Specification:

Digital Education Ecosystem (DEE)

Within the Erasmus+ Knowledge Alliance ProDiT – Projects for the Digital Transformation

621745-EPP-1-2020-1-DE-EPPKA2-KA

Authors: Areej Aldaghamin, Christian Reimann, Carsten Wolff, Ekaterina Mikhaylova

Version 1.0, 09.09.2022

Co-funded by the
Erasmus+ Programme
of the European Union

ProDiT
Projects for the Digital Transformation

Summary

A Digital Education Ecosystem (DEE) is a digital learning & teaching infrastructure which supports all aspects of digitally transformed educational system, e.g. a university or school [1]. The goal of the DEE described in this specification is to support a virtual, cross border Master School. Of course, the concepts may be transferable to any virtualized educational programme (Virtual EP). As an ecosystem, the DEE should cover the needs in a holistic way. It should support the transformation of inputs into outputs, the aggregation into outcomes and finally the generation of an impact (IOOI-logic of the result-oriented monitoring (RoM) approach).

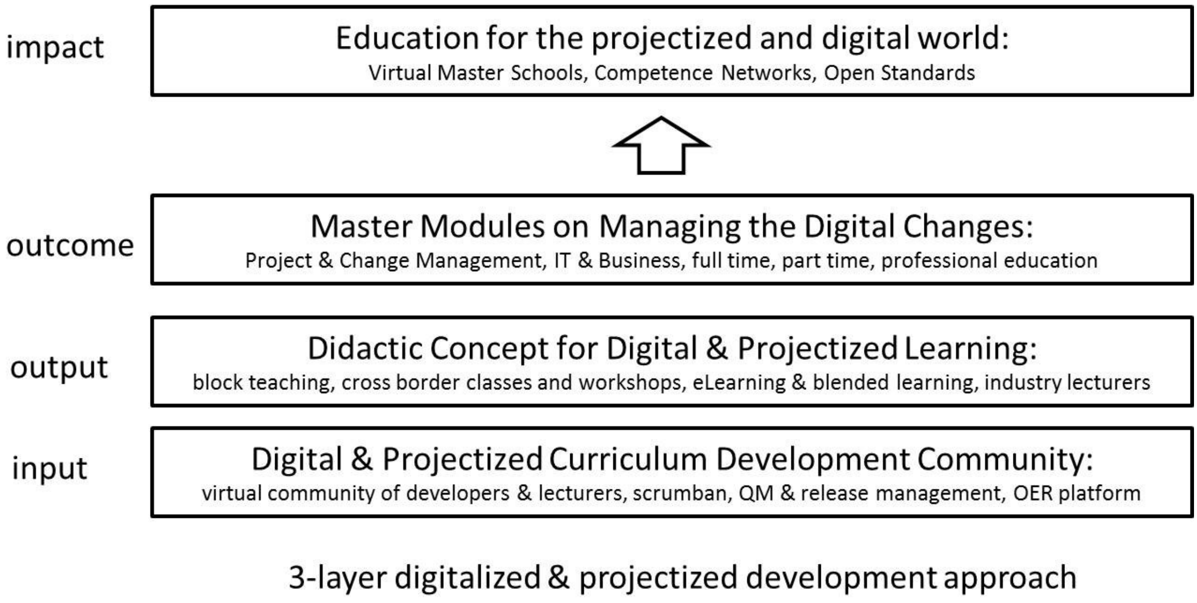
impact
Education for the projectized and digital world:
Virtual Master Schools, Competence Networks, Open Standards
outcome
Master Modules on Managing the Digital Changes:
Project & Change Management, IT & Business, full time, part time, professional education
output
Didactic Concept for Digital & Projectized Learning:
block teaching, cross border classes and workshops, eLearning & blended learning, industry lecturers
input
Digital & Projectized Curriculum Development Community:
virtual community of developers & lecturers, scrumban, QM & release management, OER platform
3-layer digitalized & projectized development approach

Figure 1: 3-layer model of a digitalized Master School according to the IOOI logic [1]

According to the 3-layer-model (see Fig. 1) the DEE has to support:

- The community of teachers and experts in the co-production of content, methods and teaching (input)
- The delivery of the competences including the teaching and learning with didactic formats (output)
- The provision of the content and the generation of the competences (outcome)

The components of the DEE (IT services) support production, delivery and representation in a virtual learning and teaching environment. It is composed out of several service layers, starting from the IT systems, the basic IT services, the Digital University services and on the top layer the actual applications which support production, information, teaching and learning. The services can be configured and tailored into a DEE which serves specific needs (e.g. the virtual cross-border Master School). Services can be defined and agreed in service-level-agreements (SLA). This specification describes the different service layers and specific services.

Table of Content

1 IT Architecture Overview

1.1 Layered IT Architecture of the DEE

This chapter will provide an overview of the complete IT Architecture of the DEE.

This includes a layered view with:

- The IT system layer which describes the server setup, including operating systems (e.g. VMware, KVM) and other (rented) systems, e.g. cloud services or hosted servers/VMs
- The Basic IT Services layer which describes the IT services like VMs, VDI-Desktops, Docker Containers, external cloud services, IT services
- The Digital University Services layer describes general educational services e.g. digital classroom (e.g. Big Blue Button, MS Teams, Discord, …), Learning Management System (moodle, both development and productive environment), user administration (students, teacher, Identity Management), Web Servers, etc.
- The Application layer describes services like Educational Programme Management/Delivery, OER Repository, OpenCoP Portal, Student Journey Configurator, Conference Portal, Co-Production Environment

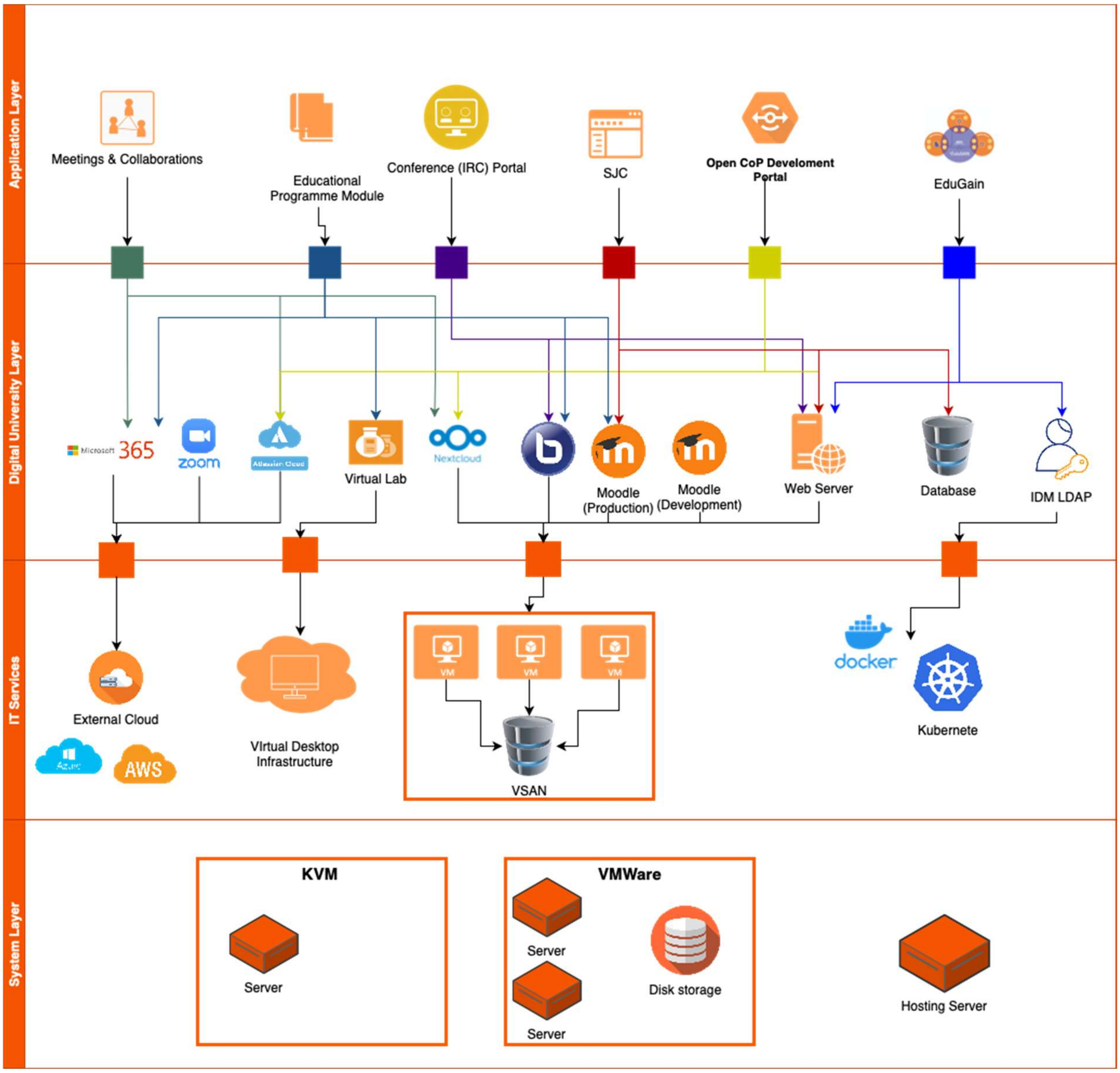

Figure 2: DEE System Architecture [own illustration]

1.2 Scenarios and Use Cases

This chapter lists relevant scenarios and use cases for the DEE, e.g.:

1.2.1 Teaching a cross border module

Use Case: Teaching Abroad

- **Brief Description:** This use case details the collection of activities that the teacher can use the system to be able to teach aboard, using the DEE system using the Mobility Planner to plan the trip aboard also the Student Journey to manage the course.
- **Actors:** System and Teachers
- **Entry condition**: Teachers must have authenticated into the DEE System.
- **Flow of events**:

1. The system provides the teachers with the mobility planner to apply for teaching mobility

2. The teachers provide his/her documents and information
3. Once the teaching mobility is confirmed from the host university, the teachers receive information about the associated systems needed for the teaching activities. (Moodle, etc.)
4. .... etc

1.2.2 Organisations of studies from student's point of view

Use Case: Create Study Preference

- **Brief Description:** This use case details a collection of responses from a student to understand and recommend study programmes that fit the needs of the student. This use case provides a set of questions to the student and provides further conditional questions based on the student's previous response(s). The use case ends with the possible best fit programmes that a student can study.
- **Actors:** System and student.
- **Entry condition**: Students must have authenticated into the DEE system.
- **Flow of events**:

1. The system provides the student with basic questions to know his or her field of interest.
2. The student provides his or her responses to the questions.
3. Once the field of interest is established, the student provides field-related questions to predict which modules will be a must-do for the student.
4. The student provides his or her take on mobility, that is, whether he or she would like to study in single or multiple locations or universities during his or her study period.
5. Based on the student's input the system provides recommended study programmes or studying combinations.
6. The student then selects a programme and continues to create a plan for his or her study period.

Use Case: Create a Study Plan

- **Brief Description:** This use case elaborates action flow and interactions a prospective student can plan his/her Master's programme. The planning involves selecting and pairing various modules and events that eventually lead to the creation of a recognizable programme. It starts with a display of modules and events (based on students' preferences) and ends with students selecting or conforming to a plan.
- **Actor:** Student, system.
- **Entry condition**: The student must have created his or her study preference.
- **Flow of events:**

1. The student reads the details of the modules and their associated events.
2. The student selects the modules that he or she prefers.
3. During module selection, the system alerts the student in cases where a new selection conflicts with an already selected module(s).
4. Upon selection of the modules, alternative modules are shown to the student with which he or she can replace a selected module.
5. The system automatically calculates the credit points of the selected modules and ensures that the student selected modules do not exceed the maximum or below the minimum.
6. The system shows an overview of all the selected modules as planned.
7. The student saves the selected modules and events as a programme plan and can be edited later.
8. The student can repeat steps 1 to 7 to create multiple plans.
9. The student compares his or her plans and selects one of them to embark on his or her student journey.

Use Case: Check Selected Studies Combination (Programme View)

- **Brief description:** This use case details possible interactions that students can have with his or her selected study combination (programme view). The use case involves checking for all the modules a student has to study in their respective semesters as well as replacing selected modules with alternatives if possible.
- **Actors:** Student, system.
- **Entry condition:**

1. The student has already selected a study combination or programme plan.
2. Students must have authenticated into the DEE-My Student Journey application.

- **Flow of events:**

1. The system organizes the student's selected study combination modules into various semesters when the student logs into the application.
2. The student sees overview details for each module including the name of the module, events, progress of the module, an upcoming event, etc.
3. The student selects a module to read more details (e.g. description, events, requirement, literature, achievements, etc.) on a particular module.
4. In case the student wants to change a module, he or she can select an alternative module from the opened module details view.
5. The system checks if it is possible for the student to select an alternative module.

Use Case: Check Journey Timeline

- **Brief description:** This use case subsumes the student's ability to see the various timelines with regards to his or her study programme's modules. The student can see timelines such as modules and events in semester, academic year, month, week or in day.
- **Actor:** Student
- **Entry condition:** Students must have authenticated into the DEE-My Student Journey application.
- **Flow of events:**

1. The student navigates to the timeline/calendar in the My Student Journey application.
2. The system generates various calendar views for an academic year, active semester, month, week, and day.
3. The student peruses the different calendar views and reads more about an event's details.

Use Case: Apply for a mobility event.

- **Brief description:** A student has to go through a sequence of interactions to apply for an event. This use case embodies the interactions.
- **Actor:** Student
- **Entry condition:**
  Students must have authenticated into the DEE-My Student Journey application.
- **Flow of events:**

1. The student navigates to the mobility in the My Student Journey application.
2. The system displays all events that the student can apply.
3. The student reads about events details (venue, duration, etc.).
4. The student then applies for the events he or she is interested in.

5. If the student's applications are accepted by the administrator, the student can monitor the application and preparation progress of the events.
6. The student can later upload the requested documents needed during the application process.

Use Case: Upload certificate/report.

- **Brief description:** This use case details the possibility of a student uploading a certificate that can be used in place of a teaching unit or as an achievement upon approval from an administrator.
- **Actor:** Student, administrator, system
- **Entry condition:**
  Students must have authenticated into the DEE-My Student Journey application.
- **Flow of events:**

1. The student navigates to the certificate view in the My Student Journey application.
2. The system displays all submitted and certificate upload drafts, in case the student has any.
3. The student creates a new certificate upload draft, adds all relevant details, and submits it.
4. The student can edit uploaded drafts but once submitted for review editing is restricted.
5. The student receives updates on the submitted certificates. These updates could be a request for additional clarifications, refusal, or acceptance.

1.2.3 Running an educational programme from coordinator's point of view

Use Case: Plan semester activities

- **Brief description:** The use cases subsumes a programme coordinator or administrator interaction with the DEE system to plan activities that will take place in a semester. It comprises specifying semester and modules duration, scheduling and re-scheduling of events, selecting modules that can be done as electives in a semester, sending elective options to students.
- **Actor:** Administrator, system
- **Entry condition:**
  The administrator must log into DEE-Administrator Configurator.
  The administrator must have permission to plan a semester.
- **Flow of events:**

The administrator:

1. enters the duration of a semester.
2. Specify the starting and end dates of the various modules in the semester.
3. Enters the period for registering for an examination.
4. Creates or updates module information.
5. Creates teaching units for the module that requires teaching units.
6. Enters or selects electives that can be offered in the semester
7. Sends elective modules or teaching units selection requests to students with a deadline.
8. Enters national holidays within the semester.

Use Case: Verify student certificate uploads

- **Brief description:** The administration sees all uploaded certificates by students and verifies the authenticity, and whether the uploaded content suffices its intended purpose.

The administrator can delegate this approval task to another colleague who is capable of verifying the certificates.

- **Actor:** Administrator, system
- **Entry condition:**

1. The administrator must log into DEE-Administrator Configurator.
2. The administrator must have permission to approve a certificate.

- **Flow of events:**

1. The system displays all uploaded certificates by students to the administrator.
2. The administrator checks the content of the certificate and provides his or her verification verdict.
3. The system sends the verification results to the respective student.
4. The administrator can delegate the verification to another administrator with certificate verification permission.

Use Case: See Overview of Programme Students

- **Brief description:** This use case subsumes the interactions of the administrator in viewing students that are registered to a particular programme.
- **Actor:** Administrator, system
- **Entry condition:** The administrator must log into DEE-Administrator Configurator.
- **Flow of events:**

1. The administrator navigates to the study programme view.
2. The system displays a list of students with their basic details such as names, email, FH Kennung, matriculation number, number of semesters studied so far as well as the number of modules, etc.
3. The administrator selects a student from the list to peruse more details (selected electives, grades, chosen modules or teaching units, attended events, etc.) about the student.
4. The administrator can export a set of student data to be used in a different context.

Use Case: See Overview Modules and Teaching Units

- **Brief description:** The administrator can see an overview of modules and teaching units and can select any of them to update, archive, or delete them.
- **Actor:** Administrator, system
- **Entry condition:** The administrator must log into DEE-Administrator Configurator.
- **Flow of events:**

1. The system displays a list of modules and teaching units.
2. The administrator can filter or search for a particular module or teaching unit.
3. The administrator can open individual modules and either read their content, update the content, archive, or delete the module.

Use Case: Import data from other applications such as HISinOne

- **Brief description:** This use case describes the administrators' ability to import data from other platforms. The DEE-Administrator Configurator ensures the importation process goes through without duplicates.
- **Actor:** Administrator, system
- **Entry condition:** The administrator must log into DEE-Administrator Configurator.

- **Flow of events:**

1. The administrator uploads the data file.
2. The system checks if the file extension is acceptable.
3. The system performs the important and reports feedback to the administrator.

1.2.4 Exchanging student’s credentials/digital twin

Use Case: Authenticating students via eduGAIN

- **Brief description:** This use case illustrates how students from different institutions but within the same federation can access the DEE system with your institution-given credentials. When a student tries to authenticate to DEE, the SP enquires from the identify federation where the user is from (WAYF), that is, the student’s institution. Once the identify federation (e.g. DFN-AAI, German identify federation) locates the student’s institution, the student is redirected to his/her IdP’s sign-on page to log in with his/her institution’s credentials, and access is granted.
- **Actor:** Identity federation, identity providers (IdPs), service providers (SPs), and student.
- **Entry condition:** The student must have login credentials from an institution recognized by the DEE system.
- **Flow of events:**

1. The student accesses one of the DEE sub-applications.
2. The application redirects the student to his/her IdP’s sign-on page to log in with his/her institution’s credentials.
3. After a successful login with the IdP, the IdP informs the DEE application to grant the student access.

1.3 Service-based overview

The DEE project is a comprehensive and holistic digital learning and teaching environment. This goes beyond classical Learning Management Systems. The goal is to support the delivery of the teaching and learning activities and to integrate the competencies and content with a fully projected and digitalized virtual environment. The EuroPIM DEE is designed as a cloud-based, service-oriented architecture that combines existing IT tools and services and adds new features. The following figure (3) shows an overview of the elements of the DEE.

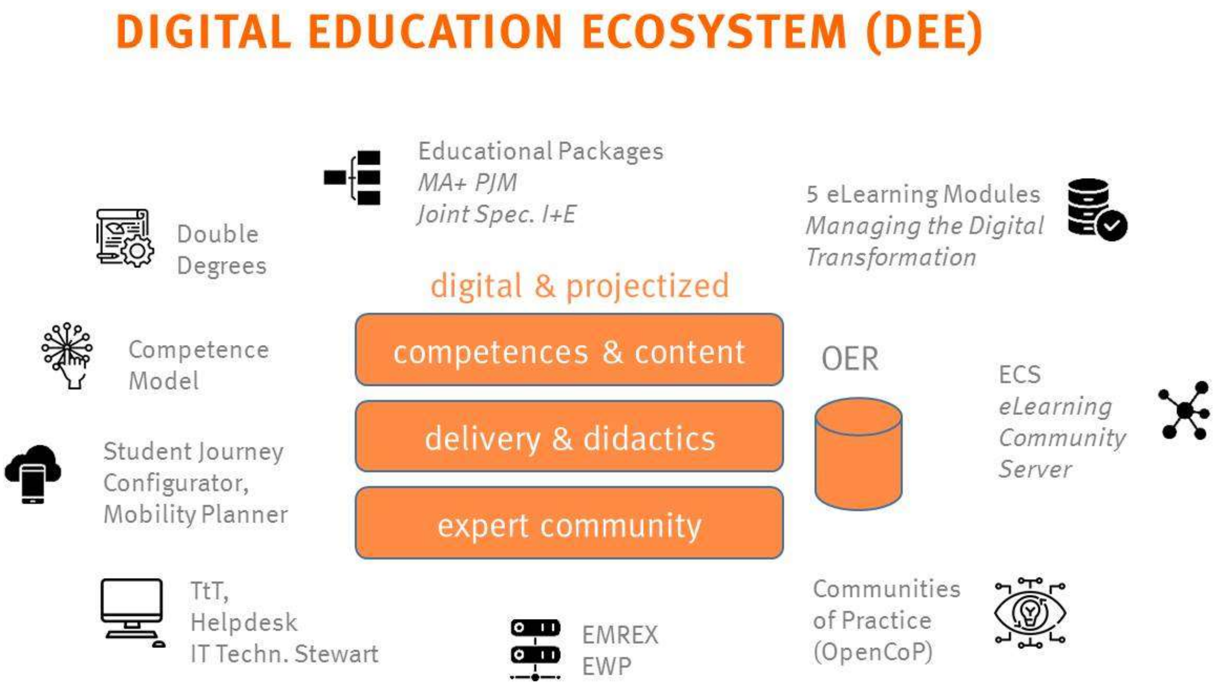

Figure 3: Overview of the Elements of the EuroPIM Digital Education Ecosystem (DEE) [1]

Strengths	**Weakness**
▪ *Delivery of the teaching and learning activities and integrate the competences and content with a fully projectized and digitalized virtual environment.* ▪ *The development and deployment team comprise expertise within the teaching and learning environment.* ▪ *Utilize SAFe to handle the complexity of DEE with Agile Release Train (ART).* ▪ *Required features and functionalities are identified by expert users*	▪ *The long-term maintenance and further development of the DEE is an issue since the development is based on short-term staff.* ▪ *Misinterpretation or misunderstanding of ideas among international teams due to the different language, cultures, regulations, etc.*
Opportunities	**Threats**
▪ *Cross-border programmes.* ▪ *The possible introduction of more specialized programmes.*	▪ *The DEE project relies on the successful acquisition of public funds.* ▪ *A continuous change of requirements and delivery of new features, services and components* ▪ *Different support systems for education in every partner country.*

Table 3: SWOT Analysis

2 IT Systems

This chapter describes the basic IT systems which are involved, their setup, their management. The IT Systems layer of DEE consists of the components described below.

2.1 Physical Servers

2.1.1 DELTA D10x-M1-PS

Intel Xeon high performance server from DELTA:1U rack chassis with 350W power supply (80Plus PlatinumLevel / 94%) and:4x 3.5" HotSwap slot for SATA/SAS1x M.2 NGFF Connector (PCI-E 3.0 x4 Slot: 22110)

2.1.2 DELTA D20x-M2-PS:

D20x-M2-PC-29-16-768GB-8x1TB-2x1TB-2x1G-2xHDR

2.1.3 DELTA D22x-M4-PC

D22x-M4-PC-19-6-96GB-10x14TB-2x480GB-2x10G-2xHDR

3 Basic IT Services

More like Infrastructure as a Service (IaaS)

3.1 Virtual Machines (VM): VMWare or KVM

DEE has vSphere which is VMWare’s virtualization platform. vSphere includes several tools and products that work together to provide a complete infrastructure for virtualization. The vSphere products used/purchased for DEE are as follows:

1. ESXi: The ESXi is a Type-1 hypervisor running on host computers and managing the execution of VMs and allocating resources to them as required.
2. vCenter Server: vCenter Server is a server application. This application runs in a windows server that is installed on a VM. This application allows the creation of new VMs, starting and stopping them, and other management tasks.
3. vSAN: VMWare’s vSAN is a software-based storage solution. Software-defined storage can help in provisions and management of storage based on policies.
4. VMWare Workstation: VMWare Workstation is a virtual machine software that is used to run multiple operating systems over a single physical host.
5. VMWare Horizon: VMWare Horizon is a virtual desktop infrastructure (VDI) solution. This component will be described in detail in section 2.1.3.
6. VMWare ThinApp: VMWare ThinApp is an application virtualization product available in the VMWare Horizon suite. ThinApp enables faster application deployment, simplifies application migration, and streamlines application management.
7. VMWare Dynamic Environment Manager: The VMWare Dynamic Environment Manager can be used to personalize any virtual Windows desktop environment. Additionally, it can also configure policies dynamically, thus simplifying end-user profile management.

The overall VMWare architecture can be described by the following diagram (see Figure 4):

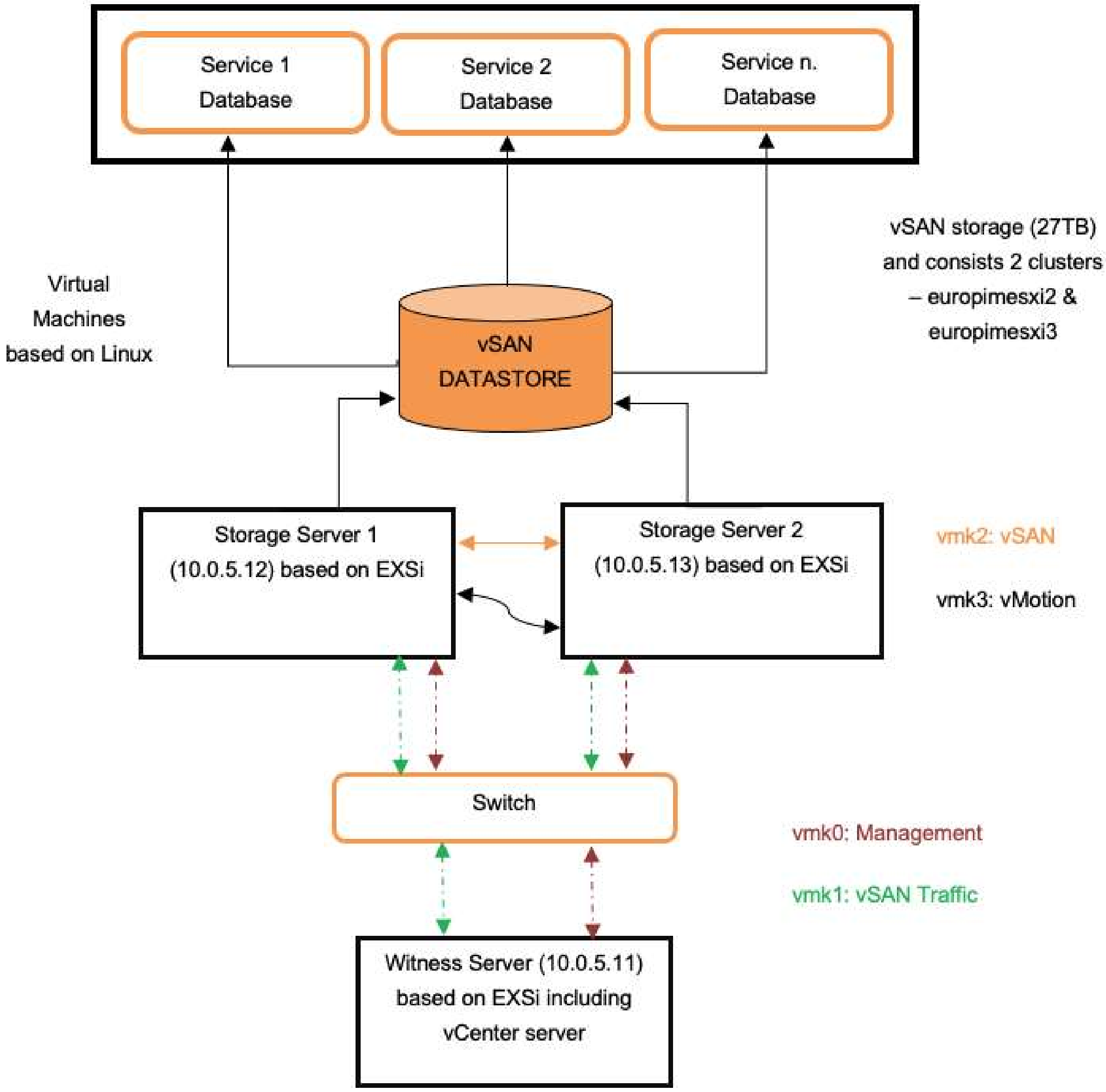

Figure 4: DEE VMWare Architecture Diagram [own illustration]

3.2 Docker Container (Kubernetes)

Docker Containers can be used to employ a microservice-based architecture where the services are loosely coupled, easily portable and easily deployable. Docker is based on a client-server architecture model and comprises of the following components:

Docker Containers can be used to employ a micro-service-based architecture where the services are loosely coupled, easily portable, and easily deployable. Docker is based on a client-server architecture model and comprises of the following components:

1. Docker daemon: The Docker Daemon also known as Dockerd listens for Docker API requests (REST API) from the docker client. Based on the request call, the daemon will build, run, or manage the docker images and containers. Docker daemon is also responsible for managing docker objects.

2. Docker client: Using the docker client (docker), is how users can interact with docker. When we execute docker commands, (such as “docker build”), the client sends them to the daemon over a docker API request.
3. Docker registries: The docker registry is a place where Docker images are stored
4. Docker objects: Docker objects include docker images, containers, volumes, networks, etc.

The containers can be configured to run on a server, a VM, or cloud instances. In our project, the Docker client and the docker daemon in the same instance was configured. It is also possible to install them in different instances. In DEE project, the Gitlab container registry was used to store the Docker images. Once the docker image has been developed for each service, it is pushed to the GitLab registry. When a deployment is required, the image will be pulled from this registry before a container is created. A container configuration requires some prerequisites, such as ensuring the required ports of the host machine are open, ensuring the paths which will need to be mounted through docker volumes, ensuing specific configurations are added as config maps, etc.

The next step is to create a container incorporating all of these as needed. In order to persist data, volumes should be created especially for the data directories of databases that are created in the containers. For example, in the SJC configurations, PostgreSQL has been used, hence we have mapped its data directory as a volume. Additionally, containers can also have multiple volumes. For instance, there is an additional volume for the journey service to persist data of the server directory. Similarly, the server directory of the User service is also mapped to a volume. Once, all the scripts and processes are in running state inside the container, we can check the webpage for validation. The webpage is reachable due to the host and container port mapping.

3.3 Virtual Desktop Infrastructure (VDI)

VDI is desktop virtualization technology where a desktop OS is running and is managed in a data center. The virtual desktop image is delivered over a network to the user’s device. This allows the user to interact with the OS and its applications as if they were running locally. VMWare Horizon is one such VDI technology. It enables users to run VM desktops and applications in the data center or cloud and remotely deliver these desktops as a managed service. VMWare Horizon comprises the following components:

1. VMWare vSphere Instant Clone Technology: Enables fast desktop provisioning
2. VMWare Dynamic Environment Manager: Enables personalization of user desktops and configure user environments dynamically based on factors such as users’ location, type of device, and user group.
3. VMWare App Volumes: Enables volumes (similar in containerization technologies) to attach to a VM at login.

The following diagram describes the overview of a VMWare Horizon implementation architecture [C. Arakelian, What Is VMware Horizon and How Does It Work? | VMware.] (see Figure 5).

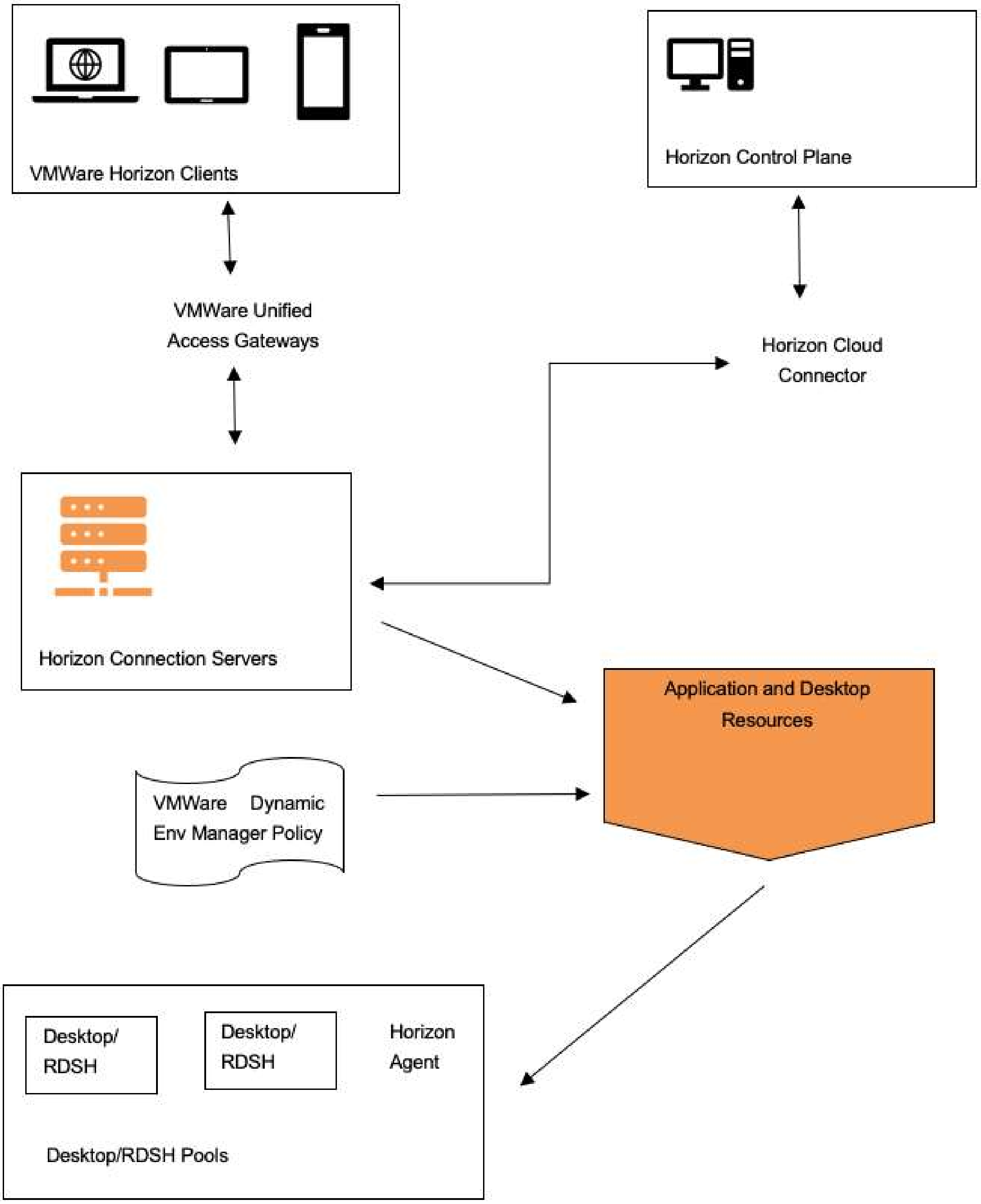

Figure 5: DEE VMWare Horizon VDI Diagram [own illustration]

3.4 eduGAIN

The exchange of user identity data between IdPs and SPs within and across federations follows a schema that outlines what user attributes are given out to the SP by the IdP and how these attributes are to be displayed. The Research and Educational Federations (REFEDS) manages the schemas used by participating federations to exchange user identity data. These schemas provided by REFEDS provides a framework for defining identity data exchange at various national level. The currently managed schemas by REFEDS are eduPerson and SCHAC (Flanagan, 2019).

EduPerson is a user identity schema designed to include all mostly used user (person) attributes and his/her organizational attributes in higher education. The schema is a Lightweight Directory Access Protocol which provides a list of attributes and definitions based on existing higher educational standards (Flanagan, 2020). The current version is eduPerson 2020-01. eduPerson contains an axillary object class for institution directories that is intended to support LDAP and facilitate communication among higher institutions. Figure 2 shows a screenshot of the format of eduPerson object class and attributes.

eduGain makes it possible for educational and research institutions and their users within different federations to share resources, thereby enhancing global research. A trusted identity given to a user by his/her institution allows him/her to access services from different federations participating in eduGain [https://edugain.org/]. The main components of eduGain are as follows:

1. Identity Federation
2. Identity Providers
3. Service Providers
4. Users

Many research and educational (R&E) institutions provide their users with a single sign-on credential for accessing the institution’s provided services. To allow institution users to have access to other services across their boundaries, the identity federation was created to ensure trust and collaboration between service providers (SPs) and identity providers (IdPs) within a federation. This collaboration offloads the burden of authentication and authorization from the shoulders of SPs.

When a user tries to access a service or a resource, the SP enquires from the identify federation where the user is from (WAYF), that is, the user’s institution. Once the identify federation (DFN-AAI, German identify federation) locates the user’s institution, the user is redirected to his/her IdP’s sign-on page to log in with his/her institution’s credentials. After a successful login with the IdP, the IdP informs the SP to grant the user access to the requested resource. Finally, the SP grants the user access to the resource object. The diagram below describes these operations (Figure x).

In DEE project, the eduGain auth. is used for the user service which is (t.b.c) …

3.5 External Cloud Platform: Azure and AWS

Third-party software applications used widely in the university such as Microsoft 365, Zoom and Atlassian cloud are hosted on external cloud platforms. These external cloud platforms include:

Microsoft Azure

Microsoft Azure is a cloud service provider owned and managed by Microsoft. It provides support for not only windows applications or services but also open source technologies allowing companies to use tools of their choice (Microsoft Azure, accessed on: 20.09.2021).

Amazon Web Services (AWS)

AWS is a cloud service provider like Microsoft Azure but they both have their advantages that influence the company’s choice of cloud service providers. In terms of infrastructure technologies, AWS has more feature-rich services than other cloud service providers (AWS, accessed on: 20.09.2021) but is relatively expensive compared with Microsoft Azure.

4 Digital University Services

4.1 IDM/LDAP/OAuth 2.0

The LDAP (Lightweight Directory Authentication Protocol) is an open and cross-platform protocol used for directory services authentication. LDAP makes the possibility to store the users, passwords, and computer accounts, and share them within the network. In the DEE project, LDAP authentication was used fron the existing authentication provided in the IDIAL institute. As a start, the LDAP was used on implementing on-premise Atlassian Jira system as a task management system.

IDM/LDAP => single sign on, but lateron extended to: Digital Twin of the Student/Teacher

4.2 Core Service (User Service)

The user service provides authentication service to the other services that require authentication by matching login credentials with the institution's LDAP directory or with eduGAIN. The user service upon successfully verifying the identity of the user generates a JSON Web Token (JWT) which allows authorization by the services the user intends to use.

The user service is responsible for handling all necessary functionalities shared between all Laravel projects within DEE. For example, one of the main functions within dee-core is authentication by sharing one account and login between all other DEE services. Using the PHP-Laravel Passport server, which implements all required functionalities (Login, Register, and Forget Password) and all users' information are stored in a central database within dee-core, we achieved the OAuth 2.0 protocol between services.

Instead of storing the username and password of any user each time the user logs in, the dee-core service is responsible for creating all required APIs and internal APIs for other applications, especially the OAuth/token API responsible for creating a token that is stored in each project and used to authorize the app. And the User flow can be described as those necessary steps:

- Users open any service that belongs to or authorised using DEE.
- The service will redirect unauthorized users into the dee-core Laravel app.
- Users will enter login details and then be redirected to service as a result of the login if it is successful.

Each service will store the details inside the dee-core oauth_clients database table, allowing the dee-core to know where to redirect the users after being successfully authenticated.

Third-parties applications/services should register within the dee-core system. Each application will be required to enter a callback URL that dee-core redirects into when the user login using dee. In addition, all clients will get a generated secret token and should store it internally within each application.

4.3 Learning Management System (LMS)

See chapter 7.

4.4 ECS (maybe LTI)

The current implementation is done based on ECS. See chapter 7.3.

4.5 Virtual Classroom System – Conferencing

This functionality is provided by MS Teams, based on the MS365 tenant of the DEE.

4.6 Virtual Lab (based on VDI)

Based on VMWare Horizon, see chapter 3.3.

4.7 Project Platform (Virtual Project Campus)

T.b.d., will be based on a portfolio of could-based project tools, e.g. GIT, JIRA.

GitLab

Gitlab is used for the code version control system. it is hosted in IDIAL and includes many projects. Using Gitlab the DEE services are grouped in the " DEE Project" Group. Every service has its repository which contains the service source code.

JIRA

JIRA is hosted also on Premise and is running on the DEE servers. It contains many projects and every project has different team members and boards. The system is used for task management and reporting of the work in progress.

Atlassian Confluence + Jira

MS Teams

4.8 Backend for Competence Management (Digital Twin of the student)

t.b.d., will be a core service for competence management tools and for the student journey configurator (see chapter 8).

- Backend for Student Journey Configurator, Mobility Planner (see chapter 8)

The backend layer of the SJC bundles the various implementation logics that enforce how data should be pulled from the database and served to the Angular frontend (user interfaces) via API calls when needed. This implementation is handled with a PHP framework called Laravel. Laravel makes implementation flexible by adopting a Model-Controller-View architecture. In a nutshell, the model in the architecture refers to the data, the controller manipulates how the data are being retrieved based on the business logic and displayed on the various user interfaces(views).

The data used in the SJC is managed with the PostgreSQL database, which is an open-sourced relational database management system that complies with Structure Query Language (SQL). Moreover, the SJC application utilises modules information from other university systems and other relevant data.

- Administrative Tools => EMREX, EWP, HISinONE, PIM Integration

4.9 Co-Production Environment, Collaboration Platform

The Co-Production & Collaboration Platform is web-/cloud-based and contains the following tools:

- Microsoft 365: The DEE contains an own MS365 tenant including MS Teams for Collaboration and Meetings, MS Forms for surveys and evaluations, and Sharepoint + Office tools for joint work on documents
- Atlassian Jira is used as a ticket and agile project management tool
- Atlassian Confluence is the main tool for documentation in Wikis and joint development of content
- Nextcloud is used as joint file repository
- Moodle is used as development platform for eLearning modules

The Digital University layer will contain the joint backend services for this platform.

4.10 Evaluation tools

t.b.d., for evaluation a could based data processing infrastructure (based on ELK or TICK stack) is planned. The main tool for evaluation surveys will be MS Forms.

4.11 Open Innovation Portal, OER

t.b.d., this will be the backend for the provision of project outcomes, e.g. Open Educational Resources (OER), white papers, documentation, tutorials etc.

4.12 Application & Registration Portal

t.b.d., first implementation is the web-based pre-check portal …

5 Application Layer

5.1 Educational Programme Management/Delivery

t.b.d., this will be the management console for the study programme directors and coordinators in order to plan and schedule the teaching and examination.

5.2 OER Repository

The OER repository will allow the management and provision of all learning materials which are published to the public as Open Educational Resources (OER, different from the internal learning resources which are provided and managed via the LMS, see chapter 7).

Core functions of the OER repository are:

- Registration for a course as a learner or teacher
- Access to all relevant digital learning resources of the course
- Administration of the users
- Publication of courses as OER
- Interaction of users with the course providers

It is planned to use an existing OER platform (e.g. Open edX) or to join one.

5.3 OpenCoP Portal

The OpenCoP Portal is the main web portal for participants of the Open Communities of Practice (OpenCoP). It supports:

- Joining and leaving an OpenCoP
- Access to all relevant digital spaces and tools
- Administration of the OpenCoP
- Publication of results
- Interaction of external people with the OpenCoP (e.g. chat, forum)

5.4 Student Journey Configurator, Mobility Planner

See chapter 8.

From an architectural point of view, the SJC runs as a single page web-based application with the support of Angular 9.0. Angular is a TypeScript based free and open-source web rontend framework. The various interaction and navigation logic of the SJC interfaces are handled with Angular components. Each component comprises mainly of three files: an HTML (Hypertext Mark-up Language) file, a CSS (Cascading Style Sheet) file and a TypeScript file. The HMTL file contains the basic design structure of the component (interface), the CSS file contains styles used to beautify the web template and last but not least, the TypeScript supports additional business logic to the components displayed on the web application. Moreover, the

TypeScript file allows data pulled from the Journey service backend to be further refined, segregated and apportioned to the various sections of the web interface.

5.5 Conference Portal

See chapter 9.

5.6 Co-Production Environment

t.b.d., this will be the web frontend for seamless access to the Co-Production Environment (see 4.9 for the backend part and chapter 10 for the frontend)

6 Software Architecture Templates

Templates for:

- How to develop a new web-based service
- How to develop a new app
- How to integrate a new tool/e.g. LMS

6.1 Service Template

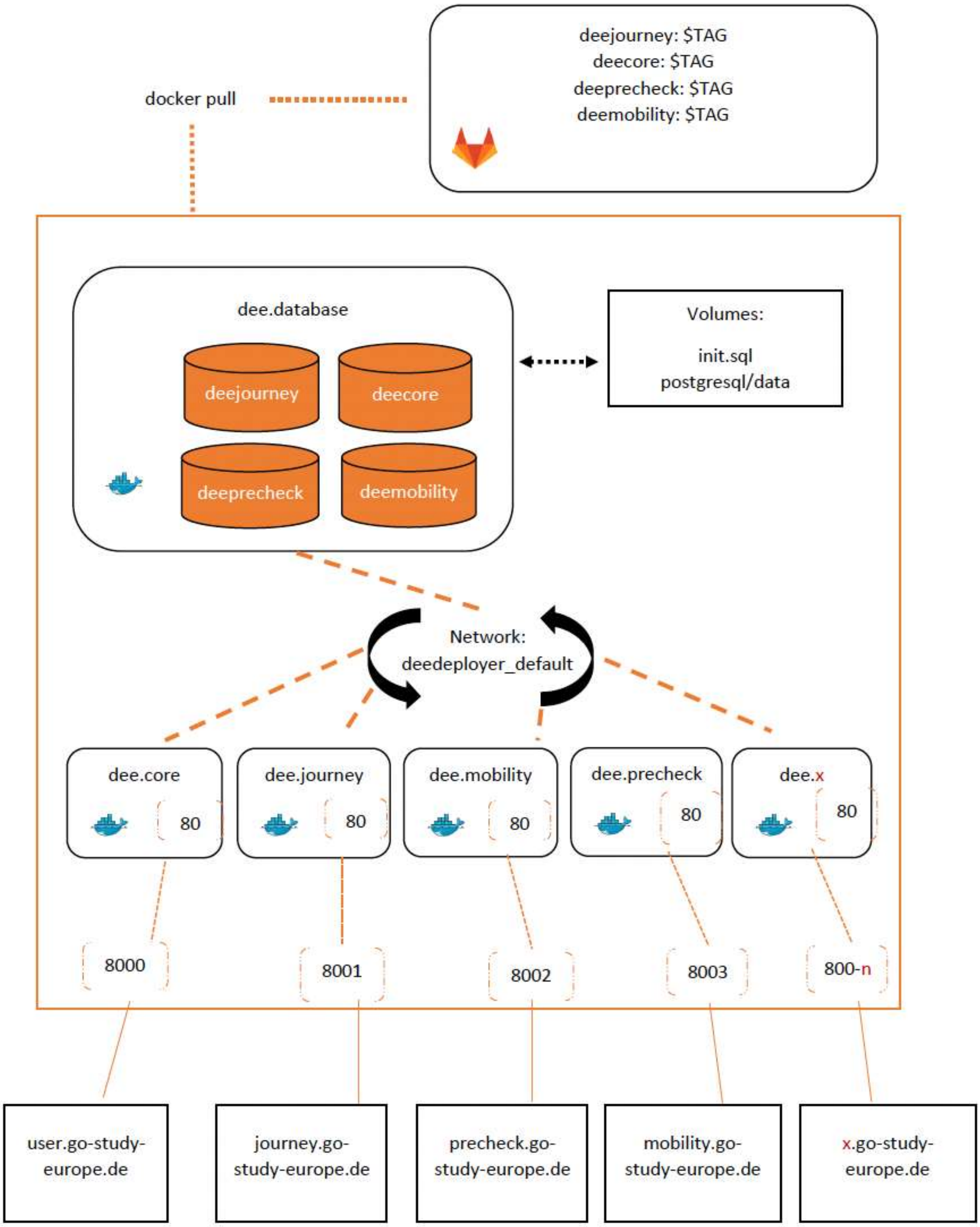

Figure 6: Service template for new DEE services [own illustration]

DEE makes use of a microservice architecture where each service depending on its use follows a predefined template. In DEE there are two types of templates a new service can be created: Backend-Service and Full-Stack service.

- Backend-Service Template: the backend service template contains only the business logic implementations that are needed by other services to provide users' functional needs. Moreover, it handles create, read, update and delete (CRUD) queries to the database of the involved service. For example, the User service used by both the Mobility Planner service and Student Journey Service uses this type of template. The diagram below illustrates the folder structure of this template.
- Full-Stack Service Template: the full-stack service subsumes a server and a client-side. The server side of this type of service is the same as the backend service explained above. On the other hand, the client-side contains various technology stacks (e.g. HTML, CSS, JavaScript, TypeScript) used for rendering user interfaces and sending requests to the server based on user interactions. The structure of this template adds up to the backend-only service template (see figure 7).

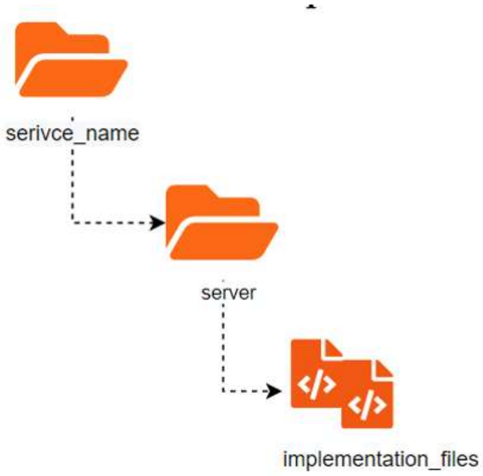

Figure 7: Service template for new DEE services [own illustration]

6.2 Integrating Other Services/Tools

The DEE services communicate with each other via REST APIs. The API available in each of the services provides a contract based upon which other services can interact with it. Newly integrated services ought to make their API accessibility known to other services to ensure smooth interactions.

7 Learning Management System (LMS)

This chapter goes more into detail compared to previous chapters and delivers a holistic vertical view over the layers in chap 4 and 5 from LMS point of view …

7.1 Moodle Development System

The Moodle development space is a section in the LMS where new digital learning resources, new courses and new eLearning modules are iteratively developed, continuously refined and maintained. It is a sandbox environment where no teaching and no access by learners or external users takes place.

The access to the respective development spaces is limited to the members of the OpenCoPs.

Educational resources which are intended for use are first reviewed and released. Then the materials are copied to the Moodle productive system.

7.2 Moodle Productive System

The Moodle productive space is intended for publication and use of the educational resources produced by the OpenCoPs.

There are 2 main spaces/areas implemented in the Moodle LMS:

- The publication space contains the full scope of the produced eLearning modules or other educational resources (e.g. case descriptions, project cases). The material published there is released and final, only bug fixes will be done. All released and published materials have a version number and change history. The publication space is not intended for learners and not used for module delivery, especially teaching. It is only used by teachers and trainers to select the materials for the actual teaching events.
- The delivery space contains actual teaching events, e.g. the delivery (teaching) of a module in a certain semester at a certain university. For this purpose, the teacher creates a new Moodle course for each delivery. Static educational resources from the publication space are linked into the actual delivery space of the module or course. Dynamic elements (e.g. an actual teaching session, an exam) are created for the specific delivery in the delivery space.

The Moodle productive system supports this process with templates and tools.

7.3 eLearning Community Server (ECS)

E-Learning Community Server (ECS) remove restriction from registration of students and teachers in the Moodle systems in partner universities. The goal is to allow for inter-platform access to courses within the various universities, that is, providing a transparent forwarding of the user from his or her home platform to a remote learning platform. Sharing is not limited to courses, but also other sharable eLearning content.

ECS is installed and deployed on virtual machines on IDiAL local network with the specifications of 8 GB minimum RAM, a minimum of 100 GB storage capacity, a virtual CPU of 4 cores, and runs on Debian 9 operating system. The installation of ECS involves the installation of “ecs4” package, Apache 2 web server and a database, in this case “postgresql”. Figure x shows the various deployment components in the ECS.

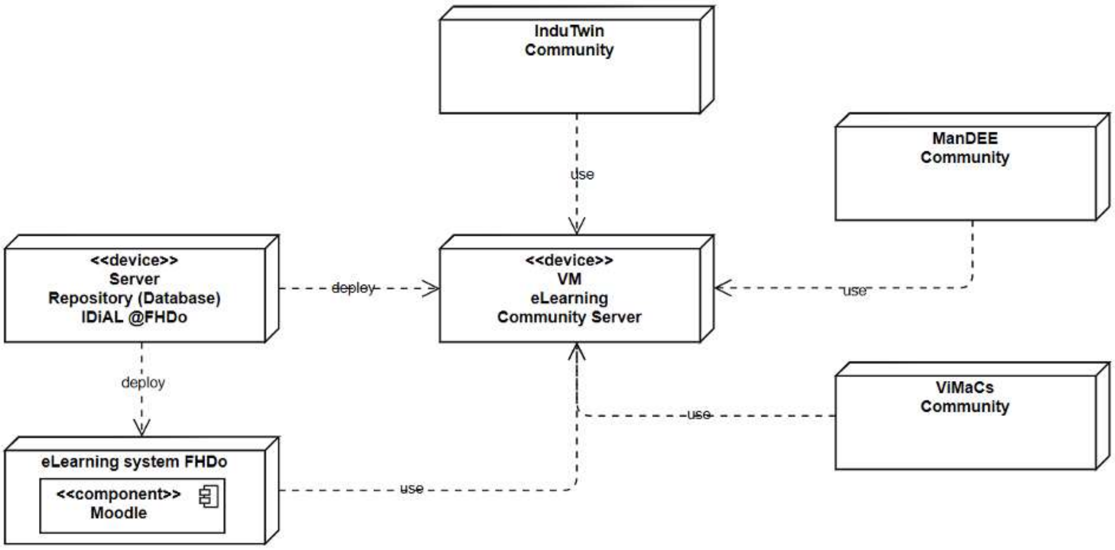

Figure 8: Setup of the DEE E-Learning Community Server (ECS) [own illustration]

8 Student Support Systems

This chapter goes more into detail compared to previous chapters and delivers a holistic vertical view over the layers in chap 4 and 5 from SJC/MP point of view …

8.1 Student Journey Configurator

The Student Journey Configurator (SJC) provides a flexible way for students to structure their programme of study content with modules from the EuroPIM partner universities that fit their career perspectives. The SJC user interfaces are designed with wireframes and tested to ensure their usability before being handed over to the development team. The SJC starts with a preference wizard that subsumes a set of questions aimed to comprehend the academic interest of the student after authenticating the student with the support of the user service. The knowledge of the student's interest is used to provide study combination suggestions which he or she can further edit (remove/add/replace modules) to suit his or her career perspective and competences.

The student journey provides enrolled students with various user interfaces for monitoring their study programme journey. The major interfaces are "My Student Journey", "Calendar", "Event/Mobilities", and "Certificates" view. My student Journey view displays the various modules (in the student's selected study combination or plan). Aside from the listing of the modules, the student can read more details about each module and as well get to see upcoming events. The calendar view presents the student with a timeline of his or her study programme duration. The calendar view has various sections that allow students to see calendar events in various periods (academic year, semester, month, week, and day). Moreover, the mobility view shows all available mobile events that can apply or have been applied to and as well monitor their status (application and preparation). It further provides a map feather that pinpoints the location of an event on a map. Finally, the certificate view displays all certificates that the student has attained during his or her study period. It further allows students to upload certificates or reports that can be used to fulfill a teaching unit or achievement of a module or a study. Further, the student journey provides an administrative panel that allows programme coordinators to handle administrative tasks such as planning a semester, selecting electives modules that can be selected by students in a semester, processing mobility events applications, verifying students' uploaded certificates, and other coordination related tasks.

8.2 Mobility Planner

The Mobility Planner assists the student who wishes to be mobile through the academic journey to plan their mobility. Additionally, supports staff mobilities within the universities. This comprises choosing where he or she wishes to study in a particular period of time at one of the partner universities, applying for such mobilities, monitoring the application, and preparation process. The architectural and technical details of the Mobility Planner are the same as that of the SJC explained in section 8.1. The mobility planner runs as a microservice and utilizes the user service for authentication and authorization of users (students/users and administrators). It as well communicates with the SJC to ascertain the mobility plans the student-created in the configurator.

9 Conference Management & Delivery System

Cross-layer description for the conference related DEE part …

9.1 Conference Portal

One of the elements of DEE is conferencing, the EuroPIM consortium host and manage online and on-site conferences. These conferences are managed via WordPress which is a content management system (CMS) platform. Using WordPress a templet was created in order to facilitate all the conferences of the EuroPIM consortium.

Examples are the portals used for the IEEE European Technology & Engineering Management Summit (ETEMS) and the Dortmund International Research Conference (IRC):

https://online.etems.digital/

https://irc.digital/

9.2 Registration & Payment

Stripe is a web payment platform that is used to manage registration and payments for conferences. Stripe provided a very simple and secure payment service also a very simple to implement service that was integrated with the portals using APIs. Stripes provide very holistic documentation of the implementation of the service.

Stripe is embedded into the conference portal.

9.3 Publication Management

EasyChair is a free web-based conference management software system. EasyChair is used to manage submissions and reviews. The tool provides a very convenient free service to manage the conferences.

Easychair is embedded into the conference management systems.

10 Co-Production Infrastructure

Cross-layer description for the OpenCoP related DEE part …

10.1 Open Community of Practice (OpenCoP) workflows

The workflows of the OpenCoPs are implemented into the OpenCoP portal (see 5.3) and the related tools for the Scrumban management process.

The modules and standards are developed and later on maintained by teams of lecturers and industry experts (OpenCoPs) in a co-production setup. These teams form a kind of a “maker community” which organizes themselves as a virtual, agile project team. Each OpenCoP has a lead owner (in the sense of the Scrum product owner), each major project outcome has an OpenCoP. The participation in the OpenCoP is flexible, but the generation of outputs follows the Scrumban cycle implemented as main PM tool. The OpenCoP conducts workshops and online meetings. It organizes the pilot teaching and the evaluation. It uses the improvement plan system for issue tracking.

The OpenCoPs follow a Scrumban management process where the development of the modules is organized in a digital & projectized way. This will be addressed with The modules and standards are developed and later on maintained by teams of lecturers and industry experts in a co-production setup. These teams form a kind of a “maker community” which organizes themselves as a virtual, agile project team. Each OpenCoP has a lead owner (in the sense of the Scrum product owner), each major project outcome has an OpenCoP (3 for the standards, 4+1 for the modules). The participation in the OpenCoP is flexible, but the generation of outputs follows the Scrumban cycle implemented as main PM tool for ProDiT (see ProDiT Methods handbook, 7.1). The OpenCoP conducts workshops and online meetings. It organizes the pilot teaching and the evaluation. It uses the improvement plan system for issue tracking.a iterative development process: based on a module specification, the eLearning content will be designed. The course will be tested in pilot sessions during block teaching weeks within the Master programmes of the involved HEI. The TtT will be tested in Train-the-Trainer-workshops. Based on the analysis of the evaluation data an improvement plan is implemented. If the evaluation and QM shows sufficient quality, the module is released (as Open Educational Resource – OER) and will be maintained on a regular basis. The approach generates improvement tickets from surveys and lessons learned which are handled and documented in Kanban style. In Scrum meetings of the OpenCoPs, these tickets are considered and prioritized for implementation. An improvement run (a long “sprint”) for a module ends with a pilot teaching and new findings for improvement. Finally, the module is released by the Internal Evaluation Board (IEB) based on a release review and checklist. The Internal Evaluation Board (IEB) is responsible for supervision and controlling of the quality process.

Relevant OpenCoP processes & workflows are:

- Joining (registration & onboarding) and leaving (offboarding) of OpenCoP members
- Development of new artefacts (e.g. educational resources as online course, eLearning modules) or update of existing artefacts in collaboration of OpenCoP members.

- Release of new artefacts for review, review and approval from IEB, release & publication for usage
- Organisation and evaluation of pilot teaching and validation activities

10.2 Tools & Platform

The processes are supported by DEE tools:

- Microsoft 365 and MS Teams for collaboration, conferencing, teamwork and online teaching
- MS Forms for evaluation surveys
- Atlassian Jira as ticket system for Scrum and Kanban
- Atlassian Confluence for documentation and Wiki development
- Moodle for development and release of eLearning modules and as LMS for pilot teaching
- Nextcloud as joint file storage
- Web portals for registration, user administration, publication of results, user communication

10.3 Publication & Dissemination

The OpenCoPs produce certain artefacts, such as:

- Moodle eLearning modules
- Online learning & teaching resources, e.g. videos, tutorials, slides
- White papers & guidelines
- Scientific publications

For publication, the following channels are used:

- Own websites, such as https://prodit-alliance.eu/ or https://go-study-europe.de/
- Scientific publication channels, e.g. conference (Dortmund IRC, IEEE ETEMS) and journals
- OpenCoP portal
- For selected modules and courses, existing (or own) OER portals

11 References

1. Wolff, C.; Reimann, C.; Mikhaylova, E.; Aldaghamin, A.; Pampus, S.; Hermann, E.: Digital Education Ecosystem (DEE) for a Virtual Master School, Proceedings of the 2020 IEEE Smart Information Systems and Technologies (SIST) conference, IEEE Xplore, Nur-Sultan, Kazakhstan, 2020
2. Mikhaylova, E.; Aldaghamin, A.; Ebberg, F.; Tokanov, O.; Wolff, C.; Reimann, C.: Digital Education Ecosystem (DEE): User-Centred Design of the Student Journey Configurator, 2021 IEEE International Conference on Smart Information Systems and Technologies (2021 IEEE SIST), Nursultan, Kazakhstan, 2021
3. Mikhridinova, N.; Badasian, A.; Aldaghamin, A.; Wolff, C.: Transforming Conferences to an Online Format: Framework and Practices, 2021 IEEE International Conference on Smart Information Systems and Technologies (2021 IEEE SIST), Nursultan, Kazakhstan, 2021
4. Wolff, C.; Tabunshchyk, G.; Arras, P.; Otegi, J.R.; Bushuyev, S.; Verenych, O.; Sachenko, A.; Reimann, C.; Hussein, B.; Vitkauskaite, E.; Mikhaylova, E.; Aldaghamin, A.; Badasian, A.; Mikhieieva, O.; Mikhridinova, N.; Myronova, N.; Hemmer, J.; Ruben, T.: Cross-border Projects in Digital Education Ecosystems, Proceedings of the 24th International Conference on Interactive Collaborative Learning (ICL2021), Springer, Dresden, Germany, 2021
5. Wolff, C.; Reimann, C.; Mikhieieva, O.; Mikhaylova, E.: Agile Development of Cross-University Digital Education Ecosystems, 28th European Conference on Systems, Software and Services Process Improvement (EuroSPI 2021), Communications in Computer and Information Science (CCIS), Springer, Krems, Austria, 2021