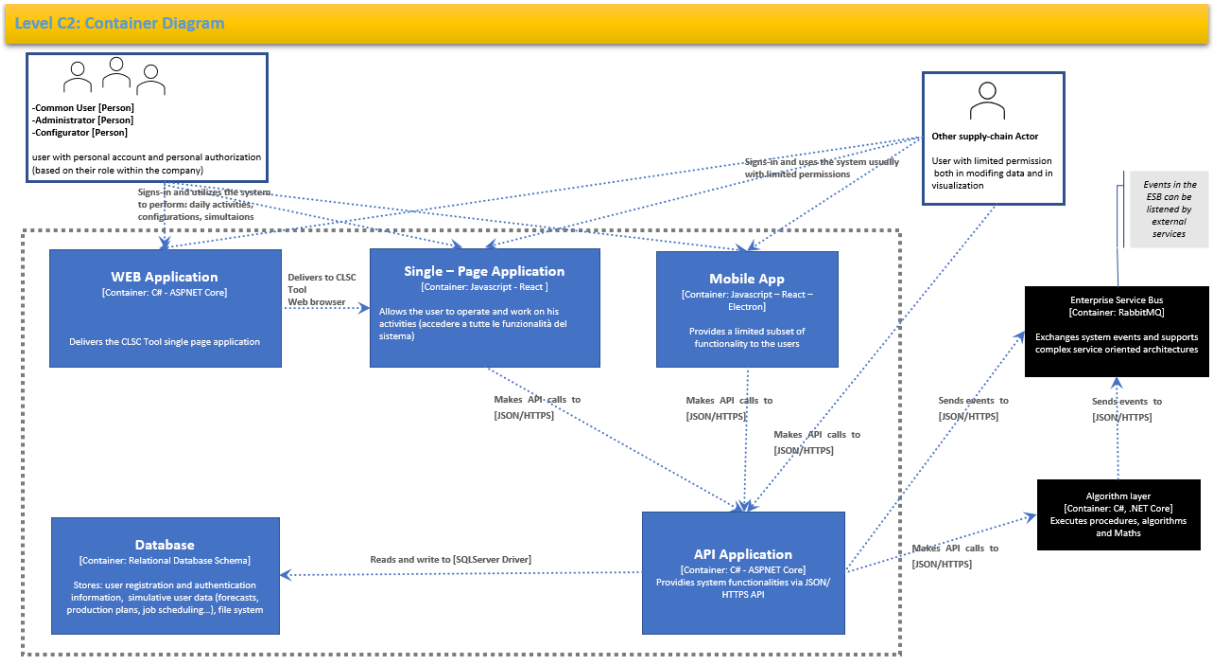

Figure 28 C4 Container Diagram

The Figure above (Figures 27 – 28) shows how the CLSC Tool could be used by different users, its dependencies to the Database, the Enterprise Service Bus, and

DigInTraCE

the Algorithm layer. Thanks to this architecture it could be considered at easy to be integrated with other components or systems.

2.3.3. External dependencies

Among the external dependencies with other DigInTraCE's modules it would be useful to consider the following schema (Figure29):

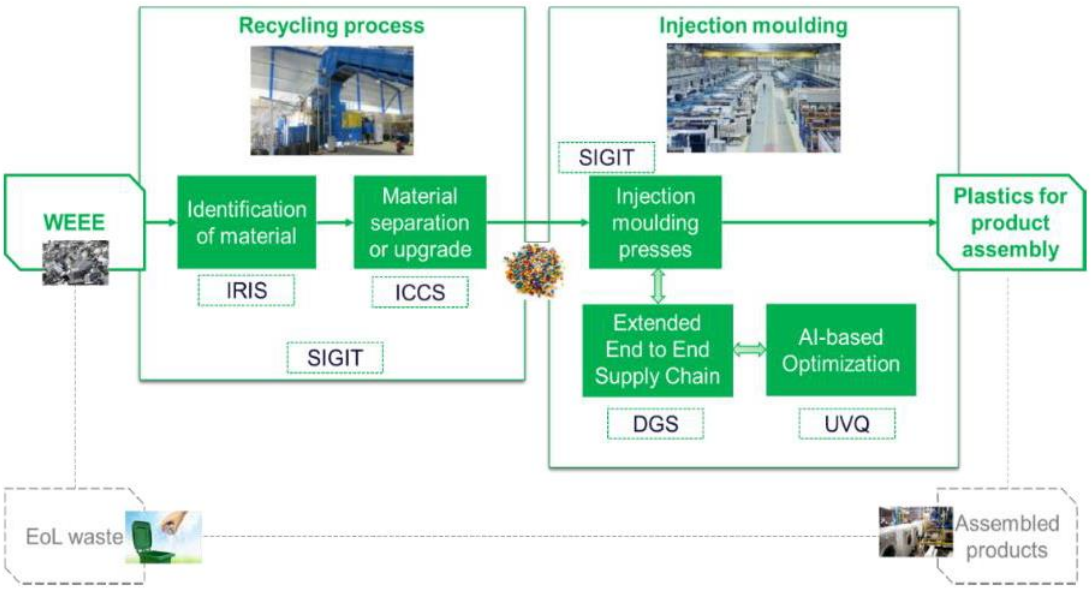

Figure 29 External dependencies diagram

The CLSC Tool should be able to access to the pieces of information related to the raw material and processes existing upstream.

However, it could also provide dynamically insights regarding the traceability of the ongoing production. To make an example into the DPP it would be possible to gather information about the i) Finished Product and the production batch it belongs, ii) its Bill of Material and the origin of the raw material, iii) the main production conditions (temperature, pressure, humidity…).

2.3.4. Technological decisions

The CLSC Tool is a Microsoft Azure-Based CLSC, and it would be useful to consider the frequency of the gathering and providing data from and to the DPP. Furthermore, it would be necessary to evaluate which method will be more effective for this use: Push Vs Pull system, evaluating if it would be more efficient managing events or a continuous polling.

2.3.5. Technical specifications

The CLSC Tool is a cloud-based platform, and its system works with Microsoft Azure.

Some of its capabilities strictly depend on the type of the Azure subscription signed, such as availability, continuity, performance, recoverability, reliability, scalability. In the following paragraphs will be described respectively the operational, structural, and cross-cutting characteristics.

DigInTraCE

2.3.5.1. Operational characteristics

The Operational characteristics are those that depend on the type and size of the Microsoft Azure subscription signed. A high-performing cloud environment typically also represents an increase to the compliance concerning the availability, continuity, performance, recoverability, reliability, scalability.

	Description
Availability	Duration for which the system needs to be available (if 24/7, steps should be in place to quickly bring the system back up in case of any failure).
Continuity	Disaster recovery capability
Performance	Includes stress testing, peak analysis, frequency of functions used, required capacity, and response times.
Recoverability	Business continuity requirements (e.g., in case of a disaster, how quickly the system needs to be back online). This will impact backup strategy and duplicated hardware requirements.
Reliability/Safety	Assessment of whether the system needs to be fail-safe or if it's mission-critical in a way that impacts lives.
Robustness	Ability to handle error and boundary conditions while running, like an internet connection failure, power outage, or hardware failure.
Scalability	System ability to perform and operate as the number of users or request increase

Table 31 CLSC Operational characteristics

2.3.5.2. Structural characteristics

The structural architecture characteristics are defined as:

	Description
Configurability	Change aspects of the software configuration
Extensibility	Ease of plug new pieces of functionality in.
Portability	It now works only with Microsoft Azure cloud. Most of its modules (from Demand to the MES) are designed to be able to run into Linux and Microsoft systems, just the Scheduling module works exclusively with Microsoft system. At this moment, the Database can be available only on SQL server and it is not portable to Oracle
Maintainability	Ease of applying changes and enhance the system
Installability	System installation on all necessary platforms is quite easy. Nevertheless, it may have some difficulties in case of a highly customized environment or context specifications that typically are evaluated during the analysis phase.

Table 32 CLSC Structural characteristics

2.3.5.3. Cross-cutting characteristics

The Cross-cutting architecture characteristics are defined as:

	Description

DigInTraCE

Accessibility	Access for all users, including those with disabilities such as colour-blindness or hearing loss it is not available
Archivability	Requirement to archive or delete data after a certain period.
Authentication	Security measures to ensure user identity verification are orchestrated with the OAUTH2.0/OPENiD standards
Authorization	Security measures to limit user access to specific functions within the application.
Legal	Legislative constraints or regulations regarding system operation, reservation rights, or application development. - ?
Privacy	It can conceal transactions from internal company employees, but typically personal and sensitive data are not used in its module (just the email is used to login)
Security	A data encryption in the database and network communication between internal systems, as well as authentication requirements for remote user access, is guaranteed. Nevertheless, it does not encrypt data between the System and the DB and within the DB itself.
Usability/Achievability	It is a very user-friendly application, therefore the level of user training required to effectively use the application/solution it is quite low.
Interoperability	Thanks to the fact that it is a Microsoft Azure cloud-based system, the integration with other systems is simplified by the tools that are available: LogicApp, as per the flows orchestration; DataFactory related to data integration; several connectors;

Table 33 CLSC Cross-cutting characteristics

2.4. MR Enabler Tool

2.4.1. Architecture overview

This module consists of two applications with similar functionalities and goals.

The first application is suitable for MR headsets, and the second is suitable for mobile devices. Both applications harness Mixed/Augmented Reality technology, which augments virtual content in the real world. Specifically, the end-user, using a mobile device or an MR headset (e.g. Microsoft HoloLens 2), can utilize the devices' cameras to observe a physical environment and augment it with 3D graphics, text, images, and sounds that appear on the screen and speakers of the mobile devices and on the MR headset, respectively.

Moreover, the end-user can interact with this augmented content through UIs and digital buttons by tapping the screen of mobile devices and using hand gestures, gaze interactions, and voice commands on MR headsets.

Both applications provide end-users with the ability to retrieve information from the DPP component of the Decentralized Traceability Platform.

Additionally, they provide access to DPP information as well as visualizations related to products along the value chain and the subsequent actions for these products.

The purpose of this module is to contribute to the intervention and engagement of end-users and to provide a more understandable presentation of the information and actions associated with a DPP.

2.4.2. Components and internal relationships

In the following paragraphs, the diagram of the components comprising the module's applications is depicted, along with the way they are interconnected and the sequence in which they are activated. Additionally, the functionality of each component is analyzed.

2.4.2.1. Components diagram

Figure 30 shows the flow diagram of the MR tool. Both mobile and headset-compatible applications follow a similar flow. The main differences appear during the visualization of the content as AR smartphones overlay digital content onto the real world using the phone's screen, whereas MR headsets integrate digital content more deeply with the real world by projecting holographic images directly into the user's field of view, allowing for more immersive and interactive experiences.

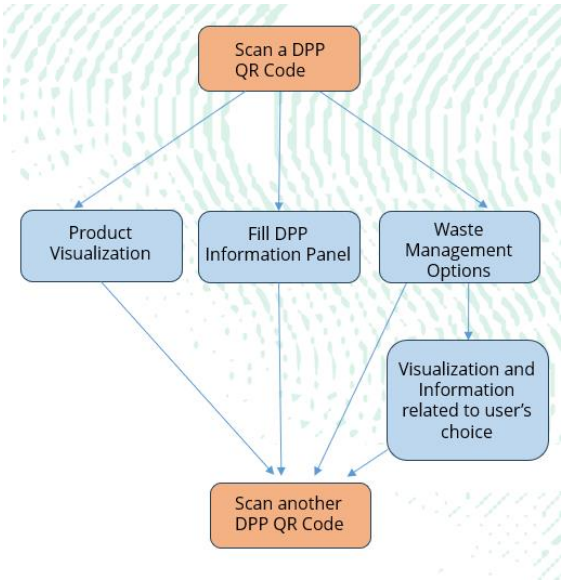

Figure 30 enabler tool flow diagram

2.4.2.2. Components description

To further assess the information within the DPPs and visualize the required data to the end-user, QR codes were utilized to track DPP information. The QR codes were selected for 2 primary reasons: 1) it is a well-known, mature, and accepted technology; 2) it is easier to scan via common Head Mounted Devices (such as, Microsoft Hololens 2[xxvii]).

According to the flow diagram, in the first step, the applications prompt the end-user to scan a DPP QR code located on the product (e.g. wood by-products, bins). When the end-user scans the DPP, visualizations related to the product are activated, along with some UI panels (digital menus containing visuals and texts) displaying DPP information and options (could appear as digital buttons in menus) either regarding possible future use of the product (e.g., wood chips may be used to produce plywood) or related to immediate next actions such as placement in a designated bin.

After the end-user selects one of the options or concludes the specified action, the corresponding visualizations and UI panels informing them about the outcomes of that specific option/action are activated.

Finally, if the end-user does not want to proceed with the selected option, they can return to the available options; otherwise, they proceed to confirm the option and then are prompted to scan the next DPP.

2.4.3. External dependencies

The MR enabler module is dependent on the DPP component of the Decentralized Traceability Platform, which integrates and incorporates all the other modules of DigInTrace. The way it depends on them involves retrieving DPP information as well as the role of end-users.

2.4.4. Technological decisions

For the implementation of the applications, the Unity3D software is utilized, which supports the C# programming language, enabling the creation of AR applications suitable for multiplatform mobile devices and MR applications for XR Head Mounted Devices (HMDs).

For AR functionality, the AR Foundation is employed, while specifically for the use of interactions on the HoloLens (hand gestures, menus, buttons, gaze interactions, voice commands, spatial sound), the Mixed Reality Toolkit 3 (MRTK 3) is utilized. MRTK 3 is based on OpenXR framework that widely supports many different HMD devices.

2.4.5. Technical specifications

The technical implementation of the applications includes algorithms for QR scanning, 3D models, and animations for the visualizations related to the product. Additionally, for communication with the DPP component of the Decentralized Traceability Platform, APIs for HTTP calls and JSON files format are utilized to retrieve the DPP information.

2.4.5.1. Operational characteristics

Operational architecture characteristics cover capabilities:

	Description
Availability	When an end-user wants to retrieve the DPP information from a product and learn regarding possible future use of the product

DigInTraCE

Robustness	Mandatory requirements: access to the internet/local network which supports Decentralized Traceability Platform services, sufficient battery, and moderate lighting conditions (avoiding extremes of brightness or darkness)

Table 34 MR Enabler Operational characteristics

2.4.5.2. Structural characteristics

Structural architecture characteristics cover capabilities:

	Description
Configurability	The root address linked to the endpoints of the services provided for information retrieval will be configurable.
Extensibility	The visualization components and functionalities are created in a generic and dynamic way so as to make it easy to plug in new pieces of functionality.
Portability	Unity supports multiplatform builds (Android, IOS, UWP) and OpenXR framework supports multiple HMD devices such as Microsoft Hololens 2
Maintainability	One common code base for both mobile and HMD implementation, supports ease of application maintenance and functionality enhancement.
Installability	Android and IOS mobile devices, HMDs supporting OpenXR (such as Microsoft Hololens 2).

Table 35 MR Enabler Structural characteristics

2.4.5.3. Cross-cutting characteristics

Cross-cutting architecture characteristics cover capabilities:

	Description
Authentication	Provide an authentication component through the Decentralized Traceability Platform.
Authorization	Provide an authorization component through the Decentralized Traceability Platform. (users access to specific functions/actions related to their role).
Privacy	Based on authorization component (users access to specific information related to their role).
Security	Communication between the AR/MR applications and the Decentralized Traceability Platform is implemented with HTTPS calls. Authentication and authorization are provided through the Decentralized Traceability Platform
Usability/Achievability	Suitable for novice users
Interoperability	Integration with the DPP component and other services of the Decentralized Traceability Platform

Table 36MR Enabler Cross-cutting characteristics

DigInTraCE

2.5. LCA Tool

2.5.1. Architecture overview

This module consists of a transparent and interoperable web platform which main objective is the traceability of various industrial processes, as well as the calculation of different environmental and circularity indicators through the usage of a friendly user interface (UI) that enables recording the stages, the input/output data and its quantities within the industrial processes the users are carrying out.

This information will help the user to reduce emissions and promote circularity by decreasing emissions and the use of secondary raw materials.

2.5.2. Components and internal relationships

The paragraphs below illustrate the components of the module's applications, showing how they are interconnected and the order in which they are activated. Additionally, each component's functionality is examined.

2.5.2.1. Components diagram

The structure and process of the LCA tool can be seen in Figure 31.

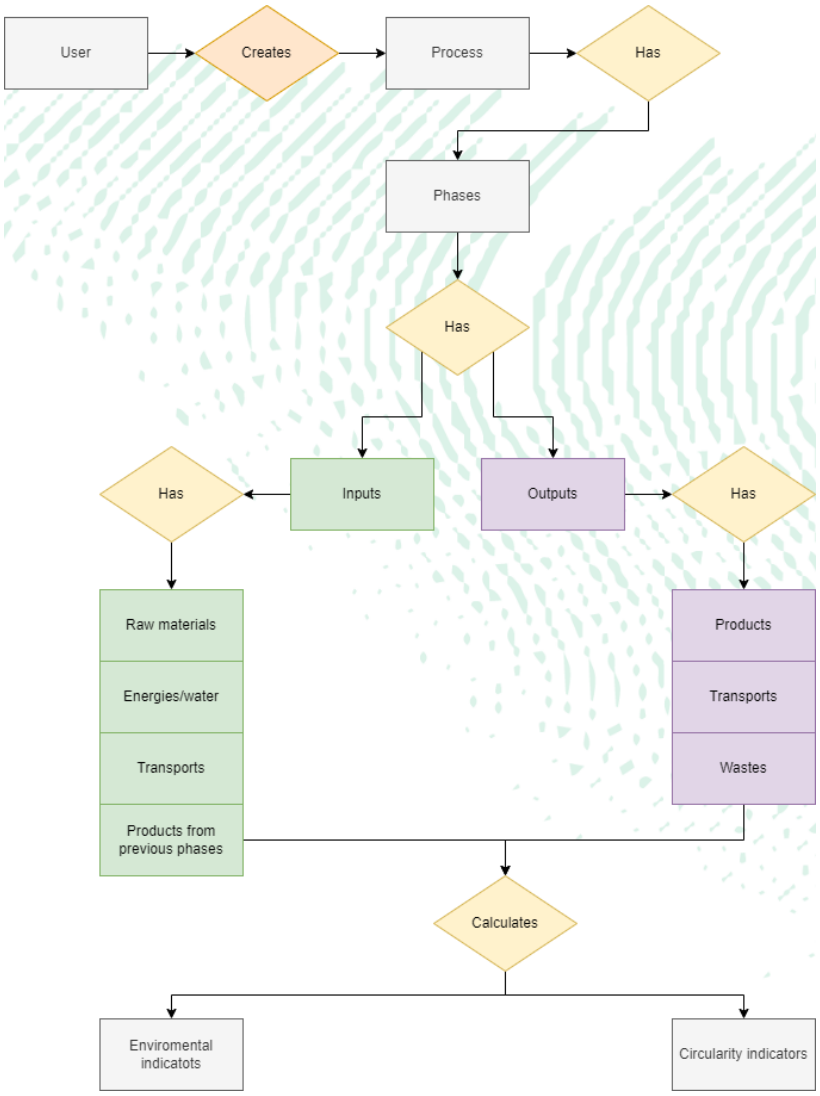

Figure 31: functionality diagram of LCA tool.

As shown in the diagram, a user creates a process, which in turn will have a series of stages representing the steps involved in creating a product. Each stage will have a series of inputs and outputs. The inputs can be classified into raw materials, energies/water, transport, and products from previous phases. As for the outputs, they can be products, transport, or wastes.

Once this information has been filled in, the user can calculate the impact of the circularity indicators and the environmental indicators.

2.5.2.2. Components description

Processes: a process can be created by a user, and this process will be the parent element that will contain the phases involved in the creation of a new product, a user can create as many processes as needed, but the calculation will be individual for each process.

Phases: a phase is a child component that can be created inside of a process, a user can create as many phases as needed, and each phase will have inputs and outputs,

this inputs and outputs will be used later to calculate the environmental indicators impact as well as the circularity indicators ones.

Inputs

- **Raw materials:** is an input that can be selected from a select form showing a list of raw materials. Once one has been selected, it will require some data like the kilograms, and additionally if needed the transport.
- **Energies/water:** is an input that can be selected from a select form showing a list of energies. Once one has been selected, it will require some data like the kilograms/kWh.
- **Products from previous phases:** if a product from a previous phase is needed on this phase, it can be added through this select input.
- **Transport:** When managing components such as raw materials, waste, or defining the end-of-life scenario, there is an option to add various transportation modes. The required specifications vary depending on the type of transportation:
    - **Air and Maritime Transport**: Only the distance in kilometres is needed.
    - **Ground Transport (Lorries)**: The following details are required:
        - Type of lorry
        - Distance in kilometres.
        - Percentage of material transported.
        - European regulation

Outputs

- **Products:** is an input that can be selected from a select form showing a list of products. Once one has been selected, it will require some data like the kilograms.
- **Wastes:** is an input that can be selected from a select form showing a list of wastes. Once one has been selected, it will require some data like the kilograms, treatment type and transports.
- **Transport:** works the same way as the transport from the inputs.

Calculations:

Once the previously seen parameters are set, the user will be able to calculate the desired impacts related to this inputted data, e.g. in the case of wanting to estimate an environmental indicator, one among other options in a list must be chosen, then, 4 graphics will display the information associated to these impacts (one general and three more detailed).

Regarding circularity indicators, the user needs to select the impact that wants to calculate, and a donut graphic will display the calculus related to the selected circularity indicator.

DigInTraCE

2.5.3. Technological decisions

When developing the web application, the technology was carefully selected to ensure high performance, scalability and ease of maintenance. Here is a brief explanation on why each component was selected.

- **React for the Frontend:** React was chosen for its flexibility and efficiency in building dynamic user interfaces. It allows the creation of reusable components, making the development process faster and more manageable. The virtual DOM in React improves performance by minimizing direct manipulation of the actual DOM. Additionally, React's extensive ecosystem of libraries and tools enhance development capabilities.
- **Node.js for the Backend:** Node.js was selected for its ability to handle asynchronous operations efficiently, making it ideal for building scalable network applications. Its non-blocking I/O model ensures that the application remains responsive even under heavy load. Furthermore, using JavaScript for both frontend and backend simplifies the development process and allows sharing code easily between client and server thus increasing productivity.
- **PostgreSQL as the Database:** PostgreSQL was chosen for its robustness, reliability, and advanced features. As an open-source relational database, it offers strong support for ACID transactions, ensuring data integrity. Its powerful querying capabilities and support for complex data types make it well-suited for handling the diverse and complex data structures of the application. Additionally, its extensibility allows the customization to meet specific needs.
- **NginX as the Web Server:** NginX was selected for its high performance, stability, and low resource consumption. It excels at handling concurrent connections, making it ideal for serving static content and acting as a reverse proxy. NginX's load balancing and caching capabilities improve the overall performance and reliability of the web application. The simplicity and flexibility in Its configuration also make it a preferred choice for modern web applications.
- **Docker for Containerization:** Docker was chosen to streamline the deployment process and ensure consistency across different environments. By containerizing the application, the team can package all dependencies and configurations ensuring its operation in different systems regardless of the environment. Docker enables easy scaling and orchestration, allowing efficient management and deployment of multiple instances of the application. Furthermore, Docker's compatibility with various cloud platforms enhances the application's portability and deployment flexibility.

2.5.4. Technical specifications

The technical implementation of the application includes algorithms for QR code creation, data reading, data calculation, and a backend API connection that allows interaction with the platform via endpoint calls, where information will be sent in JSON format.

DigInTraCE

2.5.4.1. Operational characteristics

Operational architecture characteristics cover capabilities:

	Description
Availability	If the server is not in maintenance a user would be able to access and use the tool any time.
Robustness	Error handling, logging, and monitoring, input validation, authentication and authorization, data encryption, anti-injections, cross-browser compatibility, responsive design.

Table 37 LCA Tool Operational characteristics

2.5.4.2. Structural characteristics

Structural architecture characteristics cover capabilities:

	Description
Configurability	The root address linked to the endpoints of the services provided for information retrieval will be configurable.
Extensibility	The visualization components and functionalities are created in a generic and dynamic way to make it easy to plug in new pieces of functionality.
Portability	React supports all modern browsers.
Maintainability	Platform code is modular and responsive, supporting ease of application maintenance and functionality for all kind of devices.
Installability	The web platform is accessible from any browser as long as there is internet access. The code is containerized with Docker to facilitate deployment in any environment.

Table 38 LCA Tool Structural characteristics

2.5.4.3. Cross-cutting characteristics

Cross-cutting architecture characteristics cover capabilities:

	Description
Authentication	Provide an authentication component through the Web Platform.
Authorization	Provide an authorization component through Platform. (users access to specific functions/actions related to their role).
Privacy	Based on authorization component (users access to specific information related to their role).
Security	Communication between the Platform and backend is implemented with HTTPS calls. Authentication and authorization are provided through the Platform.
Usability/Achievability	Intuitive UI suitable for novice users.
Interoperability	Web platform is accessible from any browser.

Table 39 LCA Tool Cross-cutting characteristics

DigInTraCE

3. Conclusions and future steps

In summary, D2.9 provides a technical vision of several of the components that make up the platform, the requirements necessary for its implementation and the functionalities they offer to the entire platform. This v1 represents the basis on which to build digital solutions and allows us to identify gaps or integration needs.

Throughout the document, and for each of the digital tools, relevant information has been provided in the form of functional or technical requirements, functionalities and technical diagrams, necessary to understand what they contribute to the context of the project and how they will be used to generate or obtain financial information, securely create DPPs, manage supply chain information or visualize through extended reality the information managed, together, by the DigInTraCE platform.

At the time of creation and delivery of this deliverable, there are still needs to be defined in more detail, possible changes due to the identification of new requirements in the use cases and demonstrators and other improvements that will make the DigInTraCE platform more robust and efficient. All these changes will be reflected in the final version of this deliverable.

DigInTraCE

Disclaimer of Warranties

The information and views set out in this deliverable are those of the authors and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the following information.

This project has used a standard methodology already developed in the CIRPASS project (Grant Agreement number: 101083432), following EU recommendations. Ad hoc modifications were added to comply with the Grant Agreement conditions for CIRPASS project (Grant Agreement number: 101083432).

This project has used a standard methodology already developed in the DigInTraCE project (Grant Agreement number: 101091801), following EU recommendations. Ad hoc modifications were added to comply with the Grant Agreement conditions for DigInTraCE (Grant Agreement number: 101091801).

DigInTraCE

REFERENCES

[i] https://unity.com/

[ii] https://www.khronos.org/openxr/

[iii] https://commission.europa.eu/energy-climate-change-environment/standards-tools-and-labels/products-labelling-rules-and-requirements/sustainable-products/ecodesign-sustainable-products-regulation_en

[iv] https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52022PC0142

[v] https://cirpassproject.eu/

[vi] https://cirpassproject.eu/wp-content/uploads/2023/03/CIRPASS_Benchmark-of-existing-DPP-oriented-reference-architectures.pdf

[vii] https://gaia-x.eu/

[viii] https://www.w3.org/TR/vc-data-model-2.0/

[ix] https://www.w3.org/TR/did-core/

[x] https://www.apm.org.uk/resources/find-a-resource/agile-project-management/

[xi] https://www.google.es/books/edition/OSI_Reference_Model_for_Telecommunicatio/DLNpjT3K4ooC?hl=es&gbpv=1&dq=OSI+model+reference&printsec=frontcover

[xii] https://www.enisa.europa.eu/topics/incident-response/glossary/public-key-infrastructure-pki

[xiii] https://c4model.com/

[xiv] https://www.hyperledger.org/projects/fabric

[xv] https://ref.gs1.org/standards/epcis/

[xvi] https://www.vaultproject.io/

[xvii] https://hyperledger.github.io/fabric-gateway/

[xviii] https://medium.com/@sbmeunier/when-do-you-need-blockchain-decision-models-a5c40e7c9ba1

[xix] https://connect.comptia.org/content/infographic/blockchain-decision-tree

[xx] https://www.weforum.org/agenda/2018/04/questions-blockchain-toolkit-right-for-business/

[xxi] https://ethereum.org

[xxii] https://polygon.technology/polygon-pos

[xxiii] https://solana.com

[xxiv] https://www.hyperledger.org/blog/2023/02/16/benchmarking-hyperledger-fabric-2-5-performance

[xxv] https://www.ssl.com/faqs/what-is-https/

[xxvi] https://grpc.io/docs/guides/auth/

[xxvii] https://www.microsoft.com/hololens