2024

Online platform document

D4.2 Online platform design document & publishing code libraries

Co-funded by the European Union

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 the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for

Table of Contents

Document Change History

Version	Date	Author (organisation)	Description
V1.0	21 May 2024	Mikkel Egehave & Chris Mora-Jensen (Diplomasafe)	First draft
V2.0	28 May 2024	Helen Gray, Rakel S. Hallgrimsdóttir and Fjalar Jónsson (Idan) Bianca Marie Bukh Lauridsen (UCN) Silvia Pop, Ben Stonnell and Thomas Wiegerinck (Ecolab)	Partners' feedback
V3.0	31 May 2024	Mikkel Egehave & Chris Mora-Jensen (Diplomasafe)	Final draft

List of Terms and Abbreviations

API	Application Programming Interface
CTDL	Credential Transparency Description Language
DID	Decentralised Identifier
DS	Diplomasafe
EBSI	European Blockchain Services Infrastructure
ECTS	European Credit Transfer and Accumulation System
ECVET	European Credit System for Vocational Education and Training
EDC	European Digital Credentials for Learning
EQF	European Qualifications Framework
ESCO	European Skills, Competences, Qualifications and Occupations
EU	European Union
EUDI	European Digital Identity
GDPR	General Data Protection Regulation
HR	Human Resources
JWT	JSON Web Tokens
MC	Micro-credential
REST	Representational State Transfer
URI	Uniform Resource Identifier
VC	Verifiable Credential
VET	Vocational education and training
VP	Verifiable Presentation
WCAG	Web Content Accessibility Guidelines
WP	Work Package

List of Figures

1. Introduction

The MCEU Hospitality project, funded under the Erasmus+ programme, is a catalyst for the advancement of digital and green skills within the hospitality sector across Denmark, Iceland, and Spain. By focusing on equipping 500 learners across Europe with micro-credentials (MCs), this initiative is a significant step towards a more sustainable and technologically adept workforce, contributing to the overall growth and development of the sector.

Work Package 4 (WP4) is central to the project's objectives, which involve developing a Credentials-Issuing Online Platform (Platform). This Platform is designed to streamline the process of creating, storing, sharing, and verifying certificates within the hospitality sector, ensuring user-friendliness, reliability, interoperability, and compliance. Diplomasafe, a trusted expert in Verifiable Credentials (VCs) issuance, leads this task, from defining the technical requirements to setting up and issuing credentials.

The Platform's importance within the MCEU Hospitality project is underscored by its role in advancing digital and green skills, as well as promoting the project's methodology to stakeholders in the industry. It harnesses the power of tools such as the European Digital Credentials for Learning (EDC) and the European Blockchain Service Infrastructure (EBSI), ensuring robustness, security, and interoperability with existing services. These tools are not just features, but integral components that enhance the Platform's functionality and effectiveness.

The platform will also comply with the Council of the European Union's (EU) "Recommendation on European approach to micro-credentials for lifelong learning and employability," adopted on 16 June 2022. This document sets a defined structure for micro-credentials, principles for their design and issuance, and quality assurance mechanisms. In addition, the platform will issue credentials aligned with frameworks like the European Qualification Framework (EQF) and the European Credit Transfer and Accumulation System (ECTS). Thus, enhancing collaboration between universities, Vocational education and training (VET) providers, and industries ensures that micro-credentials are fulfilling descriptions.

The Online platform design document and publishing code libraries (Deliverable 4.2) outlined in this document provide a detailed overview of the Platform design, including its architecture, data models, exchange protocols, code repository, and integration requirements with key platforms like EDC, Europass, and EBSI. This document serves as a foundational guide for the construction of the MCEU platform from July 2024 and the credential issuance during the pilot stage in 2025.

The document begins by describing the technical specifications of the Viewer and Issuer systems using visual high-level diagrams. Then, it delves into the outline of the data model and how the Platform will integrate with EDC, Europass, and EBSI. It also describes the code repository under which the Platform code will be developed starting in July 2024.

2. Data Model

The Platform data model is structured in three main sections: Viewer System, Issuer System, and Rest API System. Each system will be detailed further in Section 3, where the technical specifications for the development of the Platform are set out. Below, we describe the Data Model structure:

Viewer System: The Viewer System is a web application that allows users to manage credentials in JSON-LD format or store them in a wallet. It enables the visualisation, verification, and export of these credentials, with sharing capabilities available.

Issuer System: The Issuer System is a robust web application specifically designed to manage and issue credentials. It utilises the JSON-LD format for standardisation and interoperability. This system empowers VET Providers to create credential templates, issue credentials, and manage user roles and access. With its enhanced security protocols, credentials are stored and managed, and shared with explicit learner consent, ensuring utmost data security. The system also facilitates comprehensive user management, including capabilities to create, read, update, delete, and assign roles to users, ensuring tailored access and functionality within the application.

Rest API System: The Rest API System provides essential functionalities for managing user interactions and credential operations for both the Viewer and Issuer systems. It handles user requests, credential issuance, sharing, and verification, seamlessly integrating with a wallet system to store credentials in JSON-LD format. This centralised API ensures robust communication and data exchange across systems, facilitating secure and efficient operations tailored to support viewer access and issuer management.

2.1 GDPR Compliance

The Platform's data model is designed with a steadfast commitment to respecting the General Data Protection Regulation (GDPR) rules. The GDPR, a law of the European Union (EU), sets stringent standards for the collection and use of personal data of people in the EU. This means that the Platform handles user data with utmost care and ethics, meeting the high benchmarks that the GDPR demands to protect the rights and privacy of people.

The Platform's data model is meticulously crafted to ensure it adheres to the GDPR rules on data privacy and protection, demonstrating the Platform's unwavering commitment to handling user data with caution and ethics, in line with the strict standards required by the GDPR to protect the rights and privacy of people.

3. Technical Specifications

The Platform comprises two primary systems: the Viewer System and the Issuer System, both adopting TypeScript language and Vue.js framework. The Platform incorporates tools such as Vue router, Pinia, Vitest, and Cypress to streamline and ensure the quality of the MCEU project.

3.1 Viewer (System)

3.1.1. Tech stack

- **Language**: TypeScript
- **Framework**: Vue.js
- **Additional Tools**:
    1. Vue router: The official router for Vue.js.
    2. Pinia: The store library for Vue, provides a centralised state management solution that is simple and more easily integrated with TypeScript compared to its predecessor, Vuex.
    3. Vitest: A Vite-native unit test framework that provides a fast testing environment, capable of handling Vue component testing.
    4. Cypress: An end-to-end testing framework that enables you to write tests that run in a browser, ensuring the application works as expected from a user's perspective.

3.1.2. System breakdown

The Platform is organised in the following sections:

- **Login Page**: This is the entry point to the application. The login functionality will authenticate users based on their credentials; DID or Prof of Proof of Identity (email and password). If a user enters the correct credentials, the system will allow them to interact with the application.
- **FAQ & Support Page**: This page is designed to help users find quick answers to their questions and to request further assistance if needed. Here's what it includes:
    - **FAQ List:** This is a pre-compiled list of frequently asked questions (FAQs) along with their answers. The questions could be about how to use the Issuer app, common troubleshooting steps, user and credential management, etc. This list serves to provide immediate answers to common queries users may have.
    - **Support Email Form:** Below the FAQ list, a form where users can request additional support. It will have subject and description fields and after filling out the form, the user can hit the "send" button to submit their support request. This will trigger the system to send an email to the VET Provider team with the details provided.
- **Credential List Page:** This will be a page where the Learner will see all his credentials issued by the VET Provider. List in a table-like view.
- **Credential Detail Page:** Page with all the credential details and buttons for download and setup or update the share view that uses the Share view Component

- **Share View Component:** This is a form model component for creating and updating the share view which is used by Employers to view and verify the Learners' credentials.
- **Third-Party View Page:** This page is accessible for each credential with a share view setup. The **Employees** can view, validate and verify the **Learners'** credential.
- **Learner Profile Update Component:** This will be accessible from every page if the learner is logged in the application. It will be a form modal that can update the **Learner** information.

3.2 Issuer (System)

3.2.1. Tech stack

- **Language**: TypeScript
- **Framework**: Vue.js
- **Additional Tools**:
    1. Vue router: The official router for Vue.js.
    2. Pinia: The store library for Vue, provides a centralised state management solution that is simple and more easily integrated with TypeScript compared to its predecessor, Vuex.
    3. Vitest: A Vite-native unit test framework that provides a fast testing environment, capable of handling Vue component testing.
    4. Cypress: An end-to-end testing framework that enables you to write tests that run in a browser, ensuring the application works as expected from a user's perspective.

3.2.2. System breakdown

- **Login Page:** This is the entry point to the application. The login functionality will authenticate users based on their credentials; email and password. If a user enters the correct credentials, the system will allow them to interact with the application based on the privileges set by their user role.
- **FAQ & Support Page:** This page is designed to help users find quick answers to their questions and to request further assistance if needed. Here's what it includes:
    - **FAQ List:** This is a pre-compiled list of frequently asked questions along with their answers. The questions could be about how to use the Issuer app, common troubleshooting steps, user and credential management, etc. This list serves to provide immediate answers to common queries users may have.
    - **Support Email Form:** Below the FAQ list, there is a form where users can request additional support. It will have subject and description fields and after filling out the form, the user can hit the "send" button to submit their support request. This will trigger the system to email the support team with the details provided.
- **User Management Page:** Page button for opening a create from modal and with a table that shows user information fetched from the RestAPI. Every table row will have action buttons for updating and deleting users. Here, they can:

- **Create (C):** Add new users to the system by filling in necessary details like name, username/email, password, and role (Admin or VET Provider).
- **Read (R):** View the list of all users in the system, along with their details.
- **Update (U):** Edit the details of existing users, such as their role, password, or contact information.
- **Delete (D):** Remove users from the system if needed.

- **Assign roles:** Set the user type for each user – either Admin or VET Provider. This user type determines the user's access within the system.
- **Credential Template Page:** This page is where VET Providers can create and manage templates that are crucial for credential issuing. They can:
  - **Create (C):** Add new diploma templates from a form modal to the system by filling out the necessary details and setting the design.
  - **Read (R):** View the list of all credential templates available and their details.
  - **Update (U)**: Edit the details of existing templates from a form modal triggered from the listing table button.
  - **Delete (D):** Remove unwanted templates from the system if and only if there is no credential issued with the template.
- **Credential Issue Page:** Form page from issuing credentials based on the previously created templates. They can:
  - Select a template from a dropdown list containing all available templates.
  - Fill in the required information (such as learner details, completion date, etc.) to issue a diploma either from a file or based on filled fields (depending on the selected radio button).
  - Once all necessary details have been filled in, they can proceed to issue the credential.
- **User Request Page:** A page for viewing and managing user requests consisting of a table with all user requests that have information and approve and decline action buttons for every request. They can:
  - Respond to these requests by clicking either of the two buttons: "Approve" or "Decline", thereby managing user access/privileges in the system.
- **Credential Management Page**: This Diploma Management feature will let VET Providers have granular control over the credentials they issue and manage. This page will provide the option to manage previously issued diplomas. VET Providers can:
  - **Read:** View the details of all issued credentials in a table format. This could include details like the learner's name, course completed, date of issuance, and status of the diploma (active/revoked).
  - **Revoke:** This option allows the VET Providers to invalidate a previously issued credential. Upon revoking, the credential's status would change from active to revoked.
  - **Delete:** Completely remove a diploma record from the system.

- **Share:** This specific functionality would be triggered by opening a modal where the VET Provider can enter a description and provide email addresses or users to share the credentials with. The sharing of a credential would be subject to the learner's prior consent that is set when claiming the credential or in the credential setting in the Viewer System. The shared info will include the learner's name, the course they've completed, and other details on the credential if needed.

3.3 Rest API (System)

3.3.1. Tech stack

- **Language**: TypeScript
- **Framework**: NestJs: Build for scalable server-side applications. It is heavily inspired by Angular and supports TypeScript out of the box. NestJs embraces good programming practices, such as SOLID principles and Dependency Injection.
- **Additional Tools**:
  - Vitest: Also used on the backend, allowing for a consistent testing strategy across the full-stack of your application.
  - Mikro-orm: A TypeScript ORM for Node.js based on the Data Mapper pattern. It can work well with TypeScript due to its emphasis on strong typing and supports multiple SQL and NoSQL databases.

3.3.2. System breakdown

- **Authentication Services**
  - **Viewer Auth Guard:** Ensures that Viewers are authenticated and authorised to access the viewer-related features.
  - **Issuer Auth Guard:** Validates the credentials of Vet Providers to ensure they have the necessary permissions to manage credentials and interact with the system.
- **Activity Logging Services**
  - **Viewer Activity Log Service:** Monitors and records all activities taken by viewers within the system, storing this information for audit and tracking purposes.
  - **Issuer Activity Log Service:** Keeps a log of all issuer activities, providing an audit trail of actions such as credential issuance and management.
  - **Activity Log Repository:** Centralises the storage of activity logs for both viewers and issuers, facilitating easy retrieval and analysis of log data.
- **User Management Services**
  - **User Management Controller:** Provides a full suite of user management capabilities, including the creation, listing, updating, and deletion of user accounts and roles.
  - **FAQ Controller:** Delivers a listing service for frequently asked questions, allowing users to find answers to common queries.

- **Support Email Controller:** Handles incoming support requests by sending emails to the designated support team.

- **Credential Services**
  - **Credential Management Controller:** Enables Vet Providers to manage digital credentials, including viewing, revocation, **and** deletion.
  - **Credential Controller:** Allows authorised users to interact with credentials, including downloading, viewing, and deletion.
  - **Credential Third Party View Controller:** Grants third-party entities the ability to view credentials.
  - **Credential Template Controller:** Manages the lifecycle of credential templates, including creation, update, and deletion.
  - **Credential Issue Controller:** Responsible for the issuance of credentials and for sending notification emails to the learners.
  - **Credential Verification Controller:** Validates the authenticity and integrity of learner credentials.
  - **Credential Share Controller:** Facilitates the sharing of credentials with employers and other interested parties.
  - **User Requests Controller:** Manages user requests by providing listing functionalities and allowing learners to create, approve, and decline requests.

- **Communication Services**
  - **Email Sending Service:** It centrally manages the dispatching of emails throughout the system, ensuring timely communication with users.

3.4 Authentication and Authorization

- Keycloak, an open-source identity and access management solution, deals with the platform's authentication and authorisation services.
- **Extensions:** grnet/eidas-keycloak-extension: Allows Keycloak to integrate with the eIDAS system. To comply with eIDAS standards for electronic identification and trust services.

3.5 Database

PostgreSQL, a powerful, open-source object-relational database system, will store and manage the data workload of the platform (RestAPI and Keycloak).

3.6 High-level diagrams

In this section, we present the visual representation of the user journeys from the perspective of the Issuer (VET providers), the Employers and the Learners, as well as how they relate to each other through the platform.

In **Figure 1**, we illustrate the connections and interactions between various stakeholders in the platform ecosystem. At the centre of the diagram is the platform, which serves as the hub for managing and issuing

digital credentials. Connected to the platform are three key user groups: employers, learners, and VET providers.

Employers can use the platform to view and verify the credentials of learners, while learners can use it to manage and share their credentials with potential employers. VET providers, on the other hand, can use the platform to issue and manage digital credentials for their learners. The platform is also connected to the EBSI and the Email sending system, which facilitate secure data exchange and timely communication with users, respectively.

Figure 1: System Context Credentials Issuing Online Platform

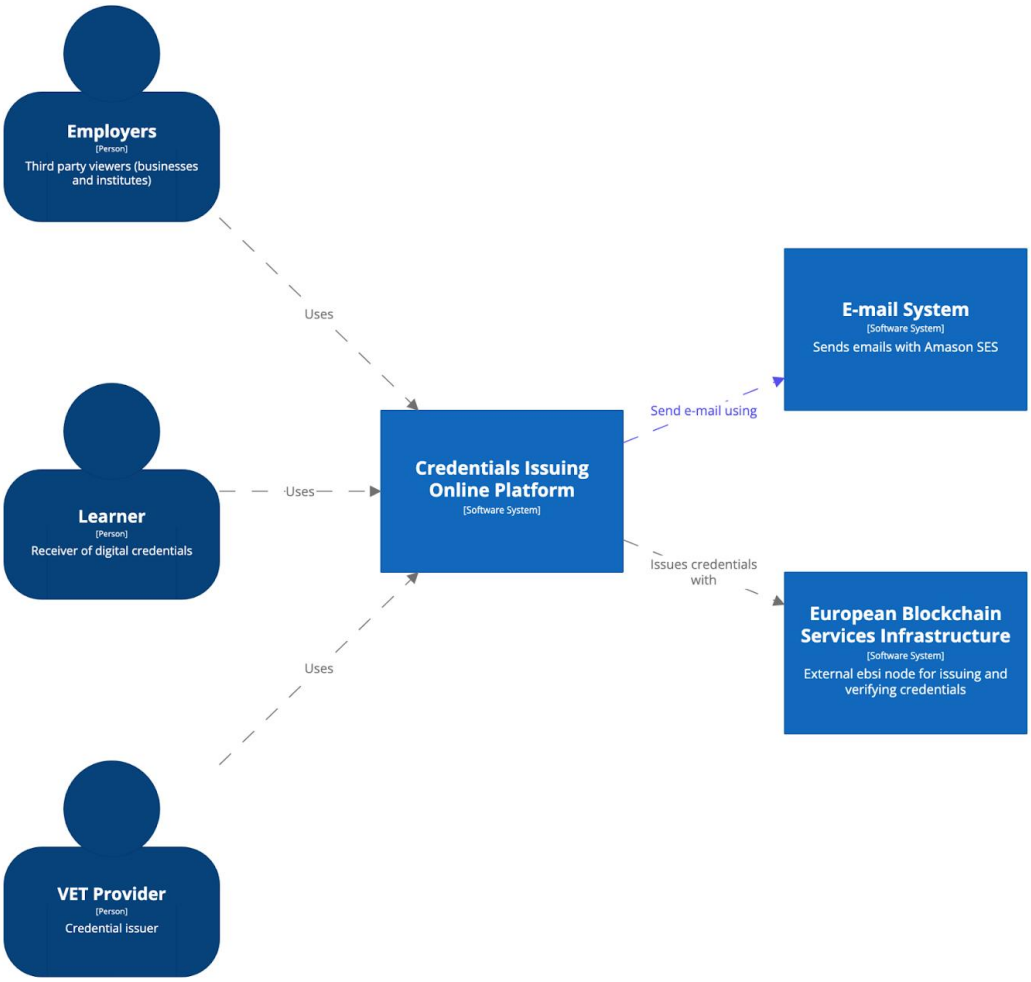


Figure 2 illustrates the connections and interactions between various components of the platform ecosystem. In the diagram is the issuing system, which is used by the VET provider to issue and manage digital credentials. The issuing system is connected to the keycloak service, which handles authentication and authorization, and the rest API, which facilitates data exchange with other systems. The keycloak service is connected to the user database, while the rest API is connected to the user database, credential database, EBSI, and email system.

Additionally, an employer is connected to the viewer system, which is used to view and verify credentials. The viewer system is connected to both the keycloak service and the rest API. Lastly, a learner is connected to the viewer system as well as a wallet system, which is used to store and manage their credentials. The wallet system

is connected to the rest API, allowing for seamless data exchange between the various components of the platform.

Figure 2: Connections and Interactions Between Components of the Platform Ecosystem

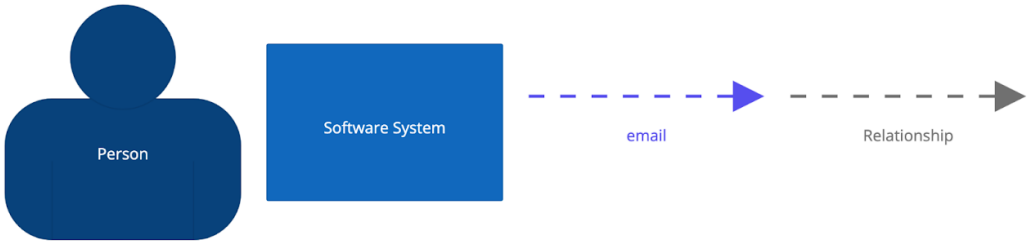


Figure 3 shows the relationship between some of the main internal components in the viewer container. The viewer container interacts with the Rest API to serve both the Employer and the Learner in order for the components to serve their purpose. Additionally, the Learner will interact with the Keycloak service through the Authentication Component.

The Viewer System is a web application that allows users to manage credentials in JSON-LD format or stored in a wallet. It enables the visualisation, verification, and export of these credentials, with sharing capabilities available. The Viewer System comprises several components, including the Login Page, FAQ & Support Page, Credential List Page, Credential Detail Page, Share View Component, Third-Party View Page, and Learner Profile Update Component. These components work together to provide a seamless and user-friendly experience for managing and sharing credentials.

Figure 3: Interaction between the Platform and Ecosystem

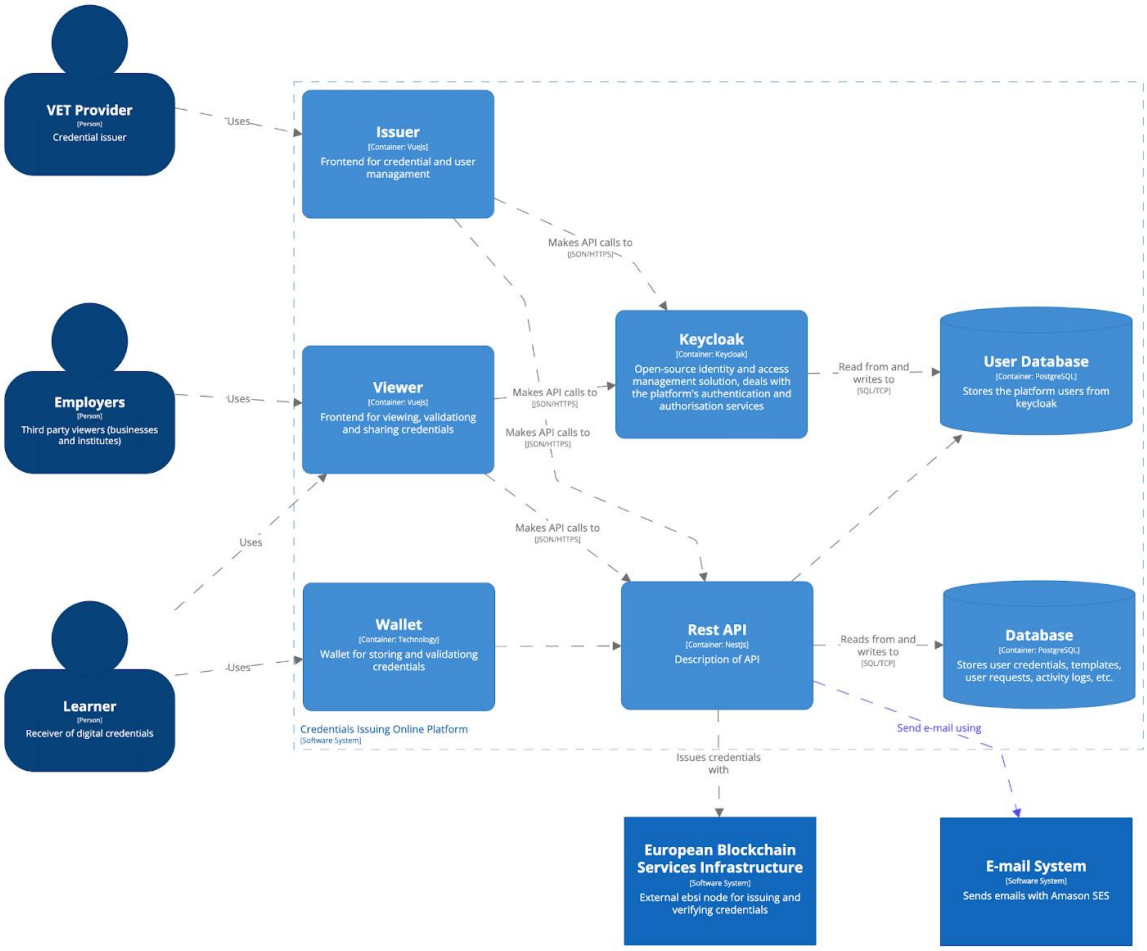
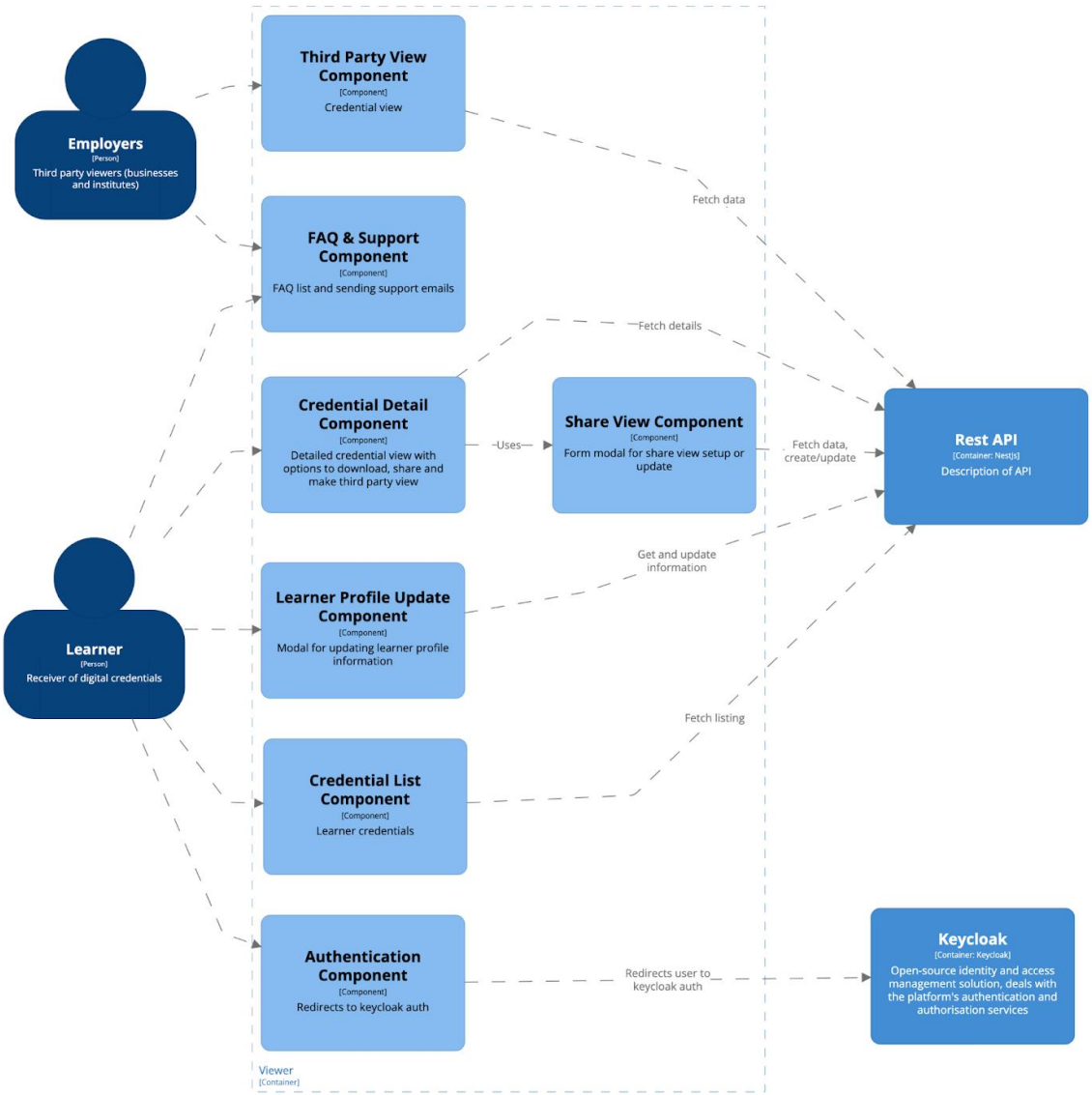


Figure 4: Credentials Issuing Online Platform Viewer Diagram



The diagram in Figure 5 shows the main components of the Issuer container in the platform, all working together with the Rest API. The only exception is the KeyCloak, which is invoked through the authentication component in the issuing container. The authentication component will, if the VET provider is successfully authenticated, be granted access to the issuing functionality. The Issuer System is a comprehensive web application designed for managing and issuing credentials, using JSON-LD format for standardisation and interoperability.

This system allows VET Providers to create credential templates, issue credentials, and manage user roles and access. With enhanced security protocols, credentials are securely stored and managed and only shared with explicit learner consent. The system also facilitates robust user management, including capabilities to create, read, update, delete, and assign roles to users, ensuring tailored access and functionality within the application.

Figure 5: Issuer diagram

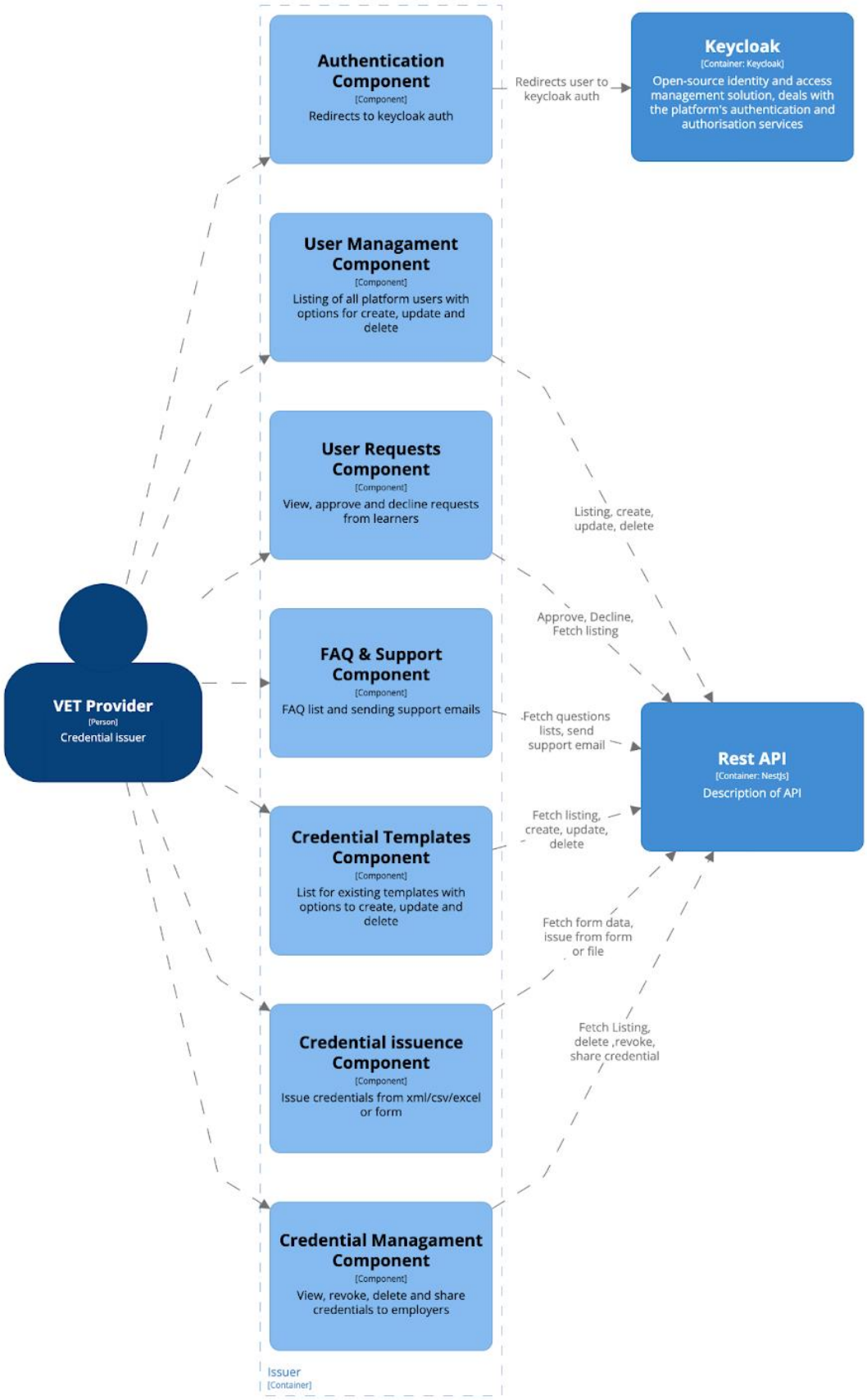
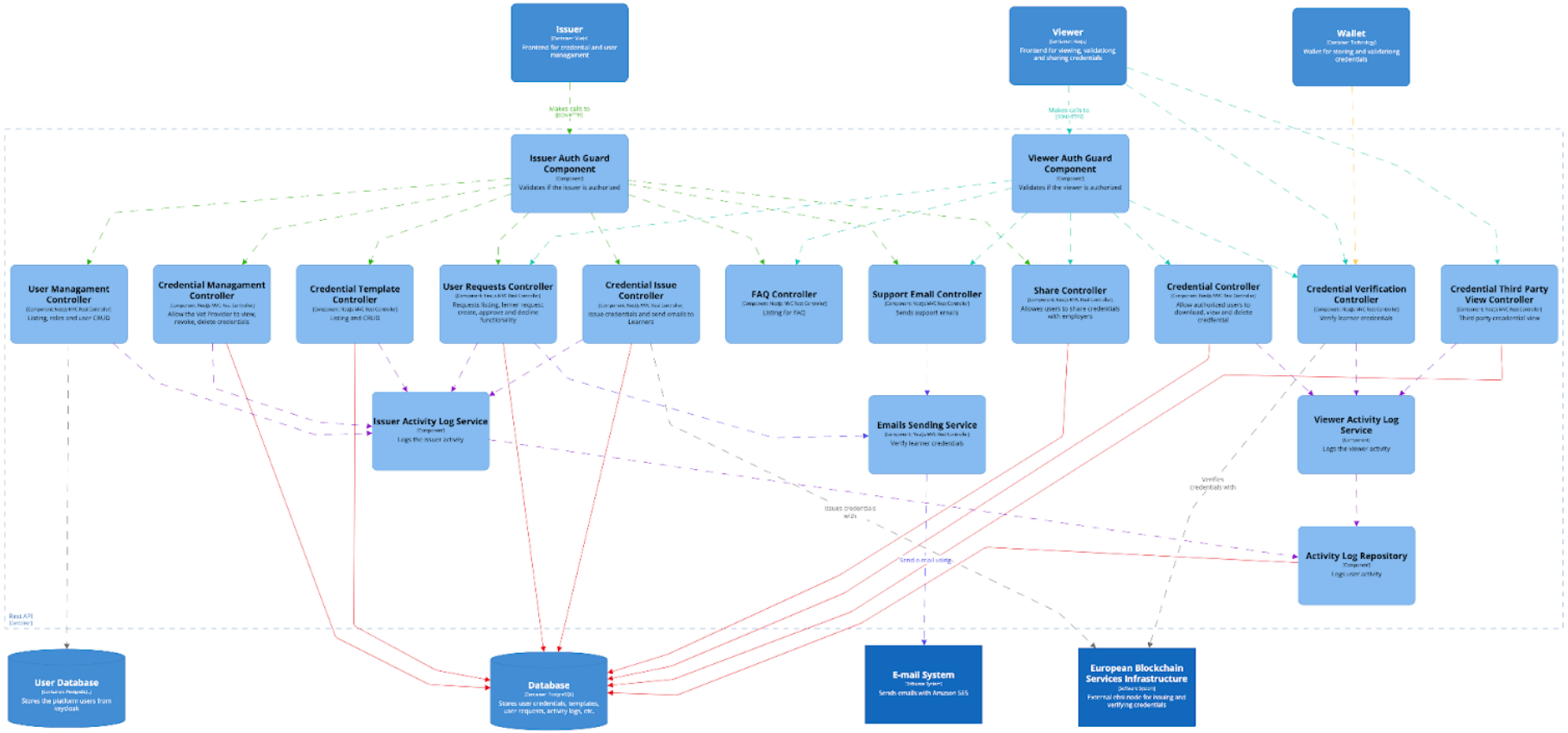


Figure 6: Rest API diagram



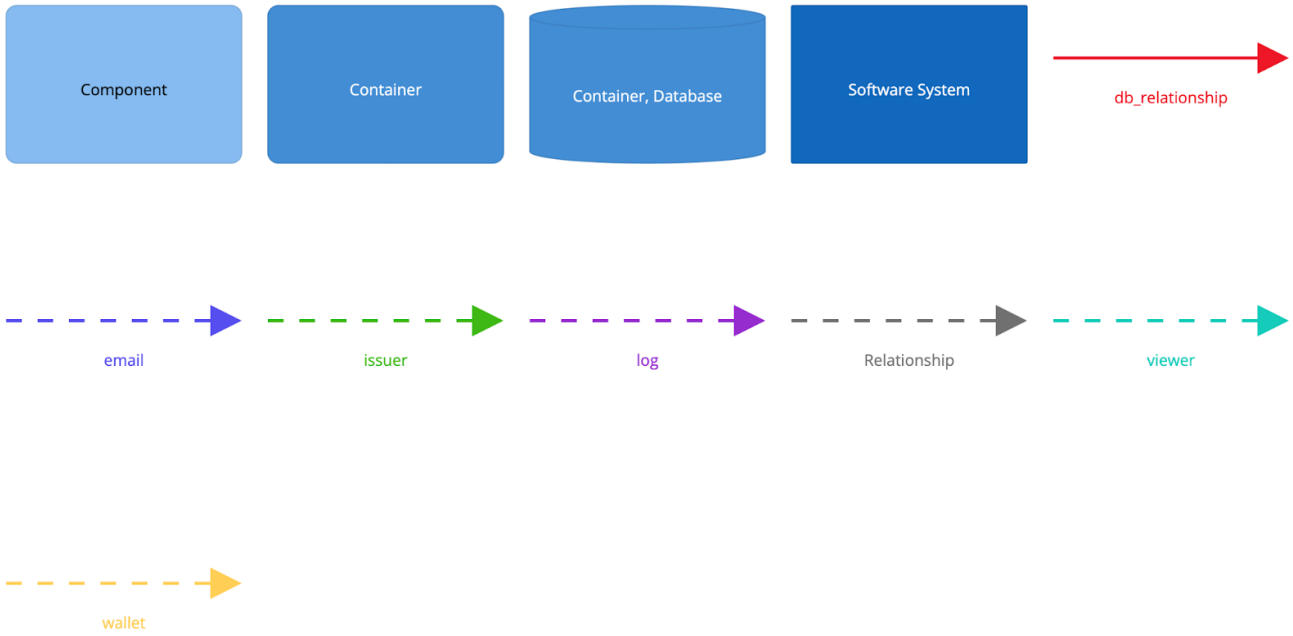
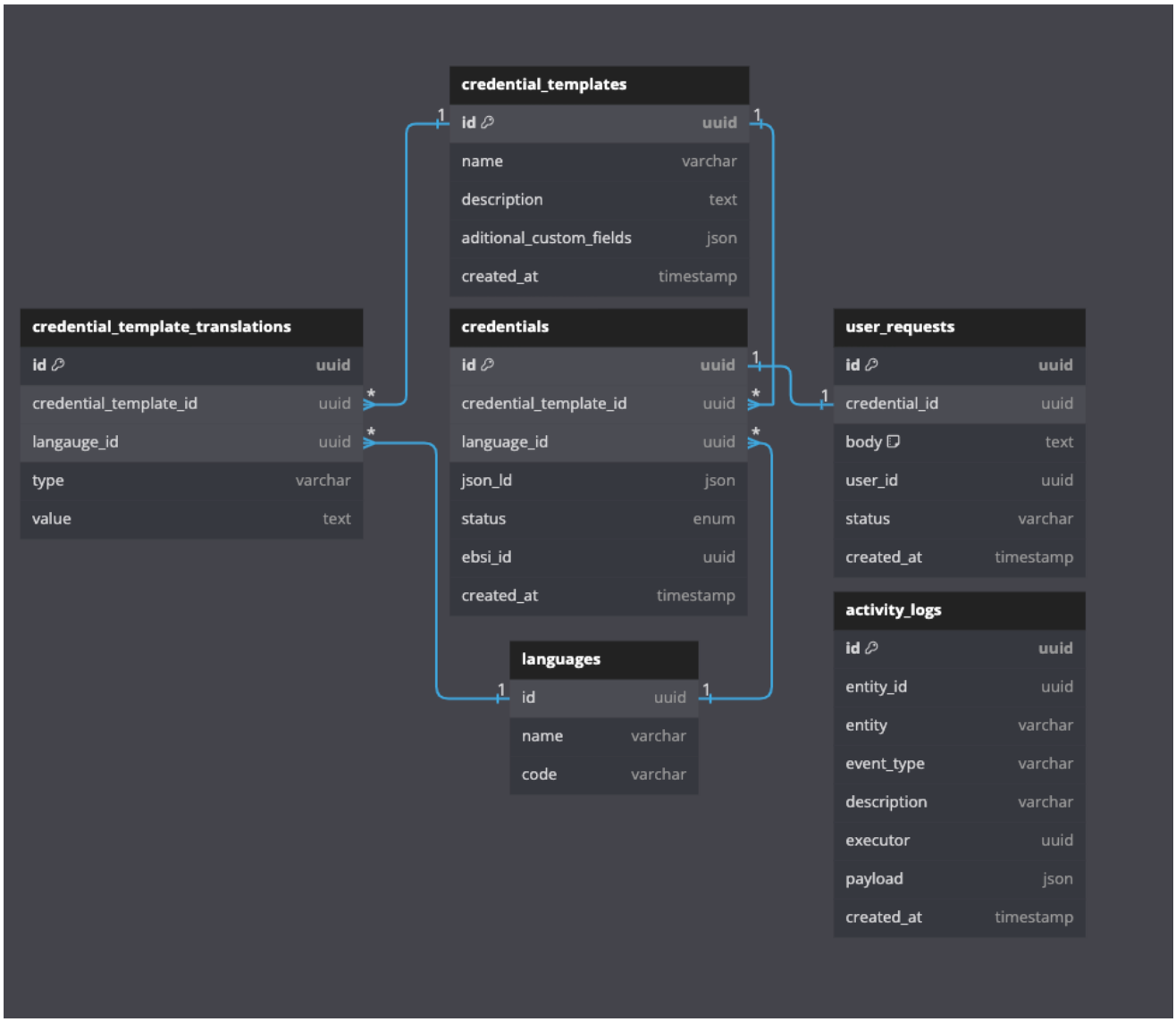


Figure 6 shows the internal components of the Rest API, which is exposed towards the issuer, viewer, and wallet on the top level while it is exposed to the databases, email system, and the Blockchain service on the bottom level. Having the external components internally connected should provide a seamless data exchange throughout the platform.

The Rest API System provides essential functionalities for managing user interactions and credential operations for both the Viewer and Issuer systems. It handles user requests, credential issuance, sharing, and verification, seamlessly integrating with a wallet system to store credentials in JSON-LD format. This centralised API ensures robust communication and data exchange across systems, facilitating secure and efficient operations tailored to support both viewer access and issuer management.

Below in Figure 7 we present the database model serving the platform excluding the authentication part as this is handled by the Keycloak service. It contains data and fields for the credential templates and data for the credentials (denoted as the json_ld field for europass interoperability). The use of the json_ld field for europass interoperability ensures that the data is stored in a standardised format, allowing for seamless data exchange and compatibility with other systems.

Figure 7: Database



4. Integration with Europass, EDC, and EBSI

The MCEU platform seamlessly integrates with Europass, EDC, and EBSI, adhering to the guidelines set forth in Deliverable 4.1. This connectivity ensures robust data exchange and compatibility, enhancing the platform's performance and efficiency.

The EDC are statements issued by educational institutions to learners, documenting their learning achievements. They certify qualifications, activities, assessments, and entitlements, offering instant and automatic verification checks to ensure their authenticity and legality. These digital credentials hold the same legal validity as traditional paper-based certificates and are recognized throughout the European Education Area.

They offer numerous benefits, allowing individuals to maintain control over their data, easily present credentials across Europe, and have them verified throughout their careers. Employers benefit from streamlined verification processes, better understanding of candidates' credentials, and trust in tamper-proof documents. Educational institutions can reduce administrative burdens and costs, accelerate issuing procedures, and gain insight into learners' credentials from diverse backgrounds.

Our system is engineered to harmonise with Europass and EDC, harnessing the capabilities of JSON-LD, a structured data format aligned with European Learning Model v3 standards. By embracing this standardised approach, we not only promote interoperability but also establish a universally recognizable format for educational credentials across Europe's diverse landscape.

We are dedicated to maintaining data integrity and transparency. Through format harmonisation, we facilitate the transferability of learner achievements, empowering individuals to navigate national boundaries effortlessly. Whether it's sharing qualifications, course catalogues, or training records, our system serves as a catalyst for seamless data exchange, nurturing a vibrant ecosystem of learning and development.

The European Learning Model v3 (ELM v3), unveiled in April 2023, signifies a significant advancement in educational data standardisation. It embodies the collective expertise of educators, policymakers, and industry leaders, ensuring the European educational framework remains adaptable, responsive, and future-ready.

For making the code publicly available, we will use either GitHub or GitLab as our repository platforms. Both platforms offer robust features for version control, collaboration, and transparency, making them suitable choices for hosting and sharing our project's source code.

5. Code Repository

The code repository for the MCEU Hospitality project serves as the central hub for hosting, managing, and collaborating on the development of the Platform's codebase. It provides a structured environment for version control, documentation, and collaboration among project stakeholders, including developers, designers, and technical leads. Below, we outline the structure and components of the code repository:

5.1 Repository Structure

- **Main Branch:** The main branch serves as the primary development branch, containing stable and tested code ready for deployment. Major releases and feature integrations are merged into this branch after thorough testing.
- **Feature Branches:** Feature branches are created for the development of specific features or enhancements. Each branch is dedicated to a particular task or user story, facilitating focused development and parallel work. Feature branches are merged into the main branch upon completion and code review.
- **Development** Environment Branches: Branches for development environments (e.g., development, staging) allow for testing new features and changes in isolated environments before deployment to production. These branches help ensure code stability and compatibility with other system components.
- **Documentation Branch:** A dedicated branch for documentation (e.g., README.md, developer guides, API documentation) ensures that project information is up-to-date and easily accessible. Documentation is maintained alongside code changes to provide comprehensive insights into the Platform's functionality and usage.

5.2 Components

- **Source Code:** The repository contains the source code for frontend, backend, and integration components of the Platform. Code is organised into directories based on functionality and technology stack (e.g., /frontend, /backend, /integration).
- **Configuration Files:** Configuration files for development, testing, and production environments are stored in the repository. These files include settings for database connections, API endpoints, authentication mechanisms, and environment-specific variables.
- **Dependencies:** Dependency management files (e.g., package.json, requirements.txt) specify the dependencies required for building, testing, and running the Platform. Dependencies are version-controlled to ensure consistent development environments across team members.
- **Scripts:** Automation scripts for tasks such as building, testing, and deployment are included in the repository. These scripts streamline development workflows and maintain consistency in development practices.
- **Tests:** Unit tests, integration tests, and end-to-end tests are stored in the repository alongside the source code. Test suites ensure code quality, functionality, and regression prevention throughout the development lifecycle.

- **Documentation:** Documentation files, including architectural diagrams, data models, and developer guides, are maintained in the repository. Documentation is continuously updated to reflect changes in codebase and project requirements.

The code repository utilises a version control system (e.g., Git) to track changes, manage branches, and facilitate collaboration among project contributors. Features such as pull requests, code reviews, and issue tracking enhance communication and coordination among team members. Access to the repository is granted to authorised stakeholders, ensuring secure and controlled access to project resources.

6. Technical and Operational Aspects

The Platform is designed to address various technical and operational considerations relevant to its development and deployment, including performance, security, and maintenance. These considerations ensure that the platform is robust, reliable, and able to meet the needs of its users. The platform is designed to allow it to accommodate growing numbers of users and increasing amounts of data.

Security is another important aspect, with the platform incorporating robust measures to protect user data and prevent unauthorised access. This ensures that the platform is secure and trustworthy, providing users with peace of mind when using its features. Additionally, the platform is designed to be easy to maintain, with updates and enhancements to ensure that it remains up-to-date and fully functional.

Overall, the technical and operational aspects of the platform are carefully considered and designed to provide a reliable, secure, and user-friendly experience. By addressing these key considerations, the platform is able to deliver a high-quality service that meets the needs of its users and provides a solid foundation for the issuance and management of digital credentials.

The platform architecture is designed to ensure compliance with both technical and semantic features outlined in Deliverable 4.1, as well as interoperability needs from technical, legal, and organisational viewpoints.

The Platform can connect with Europass and EBSI to enable smooth data sharing and compatibility, following the requirements outlined in Deliverable 4.1. This connection makes sure that the platform can interact and share data with these important platforms, improving its overall performance and efficiency.

7. Conclusion

In conclusion, the design document and publishing code libraries presented in Deliverable 4.2 represent a comprehensive blueprint for the development of the Credentials Issuing Online Platform, a pivotal component of the MCEU Hospitality project. Through meticulous attention to detail, the document has outlined the platform's user interface, architecture, data models, exchange protocols, and integration requirements with key platforms such as EDC, Europass, and EBSI.

In alignment with the European approach to micro-credentials brochure, portability Micro-credentials are owned by the credential-holder (the learner) and may be stored and shared easily by the credential-holder, including through secure digital wallets (e.g Europass), in line with the General Data Protection Regulation (GDPR). The infrastructure for storing data is based on open standards and data models, This ensures interoperability and seamless exchange of data, and allows for smooth checks of data authenticity.

By leveraging state-of-the-art technologies and adhering to industry standards, the platform is poised to streamline the process of creating, storing, sharing, and verifying certificates within the hospitality sector. Moreover, its integration with tools like the EDC and the EBSI ensures robustness, security, and interoperability.

This document lays the foundation for the construction of the platform in the second half of 2024. It serves as a guiding framework for developers, stakeholders, and partners, facilitating seamless collaboration and implementation. Ultimately, the successful execution of this platform is instrumental in advancing digital and green skills within the hospitality sector across Denmark, Iceland, and Spain, and it paves the way for the issuance of credentials during the pilot stage in 2025.

8. References

Council of the European Union. 2022. Council Recommendation on a European approach to micro-credentials for lifelong learning and employability. Retrieved from https://data.consilium.europa.eu/doc/document/ST-9237-2022-INIT/en/pdf

European Union. N.d. European Digital Credentials for learning. Retrieved from https://europass.europa.eu/en/europass-tools/european-digital-credentials

MCEU Consortium. 2024a. Requirements Document - Deliverable 4.1.

MCEU Consortium. 2024b. Grant Agreement Project 101132824.

Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance) (OJ L 119 04.05.2016, p. 1, ELI: http://data.europa.eu/eli/reg/2016/679/oj).


MCEU