

D2.2 XR2LEARN MARKETPLACE, IPR TOOLS AND PLATFORM

XR2Learn Marketplace, IPR Tools and Platform

WP2

31/12/2023



Funded by the European Union Funded by the European Union under the Grant Agreement 101092851. Views and opinions expressed are however those of the athor(s) only and do not necessarily reflect those of the European Union or European Health and Digital Executive Agency (HADEA). Neither the European Union nor the granting authority can be held responsible for them.





Author	Filisia Melissari, Alexis Lampouridis, Dimitris Kontopoulos, Panagiotis Karkazis
Work Package	WP2
Delivery Date	31.12.2023
Due Date	31.12.2023
Classification	Public

Status of deliverable

Action/role	Name	Date (dd.mm.yyyy)
Submitted by	Ioannis Chatzigiannakis, CNIT	
Responsible	Filisia Melissari, SYN	31.12.2023
Approved by (internal reviewer)	Alessio Gugliotta, EITM, Enrique Hortal, UM	

Author(s) contact information

Name	Organisation	E-mail
Filisia Melissari	SYN	melissari@synelixis.com

Document reviewers

Name	Organisation	E-mail
Enrique Hortal	UM	enrique.hortal@maastrichtuniv ersity.nl
Alessio Gugliotta	EITM	alessio.gugliotta@eitmanufactu ring.eu

© Copyright by the XR2Learn consortium



Document history

Date	Version	Author	Comments
01.11.2023	0.1	SYN	Table of Contents
11.11.2023	0.2	SYN	Adding content for section 1
01.12.2023	0.3	SYN, MAG	Adding content for sections 2, 3
20.12.2023	0.4	SYN, MAG	Adding content for sections 3-6
21.12.2023	0.5	SYN, MAG	Document ready for peer review
29.12.2023	1	SYN, MAG	Addressing the comments of the reviewers



Table of Contents

1 Introduction	12
1.1 Purpose and context of the document	12
1.2 Document structure	12
1.3 Technology	12
1.3.1 Blockchain Technology	13
2 Platform	14
2 Flationin	17
2.1 Identity and Authentication Management 2.1.1 Description	17
2.1.2 Installation and Configuration	19
2.2 Catalogue	20
2.2.1 Description	20
2.3 Community Forum	21
2.3.1 Description 2.3.2 Installation and Configuration	22
2.4 Technical Wiki	24
2.5 Installation and Configuration	25
3 Blockchain Realm	26
3.1 IPR Tool	26
3.1.1 Description	26
3.1.2 Architecture 3.1.3 Installation and Configuration	29 30
3.2 Smart Contracts	30
3.3 Future Developments	31
4 Marketplace	32
4.1 Description	32
4.1.1 Browsing contents	34
4.1.2 Creating content 4.1.3 Key Features	34 35
4.1.4 Future developments	36
4.2 Architecture, specifications, and interfaces	36
4.2.1 Installation and configuration guidelines	40
5 Integration	41
5.1 Hosting infrastructure	41
5.1.1 CI/CD workflow 5.1.2 CI/CD ninelines	41 42
5.1.2.1 Pipeline jobs	43
5.2 Develompent enviroment	44
5.3 Stage Enviroment	45
5.4 Production Enviroment	45
5.5 Monitoring resources	46
6 Conclusions	50
References	51





LIST OF FIGURES

FIGURE 1 PLATFORM LANDING PAGE	16
FIGURE 2 REGISTRATION PAGE	18
FIGURE 3 LOGIN PAGE	18
FIGURE 4 ADMINISTRATION CONSOLE OF IDENTITY AND AUTHENTICATION MANAGEMENT COMPONENT	19
FIGURE 5 XR2LEARN CONTENT CATALOGUE	21
FIGURE 6 ADDING AND ITEM TO THE CONTENT CATALOGUE	21
FIGURE 7 COMMUNITY FORUM WELCOME PAGE	22
FIGURE 8 XR2LEARN COMMUNITY LOGIN	23
Figure 9 XR2Learn Wiki	25
FIGURE 10 NFT CREATION SEQUENCE DIAGRAM	27
FIGURE 11 NFT METADATA	28
FIGURE 12 IPR TOOL ARCHITECTURE	29
FIGURE 13 MARKETPLACE LANDING PAGE	33
FIGURE 14 USER PORTFOLIO	33
FIGURE 15 MARKETPLACE WISHLIST	33
FIGURE 16 MARKETPLACE AND ASSET CREATION PROCESS MAP	35
FIGURE 17 MARKETPLACE ARCHITECTURE	37
FIGURE 18 MARKETPLACE API DOCUMENTATION	38
FIGURE 19: XR2LEARN GITHUB GROUP	42
FIGURE 20 GITHUB PROJECTS/USERS	42
FIGURE 21 CI/CD WORKFLOW	43
FIGURE 22 XR2LEARN SERVICES RUNNING ON STAGE ENV.	45
FIGURE 23 PROMETHEUS CHART	47
FIGURE 24 XR2LEARN DASHBOARDS ON GRAFANA	48
FIGURE 25 NETDATA WEB GUI	48
FIGURE 26 LIST OF RUNNING CONTAINERS IN CADVISOR GUI	49



LIST OF TABLES

TABLE 1 AUTHENTICATION AND IDENTITY MANAGEMENT CONFIGURATION	19
TABLE 2 COMMUNITY FORUM CONFIGURATION	23
TABLE 3 PLATFORM CONFIGURATION	25
TABLE 4 IPR TOOL CONFIGURATION	
TABLE 5 MARKETPLACE API	
TABLE 6 MARKETPLACE BACKEND CONFIGURATION	40
TABLE 7 MARKETPLACE FRONTEND CONFIGURATION	40
TABLE 8 XR2LEARN REPOSITORY IN DOCKER REGISTRY	
TABLE 9 STAGE SERVER FLAVOR	45
TABLE 10 PRODUCTION SERVER FLAVOR	46



EXECUTIVE SUMMARY

XR2Learn aims to create an XR technology hub, providing support in two different levels; the business level and the technical level. To provide business support XR2Learn will introduce its platform, featuring a suite of tools and assets that pave the way for new business processes and elevated interactions within a community centred around XR and education.

One of the key points of the business support is to enable the inscription of new revenue models using IPR tools based on blockchain technology. Blockchain is an emerging technology that has the potential to create novel approaches for XR developers and enthusiasts to share their knowledge, technology, and applications, through its decentralised nature and interoperability potentials. Furthermore, it enables flexible and transparent ways to handle ownership and offers new grounds for collaboration and fair redistribution of revenue.

This document will provide the description of the work to support the business strand offering of XR2Learn, providing an overview and introduction to the technological background, the technical specifications of the components developed such as the XR2Learn platform, the Marketplace and the IPR Tool, and the integration work carried for a fully functional business support system. Additionally, it will provide guidance that will enable and simplify the usage of the described tools. This document will be updated in M30, including new features and enhancements.



DoA	Description of Action
NFT	Non-Fungible-Token
DLT	Decentralised Ledger Technologies
МООС	Massive Open Online Courses
JWT	JSON Web Tokens
UI	User Interface
SSO	Single Sign On
WP	Work Package
PoS	Proof of Stake
IPR	Intellectual Property Rights
ERC	Ethereum Request for Comments
EIP	Ethereum Improvement Proposal

D2.2

Partners' names and acronyms			
CNIT	CONSORZIO NAZIONALE INTERUNIVERSITARIO PER LE TELECOMUNICAZIONI		
F6S	F6S NETWORK IRELAND LIMITED		
MAG	MAGGIOLI SPA		
LS	LIGHT AND SHADOWS		



SYN	SYNELIXIS SOLUTIONS SA
SUPSI	SCUOLA UNIVERSITARIA PROFESSIONALE DELLA SVIZZERA ITALIANA
UM	UNIVERSITEIT MAASTRICHT
НОՍ	HELLENIC OPEN UNIVERSITY
EADTU	VERENIGING VAN EUROPEAN DISTANCE TEACHING UNIVERSITIES
EITM	EIT MANUFACTURING SOUTH SRL



1 INTRODUCTION

1.1 PURPOSE AND CONTEXT OF THE DOCUMENT

The purpose of this document is to report the technical work of the business support branch of XR2Learn in the first 12 months of the project. It reports the activities carried out under the context of tasks T2.3 "XR2Learn Marketplace and IPR handling tools" and T2.4 "XR2Learn platform integration". Since blockchain is one of the key assets of our solution, a central objectives of this document is to describe the technology and its applications under the context of XR2Learn, to facilitate the easier understanding of the rest of the document. The other objective of the document is the detailed description of the technical work carried out by the tasks, with a focus on the business processes enabled by the XR2Learn platform.

Towards achieving the goals described above, the report starts with an overview of blockchain and NFTs technologies, it continues with the description of the technical work and concludes with a description of the integration. The detailed structure of the document is provided in section 1.2 below, while section 1.3 will introduce the underlying technologies of the solutions.

1.2 DOCUMENT STRUCTURE

The document splits the reporting of the technical work in three main sections and concludes it with information on the Integration of the different components. More specifically, the sections are the following:

- Section 2 describes the XR2Learn platform and its sub-components, more specifically,
 - the identity and authentication management component,
 - the catalogue,
 - the community forum,
 - and the technical wiki.
- Section 3 introduces the blockchain realm which consists of the implementations existing within or interacting with blockchain networks. In this context, two main components are detailed;
 - the IPR tool, that enables NFT creation and buying,
 - and the Smart Contracts, i.e. the code that exist within the blockchain network and provide an interface for the IPR tool.
- O **Section 4** dives into the details of the XR2Learn Marketplace, that consists of a backend and frontend component.
- O **Section 5** analyses the integration activities carried out.

1.3 TECHNOLOGY

The purpose of this subsection is to provide an overview of the key technologies adopted for creating the XR2Learn platform, and more specifically, the XR2Learn



Marketplace. This section can be used as a tool for familiarisation with such technologies, and thus pave the way for understanding the technical implementation better. The first part includes an introduction to blockchain technology in general, while the second part will focus more on the introduction to NFTs and their purposes.

1.3.1 Blockchain Technology

Distributed Ledger Technologies (DLT) is a term that is used to describe a specific set of applications that are decentralized and hold a record of the network activity in an immutable way. The most renowned application is Blockchain. It was introduced as a technology in 2011 when Satoshi Nakamoto published a whitepaper describing a peerto-peer distributed ledger system, namely Bitcoin.

At its core, Bitcoin is a decentralised application with the following functionalities:

- a. It created the first-ever digital cryptocurrency under the same name.
- b. It tracked transactions in a transparent and immutable way.

The term cryptocurrency refers to a digital currency, that can be used to exchange value within a network. It is a digital representation of value in a specific ecosystem. Each blockchain network defines its own currency that is used as the basis to enable all function within the network.

The value of blockchain as a technology is that it can be managed to handle transactions trustlessly. Meaning that in each transaction the need for a third party to provide trust, as a bank would be, is removed. Transactions are automatically enforced under the right conditions, they happen simultaneously no matter geographical limits, and each network participant can always view the transaction history of the whole network. Bitcoin was the first blockchain application to ever exist, and its emergence, paved the way for a new era of the internet.

In its formative years, blockchain was used purely for financial transactions and it did not allow much flexibility. Everything changed in 2014 when Vitalik Buterin proposed a new blockchain network called Ethereum, that for the first time, gave the ability to develop code and applications on blockchain. This was made possible through the definition of smart contracts; pieces of code written in an Ethereum native, Turingcomplete programming language called Solidity and deployed on the blockchain.

The programmatic ability paved the way for the usage of blockchain in a plethora of domains unimaginable before. Through the smart contracts, it created new ways to exploit blockchain and popularized its usage even among less tech-savvy people. The immutable nature of blockchain could be employed for enhancing the integrity of information in applications, by registering information in smart contracts. As the abilities multiplied the need for a structured and unified way of programming blockchain applications emerged. In the Ethereum community, people could propose new ideas or optimised ways to do things by creating Ethereum Request for Comments, for technical propositions, or Ethereum Improvement Proposals for improvements to the Ethereum protocol. The requests would then be discussed by the community to decide if they would evolve into standards. Other blockchain networks and communities developed their own workflows for proposing standards.

With these structures, new standards proposing new ideas were developed like the ERC-721 standard that described a way to create Non-Fungible Tokens that is described in the next subsection.



1.3.2 NFT

With the emergence of Ethereum, that inserted programmatic ability in the blockchain realm, new standards started to appear. Ethereum standards provide guidelines and technical documentation for implementing new functionalities and features in smart contracts. They outline precise rules that the smart contract developer must follow to ensure the compatibility of applications within the network. One of the most prevalent standards is ERC-721 [1], which describes NFTs.

Fungibility, as a concept, refers to the ability to exchange assets interchangeably, while maintaining parity of perceived value. Examples of fungible assets include currencies and gold. For example, a dollar is indistinguishable from and interchangeable with any other dollar. Ethereum supports a standard to create fungible tokens, namely ERC-20 [2]. Non-fungible assets, on the other hand, include the concept of uniqueness, and therefore cannot be exchanged. As an example, having two artworks, these have unique attributes, and possibly different kinds of value. Therefore, exchanging two artworks could lead to one party of the transaction having "less" while the other has "more". The uniqueness of NFTs is due to a set of properties defined for each token that are called metadata. At the time of writing there is not a unique approach for the structure of the metadata, and the schema usually depends on the underlying application. Usually, metadata are described IN JSON format where the keys are the properties, for example "content-type", or "link" which are associated with the proper values. They are a pivotal component in the NFT ecosystem, as they provide essential information about the digital asset represented and enable the interoperability among different marketplaces and platforms.

For the purpose of enhancing interoperability as well, different standards defining different functionalities emerged in the realm of NFTs. ERC-721 standard was the first standard to arrive that set the rules for creating Non-Fungible Tokens, or NFTs. Today, new standards have appeared, like ERC-1155, an upgrade of ERC-721, as well as standards implementing NFTs for other blockchain networks like PSP-34 for Polkadot [3].

NFTs are unique tokens, identified by a "token id", as described in ERC-721, that are connected to an asset through a set of metadata. ERC-721, provides an API to:

- a) Create NFTs
- b) Transfer NFTs
- c) Gain insights such as:
 - Learn who owns a specific token.
 - Get the supply of such tokens in the network.
 - Find the balance of a specific account in terms of NFTs.

NFTs at their core act in a way as a validation of the ownership of a specific asset, that is transparent widely across the blockchain network. Also, provided the tokens are uniquely identified by an id number referred to as "token id", they can also allow the tracking of the ownership of a token. NFTs can reference any type of item, physical or digital. From artworks and photos to real estate, the possibilities are constantly expanding. For instance, NFTs could be linked to:



- Images (photography or artworks)
- Videos
- Audio and music
- Tickets and coupons
- Software
- 3D models
- Physical objects

The world of NFTs made global headlines and became quickly renowned due to the enormous prices of some NFT artworks and collectibles sold. The first project bringing NFTs to the spotlight was the CryptoKitties project [4], a crypto-collectible game that allowed users to collect unique digital cats. NFTs present a diverse array of advantages to both creators and buyers:

- They empower creators by equipping them with tools to manage the ownership and copyright of their works.
- They allow creators to gain income from the reselling of their works on the secondary market with standards like ERC-2981: The NFT royalty standard [5].
- They can ensure the authenticity of the origin of works, through the transparent tracking of ownership mechanism.
- They allow for a way to transact digital or physical assets, in a secure and trustless way, breaking down geographical barriers.
- They give the opportunity to buyers to directly support their favourite artists and creators.

Currently, a broad spectrum of applications exists, encompassing different aspects of NFTs. The domain includes:

- **Applications for NFT creation:** these include tools that allow users without technical knowledge to create NFTs from their works. A popular example is Fotor [6].
- **Applications for NFT exchange:** these include NFT marketplaces, places where users can list and buy NFTs. One of the most popular examples is the OpenSea marketplace [7].
- **NFT games**: new types of games have emerged that include the NFT technology. These games allow the users to collect assets or in-game items in the form of NFTs. For example, one of the most renowned games is Axie Infinity [8].
- **Governance:** NFTs can represent voting rights in Decentralised Autonomous Organisations. These are organisations that are governed by smart contracts and not people. Every organisation sets its rules in smart contracts. These rules



define among other things, the membership eligibility criteria. Some DAOs choose to set NFTs as the way for an individual to obtain membership.

In the context of XR2Learn, NFTs are employed in the IPR Tool, that enables the flexible handling of the authorship and rights of XR related assets. Users will be able to create new NFTs of their assets using the simple and intuitive interface of the XR2Learn marketplace. These components will be analysed in sections 3 and 4.

2 PLATFORM

The XR2Learn platform acts as the central hub where users can have access to all the business and technical support services provided by the project. In detail, the platform provides access to:

- 1. **The XR2Learn catalogue:** A management system where the user can search for all the content provided by the project, described in section 2.2.
- 2. **The XR2Learn marketplace:** the place to create and exchange content, described in section 4.
- 3. **The XR2Learn community:** a forum for exchanging ideas and interacting with other users, analysed in section 2.3.
- 4. **The XR2Learn Wiki:** containing all the user guides for navigating through the platform as well as other services provided by the project, which is outlined in detail in section 2.4.

In the future version of the platform, more functionalities will be included, as for example a tool for on-demand application creation.

In Figure 1, an overview of the platform landing page when the user logins is provided. Its functionality is to supply the user with information about the different content that awaits them and guide them through what different sections contain.



FIGURE 1 PLATFORM LANDING PAGE



In the following subsections, the different components of the platform are described. Given the modular nature of the components, configuration and installation guidelines are also provided.

2.1 IDENTITY AND AUTHENTICATION MANAGEMENT

2.1.1 Description

XR2Learn manages the identity of the users based on an OAuth 2.0 mechanism. OAuth, which stands for Open Authorisation, is a protocol designed to grant authorisation to different services to gain access to user resources. It is a protocol that is compliant with the industry standards for user authentication and authorisation. The updated version used by the project, namely version 2, allows the definition of different levels of access for different services restricting the granting of access to the user resources to only the necessary per use-case. Furthermore, it can allow access to resources for users without ever sharing the user credentials with the service.

The Identity and Authentication Management component of XR2Learn is based on Keycloak [9]. Keycloak is an open-source solution for identity and access management that allows for fine-grained management of users and supports a variety of authentication protocols. Under the context of the project the OpenID Connect protocol is implemented.

OpenID Connect is an identity layer built on top of the previously described OAuth 2.0 mechanism. It allows different services to gain access to user information using JWT Access Tokens, a popular format for authenticating web services. OpenID Connect is a solution that puts user authentication as a central point, in comparison to OAuth 2.0 which is a more generic solution for resource access management, and therefore a better choice for the purposes of XR2Learn.

Keycloak features a GUI where administrators can set the different levels of access, various authentication flows that can be accommodated to different uses and endpoints where different services can be authorized. Additionally, it comes with predefined UI templates where the users can login and register. For XR2Learn, to provide a better web experience to the user, we customised these pages to conform with the branding identity of the project, ensuring a unified experience amongst all the different services provided.

When the users first visit the XR2Learn platform, they are greeted with a page that prompts them to authenticate by logging in or by registering. The registration page is portrayed in Figure 2.



First name	
Last name	
Email	
Password	
Confirm password	
Register	
× Beck to Login	

FIGURE 2 REGISTRATION PAGE

The information collected for the user at this stage is the following:

- First name
- Last name
- Email
- Password

This information is going to be shared by all the XR2Learn components and users can use these credentials to gain access to them. Users that have already registered can simply login through the form displayed in Figure 3.

Sign in to your account	
Remember me Forgot Password?	
New user? Register	

FIGURE 3 LOGIN PAGE



2.1.2 Installation and Configuration

The Identity and Authentication management component features a service based on Keycloak and PostgreSQL database for storage that are both dockerised.

The component can be configured by creating a .*env* file with the following structure:

TABLE 1 AUTHENTICATION AND IDENTITY MANAGEMENT CONFIGURATION

KEY	VALUE
POSTGRESQL_DATABASE	{Database name}
POSTGRESQL_USER	{Database user}
POSTGRESQL_PASSWORD	{Database password}
KEYCLOAK_ADMIN_USER	{The Keycloak administrator username}
KEYCLOAK_ADMIN_PASSWORD	{The Keycloak administrator password}
KEYCLOAK_CREATE_ADMIN_USER	true
KEYCLOAK_DATABASE_HOST	postgresql
KEYCLOAK_DATABASE_PORT	5432
KEYCLOAK_DATABASE_USER	{Database user}
KEYCLOAK_ENABLE_STATISTICS	true

The repository of the component can be found in

<u>https://github.com/XR2Learn/Authentication</u>. To install the component, clone the repository and run the following command:

• Docker compose up -d --build

After successful installation, the administration page to configure the service can be found in port 8000. An overview of the page is depicted in Figure 4. More instructions about the service can be found on the repository of the component, found under this link.



FIGURE 4 ADMINISTRATION CONSOLE OF IDENTITY AND AUTHENTICATION MANAGEMENT COMPONENT



2.2 CATALOGUE

XR2Learn Content Catalogue is a tool that acts as the main access point for everything XR2Learn has to offer. Every asset, technology, or knowledge is registered there, and users are able to make searches according to their needs. In the next subsection, a more detailed view of the tool is offered. As the tool is delivered as part of the main platform, there is not a dedicated part for the installation and configuration in this section.

2.2.1 Description

The Content Catalogue provides an interface to the user with the following functionalities:

- a) Search for items by name.
- b) Filter items according to type.
- c) Edit information on already registered items.
- d) Delete registered items.
- e) Add a new item.

The types of items that users can search for, or register are:

- Marketplace Assets
- Software Repositories
- MOOC

In Figure 5, an overview of the tool's interface is provided. In the figure it is depicted the first registered item, which is an XR2Learn beacon application, part of the work carried out in WP3 of XR2Learn. The information displayed is its title and description, its type as a "Software Repository" and a URL from where users can have access to this resource.



ZR2LE		gue 및 Marketplace ♡ Wishlist & Pr	rofile	📱 Technical Wiki		Ê	Welcome, User 🞗
		Catalog	gue, find everyth	ing XR2Learn ha	s to offer!		
_							
Ac	dd an Item	Search an item by title				All types	~
			Catalo	gue items			
Tit	tle	Description	Туре		Source		Actions
Be	eacon Application 3	This is the Beacon Application 3 of XR2Learn	Softw	are Repository	https://github.com/XR2Learn/Beacon-ap	p-3	
							< 1 >

FIGURE 5 XR2LEARN CONTENT CATALOGUE

Pressing the button "Add an Item" opens a pop-up window that prompts the user to fill out a form to create a new item. The form is presented in Figure 6.

ľ	Create Catalogue Item	×
	* Title	
Ľ		
_	* Description	
1		
E	* Туре	
	Туре	~
	* Source	
	Source	
	Create	

FIGURE 6 ADDING AND ITEM TO THE CONTENT CATALOGUE

From a technical perspective, the content catalogue is featured as part of the platform, built as a RESTful service employing the Django framework based on Python. More information about the related interfaces and installation is available in section 2.5.

2.3 COMMUNITY FORUM

XR2Learn provides a forum where the users can gather and exchange ideas. From the outcome of the work carried out in T2.1 and reported in D2.1, the importance of such a place to gather, learn, discuss and exchange knowledge was evident. A community forum is aligned with the core values and aspirations of the project and is a step



towards the direction of building a hub around XR technology in education. In the following subsections, the XR2Learn community forum is described and the technical guidelines for its configuration and installation are outlined.

2.3.1 Description

The XR2Learn community forum is based on Discourse [10], an open-source solution for building communities. Under the framework of XR2Learn Discourse was selected for the following reasons:

- It is adopted by many leading tech communities, such as Docker, Rust, and Django and many users will be already familiar to its interface and features.
- It is flexible in terms of configuration enabling the building of a community according to the needs of any use case.
- $\circ~$ It allows for the full refinement of the User Interface according to the branding of each project.
- It allows the authentication of the users using SSO and other popular methods.

In order to retain the identity of the project throughout its services, including the forum, Discourse is configured using a UI theme that is in accordance with the brand identity of the project, and making uses of the available branding elements such as the colour palette and logos. An overview of the main page of the forum is provided in Figure 7.

≡ 🥯 X	R2LEARN	Sign Up Login Q	
	Welcome to our community We're happy to have you here. If you need help, please search before you post. Search	٩	
all categor Topic	ries • all tags • Latest Top Categories	Replies Views Activity	
≭ Welco ■ General The first p visitors or communi	rne to our community! aaragraph of this pinned topic will be visible as a welcome message to all new and the second the second the second topic of your and the second topic of your and the second topic of your and you and more		
There are	e no more latest topics.		

FIGURE 7 COMMUNITY FORUM WELCOME PAGE

Additionally, a SSO is implemented that allows users to connect using their XR2Learn accounts, by integrating Discourse with the Identity and Authentication Management component. In Figure 8 the login page of the community forum is displayed. Users can either set up a different account for the forum and log in using this account or choose to connect with their XR2Learn identities. The second option will redirect the user to the XR2Learn login page as described in section 2.1.1.



		×
Welcome back 👏		
Log in to your account		
- Email / Username		
	➔ with XR2Learn	
Password		
I forgot my password Show		
Create your account		

FIGURE 8 XR2LEARN COMMUNITY LOGIN

Discourse is set up with an administrator account, that can be used to configure the community, topics and interface, as well as moderator accounts that can be used when the community is used in order to moderate the content uploaded.

2.3.2 Installation and Configuration

For the purposes of the project, Discourse is self-hosted and is delivered as a dockerised application. It is featured alongside the following services:

- A PostgreSQL database
- A Redis database
- Sidekiq service

To configure the service, a *.env* file must be created as depicted Table 2.

TABLE 2 COMMUNITY FORUM CONFIGURATION

KEY	VALUE
POSTGRESQL_DATABASE	{Database name}
POSTGRESQL_USERNAME	{Database user}
POSTGRESQL_PASSWORD	{Database password}
DISCOURSE_CLIENT_CREATE_DATABASE_EXTENSION	hstore, pg_trgm
DISCOURSE_CLIENT_CREATE_DATABASE_NAME	{Database name}
DISCOURSE_CLIENT_POSTGRES_USER	{Database user}
DISCOURSE_CLIENT_POSTGRES_PASSWORD	{Database password}
DISCOURSE_DATABASE_HOST	postgresql
DISCOURSE_DATABASE_NAME	{Database name}



DISCOURSE_DATABASE_PORT_NUMBER	5432
DISCOURSE_DATABASE_USER	{Database user}
DISCOURSE_EMAIL	{Admin email}
DISCOURSE_FIRST_NAME	{Admin first name}
DISCOURSE_HOST	{The domain where the forum will be hosted}
DISCOURSE_LAST_NAME	{Admin last name}
DISCOURSE_PASSWORD	{Admin password}
DISCOURSE_REDIS_HOST	redis
DISCOURSE_REDIS_PORT_NUMBER	6379
DISCOURSE_USERNAME	{Admin username}

The forum can be installed by cloning the repository and running the following command:

• docker compose up -d —build

2.4 TECHNICAL WIKI

The technical wiki of XR2Learn can be found under this link: <u>https://github.com/XR2Learn/.github/wiki</u> and the place where users will be able to find all the documentation and resources about:

- The tools of XR2Learn provided under the technical branch of the project that are the outcomes of the work carried out in WP3, specifically, the XRL2earn enablers and the Beacon applications. Since at the time of writing the work related to these is in progress, they are not featured currently in the wiki.
- The services provided under the business branch, specifically, the XR2Learn platform with its services along with the marketplace featuring the IPR handling of assets based on NFTs.

Additionally, it will feature guides on how to navigate through the platform.

The wiki is based on the solution provided by GitHub and is included in the project's repository (*https://github.com/XR2Learn*). The current structure of the wiki is depicted in Figure 9.



The XR2Learn Platform	Edit New page
fmelissari edited this page yesterday - 3 revisions	
The XR2Learn platform acts as the central hub where users can have access to all the business and technical support services provided by the project. In detail the platform provides access to:	+ Pages 6
• The XR2Learn catalogue: a management system where you can search for all the content provided by the project.	
O The XR2Learn marketplace: the place to create and exchange content, enabling blockchain functionalities with IPR	
support.	Authenticating with XR2Learn
 The XR2Learn community: a forum for exchanging ideas and interacting with other users. The XR2Learn Wiki: the current page where you will be able to find carefully curated instructions for payingting 	
through the platform as well as the other services provided by the project.	The XR2Learn Content Catalogue
In the figure below is the landing page of the XR2Learn platform upon logging in. You are greeted with a message containing information about what awaits you in the different sections of the platform.	The XR2Learn Marketplace
Party (2) FARN ≣ Catalogue ⊒ Marketolace O Woldist & Profile & Community (1) Technical Wei	The XR2Learn Platform
Hello, User!	Clone this wiki locally
Search the Catalogue to find all the new tools and content!	https://github.com/XR2Learn/.githul
Solin our community forum to share knowledge and discuss with other members.	

FIGURE 9 XR2LEARN WIKI

At the time of writing the following pages are included:

- **Authenticating with XR2Learn:** This page includes information on registering with XR2Learn and authenticating with the services.
- **The XR2Learn Community Forum:** This page guides the user on how to use the XR2Learn forum and its features.
- **The XR2Learn Content Catalogue:** The page explains the process of registering their resources on the catalogue, as well as the provided abilities for searching and sorting.
- **The XR2Learn Marketplace:** Information about the XR2Learn Marketplace and its features are outlined in this page. Additionally, the user is able to find guides on how to use the presented features as for example how to create a new NFT.
- **The XR2Learn Platform:** This is the place where the user discovers the XR2Learn platform, its aspects and functionalities, and the contents of each section.

2.5 INSTALLATION AND CONFIGURATION

The XR2Learn platform is a RESTful service based on the Django Framework. It is delivered alongside a PostgreSQL database as dockerised applications.

To configure it create a *.env* file with the structure as depicted in Table 3 Platform Configuration.

TABLE 3 PLATFORM CONFIGURATION

KEY	VALUE
POSTGRES_DATABASE	db
POSTGRES_HOST	Postgres-service
POSTGRES_PASSWORD	{Database password}



POSTGRES_USER	{The database user}
POSTGRES_PORT	5432
ADMIN_PASSWORD	{The Django admin password}
REDIS_HOST	Redis-service
REDIS_PORT	6379

To install it clone the repository found in <u>https://github.com/XR2Learn/Platform-</u> <u>Service</u> and run the following command:

• docker compose up -d --build

3 BLOCKCHAIN REALM

This section refers to the components that enable blockchain functionalities. XR2Learn handles the following as a separate layer that the marketplace utilises to unlock the decentralisation aspects of the platform. This layer includes two components:

- a) The IPR Tool, that provides an interface for blockchain functionalities to the other XR2Learn services and is described in subsection **Error! Reference s** ource not found.
- b) The Smart Contracts, decentralised applications that exist on chain and provide an interface to blockchain based assets and functions, described in subsection 3.2**Error! Reference source not found.**

3.1 IPR TOOL

This section describes the IPR tool of the XR2Learn platform analysing its different aspects. The IPR tool is responsible for handling the wrapping and distribution of items under specific IPR licenses. An overview of the tool is provided in subsection 3.1.1, the tool's architecture is outlined in subsection 3.1.2 and the technical deployment details are provided in subsection 3.1.3.

3.1.1 Description

The IPR tool is a utility component to the XR2Learn marketplace. Its role includes the creation of media as NFTs under specific IPR licenses but extends also to the exchange of NFT items and other functionalities under the context of NFTs that are described in detail in this section. The IPR tool operates as a RESTful service that exposes endpoints that can be consumed by the internal services of the XR2Learn platform, such as the marketplace.

Ethereum currently utilises two standards that describe NFTs. ERC721 [1], the Non-Fungible Token standard that was the first to emerge and is the most widely used. And the ERC1155 [11] that is the Multi Token standard. ERC-1155 was introduced as an



enhanced and optimised version of its predecessor ERC721, that allows the creation of both fungible and non-fungible tokens.

Under the context of XR2Learn the ERC1155 is the selected standard to create NFTs, for the reasons delineated below:

- It offers flexibility by allowing for the creation of both fungible and non-fungible tokens.
- It offers enhanced security in comparison to the ERC721 standard, for example it interfaces a function for secure transfers.
- It offers flexibility with the metadata.
- It allows for batch transfer of tokens, making it more cost and energy efficient.

One of the main functionalities of the IPR tool is to create JSON objects correlating to transactions that will allow the minting of an NFT to a user's account. Figure 10 depicts the process to create an NFT as enabled by the IPR Tool.



FIGURE 10 NFT CREATION SEQUENCE DIAGRAM

The tool uses the NFT data as specified by the user during the creation of an NFT in the marketplace, specifically:

- The name of the NFT
- A short description of the NFT
- The supply
- The price of the NFT
- A URL pointing to an image that displays the NFT



- $\circ~$ Optionally, a URL that can be used by the NFT creator to point to their personal website or blog
- The type of asset the NFT corresponds to, including usual media (text, image, video, sound file) or more complex data types, (repositories).
- The license the NFT will be distributed with
- The user address to mint the NFT to

The data outlined above are used to create the metadata of the NFT. An example depicting the structure of the metadata is depicted in Figure 11. The structure of the metadata is defined in order to satisfy the needs of the XR2Learn platform but also for NFTs created on XR2Learn to be able to be listed as well on the most popular NFT marketplace, OpenSea [7]. The structure is compliant with the metadata standard of the OpenSea marketplace, for it to be able to correctly display NFTs created on the XR2Learn platform and allow the users to sell their work there if they choose, enhancing this way the power of the NFTs created on XR2Learn and expanding its reach.



FIGURE 11 NFT METADATA

The tool, using the data provided by the user on the marketplace, creates a JSON file for the metadata of the NFT that it publishes on IPFS. IPFS is a decentralized storage based on a peer-to-peer network [12]. For the context of XR2Learn, an IPFS node is hosted by the project to be used as a storage endpoint and gateway to the NFT media. Afterwards, it creates the NFT smart contract based on a template contract that employs the ERC1155 standard, and specifying the metadata previously created. Finally, the tool builds the transaction object to deploy the smart contract and is going to be used by the marketplace.

Furthermore, the NFT tool includes as well other complimentary functionalities as outlined below:



- Checking if a user wallet address is the owner of a specific NFT
- Publishing files on IPFS and returning a relevant link

3.1.2 Architecture

The IPR Tool's architecture is depicted in Figure 12.



FIGURE 12 IPR TOOL ARCHITECTURE

The tool consists of a RESTful service based on Flask, a Python framework. This service is the logic layer of the tool. It implements the logic for all the basic functionalities provided. The service is responsible for interacting with blockchain networks supported by XR2Learn. Currently the networks supported are Ethereum and Polygon PoS. Information about these is provided in section 3.2 The storage layer of the component is an IPFS node. This node provides an interface to upload content on the IPFS network while simultaneously acting as a gateway for retrieving content from the network. The node can be either excluded from the



public network and act as a standalone or it can be connected to the peer-to-peer network and provide access to publicly available content. As the second option enhances decentralisation and as the content can only be accessed with the URL of the content, or more specifically, its unique id, this is the choice that is implemented in XR2Learn. The hosted node is responsible as well for interacting with the public network. This solution enhances the robustness and integrity of the storage layer. Finally, the component provides access to its services to other components through an API. The details of the interface can be found in subsection 3.1.3. The API provides documentation based on Swagger.

3.1.3 Installation and Configuration

Both the IPR tool and the IPFS node are dockerised. To install the services there are the following prerequisites:

- Docker installed
- Docker-compose installed

To configure the service, a *.env* file must be created with the structure as depicted in Table 4.

КЕҮ	VALUE
ADDRESS	The wallet address of XR2Learn
KEY	The wallet key of XR2Learn
ETHEREUM_CHAINID	The chain id of Ethereum, by default Sepolia Testnet chainid (11155111)
ETHEREUM_URL	The URL of the Ethereum RPC endpoint
GUNICORN_CMD_ARGS	"-c backend/gunicorn.py"
HOST	The address of the IPFS URL
POLYGON_CHAINID	The chain id of the Polygon network, by default Mumbai Testnet chainid (80001)
POLYGON_URL	The URL of the RPC endpoint of Polygon
PORT	The port of the IPFS URL for writing
IPFS_URL	The URL of the IPFS Gateway for retrieving files

TABLE 4 IPR TOOL CONFIGURATION

The tool can be installed by cloning the repository of the project *https://github.com/XR2Learn/IPR-Tool* and running the following command:

• docker compose up -d -build

3.2 SMART CONTRACTS

The smart contracts of XR2Learn are pieces of code written in the Solidity language that are deployed in the supported blockchain networks, namely Ethereum and Polygon. In the first version of the XR2Learn platform, the smart contracts implement functionalities specific to the NFT creation and minting.

According to the ERC-1155 Multi Token standard, as specified by the Ethereum community, XR2Learn extends its scope to fit the needs of the XR2Learn marketplace. The functionalities implemented are:



- They specify the metadata of the NFT by configuring the URL of the deployed metadata JSON file for each specific NFT. The details of the metadata are specified by the users during the NFT creation workflow.
- They specify the mandatory information of the NFT according to the standard, such as its name, supply and price. These are also specified by the user as demonstrated in section 4.
- They enable the minting functionality, allowing users to mint such tokens according to the registered supply.
- They enable transferring functionalities, allowing users to transfer such tokens. The transfer is successful if the potential buyer sends a transaction where the value is according to the price set by the NFT owner, ensuring its proper renumeration.

The NFT Smart Contracts are created and deployed dynamically according to the logic implemented in the IPR Tool. They are created according to an implemented template that is configured with the details specified by the user.

Once deployed, smart contracts are accessible to any member of the blockchain network. This functionality is enabled by the decision to use public blockchain networks, adding more value to the created NFTs, enhancing the integrity of the solution, and ensuring its interoperability with multiple other decentralised platforms compliant with the same standards, as for example OpenSea, a notable NFT Marketplace.

The template of the smart contract and the related interfaces can be found in the following repository: <u>https://github.com/XR2Learn/Smart-Contracts</u>.

3.3 FUTURE DEVELOPMENTS

For the first release of the XR2Learn platform the IPR tool focuses on implementing a flexible and basic structure that will be the basis for implementing new and more complex features in the final version of the platform.

The design and architecture of the NFT tool as foreseen, will optimise the initial architecture in terms of cost and energy by implementing a proxy schema, where the logic and storage of the NFT contracts will be separated, while implementing as well, upgradeability features.



4 MARKETPLACE

4.1 **DESCRIPTION**

XR2Learn is offering a marketplace for users to exchange their content and artistic creations. It works as a place where XR innovators and creators can advertise and sell their work, while being enumerated fairly.

The XR2Learn marketplace features:



• **The marketplace landing page:** where users can view all the marketplace items



FIGURE 13 MARKETPLACE LANDING PAGE

FIGURE 14 USER PORTFOLIO

• A wish list: where users can save items they like.

4.1.1 Browsing contents

The marketplace landing page is the place where the user has access to a preview of assets and promotion materials featured in XR2Learn. They can personalise the search towards their need by applying filters based on:

- a) The type of asset, for example, image, 3D models, software, or MOOC from the Skills.move [13] platform of EIT Manufacturing.
- b) Specific tags related to the content, for example, the user could search for software NFTs by applying the tag "software".

Also, the user can search for specific items using the name of the NFT they wish to find, or they can order the items from the most popular to the least, and from the most recent to the oldest.

Furthermore, they can click on an item that they find interesting and be provided with more information about this item, like for example a short description about the NFT, maybe an external link to the creator's personal blog and the list of tags applied to the specific item. These contents are referred to as "promotional materials" that enhance the items with branding materials of creators.

If the user considers buying the item and they wish to safekeep it for later, they can "like" it, and it is automatically added on their personal wishlist. Later, they can visit the wishlist page where they can edit it by removing the items they no longer find interesting.

From the marketplace page or from an item's page, the user can also add items to their cart, from where they can later proceed to buy them using their web3 wallet.

4.1.2 Creating content

For users that wish to create their assets as NFTs, the marketplace also features a plus button which can be used to create a new NFT. The creation of an NFT process features the following steps:

- i. The user selects the type of asset they want to create as NFT and uploads their files.
- ii. The user specifies how they wish to distribute their content, namely whether they want to create an NFT or to distribute the item for free. This choice will alter the flow of the following steps.

If they user chose to create a free item, then the process continues as:

- iii. The user inserts basic information to create a listing on the marketplace, like a title and a description.
- iv. The item is created and featured on the marketplace where users can download it.

If the user chose to create an NFT item, then the process continues as:

- iii. The user is prompted to connect to their Metamask wallet.
- iv. The user inserts details specific to the NFT, as the name, description, price and supply, as well as the license they wish to distribute it with.

- v. The user inserts details specific to the listing on the XR2Learn marketplace, as for example different displaying options for the NFT. As well as applying tags to their items. These tags can be specified by the user, or they can select tags already created by other users.
- vi. Finally, the item is minted to their wallet and added on the XR2Learn marketplace.

The entire process is shown in Figure 16 below.

FIGURE 16 MARKETPLACE AND ASSET CREATION PROCESS MAP

4.1.3 Key Features

Licensing: XR2Learn will allow the users to select the license with which they wish to distribute their content from a predefined list. The list of licenses will be decided by XR2Learn according to the needs of the project. In the first version, a limited set of licensing options will be supported. The criteria considered for defining the list include:

- Supporting different levels of openness.
- Supporting the content types included in the XR2Learn marketplace.
- Clarifying post-resell rights.

Unlockable content: The XR2LEARN marketplace will give the option to NFT creators to share unlockable content with their NFTs. The buyer or buyers of the NFT will then be able to visit the item on the marketplace again and unlock content private only to them. The unlockable content will be provided as text, but it can be:

- A link to a high-quality version of an asset that is distributed as NFT.
- Coupons or redeem codes unlocking special content.
- A link to private repository for NFT representations of software.

The possibilities are up to the NFT creator to decide. However, the unlockable content feature is part of the XR2Learn marketplace, meaning that the unlockable content will not be provided for sales outside of XR2Learn. However, users that have purchased the NFT in external markets will have the possibility to access the unlockable content, by visiting XR2Learn as the NFT owners by connecting with the wallet they have used to make the purchase.

4.1.4 Future developments

In the future versions of the platform, the marketplace will offer new functionalities such as:

- It will give the option to creators to set a royalty, so that creators can benefit from the reselling of their works on the secondary market. The functionality will be implemented according to the ERC2981, which is the NFT Royalties standard.
- It will give the NFT owner the option to sell their NFT either by setting a fixed price, as implemented in the first version, or by setting a timed auction.
- It shall give the NFT creator the option to create their NFT without paying for the gas cost, this will be paid by the buyer of the NFT. Gas refers to a unit of measuring the computational complexity of each transaction, and it is translated to fees that are paid to the miners or validators of the transaction. Therefore, each transaction, as is the NFT creation, is accompanied by a fee. The "free minting" would make the process of creating an NFT more accessible to users with limited technological expertise.
- It will feature a reputation system that will act as an achievement system where users' contribution will be measured, and they will have the possibility to be rewarded for them. Creators will have the ability to build their status in the platform through gathering badges that will be displayed on their XR2Learn portfolios.

4.2 ARCHITECTURE, SPECIFICATIONS, AND INTERFACES

From an architectural perspective, the NFT Marketplace consists of two main layers: the application layer and the storage layer. The application layer itself is comprised of two main components:

- **The backend service:** It implements the logic of the marketplace and exposes an API interface along with a Swagger documentation. It is build using the Django framework.
- **The user interface:** It implements a user-friendly and intuitive front-end application based on React JS.

The storage layer consists of a PostgreSQL database.

Figure 17 below shows the high-level architecture of the NFT Marketplace.

FIGURE 17 MARKETPLACE ARCHITECTURE

The marketplace is interacting with several other services of XR2Learn to provide its full functionality. Specifically, it is interfacing with:

- The Identity and Authentication component, that is described in section 2.1, in order to gain access to information about the users of the platform.
- The NFT tool, in order to provide blockchain functionalities related to NFTs.

The backend service offers an API to expose its functionalities, that at the time of writing, consists of 36 endpoints. The API documentation is provided using the Swagger framework, as depicted in Figure 18.

Appl Image The XR2Learn Marketplace. Contact Filinia Medisari Gruppo Maggioli ® and Synelixis Solutions ®	
	Authorize 🔒
cart	^
GET /cart/	≜ ∨
POST /cart/	â ~
PUT /cart/	
PATCH /cart/	
DELETE /cart/	
dj-rest-auth	^
POST /dj-rest-auth/login/	
POST /dj-rest-auth/logout/	≜ ∨
POST /dj-rest-auth/password/change/	
POST /dj-rest-auth/password/reset/	≜ ∨
<pre>POST /dj-rest-auth/password/reset/confirm/</pre>	≜ ∨

FIGURE 18 MARKETPLACE API DOCUMENTATION

In Table 5 Marketplace API below we provide an overview of some representative endpoints, implementing some core functionalities.

TABLE 5 MARKETPLACE API

NFT Marketplace		
	cart/	
	Description	The interface allows the handling of the user's cart. It covers functionalities related to the creation, editing and deletion of the cart.
	Protocol(s) used	HTTPS
	Methods	POST, GET, PUT, PATCH, DELETE
Interfaces	Message	<pre>POST/PUT/PATCH: {</pre>
	nft/	

Description	This endpoint handles the creation, editing and deletion of NFT, object on the side of the backend service.
Protocol(s) used	HTTPS
Methods	POST, GET, PUT, PATCH, DELETE
Message	<pre>POST/PUT/PATCH: { "name": "string", "price": 0, "license": 0, "supply": 0 } GET: [{ "id": 0, "name": "string", "price": 0, "license": 0, "license": 0, "supply": 0, "creator": "string" }]</pre>
marketplace-item/	
Description	This interface exposes functionality related to the handling of items on the marketplace.
Protocol(s) used	HTTPS
Methods	POST, GET, PUT, PATCH, DELETE
Message	<pre>POST/PUT/PATCH: { "name": "string", "description": "string", "type": "image", "owner": 0, "tags": [], "external_link": "string", "display_image": "string", "nft": 0, "unlockable_content": "string" } GET: [{ { "id": 0, "name": "string", "description": "string", "type": "image", "owner": 0, "tags": [], } </pre>

"external_link": "string", "created": "2023-10- 12T13:17:07.658Z",
--

4.2.1 Installation and configuration guidelines

Both applications are dockerised. Below are described the configuration and installation steps for each:

Marketplace Backend Application

• Create a *.env* file with the structured reported in Table 6 Marketplace Backend configuration.

TABLE 6 MARKETPLACE BACKEND CONFIGURATION

KEY	VALUE
POSTGRES_HOST	Marketplace-postgres
POSTGRES_DATABASE	db
POSTGRES_USER	user
POSTGRES_PASSWORD	{POSTGRES_PASSWORD}
GUNICORN_CMD_ARGS	"-c backend/gunicorn.py"
ADMIN_PASSWORD	{ADMIN_PASSWORD}
REDIS_HOST	marketplace-redis
REDIS_PORT	6379
AWS_SECRET_ACCESS_KEY	{S3_BUCKET_SECRET_KEY}
AWS_SECRET_KEY_ID	{S3_BUCKET_ACCESS_KEY_ID}
AWS_STORAGE_BUCKET_NAME	{S3_BUCKET_NAME}

• Run the following command: *docker compose up --build -d*

Marketplace Frontend Application:

• Create a *.env* file with the structure reported in Table 7 Marketplace Frontend Configuration.

TABLE 7 MARKETPLACE FRONTEND CONFIGURATION

KEY	VALUE
REACT_APP_AUTH_URL	{The URL of the authentication and identity management}
REACT_APP_AUTH_REALM	{The realm name}
REACT_APP_AUTH_CLIENT	{The client_id}
REACT_APP_AUTH_SECRET	{The client_secret}
REACT_APP_API	{The URL of the marketplace backend API}

• Run the following command: *docker* compose up --build -d

5 INTEGRATION

The following section is dedicated to the integration activities carried out under task 2.4 of the project. It dives into the details of the hosting infrastructure and the different environments created for testing, developing and delivering the XR2Learn services.

5.1 HOSTING INFRASTRUCTURE

This section presents the hosting infrastructure and the tools that are used for the development, testing, and deployment of the XR2Learn platform. XR2Learn uses DevOps approaches that enable the exploitation of the platform as an integrated ecosystem based on the requirements set by WP2. Building on top of these technologies, XR2Learn enables the management and control of the DevOps cycle for the development, continuous deployment and integration of added value services and their components.

However, in order to support the above-mentioned activities, a consistent infrastructure was created which supports virtualisation of HW resources, i.e. processing power, memory, storage and network resources. For this reason, the approach used is the introduction of incremental steps towards integration, validation and testing of XR2Learn components exploiting three environments (aka infrastructure versions). These infrastructures are: (a) Development infrastructure; (b) Stage infrastructure and (c) Production infrastructure. This section discusses mostly the deployment of these infrastructures as well as the description of the tools that are used in the context of XR2Learn in SYN and MAG premises.

5.1.1 CI/CD workflow

The core component of the development process XR2Learn is a public instance of GitHub which provides a collaborative environment for software development, version control and CI/CD management. Each developer has two-factor authorized access to the XR2Learn project, and she/he can create her/his own repository and add other users as members. GitHub also provides a complete CI/CD framework based on pipelines, namely GitHub Actions.

= () ×	R2Learn	Q Type [] to search	>_ + • O n @ 🔂
Overview	🖟 Repositories 🕕 🗄 Projects 🗇 Packages 🗛 Teams A People 11 🕸 Settings		
	XR2Learn Leveraging the European XR industry technologies to empower immersive learning and training		Follow
	READRE.md XR2Learn - HORIZON EU XR2Learn will deliver the XR2Learn platform around which it establishes a cross-border innovation community for XR in bringing XR technology providers, application designers, education experts, application developers, end-users and dec direct access to communicate, collaborate and matchmake interests enabling also bottom up innovation creation. XR21	learning, ision makers in Learn will go	View as: Public View as: Public View are viewing the README and pinned repositories as a public user. View can pin repositories visible to anyone. View can hide the tasks we've suggested on this page and bring them back later.
	beyond offering sound technical and business support for the creation of XR applications for educations XR2Leam will authoring tools for development of applications through its platform, deliver tools for emotion/affect detection and fo adaptation of the learning experience to the user needs and emotions, deliver guidance relevant to educational design definition, provide opportunities for piloting and user testing mediated by the large networks brought by XR2Leam pail tools that enable and boost the re-use and sharing of the learning materials/XR applications, offer business developm additionally, support IPR management through NFTs enabling novel business model implementation. XR2Leam will su (ICT-SMES) all the way from ideation to commercialisation offering them tailored business and technical support as we through FSTPs.	provide access to r automated and use case rtners, promote nt support and opport innovators II as direct funding	Discussions Set up discussions to engage with your community! Turn on discussions
	Visit <u>Tech Wiki</u>		Repositories

FIGURE 19: XR2LEARN GITHUB GROUP

At this point, there are 18 active repositories and 20 developers who create code for the XR2Learn platform and components (Figure 20). A generic view of the CI/CD workflow is shown Figure 21, in which we can see how the GitHub tool in integrated with the stage and production environments based on CI/CD pipelines.

≡ () XR2Learn		Q Type 🖉 to search	> + • O n @ 🛟
Overview 📮 Repositori	es 18 🗄 Projects 🛇 Packages 🗛 Teams 🕺 People 11 🕸 Settings		
	Q Find a repository Type -	nguage 👻 Sort 👻	Rew repository
	.github Public ♥0 ☆0 ♥1 0 Updated on Oct 24	_	·
	Authentication Private		
	Beacon-app-1 Private ● C# ♀0 ☆0 ① 1 0 Updated on Nov 14	_	
	Beacon-app-2 (Private) ● C# ♀0 ☆0 ⊙ 0 № 0 Updated last week	_	
	Beacon-app-3 (Private) ● C# ♀0 ☆0 ⊙0 ♫0 Updated last month	_	
	Community Private ¥0 ☆0 ⊙0 № 0 Updated 15 hours ago		
	en-1-interact (Private) ●C# ♀0 ☆0 ⊙0 ♫0 Updated on Nov 14	_	

FIGURE 20 GITHUB PROJECTS/USERS

5.1.2 CI/CD pipelines

Following the CI/CD approach, every time that a developer pushes small code chunks to a project hosted in a Git repository, she/he triggers a pipeline of scripts to build, test, and validate the code changes before merging them into the main branch, which natively in GitHub are mentioned as workflows. Next, the CI/CD framework deploys the new version of the component to the stage environment.

Each workflow is associated with one or more events, which are conditions that when met trigger it, and it is consisted of a set of jobs which can also be triggered manually

through the web interface by pressing the appropriate button. After triggering the job is assigned to the runner process which links to the specific repository and executes it in environment of our choice. The output of the job is displayed in real time to the GitHub web interface, from the GitHub Actions tab. This set of steps can be shifted or skipped depending on developer's requirements. Next, we describe in detail the different steps of a typical the CI/CD pipeline.

FIGURE 21 CI/CD WORKFLOW

5.1.2.1 Pipeline jobs

Container build: XR2Learn uses Docker containers as host for its components, so the first step is to build the images for all components that are developed in the repository and push them to private docker registry of XR2Learn. The building of a Docker container can be scripted as follows:

```
build_images:
    only:
    refs:
        - main
    stage: build
    script:
        - docker build -t synelixis/xr2learn-marketplace-backend:latest .
        - docker push synelixis/xr2learn-marketplace-backend:latest
    tags:
        - stage
```

Where:

- **docker build**: The instruction to build the container.
- **-t XR2Learn-repo /<container_image_name>:** The name of the container image. The first part is the internal docker registry, and the second part is the image name.
- **docker push:** Publishes the image on the docker registry.

Unit tests: During the unit test stage developers can perform software testing of an individual unit or component. This kind of tests isolates a section of code and verifies its correctness. The use of containers has a significant advantage in designing and executing unit tests since the developer is not required to create mock-ups of each component as it depends on his implementation. For example, in case of databases, sometimes it is time expensive to build a mock-up. With Docker, it is quick and straightforward to just start a docker container with the database and connect the under-test container to it. Once a container passes the unit tests, the test database can be easily removed.

Service deployment: During the deployment stage the containers of each service are deployed on the stage environment. The containers can be started separately, or the developers can use technologies like docker-compose, docker swarm etc.

deploy:
only:
refs:
- main
stage: deploy
script:
- docker compose up -d
tags:
- stage

Where:

• **docker compose up**: The command to start the docker-compose.

Production images: The images that are used in the production env are created from the last job of the pipeline of each repository. This job is triggered manually from the developer each time she/he decides to promote the current version of her/his service from the stage to the production environment. During this stage the latest version of the stage images are tagged with the appropriate version number and they are pushed to private docker registry.

```
create_prod_images:
    only:
    refs:
        - master
    stage: create_p_images
    script:
        - docker rmi synelixis/xr2learn-marketplace-backend:prod || true
        - docker build -t synelixis/xr2learn-marketplace-backend:prod -f
dockerfile/prod/Dockerfile .
        - docker push synelixis/xr2learn-marketplace-backend:prod
when: manual
    tags:
        - stage
```

In case an error occurs during the execution of one of the jobs the entire pipeline fails, and a notification mail is sent to the involved users (developers/maintainers etc) who triggered the pipeline either by pushing new code or manually through the web interface.

TABLE 8 XR2LEARN REPOSITORY IN DOCKER REGISTRY

General Tags Builds Collaborators Webhooks Settings	
synelixis / xr2learn-marketplace-backend	Docker commands To push a new tag to this repository:
This repository does not have a description 🧪	docker push synelixis/xr2learn-marketplace-backend:t agname
🔇 Last pushed: 6 minutes ago	

Following these methodologies, developers are able to catch bugs and errors early in the development cycle and ensure that all the code deployed to production complies with the established code standards.

5.2 DEVELOMPENT ENVIROMENT

The Development environment is a workspace for developers to test anything they want without worrying about affecting any other users or developers working on a live deployment. In most cases, a Development environment is set up on a local server or on the machine that developers use, so the source code is ready to be executed and modified if needed. So, in XR2Learn, developers use the tools and technologies of their choice (i.e. programming language, frameworks, IDE etc) and build the appropriate docker containers. Next, the service is executed in their local Development environment.

5.3 STAGE ENVIROMENT

The Stage environment is hosted on the Synelixis cloud infrastructure, and it consists of VMs in which all the XR2Learn services are deployed and tested. Based on specific CI/CD pipelines, each component can be built, deployed and tested in the Stage environment using the automated procedures that are installed in the Stage environment server. This approach provides the developers an area where the entire XR2Learn platform can be deployed, tested, and validated before the deployment in the production. The Stage environment is scalable, and it can be enhanced with more resources by adding new servers as the XR2Learn platform grows. Table 9 shows the predefined flavor of the servers that host the stage environment and Figure 22 presents the list of the running services in stage server.

Server flavor	
vCPUs	4
RAM memory	4GB
IP address	192.168.190.107
Storage	40GB
OS	Ubuntu 20.04
Software	Docker/Docker Compose

<pre>ubuntu@platform:~\$ docker ps - authentication-kevcloak-1</pre>	format	"{{.Names}}"
kevcloak-postgres		
forum-sidekig		
forum-discourse		
forum-redis		
forum-postgres		
xr2learn-marketplace-front-ap	р	
xr2learn-app		
backend-database		
backend-redis		
xr2learn-marketplace		
marketplace-database		
marketplace-redis		

FIGURE 22 XR2LEARN SERVICES RUNNING ON STAGE ENV.

5.4 PRODUCTION ENVIROMENT

The Production environment is hosted on Maggioli premises, and it consists of VMs in which the stable version of the XR2Learn platform is deployed and offered to the end

users. The deployment is based on a CI/CD pipeline specially created for this process and the platform can be deployed either all at once or each component can be manually deployed separately. In any case, the update process can be done without any affect to the user's data because all databases are mounted on virtual volumes which are not affected by the redeployment of the components. The CI/CD pipeline is kept in a separate repository. Following the same design approach of the stage, the production environment is also scalable and it can be easily scaled up by adding new servers to provide the appropriate resources in order to ensure the QoS level to the users. Table 10 shows the predefined flavor of the servers that are hosting the Production environment.

Server flavor	
vCPUs	2
RAM memory	8GB
IP address	40.66.55.229
Storage	60GB
OS	Ubuntu 22.04.3
Software	Docker/Docker Compose

TABLE 10 PRODUCTION SERVER FLAVOR

5.5 MONITORING RESOURCES

XR2Learn platform implements an open, service-oriented architecture which aims to cover all the operational aspects from actual realization, test, trials and production operation. To achieve these goals, it is necessary to provide the appropriate tools in order to guarantee (a) the integration of the services developed in WP2 and (b) the appropriate resources allocation for service deployment in the different environments (i.e. stage, production etc.). Therefore, in the XR2Learn platform, we designed and deployed a state-of-the-art monitoring and analysis framework based on open-source tools for collecting performance metrics from the servers in stage and production environments.

Under this perspective, it is of paramount importance to collect monitoring data from as many possible sources. In the implemented system, there are four different types of sources for collecting data:

- Containers (i.e. services running as docker containers)
- VMs (i.e. service running on VMs or VMs hosting stage/production environments)
- Network traffic (i.e. network traffic on physical and virtual level)
- Application-level metrics (i.e. service availability)

Apart from the collection and the process of monitoring data related to the performance of the XR2Learn's services and infrastructure, the monitoring framework will accommodate specific alerting rules for real-time notification events. In this respect, the monitoring framework will offer the capability to developers to define service-specific metrics and rules, whose violations will generate alerts.

The architecture of the monitoring system consists of the following components: **Monitoring tools:**

• Prometheus server [14] stands as the central point of event monitoring, storage and alerting. All performance metrics are collected, using a HTTP pull model, and stored in a time-series database (Figure 23). Some of the key features that make this server suitable for the proposed architecture are: (a) use of a flexible

query language (PromQL), which makes easier the interconnection with external systems; (b) the existence of many opensource implementations (exporters) for exposing monitoring metrics from various applications, and it is also quite easy to create new ones; (c) autonomy as there is no reliance on complex distributed storage mechanisms and (d) new monitoring targets can be easily added via reconfiguration or using the file-based service discovery mechanisms.

- The Prometheus Pushgateway [15] allows XR2Learn services, to expose their metrics to Prometheus. Since this kind of microservices may not exist long enough to be scraped, they can instead push their metrics to a Pushgateway. The Pushgateway then exposes these metrics to Prometheus server.
- The Alertmanager [16] handles alerts sent by client applications such as Prometheus server. It takes care of deduplicating, grouping, and routing them to

the correct receiver integrations such as email, PagerDuty, or OpsGenie. It also takes care of silencing and inhibition of alerts.

• Grafana [17] is an open-source solution for running data analytics, pulling up metrics that make sense of the massive amount of data and it provides interactive visualization web dashboards (Figure 24).

Ø	28 Home	40 C 40a
a +	Welcome to Grafana	Need help? Documentation Italiziala Community Public Stads
8 Q	Deabhoards Starrid dashboards Recently viewed dashboards	Latest from the Bog Orafana Labs in 2023. Top 10 moments of the year If you ask us to score 2023, we would give B a 10 out of 10. Not only because we released Grafana 10 and celebrated the 10 year anniversary of the Grafana project with the first-over Golden Graf Awards and the lawards of a face part documentary units. But also because we requireded our OSB offenting: veelcome
@ D	XRDLeamEnvironments Cr XRDLeamEnvironments Cr Promethieus 2.0 Overview Cr Advits Cr	to the turing, data and synthesis and unitate asymptity and but hully interaction or unitate documentation, which now includes interaction of OpenTelemetry best practices: A user's guide to getting started with OpenTelemetry. Due to if you've landed on this bidg you've likely effer considering starting your OpenTelemetry journey or you are well on your way. As OpenTelemetry bases grown, nodel wy with the observability constraining of openTelemetry journey or you are well on your way. As OpenTelemetry bases grown, nodel wy with the observability constraining of openTelemetry journey or you are well on your way. As OpenTelemetry bases proven, nodel wy with the observability constraining of openTelemetry journey or you are well on your way. As OpenTelemetry bases and the open Telemetry adoption of the open Telemetry and the openTelemetry journey or you are well on your way. As OpenTelemetry bases and the openTelemetry adoption of the open Telemetry and the openTelemetry journey or you are well on your way. As OpenTelemetry bases and the openTelemetry adoption of the openTelemetry and the openTelemetry journey or you are well on your way. As OpenTelemetry bases and the openTelemetry adoption of the openTelemetry bases and the openTelemetry adoption of the openTelemetry adoption of the openTelemetry adoption of the openTelemetry adoption of the openTelemetry adoption of the openTelemetry adoption of the openTelemetry adoptio
		Imperiment an Uppin learning synaptic cristina uses is tury committee to uppin learning compatibility into our opin source projects and our products while its ormaning activity movied in the Cit Carommunity. New In Grafana kit. The Intest OSS features in viol.48.0 and user-defined project limits in Grafana Cloud kits Dec 14. Orafana kit viol.48.0 has been released, featuring support for memory-efficient handling of large files in tests and tone of other improvements for Grafana kits OSS intervis a quick overview of the latert features in Containa kits viol.48.0, as well as some other exciting updates related to trafana Cloud kits and the kit community. Au Containa kit viol. Others the latert features in the relative activers. A kit reve subcommand to generate a test cospt.
		Easily page participants to accelerate incident response in Grafana IRM Det 13 incidents almost never happen in a vacuum. When you nooke an afert about a potential issue, odds are pretty good that you'll need to nanigate between different tools and fearms to get things resolved. Of course, trining is critical in these situations, so the easier it is to communicate – between both tools and tearms – the better off you'll be. That was the inspiration for a new capability in Grafana incident Response & Management (Grafana IRM) that ties Grafana Orical and Grafana incident closer together.

FIGURE 24 XR2LEARN DASHBOARDS ON GRAFANA

Monitoring Agents:

• Netdata.io [18] is a powerful real-time monitoring agent which collects thousands of metrics from systems, hardware, virtual machines, and applications with zero configuration. It runs permanently on the physical/virtual servers, containers, cloud deployments, and edge/IoT devices, and is perfectly safe to install on your systems mid-incident without any preparation (Figure 25).

← → C 🔺 Not secure api-db-stage-tender.maggiolicloud.it:19999/#menu_system_submenu_cpu;theme=slate;help=	=true	🚖 🕈 🗯 🙆
≡ 9219149672d8 1 2	🗊 Update Now ⑦ 🕁 🋧 昌 🗳	SIGN IN TO CLOUD
		ည် Last 6 minutes ÷
CPU The set of the let of y activity interests CPU CPU CPU CPU CPU CPU CPU CPU	Not Output 0.01 Provide A	
CPU Total CPU utilization (all cores). 100% here means there is no CPU idle time at all. You can get per core usage at: Applications Monitoring section. Keep an eye on inwait	the CPUs section and per application usage at the re a bottleneck and they slow your system down, tly high percentage of softirg may indicate network	
		 Networking Stack IPv4 Networking IPv6 Networking Network Interfaces
reación (* 1250) (* 1250) (* 1250) (* 1400) (* 1400) (* 1400) (* 1410) (* 1410) percentage ■ softw 8.3 ■ user 10.8 ■ system 2.5 ■ nice 0.8 ■ lovait 0.0		 Firewall (netfilter) systemd Services

FIGURE 25 NETDATA WEB GUI

• cAdvisor [19] provides metrics of the resource usage and performance characteristics of the running containers. It is a running daemon that collects, aggregates, processes, and exports information about running containers. Specifically, for each container it keeps resource isolation parameters, historical

resource usage, histograms of complete historical resource usage and network statistics (Figure 26).

FIGURE 26 LIST OF RUNNING CONTAINERS IN CADVISOR GUI

6 CONCLUSIONS

The document details the activities executed under tasks 2.3 "XR2Learn Marketplace and IPR handling tools" and 2.4 " XR2Learn platform integration" of WP2. It provides insights into the current status of the development, focusing on the XR2Learn platform which encompasses the marketplace featuring tools for flexible IPR management. The reported services are delivered in the production environment of XR2Learn hosted at the premises of Maggioli and are offered as the business branch of the project.

The current version of the platform will be refined and enhanced with new functionalities per component as described in the relevant sections. For example, the enhancements will include a tool for on-demand application creation and the IPR Tool will be updated to optimise the cost and energy efficiency. An updated version of the present document will be delivered in M30 of the project. The updated version will incorporate enhancements and enrichments to the described services informed by feedback gathered during the usage period of the platform in the interim, as well as new functionalities that will be enabled.

D2.2

REFERENCES

- W. Entriken, D. Shirley, J. Evans and N. Sachs, "Ethereum Improvement Proposals," 24 01 2018. [Online]. Available: https://eips.ethereum.org/EIPS/eip-721.
- [2] F. Vogelsteller and V. Buterin , "ERC-20: Token Standard," 19 11 2015. [Online]. Available: https://eips.ethereum.org/EIPS/eip-20.
- [3] P. Ossun, G. Baneling and Markian, "Non-Fungible Token Standard for Substrate's contracts pallet," 22 11 2021. [Online]. Available: https://github.com/w3f/PSPs/blob/master/PSPs/psp-34.md.
- [4] "CryptoKitties," Dapper Labs, [Online]. Available: https://www.cryptokitties.co/.
- [5] Z. Burks, J. Morgan, B. Malone and J. Seibel, "ERC-2981: NFT Royalty Standard," 15 09 2020. [Online]. Available: https://eips.ethereum.org/EIPS/eip-2981.
- [6] "fotor," [Online]. Available: https://www.fotor.com/.
- [7] "OpenSea," [Online]. Available: https://opensea.io/.
- [8] "Axie Infinity," [Online]. Available: https://axieinfinity.com/.
- [9] "Keycloak," [Online]. Available: https://www.keycloak.org/.
- [10] "Discourse," [Online]. Available: https://www.discourse.org/. [Accessed 20 12 2023].
- [11] W. Radomski, A. Cooke, P. Castonguay, J. Therien, E. Binet and R. Sandford, "ERC-1155: Multi Token Standard," 17 06 2018. [Online]. Available: https://eips.ethereum.org/EIPS/eip-1155.
- [12] "IPFS," [Online]. Available: https://ipfs.tech/. [Accessed 20 12 2023].
- [13] "Skills.move," [Online]. Available: https://www.skillsmove.eu/.
- [14] "GitHub prometheus/prometheus: The Prometheus monitoring system and time series database," [Online]. Available: https://github.com/prometheus/prometheus. [Accessed 20 12 2023].
- [15] "GitHub prometheus/pushgateway: Push acceptor for ephemeral and batch jobs," [Online]. Available: https://github.com/prometheus/pushgateway. [Accessed 20 12 2023].
- [16] "GitHub prometheus/alertmanager: Prometheus Alertmanager," [Online]. Available: https://github.com/prometheus/alertmanager. [Accessed 20 12 2023].
- [17] "GitHub grafana/grafana," [Online]. Available: https://github.com/grafana/grafana. [Accessed 20 12 2023].
- [18] "GitHub netdata/netdata: Real-time performance monitoring," [Online]. Available: https://github.com/netdata/netdata. [Accessed 20 12 2023].
- [19] "GitHub google/cadvisor," [Online]. Available: https://github.com/google/cadvisor. [Accessed 20 12 2023].