Ref. Ares(2025)11555989 - 23/12/2025

D4.5
Marketplace ver.2

Funded by
the European Union

XR4ED Project Profile

Project title	eXtended Reality for EDucation
Project acronym	XR4ED
Grant agreement No.	101093159
Start date of project	1 January 2023
Duration of project	36 months
Project website	http://www.xr4ed.eu/

Funded by the European Union

Partners

CYENS CENTRE OF EXCELLENCE	CYENS	RESEARCH	CYPRUS
konnektable TECHNOLOGIES	KONNEKTABLE	SME	IRELAND
21 Sci science 27 Co communication	SciCo	SME	GREECE
F6S	F6S	SME	IRELAND
INFINITIVITY DESIGN LABS	Infinity Design Labs	SME	FRANCE
KAINOTOMIA	KAINOTOMIA	SME	GREECE
NUROGAMES	NUROGAMES	SME	GERMANY
VASS	VASS EU Services	INDUSTRY	Luxembourg
SIMAVI	Software Imagination & Vision SRL	SME	ROMANIA
UPC	Universitat Politecnica de Catalunya	RESEARCH	SPAIN
CELLOCK	CELLOCK Ltd	SME	CYPRUS

Document information

Related work package	WP4	
Document reference	T4.5	
Related deliverables	D4.5	
Type		DEM - Demonstrator, Pilot, Prototype
		DMP - Data Management Plan
	✓	R - Document, Report
Status	Final	
Version	5	
Due date	31/12/2025	
Submission date	16/12/2025	
Lead partner	Cellock Ltd (CLK)	
Contributing partners	CYENS, F6S	
Reviewers	SIMAVI, CYENS	

Dissemination level	
✓	PU: Public
	SEN: Sensitive

List of contributors	
Name	Partner (short name)
CELLOCK LTD	CLK
CYENS - CENTRE OF EXCELLENCE	CYENS
KONNEKTABLE TECHNOLOGIES LTD.	KT
SOFTWARE IMAGINATION & VISION SRL	SMV
UNIVERSITAT POLITECNICA DE CATALUNYA	UPC

Revision History

Document History			
Version	Date	Change editors	Changes
1	10/06/2025	H. Zacharatos, N. Vlotomas	Content and format
2	15/09/2025	H. Zacharatos, N. Vlotomas	First draft
3	14/10/2025	H. Zacharatos, N. Vlotomas	Second draft
4	30/11/2025	H. Zacharatos, N. Vlotomas	Final draft
5	22/12/2025	Fotis Liarokapis	Final edits and review

Quality control		
Role	Partner (short name)	Approval date
Deliverable leader	CLK	20/12/2025
Quality manager	CYENS	22/12/2025
Project coordinator	CYENS	22/12/2025

Table of Contents

XR4ED Project Profile 2
Partners 3
Document information 4
Revision History 6
List of Tables 10
List of Acronyms 10
1 Introduction 12
1.1 Purpose of the document 12
1.2 Scope and positioning within XR4ED 12
1.3 Evolution from the first version of the Marketplace 13
1.4 Methodology and inputs 13
1.5 Structure of the document 14
2 Marketplace concept and role in the XR4ED platform 16
2.1 Marketplace objectives and value proposition in XR4ED 16
2.2 Marketplace within the XR4ED platform backbone and layers 17
2.3 XR Solutions Catalogue and Marketspace 20
2.4 Links to open calls and mini-pilots 22
2.5 Target user groups and stakeholder roles 25
3 Marketplace architecture – final implementation 28
3.1 Updated technology stack 28
3.2 Logical architecture and component view 29
3.3 Deployment architecture 35
3.4 API design and integration pattern 38
3.5 Security, performance and scalability measures 41
4 Implemented features and functionalities 46

4.1 User roles and access control model 48
4.2 Solution onboarding workflow 48
4.3 Download, installation and access flows 50
4.4 Authoring and asset-centric flows 51
4.5 API exposure for third-party applications and services 53
4.6 Administration features 54
5 XR services, tools and components in the Marketplace 56
5.1 D4.2 Marketplace baseline 56
5.2 Content and services exposed through the Marketplace 57
5.3 Key differences between Version 1 (D4.2) and Version 2 (D4.5) 61
6 User experience, operations and compliance 65
6.1 User experience and accessibility as implemented 65
6.2 DevOps, environments and operational processes 68
6.3 Compliance, privacy and content licensing in the Marketplace 71
6.3.1 Data protection and privacy 71
6.3.2 Intellectual Property Rights and content licensing 71
6.3.3 Terms of use and ethical framework 72
7 Validation, lessons and roadmap – from prototype to final Marketplace 73
7.1 Technical verification and integration status 73
7.2 Feedback from developers, pilots and internal users 75
7.3 Lessons learned and open points 77
7.4 Roadmap and sustainability snapshot 80
8 Conclusions 83
9 References 87
10 Annexes 88

List of Figures

Figure 1: Marketplace objectives and value proposition in XR4ED 15
Figure 2: Main aspects of XR4ED Marketplace platform 17
Figure 3: XR4ED platform overview highlighting the role of the Marketplace as a central access and brokerage layer 18
Figure 4: XR4ED Platform backbone (D4.5) 19
Figure 5: XR4ED platform overview highlighting the Marketplace as the main user-facing portal 20
Figure 6: XR Solutions Catalogue and Marketspace 21
Figure 7: Links to Open Calls and Pilots 24
Figure 8: Microservices and Backend Services 33
Figure 9: Microarchitecture framework for XR4ED Marketplace microservices 34
Figure 10: XR4ED three-tier architecture underpinning the Marketplace deployment 38
Figure 11: XR4ED Marketplace UX 46
Figure 12: User roles and access control model 47
Figure 13: Asset Onboarding Workflow 49
Figure 14: Key device-specific flows 52
Figure 15: Microservices - Products 58
Figure 16: Microservices – File management 60
Figure 17: Marketplace services and third-party contributions 61
Figure 18: XR4ED architecture and Marketplace milestones from design to Version 2 64
Figure 19: User experience & Accessibility implemented (WP2/WP3) 65
Figure 20: XR4ED 3-tier architecture underpinning the Marketplace deployment 70
Figure 21: Technical Verification & Integration Status 73
Figure 22: Roadmap and sustainability Snapshot 81
Figure 26: List of Products 88
Figure 23: XR4ED Platform 88
Figure 24: Check out Product 89
Figure 25: Product Details 89
Figure 28: View my Orders 90
Figure 27: Edit Product_1 90
Figure 29: Edit Product_2 91
Figure 32: Edit Product_3 92
Figure 31: User Management 92

Figure 33: List of Roles and Permissions 93
Figure 30: Logging 93

List of Tables

Table 1: XR4ED Marketplace – Version 1 (D4.2) vs Version 2 (D4.5) at a glance 63

List of Acronyms

ACRONYM	DESCRIPTION
AI	Artificial Intelligence
API	Application Programming Interface
AR	Augmented Reality
CDN	Content Delivery Network
CI/CD	Continuous Integration and Continuous Delivery/Deployment
DEVOPS	Development and Operations (combined development and operations practices)
DPIA	Data Protection Impact Assessment
DOA	Description of Action
EU	European Union
FPS	Frames Per Second
FSTP	Financial Support to Third Parties
GA	Grant Agreement
GDPR	General Data Protection Regulation
HA	High Availability
HMD	Head-Mounted Display

HTTP	Hypertext Transfer Protocol
HSTS	HTTP Strict Transport Security
JWT	JSON Web Token
K8S	Kubernetes (container orchestration system)
KPI	Key Performance Indicator
LMS	Learning Management System
MR	Mixed Reality
NGO	Non-Governmental Organisation
OER	Open Educational Resource(s)
OAUTH 2.0	Open Authorization protocol (version 2.0)
PU	Public (dissemination level)
SDK	Software Development Kit
SME	Small and Medium-sized Enterprise
STEM	Science, Technology, Engineering and Mathematics
TC	Test Case
TLS	Transport Layer Security
UI	User Interface
UX	User Experience
VR	Virtual Reality
WP	Work Package
XR	Extended Reality
XR4ED	eXtended Reality for Education (project acronym)

1 Introduction

1.1 Purpose of the document

The XR4ED project brings together the European EdTech and extended reality (XR) communities to create a sustainable, on-demand education platform and a European reference point for XR-based learning and teaching. A central pillar of this vision is the XR4ED Marketplace, which acts as a one-stop shop for XR content, tools and services for learning, training and education

This deliverable, D4.5 *Marketplace – Version 2*, builds on the first Marketplace deliverable D4.2 *Marketplace – Version 1*. Whereas D4.2 focused on specifying and implementing the initial architecture, user experience and feature set, this second version documents the evolution of the Marketplace towards its more mature state in the second half of the project. It captures the implementation decisions taken in WP4 and in particular in T4.1 *Platform backbone development*, as well as the refinements introduced in response to requirements from WP3, the open calls and the project pilots.

The document has three main purposes:

- **To present the updated conceptual role of the Marketplace inside the XR4ED platform backbone and its contribution to project objectives.**
- **To describe the final implementation of the Marketplace architecture, features and user experience as they stand towards the end of the development cycle.**
- **To provide a basis for validation, exploitation and future evolution of the Marketplace beyond the XR4ED project lifetime.**

As such, D4.5 is both a technical reference for developers and integrators and a high-level overview for project partners, reviewers and external stakeholders. It documents the Marketplace as a key XR4ED asset.

1.2 Scope and positioning within XR4ED

Within XR4ED, WP4 focuses on the development of the platform backbone and the integration of core modules, ensuring that the various XR authoring, delivery and support components operate as a cohesive whole. The Marketplace is one of the main exploitable assets identified in the project, alongside the authoring tools, XR glasses components and DevOps environment.

T4.1 is responsible for the platform backbone and the implementation of the Marketplace web application and its supporting APIs and microservices. This work is closely linked to:

- **D4.1 – Platform backbone and integration** (architecture and common services),
- **D4.2 – Marketplace – Version 1** (initial Marketplace architecture, UX and features),
- **D4.3 and D4.4** (integration and validation of XR modules), which rely on the Marketplace as an access point and discovery layer for XR4ED tools and solutions.

D4.5 therefore sits at the intersection of these deliverables. It documents the Marketplace as a concrete, deployed component of the XR4ED platform and explains how it interacts with the other backbone services, the authoring tools, the DevOps environment and the pilot activities.

1.3 Evolution from the first version of the Marketplace

The first version of the Marketplace, described in D4.2, introduced the core concept and implementation of a web-based Marketplace for XR applications, components and assets. It provided:

- A modern web architecture based on a React/Next.js frontend and a Node.js/Express backend, with a microservices-oriented design.
- A first version of the user experience, including browsing, search and basic discovery features for XR applications and components.
- Initial submission and onboarding processes for XR applications and assets, including support for VR, AR and mobile platforms.
- A deployment and hosting approach using scalable cloud infrastructure, continuous integration and deployment and basic monitoring and backup strategies.
- High-level compliance and legal considerations covering GDPR, IPR and terms of service.

Version 2 builds on this baseline and reflects:

- The consolidation and hardening of the Marketplace architecture and deployment.
- The alignment of user experience and functionality with requirements coming from pilots, open calls and ecosystem activities.
- The population of the Marketplace with XR4ED consortium tools and, where available, third-party solutions from funded projects and external contributors.
- The refinement of compliance, ethics and sustainability aspects as the project approaches its final phase.

Whereas D4.2 was largely forward-looking and blueprint-oriented, D4.5 aims to document the Marketplace as a mature and operational component, while still acknowledging planned evolutions beyond the project.

1.4 Methodology and inputs

The content of this deliverable is grounded in the collaborative work carried out across several XR4ED work packages:

- **WP3 – Requirements and architecture**, which provided the user, technology and integration requirements as well as the overall XR4ED architecture in which the Marketplace is embedded.
- **WP4 – Platform backbone and XR modules**, which implemented the Marketplace web application, APIs, deployment and operations and integrated it with other XR4ED components.

- **WP5 – Open calls and pilots**, which informed the design of onboarding, catalogue structure and usage patterns through the funded developer and pilot projects and their feedback.
- **WP2 – Ethics, human-centred design and ecosystem of trust**, which shaped the accessibility, inclusiveness and data protection principles applied in the Marketplace.
- **WP6 – Dissemination and exploitation**, which defines the role of the Marketplace in XR4ED's exploitation and sustainability strategy.

The deliverable combines:

- Technical documentation and source code artefacts from the Marketplace implementation.
- Design and UX documentation from the human-centred design activities.
- Inputs from the open calls, pilots and external stakeholders, as these become available.
- Public communication about XR4ED and its Marketplace, as published on the project website and related channels.

Public communication about XR4ED and its Marketplace, as published on the project website and related channels (see References [4] - [6]).

Where specific quantitative data, configuration snapshots or detailed feedback are required, the relevant sections explicitly indicate that they will be finalised once the corresponding inputs are provided by the responsible partners, in order to avoid speculative or non-verified content.

1.5 Structure of the document

The document is organised as follows.

- **Section 2** presents the concept of the XR4ED Marketplace and its role within the overall XR4ED platform and ecosystem. It explains the objectives and value proposition of the Marketplace, its position in the platform backbone, the distinction between the XR Solutions Catalogue and the Marketspace, its links to open calls and mini-pilots and the main stakeholder groups it targets.
- **Section 3** describes the Marketplace architecture and final implementation. It outlines the updated technology stack, the logical architecture and component view (web and admin interfaces, APIs, microservices), the deployment architecture across environments and domains and the main security, privacy and data-protection by-design mechanisms.
- **Section 4** details the implemented features and functionalities of the Marketplace in Version 2. It focuses on the user roles and access control model, the solution and asset onboarding workflow, download and access flows for XR content, search and discovery mechanisms and the key administration features.
- **Section 5** summarises the XR services, tools and components that are exposed through the Marketplace in Version 2 and positions them against the baseline defined in D4.2. It highlights how XR4ED consortium tools, reusable assets and Marketspace services are represented in the catalogue and provides a concise comparison between Version 1 and Version 2.

- **Section 6** describes the implemented UX and accessibility approach, the DevOps, environment and operational processes that support the Marketplace in production and the main compliance, privacy and content-licensing arrangements.
- **Section 7** reports on technical verification, feedback and lessons learned, and outlines the roadmap and sustainability snapshot for the Marketplace beyond the end of the project. It links the evolution from prototype to final service with the validation activities carried out in XR4ED.
- **Section 8** concludes the deliverable, summarising the contribution of the Marketplace to the XR4ED objectives and the wider European XR in education landscape.

The document ends with the list of references and annexes, which provide additional technical details, figures and examples for readers who require a deeper technical view

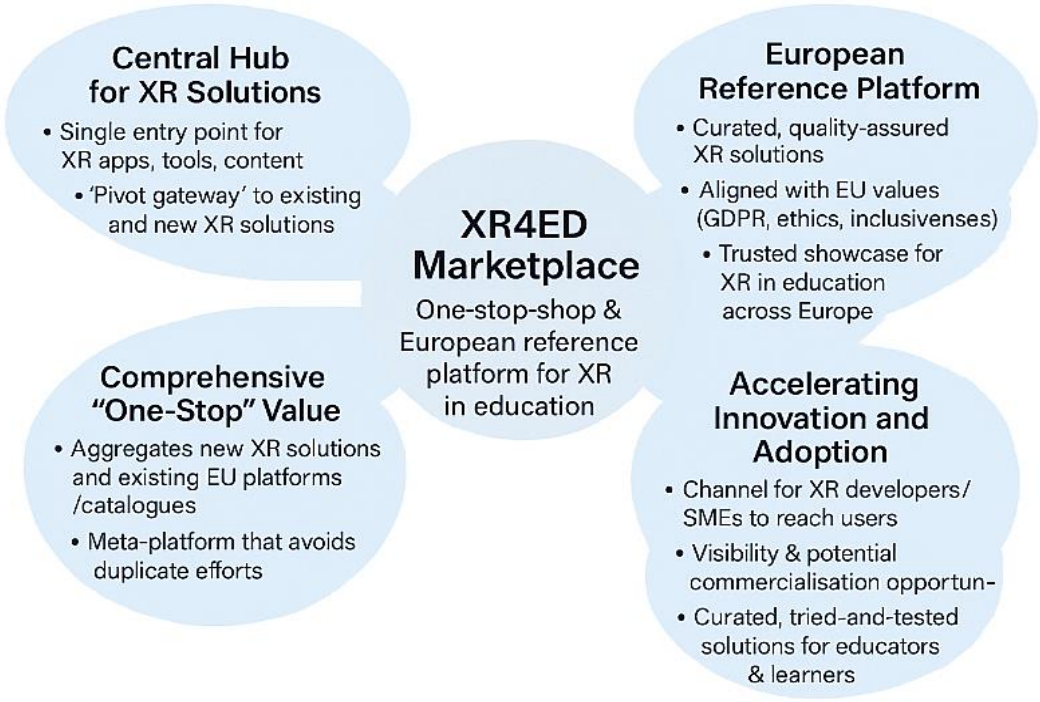


Figure 1: Marketplace objectives and value proposition in XR4ED

2 Marketplace concept and role in the XR4ED platform

The XR4ED Marketplace is conceived as the **one-stop-shop** for XR in education – a central hub where the fragmented EdTech and XR communities can come together. In the XR4ED platform, the Marketplace serves as the **European reference platform** for learning and teaching with XR, providing a **single point of access** to XR solutions, tools and resources across Europe. This section outlines the Marketplace's objectives and value proposition within XR4ED, its place in the overall platform architecture, its key components (the XR Solutions Catalogue and Marketspace), links with XR4ED open calls and pilot activities and the target user groups it engages.

2.1 Marketplace objectives and value proposition in XR4ED

The primary objective of the XR4ED Marketplace is to **unify the XR for Education ecosystem**. By acting as a one-stop-shop, it **overcomes fragmentation** in the European EdTech and XR sector and accelerates innovation in XR-based learning. In practice, this means creating a **central, open Marketplace** for XR applications in learning, training and education that becomes the go-to reference platform in Europe. The key value propositions of the Marketplace can be summarized as follows:

- **Central Hub for XR Solutions** – The Marketplace provides a **centralized platform** where educators, learners and developers can find XR content, applications and tools in one place. It serves as a **"pivot gateway"** to existing XR solutions and new offerings alike, making it easier to discover and access XR resources without searching across disparate repositories. By offering a **central access point** to XR applications and content (e.g. virtual labs, AR learning apps, 3D assets), XR4ED positions Europe at the forefront of educational technology innovation.
- **European Reference Platform** – As a reference platform on XR for education, the Marketplace curates and showcases quality XR solutions aligned with European values and standards. It is envisaged as the **European-wide showcase** of XR in education, setting benchmarks for privacy, ethics and inclusiveness in educational XR offerings. By **mobilising the EdTech and XR community** across the EU, XR4ED ensures that the Marketplace reflects Europe's diversity of innovations while upholding common values (e.g. GDPR compliance, accessibility). In this way, the platform not only aggregates solutions but also builds trust as an authoritative source for XR educational content.
- **Accelerating Innovation and Adoption** – The Marketplace is designed to **spur innovation** by giving XR developers and solution providers a ready channel to reach users and receive feedback. It lowers barriers to entry for new XR learning applications by providing visibility and potential commercialization opportunities (through app listings, subscriptions, etc.). At the same time, it helps **educators and learners adopt XR** by offering a curated selection of tried-and-tested solutions and clear usage information. By bringing early prototypes to a market-ready stage and broadcasting them to a broad audience, XR4ED helps startups and SMEs in EdTech **advance their XR innovations**

more efficiently. Ultimately, the Marketplace aims to **boost the deployment of XR in education** by connecting supply and demand in a dynamic, user-driven environment.

- **Comprehensive "One-Stop" Value** – True to the one-stop-shop concept, XR4ED's Marketplace not only lists new XR applications, but also **links to existing initiatives and repositories** to avoid reinventing the wheel. It acts as a **meta-platform**, aggregating resources from other EU platforms, catalogues and libraries alongside its own offerings. This ensures that educators and developers see XR4ED as a *reference point* – a single place to find **both new and existing XR educational solutions** and related knowledge. By providing this comprehensive coverage, the Marketplace increases efficiency (users don't need to search multiple sites) and fosters a sense of community and shared knowledge base in the XR for Education domain.

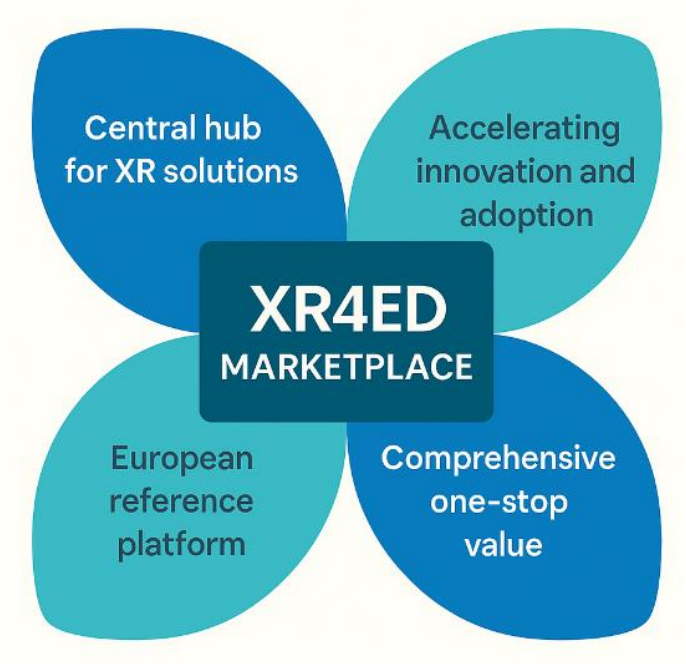


Figure 2: Main aspects of XR4ED Marketplace platform

Overall, the Marketplace's value proposition within XR4ED is to **unite community and content**: it brings together the XR and education stakeholders, curates' quality XR learning tools and simplifies access to those tools, thereby accelerating the adoption of XR in education across Europe. It is a cornerstone of the XR4ED vision of a sustainable, on-demand platform for XR education innovation.

2.2 Marketplace within the XR4ED platform backbone and layers

Within the XR4ED platform architecture, the Marketplace is a key component of the **platform's backbone**. Task T4.1 of the project focuses on the **Platform Backbone Development**, which includes building the Marketplace and integrating it with all underlying services and components of XR4ED. In essence, the Marketplace is the **user-facing layer** of the XR4ED platform that leverages a robust multi-layer backend infrastructure put in place by T4.1.

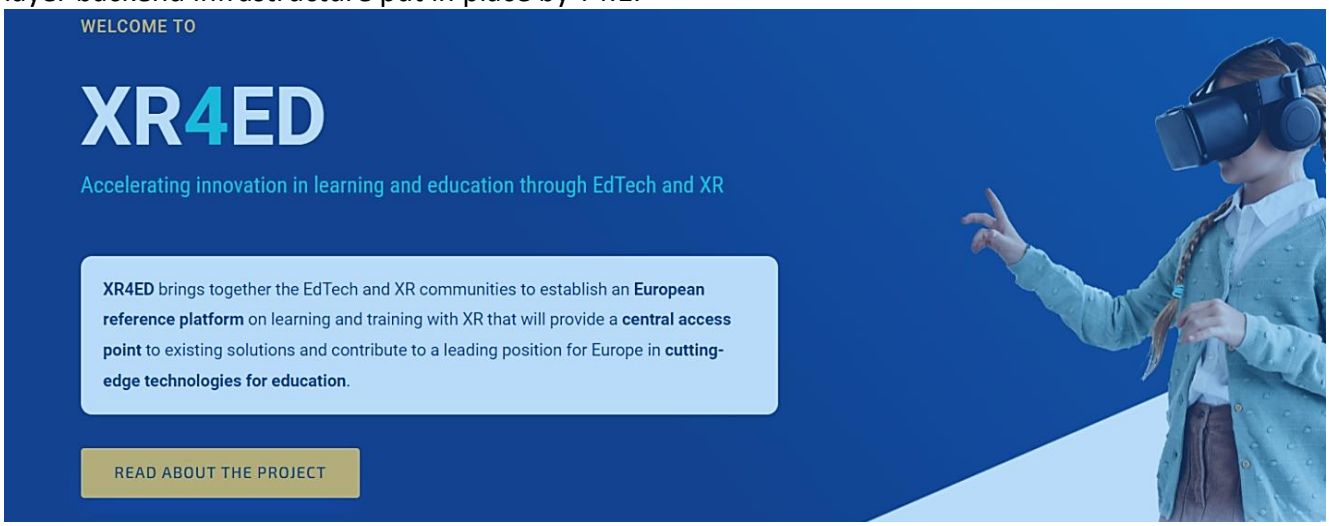


Figure 3: XR4ED platform overview highlighting the role of the Marketplace as a central access and brokerage layer

As illustrated in Figure 2, the Marketplace is positioned as the main entry point to XR4ED platform capabilities such as the DevOps environment, authoring tools and reusable XR components.

From a high-level architecture perspective, the XR4ED platform is structured in layers to ensure modularity and scalability. At the base is a **cloud-based infrastructure** providing secure, scalable hosting, databases and APIs for all platform functions. On top of this infrastructure lies the **core services layer**, which includes the XR4ED common services (user management, content management, data storage, etc.) and a suite of integrated XR tools and services. As part of T4.1, the project is implementing a **comprehensive set of flexible XR services and tools** into this backbone – such as game authoring tools, XR content creation tools, XR hardware support (e.g. HMD motion capture integrations) and a DevOps environment for XR developers. These services form the **backbone layer** that all front-end applications will use.

The Marketplace sits on top of these core services as a **distinct application layer** within the XR4ED platform. It is closely integrated with the XR4ED backend – for example, it uses the common

authentication system for user accounts, interacts with the XR content repository/database to store and retrieve solution listings and connects with the DevOps platform to enable developers to publish their outputs directly. In this layered model, the Marketplace can be seen as the **front-door** of the XR4ED ecosystem: it presents the XR solutions and services to end-users through an intuitive interface, while relying on the backbone for data and functionality. This separation of layers ensures that improvements in core services (e.g. new XR components or tools from the consortium) automatically enrich the Marketplace offerings and vice versa, any user activity in the Marketplace (e.g. a new solution submission) triggers the appropriate backend processes.

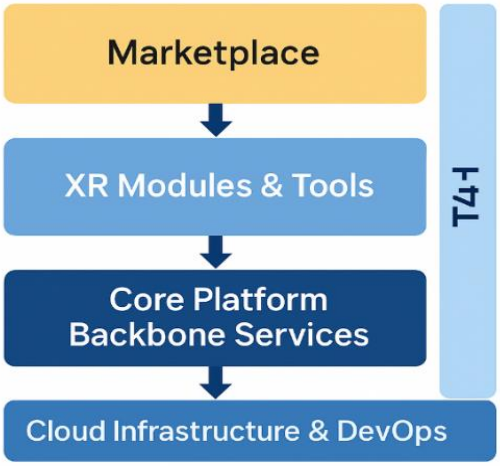


Figure 4: XR4ED Platform backbone (D4.5)

Crucially, the Marketplace is not a standalone silo but an **integral part of the XR4ED platform**. It shares the same **platform backbone** with other XR4ED components (such as the Developer Space and the Ethics Observatory). This means, for instance, that a developer using XR4ED's cloud-based DevOps environment to create an XR learning app can seamlessly move to the Marketplace layer to list that app in the Solutions Catalogue. Likewise, a teacher browsing the Marketplace to find a VR training module is indirectly interacting with the platform's underlying services (such as recommendation engines or metadata repositories implemented in the backbone). The Marketplace thus acts as the **convergence layer**: it brings together the outputs of various XR4ED platform layers and presents them in a unified experience.

From the T4.1 perspective, developing the Marketplace involved ensuring it aligns with the platform's overall architecture and layers. This entailed adopting a modern web application stack for the Marketplace (e.g. a Next.js frontend with a scalable Node/Express backend and database) that fits into XR4ED's microservice-oriented backbone. The outcome is a **modular Marketplace application** that can evolve with the platform: new backend services or XR components can be plugged in with minimal changes to the Marketplace interface.

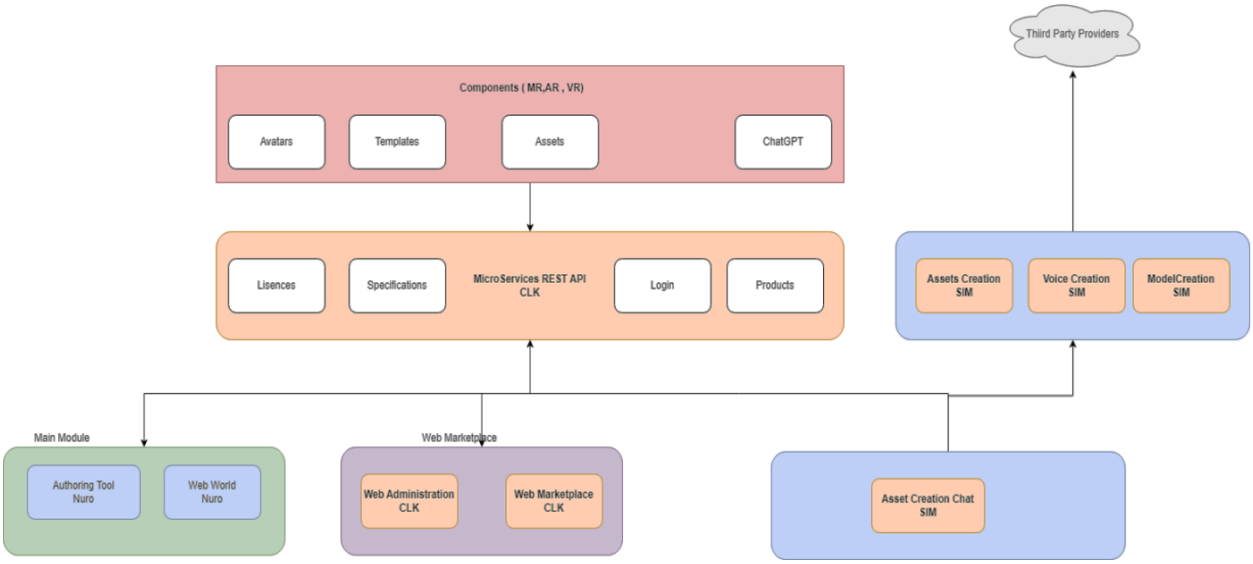


Figure 5: XR4ED platform overview highlighting the Marketplace as the main user-facing portal

In summary, the Marketplace is **embedded in the XR4ED platform's backbone** as the layer where all the platform's capabilities coalesce into a user-facing portal. It is both supported by the lower layers (infrastructure, XR services) and supporting higher-level goals (user engagement, content dissemination) of the XR4ED ecosystem.

The Marketplace is therefore not a standalone website but the user-facing portal of the XR4ED platform backbone. It is positioned above shared infrastructure and XR services, and exposes capabilities such as authoring tools, DevOps, reusable components and evaluation resources to end-users. Figure 2.X provides an overview of the XR4ED platform and highlights the role of the Marketplace within this layered architecture.

2.3 XR Solutions Catalogue and Marketspace

The XR4ED Marketplace brings together two complementary dimensions: a curated catalogue of XR solutions and an offer–demand space for XR services and collaboration.

The XR4ED Marketplace comprises two primary functional areas: the **XR Solutions Catalogue** and the **Marketspace**. Together, these components address both sides of the supply–demand equation for XR in education – the Solutions Catalogue focuses on ready-made XR products/solutions, while the

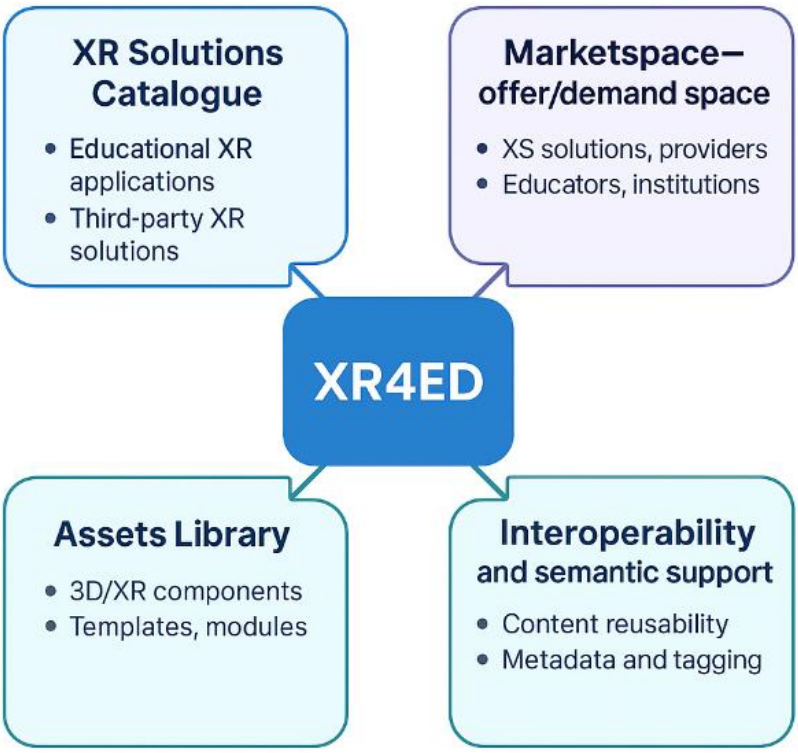


Figure 6: XR Solutions Catalogue and Marketspace

Marketspace facilitates community interaction around needs and services.

- **XR Solutions Catalogue:** This is the curated repository of XR applications, content and tools for education that are available through XR4ED. The Solutions Catalogue serves as a **showcase and database** of XR solutions, where providers can **submit and promote their ready-to-market XR solutions** (e.g. apps, platforms, modules) and users can discover and access them. Each entry in the catalogue includes metadata like the educational domain, target age group, XR technology used (VR/AR/MR), usage requirements, etc., enabling teachers or learners to easily find solutions that fit their needs. The Catalogue is open to *all European players* in the XR for Education space – startups, SMEs, research teams, etc. – who meet the platform's quality and interoperability criteria By aggregating solutions in one place, the XR Solutions Catalogue supports the **promotion of European digital educational XR solutions**, helping innovative tools gain visibility across Europe It also

emphasizes openness and standards compliance: solutions leveraging open standards like **OpenXR and WebXR** are highlighted to encourage interoperability in the ecosystem. In short, the XR Solutions Catalogue is the **Marketplace's library of XR content**, where supply-side participants publish their offerings and demand-side participants browse and acquire them.

- **Marketspace (Offer/Demand space):** Complementing the catalogue of products, the Marketspace is a dedicated **offer-and-demand forum for XR4ED-related services and needs**. This feature acknowledges that beyond ready-made apps, the community may seek collaborations, expertise, or custom solutions. The Marketspace provides a space were, for example, a teacher or school can post a *request* for a certain XR content ("Looking for a chemistry AR app in Greek language") or a developer can *offer* their services ("Offering Unity development for educational VR projects"). In essence, it's an XR4ED-specific classifieds or matchmaking area. The XR4ED project explicitly envisions this as a **facilitator of cooperation and knowledge sharing**: the platform **allows offers and requests for XR services** in education, enabling stakeholders to find partners or resources for their XR initiatives. This can include services like content adaptation, technical support, training services, or even partnerships for research pilots. By hosting this Marketspace, the Marketplace goes beyond static content listing and actively **nurtures an XR education ecosystem** – it helps connect those who *have* something (a skill, a service, an idea) with those who *need* something in the XR education realm. This offer/demand exchange is moderated and structured within the platform to ensure relevance and quality (for instance, categories for different service types, response tracking, etc.). Ultimately, the Marketspace transforms the Marketplace from a one-way catalogue into a **dynamic community market**, aligning with XR4ED's goal of fostering collaboration among the EdTech and XR stakeholders.

By integrating the XR Solutions Catalogue and the Marketspace, the XR4ED Marketplace serves both products and services, content and community. Visitors can not only find existing XR applications for immediate use, but also connect with the community to request custom solutions or offer expertise. This two-pronged approach makes the Marketplace a living, interactive environment rather than a static app store and reinforces XR4ED's role as a one-stop reference hub for XR in education.

2.4 Links to open calls and mini-pilots

A significant aspect of XR4ED's strategy to populate and validate the Marketplace is through its **open calls** and subsequent mini-pilots. The project has launched a substantial open call program (with a total budget of €4.6 million) to engage external innovators, structured into two tracks: a **Developers track** and a **Pilots track**, funding 20 projects in total (10 per track). These open calls are directly linked to the Marketplace concept, as they are a primary mechanism to **enrich the platform's content and services** and to ensure real-world testing of its offerings.

- **Open Call for Developers:** This track focuses on bringing new XR solutions and components **into** the XR4ED platform. The objective is to **enrich the XR4ED platform** through the development of innovative XR-based learning solutions that fill gaps or introduce new capabilities. Teams (start-ups, SMEs, universities, etc.) selected in the developers' open call receive funding and technical support to build their XR solutions over a 12-month period. Importantly, these solutions are **required to integrate with the XR4ED platform and be published in the XR Solutions Catalogue upon completion.** In practice, this means the winners might develop, for example, a virtual field trip application, a new authoring tool feature, or a specialized AR educational game aligned with XR4ED's thematic priorities. As they develop, they use XR4ED's tools (ensuring compatibility with OpenXR/WebXR standards) and finally **on-board their product into the Marketplace** for the wider community to access. Through this process, the open call for developers injects fresh content into the Marketplace – by project's end, at least 10 new ready-to-deploy XR solutions for education are expected to be added. This greatly expands the catalogue and demonstrates the platform's capacity to attract cutting-edge solutions.
- **Open Call for Pilots:** This track aims at **validating and demonstrating XR solutions in real educational environments**. The pilots involve consortia of educators, schools or other end-users who adopt XR solutions (including those from the developers track or existing catalogue) and implement them with learners to gather feedback. The goal is to fund projects that **deploy and customize XR4ED solutions in educational settings**, essentially serving as mini-pilots that prove the value of XR in day-to-day learning. Each pilot project runs for around a year, during which the team will integrate XR content into curricula or training activities, monitor outcomes and provide usage data and user feedback. This has a twofold benefit for the Marketplace and platform: (1) **Validation:** It shows how the XR solutions perform in practice, informing improvements to those solutions or to the platform's support features. (2) **Enrichment of knowledge:** The pilot results (best practices, implementation guides, user testimonials) can be fed back into the Marketplace as additional content – for example, as case studies or ratings/comments on the solutions used. In short, the pilot track ensures the Marketplace is **not just populated with solutions, but with proven solutions** that have been tested by teachers and learners in realistic scenarios. The XR4ED consortium tightly links these pilot projects with the platform's evolution. Feedback loops are established so that issues discovered in the field can be addressed in platform updates (WP4 development cycles) and so that

successful pilot outcomes are disseminated via the Marketplace (e.g. showcasing a "featured success story" solution that a pilot validated).

Both open call tracks reinforce the XR4ED Marketplace's growth and credibility. The developers' projects fill the Marketplace with new XR offerings aligned to six thematic educational areas prioritized by XR4ED (including virtual field trips, STEM content creation, climate change awareness, vocational training, distance learning and inclusion). The pilots' projects, in turn, ensure these offerings are **verified in diverse educational contexts**, from schools to vocational training centres, across different countries. Together, these efforts create a pipeline: **innovation → integration → validation**.

Together, these efforts create a clear pipeline:

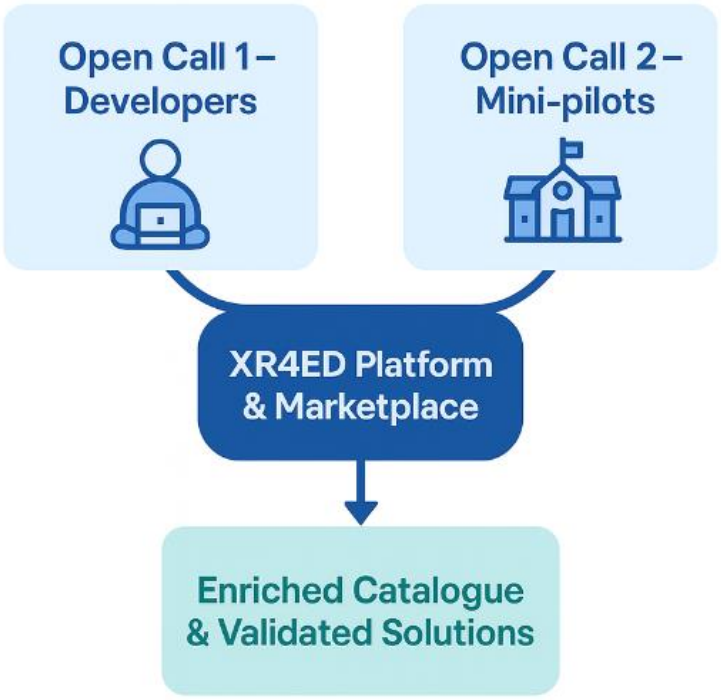


Figure 7: Links to Open Calls and Pilots

- **Innovation** – external teams design and develop XR solutions through the Developers track.
- **Integration** – funded solutions are on-boarded into the XR4ED platform and published in the XR Solutions Catalogue.
- **Validation** – Pilots deploy these solutions in real educational contexts and feedback evidence, best practices and user feedback into the Marketplace.

XR4ED uses the open calls to source innovation externally, integrates those innovations into the Marketplace (making them accessible to all) and then validates them via real use, enriching the platform with user experience insights. By the end of the project, the open call mechanism will have directly contributed a sizeable portion of the Marketplace's content (at least 20 new solutions) and a community of early adopters who have piloted XR in education, thus jump-starting the Marketplace's user base. This approach ensures that the XR4ED Marketplace is **populated, proven and poised for sustainability**, supported by a growing ecosystem of developers and practitioners brought in through the open calls.

2.5 Target user groups and stakeholder roles

The XR4ED Marketplace is designed with a **multi-stakeholder audience** in mind, reflecting the varied landscape of educational XR. Key target user groups and their roles in the Marketplace include:

- **Teachers and Educators:** As primary end-users, teachers (and instructors/trainers in vocational or higher education) use the Marketplace to find XR content that can enhance their teaching. They might search the XR Solutions Catalogue for curriculum-aligned apps (e.g. a VR anatomy lab for biology class) or browse reviews from other educators. Teachers can also play a role in the Marketspace by requesting specific XR solutions or by sharing feedback and use-cases for solutions they have tried. In XR4ED's vision, **educators are central beneficiaries** – the platform is meant to empower teachers to easily adopt XR without needing specialized development skills. Thus, the Marketplace's user experience is tailored to be educator-friendly (e.g. filters by subject or age group, guides on using XR in class). Teachers ultimately act as the **gatekeepers for adoption** of XR solutions in classrooms and the Marketplace supports them by providing trusted, curated options and a community forum for exchange of tips and resources.
- *Teachers and learners (end-user group)*
- *XR solution providers (providers)*
- *Developers and technical contributors (developers)*
- *Intermediaries and enablers (intermediaries)*
- **Learners and Students:** While students may not directly download apps from the Marketplace in all cases (often teachers will facilitate access), they are nevertheless key stakeholders as the end-users of XR learning experiences. The Marketplace's success is measured by how well the XR content engages and benefits learners. Thus, indirectly, **learners influence demand** on the platform – popular age-appropriate content or content that demonstrably improves learning outcomes will be sought after by schools. In some contexts (such as higher education or self-learning), learners themselves might browse the Marketplace (for example, a university student looking for an AR module to practice chemistry at home). The platform is inclusive of learners' needs: it includes categories like K-12, higher education, vocational training and it prioritizes accessibility features so

that content is usable by learners of different abilities. Learners also participate in pilots and can provide feedback or ratings for solutions. In summary, learners are the **ultimate end-users** of the XR solutions and their positive experiences and outcomes drive the continued use and credibility of Marketplace offerings.

- **XR Solution Providers (EdTech SMEs and Content Creators):** This group includes companies, start-ups and even research labs that have developed XR applications or content for education. For providers, the XR4ED Marketplace offers a **distribution channel and a visibility platform**. They can submit their XR products to the Solutions Catalogue, reaching a Europe-wide audience of educators and users that would be hard to assemble independently. Providers benefit from the *European reference platform* effect – being listed in XR4ED can be seen as a mark of credibility and alignment with EU standards. Their role in the Marketplace is as **suppliers** of quality content; they are expected to maintain their listings (updates, support information) and respond to user questions or needs via the platform. Many such providers are SMEs who struggle to get visibility in a fragmented market – XR4ED addresses this by **aggregating demand** in one place. Moreover, providers can also observe the Marketspace (offers/requests) to sense what new needs are emerging (e.g. if many educators request a certain type of XR content, that signals a market opportunity). Thus, XR solution providers use the Marketplace both to **showcase their solutions and to align offerings with user demand**, making them key contributors to the platform's content.
- **Developers and Tech Experts:** Beyond companies, individual developers or small teams (e.g. indie XR developers, students or researchers in XR) form another target group. XR4ED dedicates a **Developer Space** and DevOps infrastructure for them in the broader platform and the Marketplace is the venue for them to **share or monetize the XR components they create**. Developers use the Marketplace to publish not only complete apps but also reusable components or assets (such as 3D models, code libraries, plug-ins) that others in the community might use. They are often the participants in the open call Developer track, meaning many developers on the platform will have funding and support to integrate new solutions. Their role is to expand the platform's offerings technologically – pushing the envelope with novel XR interactions or tools. The Marketplace gives them a **community of practice**: they can get feedback from educators, collaborate with other developers (e.g. via the Marketspace if they seek partners) and gain recognition. In sum, developers are the **innovators and content creators** whose work populates the Marketplace and the platform in turn provides them with tools, standards (OpenXR/WebXR compliance) and an audience to facilitate development and dissemination.
- **Intermediaries and Enablers:** This category encompasses a range of stakeholders who facilitate the integration of XR in education but are neither end-user educators nor primary developers. It includes roles such as **instructional designers, teacher trainers, system integrators, educational consultants and even policymakers or standardization bodies**. These intermediaries might use the Marketplace

to stay informed about the latest XR solutions (so they can recommend them or build services around them). For instance, a teacher training organization might browse the Marketplace for examples of XR apps to include in their training curriculum, or a system integrator might look for components to include in a larger educational platform. XR4ED also identifies "social innovators" (e.g. NGOs or regulators focused on digital education ethics) as a stakeholder group – they can use the Marketplace as a reference for what's available and what best practices are emerging. Intermediaries often act as **multipliers**: a system integrator could incorporate several XR4ED solutions into a package for schools, thereby bringing more users to the platform, or a policymaker could use successful Marketplace cases to inform broader adoption strategies. The Marketplace supports them by providing **comprehensive information and networking opportunities**. Through features like reviews, ratings and the Marketspace, intermediaries can gauge solution efficacy and even connect with developers or providers. Business community members (investors, hardware providers) also fall here – for example, an XR hardware manufacturer could list their device compatibility in solution descriptions, or an investor could scout emerging XR EdTech talent via the open call winners showcased. In essence, intermediaries ensure that the XR4ED Marketplace doesn't operate in isolation but is linked with the wider educational and technological ecosystem. By engaging these stakeholders, XR4ED aims to build a **sustainable ecosystem** where Marketplace content translates into real adoption on the ground with the support of these enabling players.

All these groups – teachers, learners, providers, developers and intermediaries – interact on the XR4ED Marketplace, creating a **multi-sided platform**. XR4ED's design acknowledges their different needs and roles: for example, a teacher gets an intuitive browsing and evaluation interface, a developer gets documentation and API support and a provider gets analytics on their listing's engagement. By targeting a diverse user base, the Marketplace strives to generate network effects: more developers and providers attract more teachers (due to richer content) and more teacher uptake attracts more providers and investment and so on. This virtuous cycle is critical for XR4ED's long-term impact and is why outreach and engagement (WP5 and WP6 activities) are focusing on all these stakeholder groups. In conclusion, the XR4ED Marketplace is a platform *for* the community and *by* the community – its role is to serve the needs of each stakeholder group while also facilitating interactions between them, thereby strengthening the overall XR for Education ecosystem.

In conclusion, the XR4ED Marketplace is a platform for the community and by the community. It is designed to serve the needs of each stakeholder group while also facilitating interactions between them. By doing so, it strengthens the wider XR for Education ecosystem and supports the project's ambition to act as a sustainable European reference point beyond the project's lifetime.

3 Marketplace architecture – final implementation

The final XR4ED Marketplace architecture is built on a modern, robust technology stack and a modular design that reflects the platform's backbone role in the XR4ED ecosystem. This section describes the architecture's implementation, covering the updated technology stack, the logical component breakdown, the deployment approach, API design and integration patterns and key measures for security, performance and scalability. The design choices are fully aligned with the documented WP4 plans (Task 4.1) and the project's layered architecture, ensuring the Marketplace integrates seamlessly as the central platform component for XR content distribution and services.

3.1 Updated technology stack

Frontend: The Marketplace front-end is implemented with **React** and the Next.js framework, using **TypeScript** throughout. Next.js enables server-side rendering (SSR) and static site generation, which significantly improves page load times and search engine optimization for dynamic content. The use of TypeScript on the client-side helps catch errors early and ensures a type-safe codebase, boosting maintainability and scalability. React serves as the core UI library for building responsive, reusable components on both the end-user web interface and the administrator interface. This consistent front-end stack delivers a rich, interactive user experience across devices while adhering to accessibility and design guidelines from the project's human-centred design framework.
Backend: The back-end is built on a Node.js runtime with an Express.js web application framework. Node.js offers a non-blocking, event-driven server environment well-suited for handling many concurrent requests, as expected in a busy Marketplace. Express provides a lightweight, flexible structure for defining the Marketplace's RESTful API endpoints, routing and middleware, forming the "Main API" that powers all client interactions. The backend code is written in TypeScript as well, ensuring type consistency between front-end and server logic and reducing runtime errors. This uniform use of TypeScript across the stack simplifies the development process and improves code quality. In addition to the core Node/Express framework, the Marketplace backend integrates with a set of third-party services via API clients for specific functionalities such as payment processing, email notifications and usage analytics. These external integrations are handled in a controlled way through the backend API layer, which acts as an intermediary to manage external communication and data exchange while keeping the internal architecture secure and consistent.
Database: All persistent Marketplace data is stored and managed in a **PostgreSQL** relational database. PostgreSQL was chosen for its reliability, SQL compliance and ability to handle complex queries at scale. The database schema covers users (accounts and profiles), XR solutions and components metadata, content files or links, transaction records (if applicable for downloads or purchases) and licensing information. PostgreSQL ensures data integrity through ACID-compliant transactions and supports

advanced indexing and full-text search features, which are beneficial for implementing robust search and discovery in the Marketplace. By leveraging PostgreSQL's scalability and robustness, the platform can support a growing catalogue of XR resources and a high volume of user interactions (search queries, submissions, downloads) with efficient data retrieval.
Infrastructure and Deployment: In the final implementation, the Marketplace runs on a cloud-based, containerized infrastructure that emphasizes scalability and high availability. The application is packaged using **Docker** containers, allowing consistent runtime environments across development, testing and production deployments. For orchestration, the project uses **Kubernetes** (K8s) clusters to manage container deployment, scaling and health. This modern cloud infrastructure (provided via CELLOCK's state-of-the-art cloud services environment) enables flexible resource allocation and on-demand scaling to meet variable workloads. The architecture is designed with redundancy and load balancing in mind, meaning no single point of failure exists in production: multiple instances of critical services run in tandem and traffic is distributed so that if one instance goes down, others continue serving requests. By leveraging managed cloud services for the database, storage and networking, the platform benefits from built-in resilience and performance optimizations (such as SSD-backed storage and fast network pipes). In summary, the technology stack remains consistent with the initial design (Next.js/React, Node/Express, PostgreSQL), but it has been fully realized on a scalable cloud infrastructure with containerization and orchestration. This stack selection and deployment setup provide a strong foundation for the Marketplace's final release, balancing developer productivity with the performance and reliability needs of a pan-European XR platform.

3.2 Logical architecture and component view

The Marketplace's architecture follows a **modular, service-oriented design** in line with XR4ED's layered platform architecture. The system is conceptually divided into distinct layers and components, each responsible for specific functionalities, which work together as the backbone of the XR4ED platform (Task 4.1). The major components include: (i) the user-facing web interface, (ii) the administrator (and content provider) interface, (iii) a central API layer and (iv) a collection of back-end microservices and modules that encapsulate core services. Figure 3.1 provides a high-level illustration of these components and their interactions.

- **Web User Interface:** The web interface is the front-end that end-users (such as educators, learners, or any visitor) interact with to browse and consume XR content. Implemented as a Next.js web application, it delivers a dynamic single-page application experience with server-side rendering for performance. Through this interface, users can search for XR applications and components, explore categories and recommendations, view detailed descriptions and initiate downloads or access content. The web UI communicates with the back-end via the Marketplace's RESTful API for all data needs – for example, fetching the list of available XR learning applications, querying search results,

or loading a user's profile and content library. The interface is designed to be responsive and accessible, adhering to WCAG guidelines (as detailed in Section 5) and to support multiple device types (desktop, mobile, XR headsets via web when possible). Importantly, the web interface also serves as the entry point for XR4ED's integrated platform services: when a user wants to launch an XR experience or open an authoring tool, the web UI handles that request (often by redirecting to or embedding the appropriate service) while maintaining a unified user session and consistent look-and-feel. Overall, the user-facing front-end is the presentation layer of the Marketplace, focused on intuitive navigation and engagement, while relying on the underlying API for business logic and data.

- **Provider/Admin Interface:** In addition to the public catalogue interface, the Marketplace offers a secure interface for content providers and system administrators. This may be provided as a dedicated **Admin Dashboard** (which is built with the same React/Next.js stack for consistency) or as specific privileged views within the main web application. Authenticated XR solution providers (e.g. open call winners, project partners) can access this interface to submit new XR solutions or components, manage their listings (edit descriptions, upload new versions) and monitor usage analytics for their content. Meanwhile, platform administrators use the admin interface for curation and governance tasks: reviewing and approving submissions before they become public, categorizing content, managing user accounts and roles, configuring Marketplace settings and monitoring system metrics. The admin interface surfaces controls for these tasks in a user-friendly manner, backed by appropriate authorization checks in the API. Both the provider and admin functions benefit from React's component reusability (many UI elements are shared with the main interface) and from Next.js's ability to pre-render pages for fast loading, even in management sections. By providing separate views and controls for different user roles, the Marketplace ensures that each stakeholder (end-user, provider, admin) has an experience tailored to their needs, all within the same overall web application ecosystem.
- **Central API Layer:** At the heart of the architecture is the **Main API**, which serves as the single-entry point for all client requests and orchestrates the application's core logic. This API is implemented as a set of RESTful web services (using Express.js on Node.js) that handle operations such as searching the catalogue, user authentication, content submission, retrieval of assets and so on. The API is the intermediary between the front-end interfaces and the back-end services or database. For example, when the web interface needs to display a list of VR learning applications in a certain category, it calls an API endpoint (e.g. /api/solutions?category=VR), which in turn retrieves the data from the database and/or relevant microservice and returns it as JSON. The Main API enforces validation, business rules and access control on every request. It is designed according to REST principles with clear resource-oriented endpoints (for XR solutions, users, comments, etc.) and uses standardized HTTP response codes and structures. This makes integration and debugging easier and it also allows external systems to use the API in a consistent way (discussed further in Section 3.4). Importantly,

the API layer is where cross-cutting concerns like logging, error handling and security checks (authentication and authorization) are centrally applied. The Main API was documented comprehensively using an OpenAPI/Swagger specification in the project, ensuring that both internal developers and third-party integrators can easily understand the available endpoints and their usage. In the final implementation, this API layer is deployed as a containerized service behind a load balancer, meaning it can be scaled horizontally if needed to handle high request volumes. It represents the **integration glue** of the Marketplace – decoupling the front-end from direct database access and insulating the client from the complexity of underlying microservices.

Microservices and Backend Services: Behind the central API, the Marketplace's server-side functionality is decomposed into a set of specialized services (microservices) that handle specific domains of responsibility. This microservices architecture, introduced in the first version, has been fully realized in the final implementation to improve modularity and scalability. Each core service can operate and be deployed independently, communicating with others through well-defined internal APIs. The key backend services include:

- **Authentication Service:** Manages all user login and token issuance processes across the platform. This service verifies user credentials, generates secure tokens (e.g. JWTs) and handles session validation for subsequent API calls. It enforces role-based access, ensuring that, for example, admin-only endpoints can only be accessed by users with administrator privileges. By isolating authentication logic here, the platform can easily integrate with single sign-on or OAuth2 flows and maintain a consistent security policy across the XR4ED platform.
- **User Profile/Account Service:** (Tightly related to Authentication) Maintains user account data and profile information. It provides APIs for user registration, profile updates, retrieving user details and managing roles/permissions. This service underpins the user management needs of both the Marketplace and other XR4ED components, acting as part of the platform backbone user directory.
- **XR Asset Management Service:** Handles the storage, organization and retrieval of digital content (assets) in the Marketplace. XR assets can include application binaries, 3D models, AR templates, avatars and any components uploaded to the platform. This service interacts with cloud storage systems to save files and with the database to store metadata (such as asset names, descriptions, owner, format, tags). It also generates and manages metadata to ensure assets are properly catalogued and searchable. When a developer submits a new XR solution, the asset service processes the upload (or links to an external repository if applicable), scans for any required metadata and makes the asset available for download through secure URLs. This service is crucial for ensuring content is stored efficiently and delivered quickly to users.
- **Content Delivery Service:** Optimizes the distribution of content to end-users, especially important for larger XR applications or high-fidelity 3D assets. This service works closely with the Asset Management Service: once an asset is stored, the Content Delivery Service ensures it can reach users

with low latency and high reliability. It employs content delivery networks (CDNs) and caching strategies to serve content from edge locations closer to users, which is beneficial given XR4ED's Europe-wide user base. For instance, if a teacher in Spain is downloading a large VR lesson module, the CDN might serve it from a European edge server rather than from a single central server, reducing download times. The Content Delivery Service monitors the user's device and network conditions to possibly adjust delivery (for example, choosing an optimized format or resolution for a mobile device). This microservice ensures that the end-user experience of accessing XR content is as smooth as possible, even under network constraints.

- **Licensing and Compliance Service:** Manages license enforcement for content and tools. Many XR applications or components may have specific license terms (open-source licenses, creative commons for assets, or commercial licensing for paid content). This service tracks which users or organizations have rights to use a given piece of content and enforces any usage limitations. For example, if a particular AR training app is only available to verified educators, the licensing service will ensure only accounts with the "educator" status (or a purchased license) can download it. It integrates with the transaction records (if the Marketplace supports purchases or subscriptions) and with the user roles. The licensing service thus upholds the business rules around content usage and provides auditing capabilities (e.g. who downloaded what, license expiration, etc.), which is also important for GDPR compliance and content provider assurance.
- **Communication/Notification Service:** Facilitates platform-wide notifications and messaging. Whenever certain events occur – such as a successful submission approval, a new software update for a component, or a user receiving an invitation – this service sends out notifications accordingly. Notifications can be in-app (displayed on the web interface), via email, or other channels as configured. For instance, when an open-call developer's new XR application is approved and published on the Marketplace, the platform can send them a confirmation email and also notify users who follow the "new releases" category. This service ensures timely and reliable delivery of such messages. It also provides any real-time communication needs; for example, if the Marketplace includes a chat-based support or an "Asset Creation Chat" feature as conceptually described in D4.2, this service would manage those message exchanges, possibly using WebSocket connections for real-time feedback.
- **Integration Service:** To extend functionality, the Marketplace includes a microservice dedicated to integrating third-party systems and external APIs. This service acts as an adapter or broker for calls to external platforms – for example, connecting to a third-party *text-to-speech* engine for generating voiceovers in an XR scene, or interfacing with an educational repository API to import an existing list of XR resources. It also handles payment gateway interactions if the Marketplace processes transactions (for subscriptions or content purchases) and connects to analytics services for usage tracking. By encapsulating these external interactions in a separate service, the architecture keeps

external dependencies isolated, making the system more modular. It allows new integrations to be added or changed without disrupting the core Marketplace functionality. This "Third-Party Integration Service" ensures the XR4ED platform can leverage external innovations (such as cloud rendering services or AI-based tools) by communicating through secure APIs, effectively broadening the platform's capabilities beyond what is built in-house.

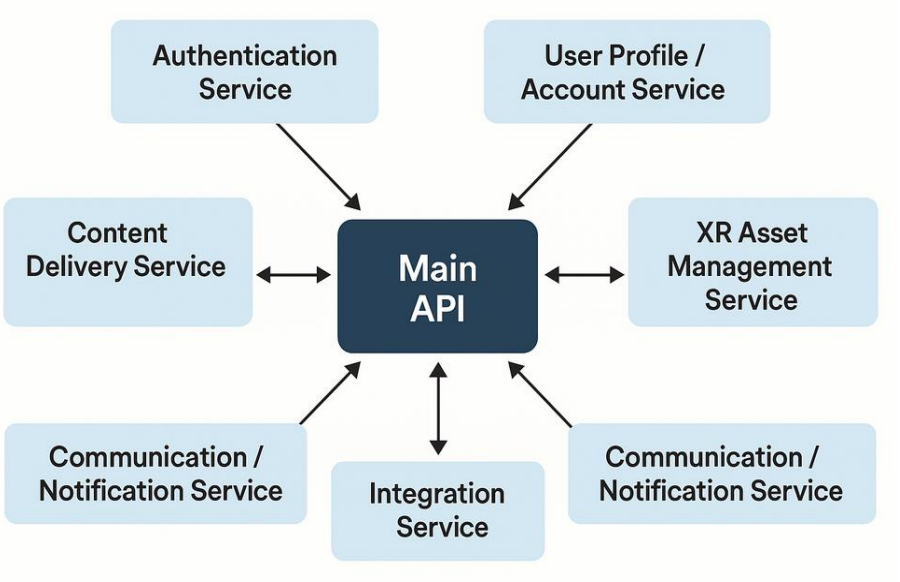


Figure 8: Microservices and Backend Services

All these microservices collectively form the **backbone of the Marketplace's logical architecture**. They interact with each other and with the central API in a carefully orchestrated manner. For example, when an administrator approves a new XR app submission: the sequence might involve the Admin UI calling an approval API endpoint; the Main API then invokes the Asset Management Service to mark the content as published, the Licensing Service to assign any default license and the Notification Service to alert the submitter that their app is now live. These services communicate predominantly via RESTful internal APIs or lightweight message queues where asynchronous processing is needed. Each service can be scaled or updated independently, which greatly enhances maintainability and the ability to introduce new features. The modular approach also aligns with the platform's layered design principle – separating the presentation layer (UIs), the application logic layer (API and services) and the data layer (database). This clear separation of concerns makes the system more understandable and robust. It also means that components developed in other work packages (e.g. the XR authoring tools, DevOps environment, etc.) can interface with the Marketplace via these well-defined service APIs, integrating into the platform

backbone without needing to be tightly coupled in code. In essence, the logical architecture of the Marketplace – comprising the Web interface, Admin interface, API and microservices – realizes the project's vision of a flexible, scalable XR platform where new modules and tools can plug in at the appropriate layer (front-end or service level). The final implementation confirms the viability of this architecture, as each component has been implemented and integrated, providing the foundation for the unified XR4ED platform experience.

The Figure 8 below illustrates the microarchitecture framework adopted in XR4ED, in which the Marketplace microservices are deployed as independent but interconnected components behind an API layer and supported by shared infrastructure services (e.g. logging, monitoring, authentication). This aligns with the logical component view presented above and underpins the Marketplace's scalability, resilience and ease of maintenance.

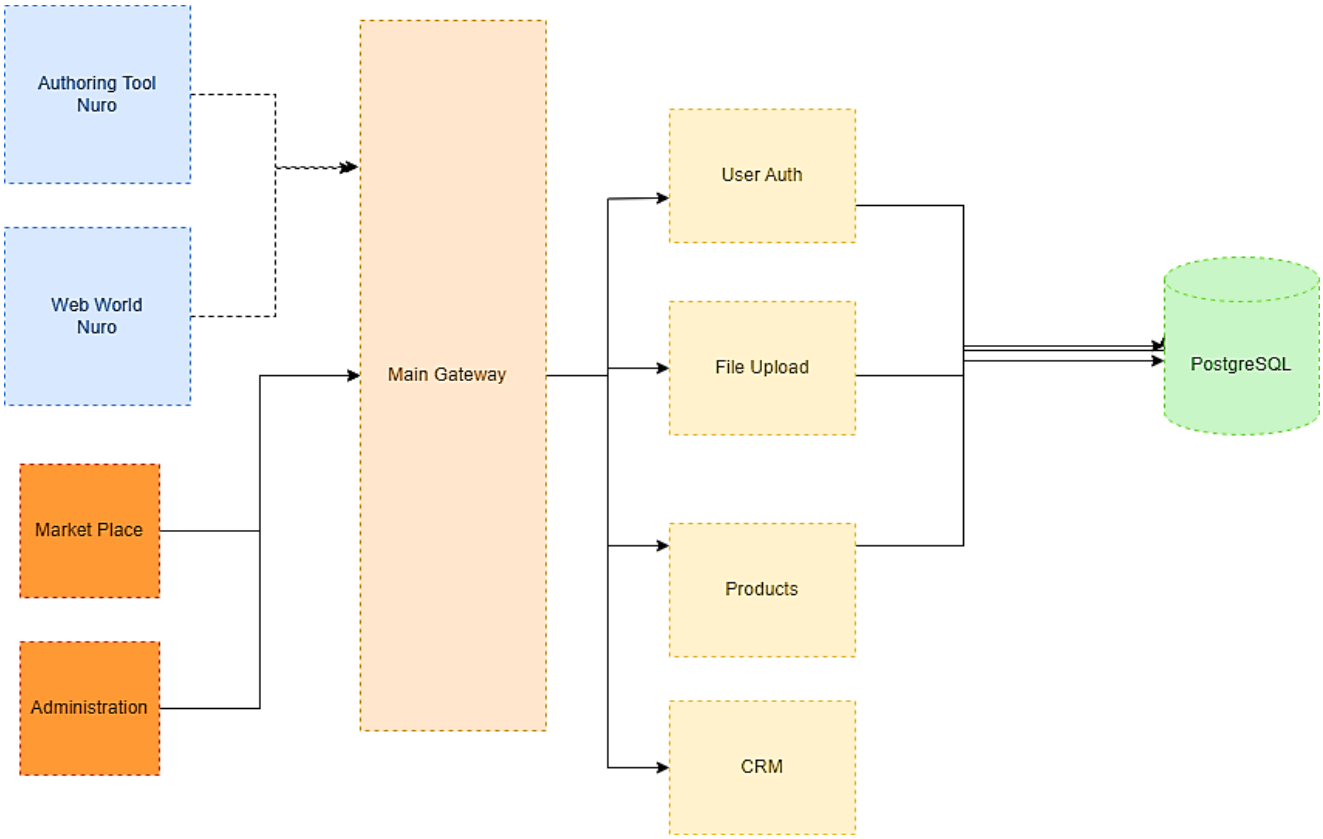


Figure 9: Microarchitecture framework for XR4ED Marketplace microservices

3.3 Deployment architecture

The deployment architecture translates the logical design into a physical setup in the cloud, with an emphasis on **multiple environments, container clusters and integration within the wider XR4ED platform**. The development team established separate deployment environments to support the software lifecycle: a development environment for internal testing, one or more staging/pre-production environments for integration testing and pilot evaluations and the production environment for the live platform accessible by end users. Each environment is configured to mirror the production setup as closely as possible, ensuring that testing is reliable and that deployments behave consistently. For example, a **sandbox environment** was provided to third-party developers (such as the open call winners) so they could test the Marketplace API without affecting real data. This sandbox uses the same API contract but operates on a separate database and storage bucket, isolating test data from the live catalogue. Maintaining these distinct environments has been key to validating new features (through WP5 mini-pilots and internal QA) before rolling them out to all users.

In the production environment, the Marketplace runs on a **containerized cluster**. All components – the Next.js front-end, the Node.js API server and each microservice – are packaged as Docker containers, which are deployed on a Kubernetes cluster spread across multiple cloud servers. Kubernetes manages the deployment, scaling and health of these containers. The cluster is configured with *multiple replicas* of critical services (e.g. the API server and any stateful microservices), along with an ingress controller and load balancers to distribute incoming requests. This arrangement underpins the platform's **high availability (HA)** strategy: even if one container instance or even a whole node in the cluster fails, another instance on a different node can continue serving requests, thereby avoiding downtime. The infrastructure explicitly avoids single points of failure – for instance, the PostgreSQL database is deployed in a high-availability mode with a primary instance and a hot standby replica and automated backups – so that a database outage will trigger a failover with minimal interruption. Similarly, the file storage (for XR assets) uses redundant storage services and the Kubernetes control plane will restart or reschedule containers if any application instance crashes. This robust setup meets the project's requirement for reliability: the Marketplace should be continuously accessible to users across Europe, even under maintenance or unexpected issues.

The Kubernetes cluster also enables **elastic scaling** to handle varying load. Each microservice deployment can scale horizontally (adding more container instances) based on demand. For example, if user traffic spikes during a pan-EU virtual classroom event (many users downloading a particular XR experience simultaneously), the Content Delivery Service and the API service can automatically scale out to add capacity. Kubernetes leverages cloud auto-scaling groups to provision additional computing resources when thresholds are exceeded and it can later scale back down to conserve resources when the load subsides. This on-demand scaling ensures consistent performance without over-provisioning hardware. It is complemented by global Content Delivery Networks at the edge (as mentioned, caching content

closer to users), which reduces the direct load on the cluster for serving large files. Together, the cluster auto-scaling and CDN integration allow the Marketplace to serve a growing user base and content library with stable response times.

- **Domain and Access Configuration:** The Marketplace is deployed under the XR4ED platform's domain structure to present a unified platform to users. In practice, the production front-end is accessible via an XR4ED subdomain (for example, Marketplace.xr4ed.eu or a similar address under the main project domain) and the API endpoints are exposed under a corresponding subdomain (e.g. api.xr4ed.eu) or path. The use of a custom domain with TLS encryption means users connect securely to the service with all communications encrypted (HTTPS enforced). Internally, the cluster uses service discovery and internal DNS for inter-service communication, but externally the API is reachable through the public domain so that external applications (and the front-end in the user's browser) can call it. Each environment typically has its own domain or namespace (for instance, a staging environment might be on a non-public domain or behind VPN for internal testing).

Integration with the overall XR4ED platform is a core consideration in the deployment architecture. The XR4ED project envisions the Marketplace not as a standalone website, but as the central hub within a broader platform that also includes the **XR DevOps environment** (developer space), various XR tools/services (like the game authoring tool, XR authoring components, etc.) and the Ethics Observatory resources. To achieve this integration, the deployment has been configured such that the Marketplace and these other components can operate under a unified authentication and user management system and share data where appropriate. Practically, this means that a single sign-on covers the Marketplace and other platform modules – a user who logs into the Marketplace (via the Auth service) is also authenticated to use the DevOps portal or other connected services, eliminating the need for multiple accounts. From a deployment perspective, some of these other components might be deployed on the same Kubernetes cluster or on federated clusters, depending on technical requirements. For instance, if the XR authoring tool is a web application, it could be containerized and deployed in the same cluster, accessible through a different path on the main domain. Alternatively, if some tools are external or provided by partners (like a VR engine or an external repository), the integration might simply be through APIs rather than co-deployment.

To coordinate these, **integration points** are set up: the Marketplace's API includes endpoints and authentication mechanisms that the other XR4ED platform components use. Task 4.3 of the project addressed this integration of components and APIs to ensure everything works together as a coherent system. For example, when an open-call	From a **network topology standpoint,** the deployment architecture isolates internal traffic from public traffic. User requests hit the cluster via the ingress controllers and public endpoints are exposed only for the front-end and main API. The microservices behind the scenes are not directly reachable from the internet; they

developer uses the XR DevOps pipeline to build an XR application, upon completion they can invoke the Marketplace API (authenticated via their platform credentials) to submit the new application to the catalogue. Conversely, the Marketplace can pull in data from other components – e.g. display a "Launch in XR" button that opens an experience in the XR Web World viewer. All these interactions are facilitated by deploying the platform components in a network configuration where they can securely talk to each other (using internal API endpoints, shared secrets or tokens and consistent user identity tokens).	communicate over the cluster's internal network. This adds a layer of security and simplifies compliance with data protection rules, as only the hardened API surface is exposed. Additionally, the deployment includes monitoring and logging components (such as Prometheus for metrics and ELK stack or similar for log aggregation) as sidecar services in the cluster, which help the operations team keep track of system health and performance in real time (elaborated in Section 7 on operations).

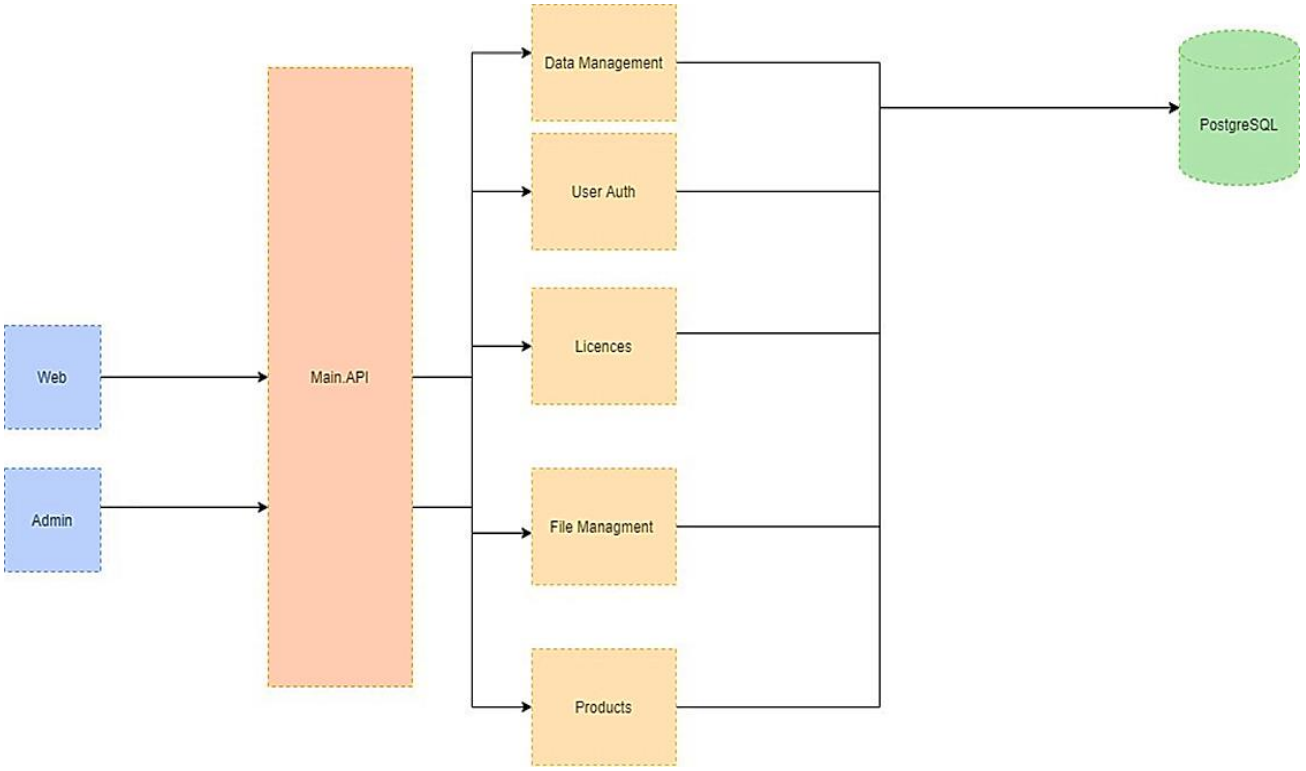


Figure 10: XR4ED three-tier architecture underpinning the Marketplace deployment

In summary, the deployment architecture uses container orchestration and cloud infrastructure to ensure the Marketplace is **robust, scalable and well-integrated** with the XR4ED platform. Multiple environments and careful domain management support the development workflow and user access needs. The use of Kubernetes clusters with auto-scaling and load balancing achieves the required scalability and high availability, while integration configurations (like shared auth and API endpoints) tie the Marketplace into the overall XR4ED ecosystem. This deployment setup has been tested during the project's pilot phases and refined to address any integration issues, resulting in a smoothly running production system that anchors the XR4ED platform backbone.

3.4 API design and integration pattern

The XR4ED Marketplace was developed with an **API-first design**, meaning that all functionality is accessible through well-defined APIs. This approach facilitates integration with other XR4ED components and allows third-party applications to interact with the Marketplace programmatically. The final

implementation's API design builds upon RESTful principles and adheres to industry standards for web services, ensuring clarity and compatibility.

RESTful API Design: The main Marketplace API is organized around RESTful endpoints that represent key resources in the system (such as /users, /solutions, /components, /categories, etc.). It uses clear HTTP methods (GET for retrieval, POST for creation, PUT/PATCH for updates, DELETE for removals) and standard HTTP response codes to indicate outcomes (200 OK, 201 Created, 401 Unauthorized, etc.). All API responses are in JSON format, making them easy to consume by web front-ends and other clients. The URLs are structured to be intuitive – for example, an external app can GET /api/solutions/123 to retrieve details of the XR solution with ID 123. The API design also includes support for query parameters for filtering and pagination (e.g. /api/solutions?category=STEM&limit=10 to get 10 items in the STEM category), enabling efficient client-side usage without over-fetching data. Throughout the project, the API endpoints and data models were documented using **Swagger (OpenAPI)**, which serves as a live reference for developers. The Swagger documentation lists each endpoint with its purpose, required parameters, request/response schemas and example calls, providing a self-descriptive interface for integration. This proved valuable not only for internal development but also for external teams (such as the open call winners) who needed to understand how to connect their applications with the XR4ED platform. By following a standard REST/OpenAPI design, the Marketplace API ensures a low learning curve and leverages a wide range of tools (automatic client code generation, testing tools, etc.) for integration.

Integration with XR4ED Components: Within the XR4ED platform, various components (the Marketplace, the DevOps environment, XR authoring tools, etc.) communicate via the API to share functionality. The Marketplace API exposes endpoints that other internal components use. For instance, the XR Authoring Tool (developed in WP3/WP4) can use the Marketplace's Asset Management API to upload and store new content. When a user in the authoring tool chooses to "publish" an XR lesson, the tool calls the Marketplace's submission API (with the user's auth token) to create a new entry in the catalogue and transfer the content package. Similarly, the Developer DevOps platform, which might manage build pipelines and version control for XR apps, could invoke Marketplace APIs to update the status or metadata of a solution (for example, marking a solution as updated after a new build). Thanks to a unified authentication scheme, these cross-component API calls are authorized seamlessly – a token issued by the Auth service is trusted across all services. The integration pattern here is predominantly **service-to-service REST integration** using the same API interface that the front-end uses, albeit with privileged credentials or service accounts where appropriate. In some cases, asynchronous integration is used: for example, if an XR application is being tested in a pilot (WP5) and generates user feedback or usage data, those might be sent to the Marketplace in batch or via a message queue to update ratings or usage statistics.

In aligning with the project's open platform goals, the Marketplace also provides ways to link with existing external XR repositories and catalogues. While not a direct real-time integration, the design allows for import/export of listings via standard formats (e.g. JSON or CSV export of the solutions catalogue, or ingestion of entries from another source). This can be facilitated by dedicated API endpoints (e.g. an admin-only endpoint to bulk import solutions from a sister project's database) or through the Integration Service calling external catalogue APIs. For instance, XR4ED might fetch a list of publicly available educational XR apps from another EU project's repository and then list them (with attribution) in the Marketplace – the Integration microservice can periodically call that external repository's API and then use internal APIs to update XR4ED's catalogue. By using APIs both for internal and external data exchange, the platform maintains a loose coupling with other systems and can evolve interfaces independently.

External Systems Integration: The Marketplace's back-end interacts with several external systems to provide ancillary functionality and these interactions follow a secure integration pattern. Key examples include: **Payment gateway integration** – if the Marketplace supports paid content or subscription models, it integrates with an external payment service (such as Stripe or PayPal) via that service's API. The backend (through the Integration Service or directly in the API layer) will create payment intents, handle callbacks/webhooks for payment confirmation and update Marketplace records (e.g. granting a license on successful payment). This integration is done server-to-server for security, keeping payment data off the client. **Email and Notification services** – for sending confirmation emails, password resets, or announcements, the Marketplace uses an external email service API (like SendGrid or similar). The Notification Service formats the messages and calls the external API to dispatch emails. This decouples email delivery from our system (we don't need to run an email server) and improves deliverability. **Analytics and Logging** – to gather usage analytics, the front-end may use an external analytics SDK (like Google Analytics or Matomo) and the back-end may send events to a monitoring service. These are integrated in compliance with privacy rules (ensuring user consent and anonymity where required). By leveraging such third-party APIs for non-core features, the Marketplace avoids reinventing the wheel and ensures robust performance in those areas. All external integrations are managed through the backend (not directly from the client), which adds a layer of control: API keys for third-party services are kept secure on the server side and calls to external systems are abstracted so they can be swapped or modified without impacting the client or other parts of the system.

3rd-Party Applications and Open API Access: A major benefit of the XR4ED Marketplace's API-centric design is that it allows third-party applications (outside the XR4ED consortium) to interface with the platform. In the final implementation, developers from ed-tech companies, research organizations, or even other EU projects can obtain API credentials and use the Marketplace API to, for example, query available XR resources or integrate XR4ED content into their own applications. The API supports both API key and OAuth 2.0 based authentication for external clients. In practice, this means an application can

register (through a developer portal or by contacting the XR4ED team) to receive an API key or client credentials. Using these, the external app can request data. The OAuth support allows for more granular, user-consented access. For instance, a third-party learning management system (LMS) could integrate with XR4ED: a teacher using that LMS could click "Import XR content from XR4ED", which would redirect them to XR4ED to authorize the LMS (using an OAuth 2.0 authorization code flow). Upon consent, the LMS would be able to call certain Marketplace APIs (like searching the catalogue or retrieving content info) on behalf of that teacher's account. This pattern ensures security (the user approves the access and can revoke it) and aligns with modern web integration practices. The Marketplace API is rate-limited to protect the service – each external API client must respect certain call quotas (e.g. number of requests per minute). These limits are documented and enforced by the API gateway to ensure that no single integration can overload the system or abuse the service. If higher usage is needed (say an institutional partner wants to regularly sync the entire catalogue), arrangements can be made to provide higher limits or a data dump, but by default the rate limiting preserves fair use and stability.

The integration pattern for third-party apps is fundamentally **open and standard-based**. The use of REST/JSON means nearly any programming environment can interact with XR4ED. The comprehensive Swagger documentation and sandbox environment mean developers can experiment and build against the API easily. We have also ensured the API embraces relevant **standards for XR content metadata** – for example, using standard schema for educational metadata where possible and supporting content formats consistent with OpenXR or WebXR so that external apps retrieving content know how to handle it. As a result, XR4ED can act as a **hub in a larger ecosystem**: sister projects or regional XR initiatives can pull data from XR4ED to populate their local catalogues (with attribution and via either API calls or periodic exports) and conversely XR4ED can include references to external content but present them through the Marketplace interface for a one-stop user experience.

In summary, the final API design of the Marketplace is **RESTful, well-documented and secure**, facilitating smooth integration both internally (among XR4ED platform modules) and externally (with third-party systems and applications). The integration patterns emphasize decoupling and use of standard protocols, which not only helped in assembling the XR4ED platform from multiple components in WP4, but also paves the way for the Marketplace's continued evolution and adoption beyond the project (allowing others to build on the platform as an open Marketplace for XR in education).

3.5 Security, performance and scalability measures

From the outset, the XR4ED Marketplace architecture was designed with strict security and high-performance requirements, given its role as a pan-European platform handling potentially sensitive user data and heavy XR content. The final implementation incorporates multiple layers of security controls, optimizations for performance and scalability features to ensure the platform is reliable and responsive under growing demand. This subsection outlines the key measures in place, focusing on authentication

and authorization, rate-limiting, high availability and related aspects like performance tuning and scalability strategies.

Authentication and Authorization: The Marketplace uses a robust authentication system based on modern token-based methods. When users log in, the system issues a secure **OAuth 2.0 bearer token** (JSON Web Token) that represents the user's session. This token must accompany subsequent API requests (in an HTTP Authorization header) and it is checked by the API gateway and relevant microservices to verify the user's identity and permissions. The platform supports both standard username/password login as well as third-party authentication mechanisms if needed (for example, institutional logins or single sign-on, which could be added via the OAuth2 framework). Authorization is enforced via a **role-based access control (RBAC)** model. Each user account is associated with one or more roles (e.g. *end-user*, *content provider*, *admin*, *pilot evaluator*, etc.) and each API endpoint and front-end feature is protected by role checks. For instance, only users with the "admin" role can access the administration UI routes or perform actions like approving content and only authenticated providers can initiate a content submission workflow. The Auth Service and the Main API coordinate on these checks: the Auth Service issues tokens containing the user's role claims and the API trusts those claims to grant or deny access accordingly. This arrangement allows fine-grained control; as requirements evolved (such as introducing a new role for FSTP project members to access specific DevOps tools), the RBAC configuration was updated without significant code changes. Moreover, all authentication-related communications are done over **secure channels (HTTPS)** with strong TLS encryption. The platform enforces HTTPS for all user interactions and even employs HSTS (HTTP Strict Transport Security) headers so that browsers will refuse any non-HTTPS connection. By doing so, we prevent eavesdropping and man-in-the-middle attacks, ensuring credentials and tokens cannot be intercepted. Sessions are typically stateless (the server trusts the token and does not store large session data), but token expiration and refresh mechanisms are in place to balance security and usability (e.g. tokens might expire after an hour, with refresh tokens or re-login required, to mitigate risk of leaked tokens). Additionally, the system implements **CORS policies** for the API: only allowed origins (such as the official front-end domain and trusted partners) can call the APIs from a browser context, which helps prevent unauthorized third-party web pages from making API calls with a logged-in user's credentials.
Protection Against Abuse (Rate-Limiting and Throttling): To maintain service quality and security, the Marketplace enforces rate limits on API usage. Every API client (whether it's the front-end application or an external integration) is subject to defined thresholds, such as a maximum number of requests per minute from a single IP or a limit on burst requests. If a client exceeds the allowed rate, the system responds with HTTP 429 Too Many Requests and temporarily throttles further requests. These measures prevent any single user or integration from overwhelming the backend, whether intentionally (as in a denial-of-service attack) or unintentionally (due to a bug causing rapid repeated

calls). The limits are tuned based on expected normal use; for example, the average teacher browsing the catalogue will never hit the limit, but a malicious script trying to scrape the entire database quickly would be stopped. Rate-limiting is implemented at the API gateway level and in the microservices for critical resources. In addition to generic rate limits, there are safeguards like **input validation and sanitization** on all endpoints to fend off injection attacks. All user-supplied data (query parameters, form inputs, JSON bodies) are validated against expected formats and cleaned of any potentially dangerous content (e.g. stripping script tags to prevent XSS, escaping SQL wildcards to prevent SQL injection). This not only secures the system but also improves performance by rejecting bad requests early. The platform also monitors for unusual usage patterns (via logs and analytics); if an API key or user account is exhibiting suspicious behaviour (e.g. thousands of requests in a short time), the ops team can revoke credentials or ban the IP as needed. Collectively, these measures ensure **fair usage** and system stability and they contribute to security by mitigating brute-force attempts (e.g. rapid password guessing or token brute forcing is made impractical due to these controls).

High Availability and Redundancy: The XR4ED Marketplace is engineered for high availability (HA), recognizing that educators and learners across different time zones may rely on it at any hour. As described in the deployment section, the production setup avoids single points of failure: multiple instances of services run concurrently and infrastructure redundancies are in place. If an API server instance crashes or a hardware node goes down, the load balancer automatically diverts traffic to other healthy instances. Automated health checks (liveness and readiness probes in Kubernetes) ensure that any unresponsive instance is taken out of rotation. For stateful components like the database, replication is used; the primary database is shadowed by replicas so that a standby can take over in case of primary failure, with minimal downtime. Similarly, the storage for assets employs redundant copies (and we perform regular backups of the database and file storage, as part of our disaster recovery plan). The system's HA is further supported by the distributed nature of the microservices: for example, even if the notification service is momentarily down, that doesn't block a user from browsing content (they might just not receive a notification immediately), which is an inherent advantage of a decoupled design. During high-load events, the auto-scaling mechanisms ensure additional service instances are spawned to maintain responsiveness, thereby maintaining availability under stress. We have also set up **disaster recovery procedures**: periodic snapshots of data, tested restore processes and runbooks for recreating the environment in a new region or provider if needed. While these are contingency measures, they bolster confidence that the Marketplace can recover from serious incidents (like data centre outages or cyber-attacks) with minimal data loss and downtime. Overall, these HA efforts aim for an uptime target in line with industry standards (e.g. >99.5% availability). The final months of the project included resilience testing – intentionally taking down instances and simulating failures – to verify that the system self-heals and continues operating as expected, which it did.

Performance Optimizations: Delivering rich XR content quickly to users is a priority. Beyond the inherently performance-conscious tech stack (SSR for fast first paint, efficient Node.js asynchronous handling, PostgreSQL indexing, etc.), the team implemented several optimizations. Static resources (images, scripts, style sheets) are served via a CDN, reducing latency for users across Europe. Dynamic API responses are cached where feasible: for instance, popular listings or frequently accessed data (like category lists) may be cached in memory or at the API gateway level for short durations, so that repeated requests don't always hit the database. We leverage browser caching hints as well; the front-end sends appropriate cache headers for content that doesn't change often. The search functionality benefits from database full-text search indices and, in future iterations, could be backed by a dedicated search engine if needed (the design considered using Elasticsearch or similar if catalogue size grew extremely large, though the current PostgreSQL-based search with proper indexing has been sufficient for the initial scale). Additionally, the use of microservices means computationally heavy tasks (such as generating a 3D preview thumbnail or compressing an asset) can be offloaded and processed asynchronously without delaying user requests. The platform also uses **lazy loading** and progressive content loading on the front-end – for example, on a solution's detail page, the heavy 3D model preview might load after the basic info is shown, so the user sees something quickly. During development, tools like Lighthouse and JMeter were used to identify bottlenecks on the client and server sides, leading to code optimizations (for example, eliminating redundant API calls, optimizing database queries with joins or using CDNs for large file distribution). As a result, the Marketplace achieves a responsive user experience: page navigation and search queries typically complete within a couple of seconds even under load and content downloads/launch are bounded mostly by user bandwidth thanks to the content delivery optimizations.

Scalability Approach: Scalability has been woven into both the software architecture (through microservices and stateless design) and the infrastructure (through container orchestration and cloud elasticity). Horizontally, the system can scale out by increasing instances of services behind the load balancer – this is how we handle increased concurrency or throughput needs. Vertically, services can be allocated more resources (memory, CPU) on the cluster if monitoring indicates they need headroom for performance (though horizontal scaling is preferred for better redundancy). The microservices architecture ensures that a surge in one type of workload only affects the relevant service. For example, a spike in asset downloads will primarily tax the Content Delivery and Asset services; these can scale up independently, without needing to scale the entire application stack. This makes scaling more efficient and cost-effective. The cloud infrastructure allows us to scale to a large number of users (well beyond the initial target of the project's pilot users) without redesigning the system – by simply allocating more nodes to the Kubernetes cluster and configuring higher auto-scaling limits. We have tested the system with simulated loads representative of, say, an entire school network accessing content simultaneously and it handled the load by automatically distributing it

across multiple containers. In terms of software scalability, the design also accounts for feature scalability – meaning new features can be added by introducing new microservices or extending the API without refactoring monolithic code. This kind of scalability (extensibility) is evident in how the platform integrated the consortium's XR tools: new components were plugged in via APIs (Task 4.1 deliverables) rather than hacking into a single codebase, demonstrating the architecture's capacity to grow in functionality.
Security Monitoring and Maintenance: Finally, recognizing that security is not a one-time setup, the platform includes ongoing security monitoring. All access to the system is logged (with appropriate anonymization and aggregation to respect privacy). We maintain detailed audit logs for sensitive actions (e.g. admin actions like content approval or user role changes), which are periodically reviewed. Automated security scanners are run on the infrastructure and code (as part of CI/CD) to catch vulnerabilities early. During runtime, intrusion detection systems monitor for unusual patterns that might indicate an attack (for instance, a sudden flood of malformed requests). The Marketplace undergoes regular updates to its dependencies to patch any security issues – thanks to the CI/CD pipeline, we can deploy updates quickly and the microservice separation means updates to one part (e.g. the authentication module) can be rolled out without affecting the entire system. In the unfortunate event of a security incident, an incident response plan is in place aligned with the project's data protection protocols (these ties into WP2 efforts on privacy and the "ecosystem of trust"). Data encryption is applied both in transit (TLS as mentioned) and at rest for sensitive data (the database encrypts certain fields like passwords – hashed and salted – and any personal identifiable information and full-disk encryption is enabled on storage volumes). These measures protect user data even if infrastructure were compromised.

In conclusion, the final Marketplace implementation meets high standards for **security, performance and scalability**. It authenticates and authorizes users rigorously, protects against abuse with rate-limits and validations and is engineered to remain available and performant through robust scaling and failover mechanisms. These qualities ensure that the XR4ED Marketplace can serve as a trustworthy and efficient backbone for XR applications in education – handling growth in users and content while safeguarding the data and experiences of its community. Together with the measures described, this creates a resilient platform prepared for both current usage and future expansion as XR4ED's user base and content offerings continue to grow.

4 Implemented features and functionalities

This section summarizes the key features and workflows now live in the XR4ED Marketplace (Version 2), as verified in the current implementation. The Marketplace offers a unified web interface where different user roles (educators, learners, developers, etc.) can register, browse and access XR solutions. All users begin by registering an account and setting up a profile (e.g. providers supply their organization details, educators supply school information). Once logged in, users see a modern, responsive web UI that supports desktops, mobile browsers, and VR/AR devices. Crucially, the interface provides powerful search and navigation tools to find XR learning content across the entire catalogue.

Figure 11: XR4ED Marketplace UX

- **Search & Filtering:** The Marketplace features a robust keyword search with advanced filters. Users can search on names, keywords or tags and narrow results by criteria (e.g. subject category, device compatibility, ratings, etc.). A powerful internal search engine and SEO-friendly metadata help users find relevant XR apps and assets quickly.
- **Categorization & Tags:** All XR solutions and assets are organized into well-defined categories and tags. The system supports a dynamic category hierarchy (e.g. “STEM”, “History”, “Arts”) and assigns relevant keywords (e.g. platform, learning topic) to each item, making discovery intuitive.

- **Personalized Recommendations:** The interface provides personalized suggestions. For example, educators see recommended products based on their profile and past activity. Product pages include detailed descriptions, usage instructions and user reviews to help decision-making.
- **Browsing & Navigation:** Users can browse by theme or collection (e.g. all "VR training apps" or all "Math learning modules"), with featured and recently added content highlighted. A shopping-cart and checkout flow allow for free or paid downloads as applicable (we support institutional payments or card transactions).
- **Multi-Platform Access:** XR content is made available in forms appropriate to each device. VR applications can be downloaded and installed directly on headsets, and AR apps can be installed via QR-

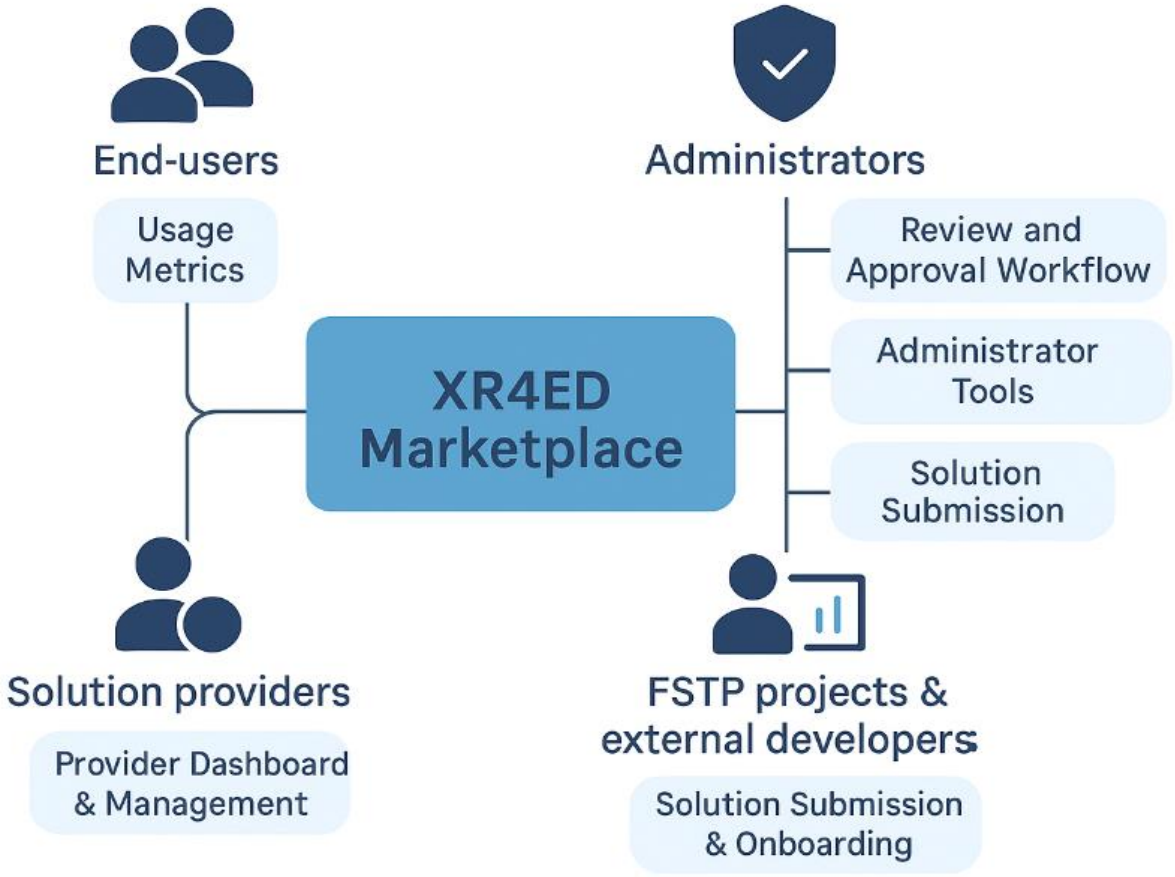


Figure 12: User roles and access control model

code links (scanned from the user's mobile device or AR glasses). All downloads and launchers integrate seamlessly with VR/AR/MR environments and mobile devices.

4.1 User roles and access control model

Beyond the public storefront, the Marketplace includes dedicated functionality for content providers (XR developers and project teams) and platform administrators. Registered **providers** access a protected submission portal (built into the same web app) where they can create and manage their XR offerings. The portal is integrated with the underlying APIs and follows the published workflow (onboarding → review → publication). **Administrators** use a separate admin dashboard to review content, manage the catalogue taxonomy, and configure user roles and settings. This ensures that all published solutions meet quality and policy standards.

- **Solution Submission & Onboarding**: Providers submit new XR applications or components through a guided form. They upload assets (e.g. app binaries, 3D models), enter metadata (title, description, supported device/platform, etc.) and specify pricing or licensing terms. The submission process captures all needed fields and invokes validation checks before sending the entry for review.
- **Review and Approval Workflow:** Every submission is routed to the administrative team for validation. Administrators review content details and can approve or request changes. The system records each submission's status and notifies providers of approval or rejection. For example, new solutions are only published publicly after explicit admin approval.
- **Provider Dashboard & Management:** Once approved, each provider sees a dashboard showing all their published solutions. From here they can edit listings (update descriptions, release new versions of an app or asset), manage downloads, and track basic usage metrics. Providers also see analytics about their content's performance (number of views/downloads, user ratings, etc.) so they can refine their offerings.
- **Administrator Tools:** Administrators use rich catalogue-management tools to curate content – for example, editing categories/tags, deactivating outdated items, and managing featured listings. They also handle user and role management (e.g. assigning "provider" or "admin" roles) and system-wide configurations (e.g. payment settings, content guidelines). Reporting dashboards give admins an overview of system health and content inventory.
- **Usage Metrics*:*** Key usage data (number of solutions, downloads, active users) is being collected under WP5/WP6 and will be documented in the corresponding impact and pilot deliverables.

4.2 Solution onboarding workflow

The Marketplace implements a guided onboarding workflow that allows providers and FSTP projects to submit solutions and assets.

Key steps in the onboarding workflow include:

- **Submission:** The provider logs into the Marketplace and fills out a submission form, entering all required metadata (educational category, XR platform, licensing, etc.) and uploading the asset files (application executables, AR/VR content packages, 3D models, etc.). The UI collects this information in structured fields to optimize discovery.
- **Automated Processing:** Upon submission, the Asset Management Service stores the files and automatically generates and normalizes metadata (e.g. ensuring tags and categories are complete). It also checks any external repository links if the content is hosted elsewhere.
- **Moderation & Validation:** An administrator receives a notification and examines the submission. The admin can review descriptions and preview the asset. The system enforces rules (e.g. verifying ownership or licenses) and may scan for malware or policy violations. Providers are required to certify they hold rights to the content. Once satisfied, the admin **approves** the submission.
- **Publication:** After approval, the main API invokes the Asset Service to mark the content as published and to apply any default licensing or access controls (for example, tying downloads to educator-only status if appropriate). A notification is sent to the provider and the listing becomes visible in the public catalogue.
- **Versioning & Updates:** Providers can later edit or update their listings. New versions (e.g. patched binaries or revised 3D models) can be uploaded via the Provider UI, triggering a lightweight review before the updated content replaces the old version.

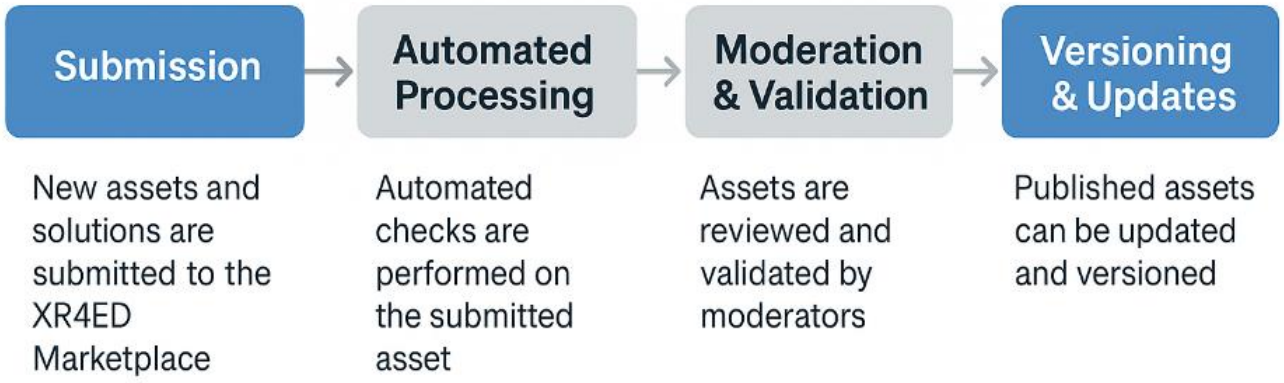


Figure 13: Asset Onboarding Workflow

The Marketplace's backend services include support functions that enhance user interaction and system responsiveness. A **Notification Service** ensures that important events (submission outcomes, new releases, account messages) generate alerts. For example, when a provider's app is approved or when a new version of a followed solution appears, users receive in-app notifications and/or email updates. The

platform frontend is built with accessibility and responsiveness in mind: it follows WCAG guidelines and is compatible with assistive technologies (keyboard navigation, alt-text for images, etc.). This means educators and learners can navigate the site effectively on various devices and under different needs.

- **Notifications and Messaging:** A real-time event system keeps users informed. In-app banners and email alerts are sent for key activities (submission approval, new content in subscribed categories, password resets, etc.). These notifications improve engagement and ensure that no important update is missed.
- **Accessibility and Inclusivity:** The UI is designed to be fully responsive (desktop, tablet, mobile) and meets common accessibility standards. For instance, the text and UI controls adhere to WCAG contrast and keyboard-navigation requirements, making the site usable by screen readers and other assistive tools. The platform also supports multiple languages/locales (English is complete; additional languages can be added). *(Details of implemented accessibility features to be confirmed by partners.)*
- **API Access and Integrations:** A comprehensive RESTful API provides programmatic access to Marketplace data and services. Third-party applications or XR tools (e.g. the consortium's authoring tools, DevOps platform) can query this API to search the catalogue, retrieve solution metadata, or manage user actions. The API is fully documented (using OpenAPI/Swagger) so that external developers can easily integrate XR4ED content into their own apps or workflows.
- **Support and Feedback Channels:** The system is ready to incorporate user feedback. Features like built-in help/contact forms, pilot questionnaires or a community forum can be plugged in. Inputs from pilot users and workshops will be collected to improve the platform.

4.3 Download, installation and access flows

The Marketplace offers comprehensive discovery tools so that educators and students can quickly find relevant XR content. Users can perform **full-text searches** over titles, descriptions, and tags, with natural-language query support for intuitive use. Advanced **filtering** lets users narrow results by multiple criteria: for example, educational domain (e.g. STEM, languages), target age group, platform (VR/AR/MR), technical requirements, or price range. Behind the scenes, every product is organized into a dynamic category hierarchy and is richly **tagged** by providers; the Marketplace leverages this taxonomy to power filters and faceted navigation. Search results are ranked using relevance to keywords, and the system supports sorting by popularity, recency, or user rating.
To boost discoverability, all listing pages are optimized for **SEO**: titles, descriptions and metadata are structured to be readable by web crawlers, so content appears in both internal search and external web

search results. In practice, each solution's page includes standardized metadata fields (educational level, learning objective, languages, etc.) that improve search rankings.
In v2 the platform also integrates **personalized recommendations**. An underlying recommendation engine (part of the XR4ED backbone) analyses user profiles, past activity, and solution metadata to suggest relevant apps to each visitor. For example, a teacher who frequently searches for "chemistry VR" might see a "Recommended for you" section featuring similar science apps. Community-driven features (like user ratings and comments) further refine suggestions. The UI highlights curated lists ("Top STEM Apps", "Latest Releases", etc.) based on usage trends.

- **Search features:** keyword query with support for synonyms and language variants; advanced filters (category, subject, age group, technology, license type, etc.) to refine results; auto-complete suggestions. All filters update results on-the-fly.
- **Categories and tags:** Content is organized into browsable categories (e.g. "Virtual Laboratories", "AR Field Trips") and tagged with attributes (e.g. "human anatomy", "interactive"). This allows teachers to browse by broad topics or click on tags from a solution page to find related items.
- **Recommendations:** A "Recommended" section leverages profile data and solution metadata. This was not present in D4.2; in v2 it uses the platform's recommendation service to suggest apps based on user interests.
- **SEO and metadata:** Product listings include structured metadata (OpenXR/WebXR compatibility, language, age range, etc.) to improve search results. All Marketplace pages are crawlable and use search-friendly URLs.

4.4 Authoring and asset-centric flows

The Marketplace supports multiple device types and ensures that once a solution is acquired, end-users can install and launch it smoothly. Across all platforms, the download process is initiated from the user's account (where purchased or granted items are listed) or via emailed links.

Key device-specific flows are:

- **VR Headsets (Oculus, SteamVR, etc.):** After purchasing a VR application, the user's dashboard provides a **device-specific download link**. For PC-based VR (e.g. SteamVR on Windows) the link may point to an installer or to the Steam store; users click it to begin download via their VR platform client. For standalone headsets (e.g. Oculus Quest), the user can download directly on the headset's browser or via a desktop app. Once downloaded, the user runs the installation (often auto-launched by the VR platform). The new VR app then appears in the headset's library or menu, ready to launch.

The system notifies users of updates (version upgrades or patches) in the same way – the Marketplace flags the update in the user's account, and they can download/install it on demand.

- **AR Devices and Mobile:** For AR experiences (and related mobile apps), the Marketplace uses **QR codes** to streamline installs. Upon purchase, the user is presented with a unique QR code (also sent by email or available on their dashboard) that encodes an installation URL. Scanning this QR with a smartphone or tablet redirects the user to the appropriate app store (Google Play or Apple App Store) or to a direct download page. The user then installs the app as usual; permissions (camera, sensors, etc.) are requested during installation. Once installed, the AR app runs on the device (for smart glasses, it may sync via a companion phone app). Updates to AR apps are also handled via the app store, with notifications delivered through the Marketplace.

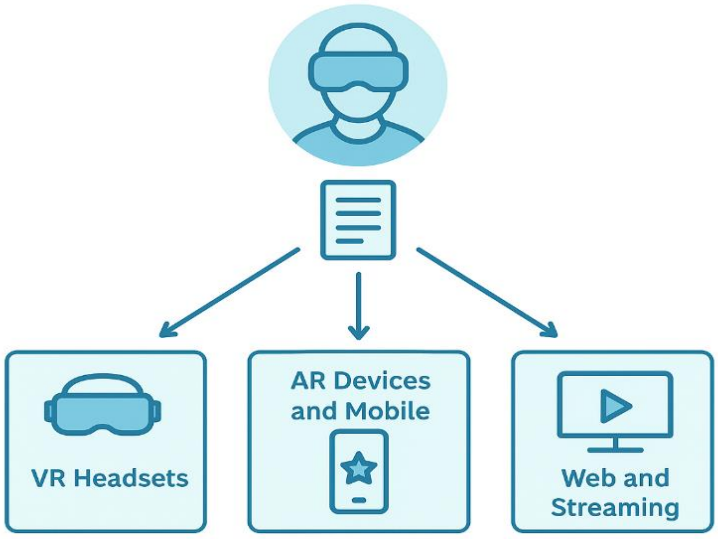


Figure 14: Key device-specific flows

- **Mobile (Android/iOS) Apps:** Similarly, mobile XR-related apps or companion apps can be downloaded via app-store links. After purchase the dashboard provides a link to Google Play/App Store. Clicking it opens the store page where the user taps to install. For advanced use cases (beta testing or offline install), users can also **manually download APK/IPA** files directly from the Marketplace and install them (after enabling "unknown sources"). Updates are delivered through the standard app store channels or via new file downloads on the dashboard.
- **Web and Streaming:** Some lightweight XR content (e.g. WebXR experiences) can be launched directly in a browser. The Marketplace provides "Launch in WebXR" buttons when supported. The web UI ensures the content is loaded in the user's profile with appropriate session credentials.

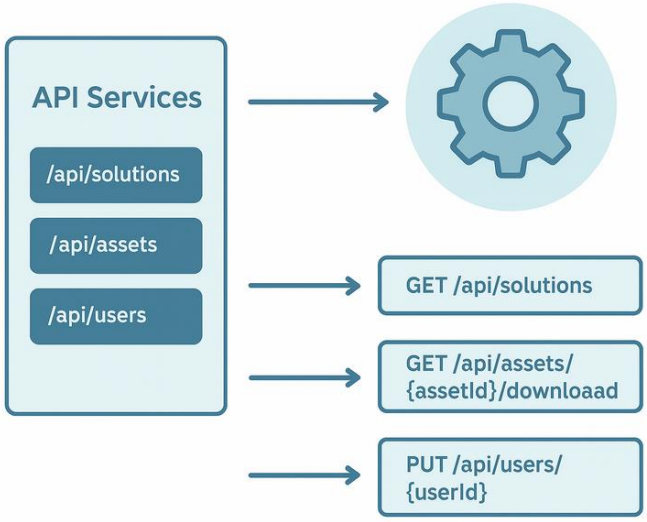
Throughout these flows, **access control** is enforced by the Licensing and Compliance service: it checks user roles and entitlements before allowing a download. For example, if an app is restricted to educators only, only accounts marked as teachers (or with a valid purchased license) can retrieve the filefile://file_00000000c52871f4b21d1f453407e2c7/. All downloads are served over secure URLs, and large assets are delivered via a Content Delivery Network for low latency. Supported file formats include VR executables (e.g. Unity package, APK, Windows installer), AR app bundles, and asset formats like glTF/FBX for 3D models; the system validates these formats on upload.

4.5 API exposure for third-party applications and services

A dedicated Administration interface (or privileged views in the main app) gives platform administrators full control over the Marketplace. Using this interface, admins can **moderate content and users, curate the catalogue, and monitor system usage**. Key admin functions include:

- **Submission Review:** Administrators see a queue of newly submitted solutions or assets. They can inspect each listing (preview uploaded files, read descriptions) and either approve or reject it. Approved content is published to end-users; rejected content is sent back for revision. This manual review step enforces quality standards.
- **Content Management:** Admins can edit any listing's metadata (title, description, tags, category), reassign content to different categories, or remove outdated items. They can also *feature* certain solutions (making them appear in highlighted sections on the homepage). Bulk import tools allow populating the catalogue from external sources if needed.
- **User and Roles Management:** Through the admin dashboard, administrators can add or suspend user accounts, assign roles (e.g. teacher, provider, super-admin), and manage permissions. For example, they can promote a new content provider account so it can submit listings. Role assignments are enforced in the API layer.
- **Reporting and Analytics:** The admin interface includes dashboards showing usage metrics and content performance. For instance, admins and providers can view how many times a solution was downloaded or viewed, track user ratings/comments, and see trends over time. These analytics help gauge demand and identify popular or problematic content. System metrics (server load, API call volumes, etc.) are also accessible to IT staff.
- **Moderation Tools:** Admins can respond to user flags or complaints (e.g. if an educator reports an issue with content). There are also workflows for takedowns or IP disputes: although D4.2 envisioned formal procedures, v2 has implemented basic tools for removing content and logging takedown requests.

- **Configuration:** The admin can adjust system-wide settings such as default license templates, featured category ordering, or even theme colors. Critical platform configurations (e.g. SMTP server for notifications, payment gateway keys) are managed here under strict admin-only access.



4.6 Administration features

The XR4ED Marketplace exposes a comprehensive RESTful API that enables integration with third-party applications, the XR4ED platform components, and external systems. The **central API layer** provides endpoints for all core operations – search, browsing, content submission, user management, and more – and is documented via OpenAPI/Swaggerfile://file_00000000c52871f4b21d1f453407e2c7/. External developers (e.g. open-call teams) and other XR4ED modules use these APIs, authenticating with OAuth2/JWT tokens issued by the Auth Service.

Key API services include:

- **Solution Catalogue endpoints:** e.g. GET /api/solutions (with query parameters for filtering by category, technology, etc.) returns a JSON list of matching solutions. Individual solution details can be fetched by ID (GET /api/solutions/{id}). Providers can POST or PUT to /api/solutions to create or update listings (this triggers the same onboarding workflow described above).
- **Asset download endpoints:** e.g. GET /api/assets/{assetId}/download provides a secure URL or stream for downloading the asset file. These endpoints check the user's permissions (via the Licensing Service) before returning content.

- **User and Auth endpoints:** e.g. POST /api/auth/login, GET /api/users/{userId}, PUT /api/users/{userId} for retrieving/updating user profiles. Role-based access is enforced on each call by the API middleware.
- **Search endpoint:** GET /api/search?query=...&filter=... wraps the full-text search and filter logic, returning paginated results.
- **Metadata endpoints:** e.g. /api/categories, /api/tags, to populate search filters dynamically.
- **Developer integrations:** The API allows external systems to trigger Marketplace actions. For example, as pilots finished in the DevOps environment, they could **invoke the Marketplace API** (using their XR4ED credentials) to submit a completed XR app to the catalogue. Conversely, the Marketplace can call other services' APIs (e.g. instruct an XR Web viewer to launch a scene).
- All endpoints return JSON and use standard HTTP methods and status codes for simplicity. The API is *versioned* and served under a dedicated subdomain (e.g. api.xr4ed.eu) with HTTPS. A complete Swagger specification (generated during development) is published so that integrators can see request/response schemas and try out calls. For example, one might retrieve STEM apps by calling: GET /api/solutions?category=STEM&limit=10 (as shown in developer docs).

5 XR services, tools and components in the Marketplace

This section summarises the content and services that the XR4ED Marketplace exposes in its second version and positions them against the baseline defined in D4.2 Marketplace – Version 1. It focuses on the types of XR solutions, tools and assets that are accessed through the Marketplace, how they relate to the platform capabilities defined in the XR4ED proposal and how the offer–demand space for XR in education has been consolidated in Version 2.

5.1 D4.2 Marketplace baseline

D4.2 Marketplace – Version 1 provided the initial blueprint for the XR4ED Marketplace as the "offer and demand space" for XR4ED-related services and solutions. In line with the proposal, it defined the Marketplace as an online environment where European XR providers could submit and promote their ready-to-market XR solutions for learning, training and education, and where users could discover and access them through a user-friendly interface.

At content and service level, the D4.2 baseline can be summarised along four main dimensions:

- **XR Solutions Catalogue and Marketspace.** D4.2 formalised the idea of a European XR4ED solutions catalogue, open to external players, where XR applications and services can be listed with rich metadata (educational domain, target group, XR technology, etc.). In parallel, it specified a Marketspace where providers and users could publish offers and requests for XR4ED-related services, including integration, customisation and support.
- **Platform capabilities surfaced through the Marketplace.** The document linked the Marketplace to a number of core XR4ED capabilities that would eventually be exposed as services or content entries, namely:
  - the **DevOps environment** for XR developers,
  - XR **software development kits (SDKs)** and build tools (e.g. game engines),
  - an **Assets library** aggregating XR assets from EU and national projects and open repositories,
  - an **Ethics observatory** for privacy, ethics and inclusiveness resources,
  - links to XR-related **repositories, catalogues and libraries**.

- **Core Marketplace features and user experience.** D4.2 described the first version of the browsing, search and discovery experience for XR applications and components, the initial submission and onboarding workflows for XR content, and the basic mechanisms for metadata management, categorisation and access.
- **Technical and operational services behind the scenes.** D4.2 outlined the technology stack and hosting approach for the Marketplace (modern web UI, microservice-oriented backend, scalable cloud deployment), as well as high-level compliance and legal considerations (GDPR, IPR, terms of service), positioning the Marketplace as a robust but still evolving platform component.

Overall, D4.2 was largely **forward-looking**: it provided a comprehensive architectural and functional blueprint, defined the main service categories to be supported by the Marketplace and described how XR4ED platform capabilities (DevOps, assets library, authoring tools, observatory) would eventually surface through this "one-stop-shop" for XR in education.

5.2 Content and services exposed through the Marketplace

Version 2 of the Marketplace, as documented in this deliverable, reflects the transition from a blueprint-oriented Marketplace to a **populated and operational** component of the XR4ED platform backbone. It consolidates the initial content and service categories from D4.2 and aligns them with the XR4ED platform architecture, microservice framework and the project's open calls and pilots.

From a content and services perspective, the Marketplace Version 2 focuses on three main groups:

a) XR solutions, tools and modules from the XR4ED consortium

The Marketplace now acts as the primary access point to XR4ED's own technological assets that were planned in the proposal and implemented through WP3 and WP4. These include, in particular:

- **XR Game Authoring Tool and XR Authoring Tool**

  Consortium-developed authoring tools for building XR learning experiences are surfaced through Marketplace entries that describe their functionality, target users (e.g. teachers, instructional designers, developers), supported XR hardware and integration prerequisites. Through the Marketplace, these tools can be:

  - discovered by educators looking for authoring environments,
  - accessed by developers seeking to build on top of existing XR4ED modules,

- linked with open-call projects that reuse or extend them.

❑ **XR4ED DevOps platform**

The XR4ED DevOps platform, conceived as a web-based environment integrating multiple third-party tools for development and collaboration, is exposed as a service entry. The Marketplace provides:

- a **service description** of the DevOps environment and its scope,
- links or workflows for developers (including funded third parties) to onboard their projects,
- alignment with the **Products** and **File management** microservices that support product builds, content uploads and metadata management.

❑ **XR/AR glasses UI/UX components and XR avatars**

Reusable UI/UX components for XR/AR glasses and XR avatar technologies, developed within XR4ED, are published as reusable components or SDK-like modules. The Marketplace presents:

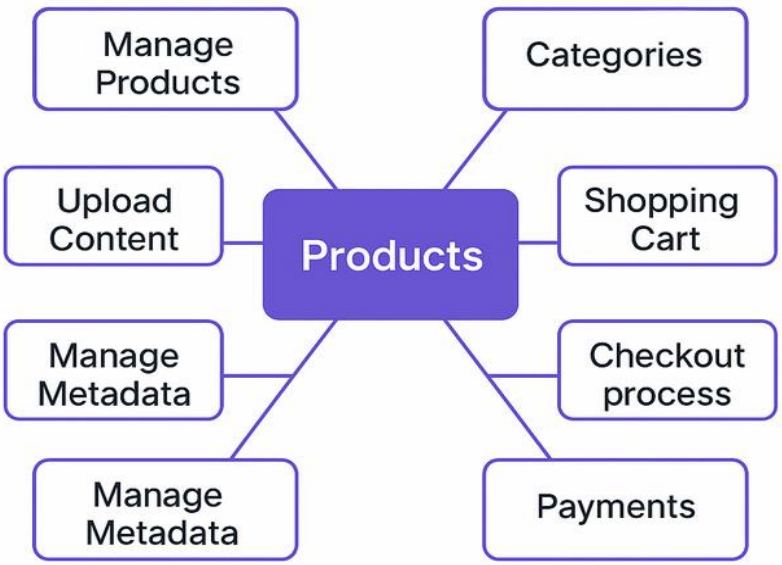


Figure 15: Microservices - Products

- concise summaries of what each component provides (e.g. multi-device compatible UI patterns, collaborative avatar capabilities),
- supported platforms and integration hints,
- references to documentation and licensing conditions.

Together, these entries ensure that XR4ED's own building blocks are visible, citable and re-usable by both internal and external stakeholders via a single catalogue.

b) Assets library and reusable XR resources

In line with the proposal's ambition to create an **Assets library** that aggregates XR resources from EU and national projects, Version 2 of the Marketplace integrates content types that go beyond complete XR applications:

- **3D and graphical assets**, including models and visual elements relevant to XR learning experiences.
- **XR learning resources and open educational resources (OERs)**, sourced from projects and repositories that allow re-use under appropriate licences (e.g. Creative Commons).
- **XR accessibility resources** and guidelines that support inclusive XR design for education.

These resources are managed and delivered through the **File management** microservice layer, which handles upload, type detection, storage and download of content, and are exposed in the catalogue with appropriate metadata and licensing information.

c) Marketspace services and third-party contributions

The **Marketspace** dimension of the Marketplace – the offer/demand space for XR4ED-related services – is aligned with the project's cascade funding scheme and the two waves of open calls (developers and mini-pilots).

In Version 2, this translates into:

- **Service-oriented entries** where XR4ED partners and third parties can describe services such as XR content development, integration support, training or customisation.

- **Listing and showcasing of solutions** produced by funded developer projects, which are required to use open standards (e.g. WebXR, OpenXR) and to publish their outcomes in the European XR4ED Solutions Catalogue.
- The possibility to link **mini-pilot deployments** and lessons learned to the corresponding Marketplace entries, illustrating how specific XR solutions have been tested in real educational contexts.

Figure 16: Microservices – File management

As a result, the Marketplace in Version 2 is not only a catalogue of artefacts, but also a **brokerage layer** where offerings from XR4ED partners and funded projects can be matched with concrete educational needs.

d) Supporting microservices and operational services

Although this section focuses on content and services, it is important to note that Version 2 leverages the microservice framework designed under WP3 and implemented in WP4. In particular:

- **User Authentication and Licences** microservices handle authenticated access, licence types, licence keys and expiry, ensuring that protected content and services are accessed under appropriate conditions.

- The **Products** microservice manages product creation, metadata, categorisation, shopping-cart and checkout processes where relevant, even when monetisation is not yet fully activated.
- The **File management** microservice underpins the upload, storage and download of XR content and assets, enabling the Assets library to function at scale.

These microservices make the Marketplace content **operational and governable**, providing a clear separation between user-facing catalogue entries and the backend services that enforce policies, security and interoperability.

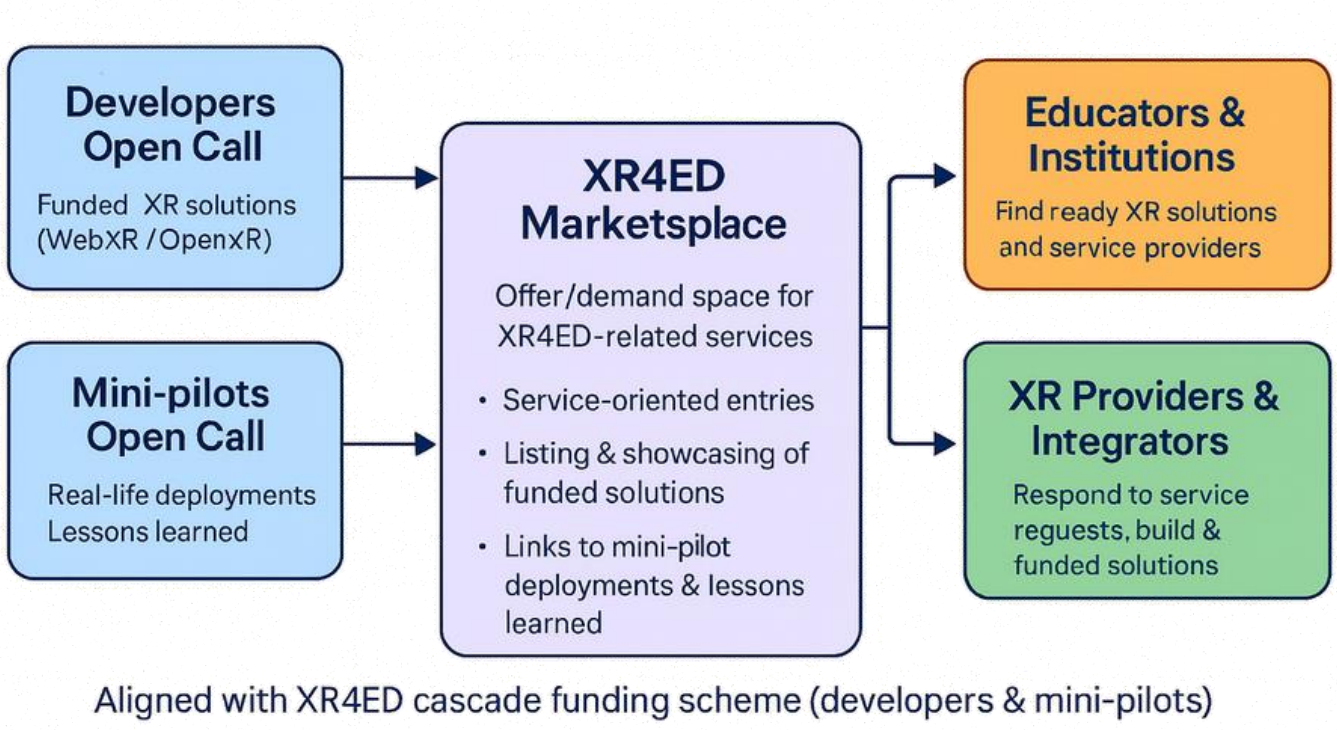


Figure 17: Marketplace services and third-party contributions

5.3 Key differences between Version 1 (D4.2) and Version 2 (D4.5)

From a content and services perspective, the evolution from D4.2 to D4.5 can be summarised as a shift from **conceptual design and initial implementation** to a **consolidated, populated and operational** Marketplace.

At a high level, the key differences are:

- **From blueprint to deployed catalogue and marketspace**
  - *Version 1 (D4.2):* Defined the XR Solutions Catalogue and Marketspace conceptually, with initial UX flows, metadata structures and submission processes, but with limited concrete population.
  - *Version 2 (D4.5):* Documents a running catalogue that exposes XR4ED consortium tools (authoring tools, DevOps, avatars, XR/AR UI components), assets and resources, as well as entries linked to open-call projects and mini-pilots where available.
- **From planned platform capabilities to surfaced Marketplace entries**
  - *Version 1:* Identified platform capabilities such as the DevOps environment, assets library, ethics observatory and external repositories as future elements to be linked to the Marketplace.
  - *Version 2:* Describes how these capabilities are actually surfaced, for example through Marketplace entries for the DevOps platform, reusable XR components and curated asset collections, supported by dedicated microservices for products, files and licences.
- **From generic open-call linkage to concrete FSTP integration paths**
  - *Version 1:* Stated that third-party projects funded via open calls would have to publish their solutions in the XR4ED Solutions Catalogue.
  - *Version 2:* Aligns the Marketspace and catalogue with the open calls and mini-pilots, providing a structured way to host, describe and showcase outcomes from funded developer projects and to connect them to their pilot deployments and thematic pillars.
- **From high-level compliance to operational governance**
  - *Version 1:* Outlined high-level GDPR, IPR and terms-of-service considerations.
  - *Version 2:* Places these aspects in the context of an operational Marketplace, supported by authentication, licensing and file management services and by clearer roles for content owners, providers and users, as further detailed in Sections 4 and 6.

In summary, **Version 1** of the Marketplace defined *what* XR4ED wanted to offer through a one-stop-shop for XR in education, while **Version 2** shows *how* this vision is instantiated in a real, populated platform component that combines consortium technologies, aggregated XR resources and third-party outcomes under a coherent catalogue and marketspace.

Table 1: XR4ED Marketplace – Version 1 (D4.2) vs Version 2 (D4.5) at a glance

Aspect: Overall role **Version 1 – D4.2 (Marketplace v1):** Conceptual blueprint and first implementation of the XR4ED Marketplace (one-stop-shop, marketspace, catalogue). **Version 2 – D4.5 (Marketplace v2 – final):** Consolidated, operational component of the XR4ED backbone, integrated with platform services and ecosystem activities.	**Aspect:** Architecture & stack **Version 1 – D4.2 (Marketplace v1):** Target architecture, technology stack and microservices design specified; initial web UI and backend in place. **Version 2 – D4.5 (Marketplace v2 – final):** Architecture implemented and deployed on the XR4ED infrastructure (frontend, APIs, microservices, storage).	**Aspect:** Content & services exposed **Version 1 – D4.2 (Marketplace v1):** Planned XR content types, tools and services (solutions catalogue, assets library, DevOps, observatory, etc.). **Version 2 – D4.5 (Marketplace v2 – final):** Marketplace populated with XR4ED tools/modules, assets and service entries; concrete examples visible in the catalogue and Marketspace.
Aspect: Integration of XR4ED capabilities **Version 1 – D4.2 (Marketplace v1):** Platform capabilities (DevOps, authoring tools, assets, observatory) identified as future integration targets. **Version 2 – D4.5 (Marketplace v2 – final):** Capabilities surfaced as actual Marketplace entries (e.g. authoring tools, DevOps access, reusable components).	**Aspect:** Third-party / FSTP contributions **Version 1 – D4.2 (Marketplace v1):** Open calls and third-party solutions foreseen to feed the catalogue; integration path described conceptually. **Version 2 – D4.5 (Marketplace v2 – final):** Marketspace aligned with cascade funding scheme; clear paths for funded solutions and pilots to be listed, showcased and linked to deployments.	**Aspect:** Governance, compliance & access control **Version 1 – D4.2 (Marketplace v1):** High-level GDPR, IPR and terms-of-use principles defined; roles and responsibilities outlined conceptually. **Version 2 – D4.5 (Marketplace v2 – final):** Compliance, licensing and access control embedded in operational services (auth, licenses, file management) and user role

		model (end-users, providers, admins, FSTP).
Aspect: Maturity level **Version 1 – D4.2 (Marketplace v1):** Early implementation and design baseline; focus on vision, requirements and initial features. **Version 2 – D4.5 (Marketplace v2 – final):** "As-built" view of the Marketplace at project maturity, ready to support exploitation, sustainability and reuse.		

This timeline, originally defined in WP3, shows how the final platform architecture design (M12), the FSTP project designs and the two Marketplace milestones (D4.2 and D4.5) fit into a coherent roadmap from blueprint to operational platform.

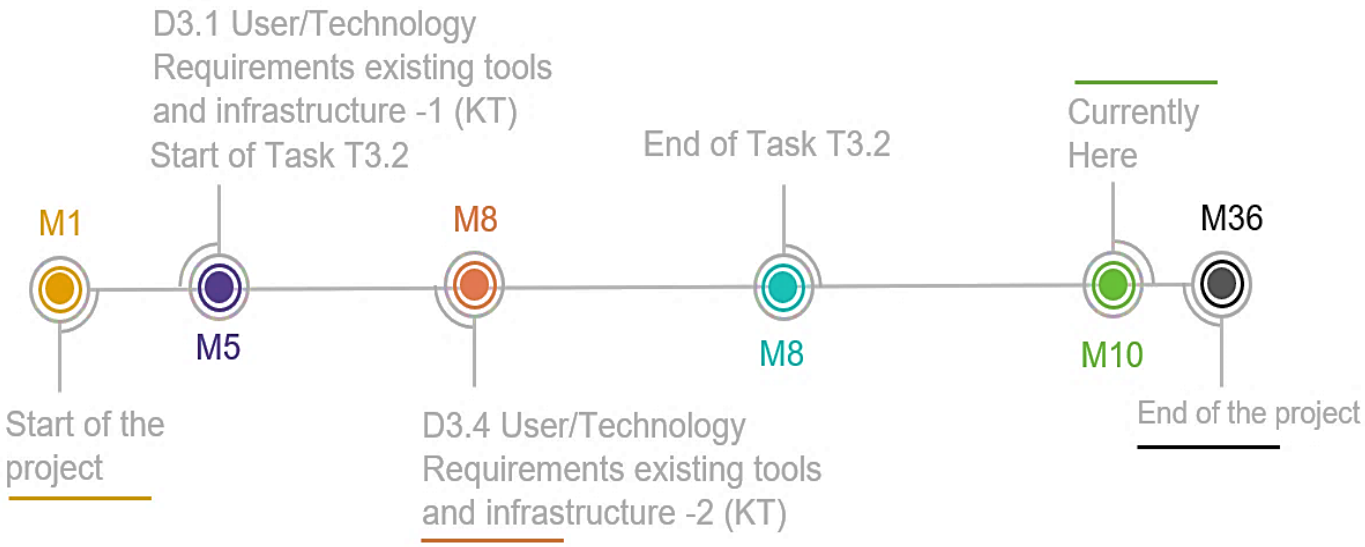


Figure 18: XR4ED architecture and Marketplace milestones from design to Version 2

6 User experience, operations and compliance

The previous sections have described the Marketplace concept, architecture and implemented features in detail. This chapter consolidates these aspects from three complementary perspectives that are critical for a production-grade educational platform: user experience and accessibility, operational robustness and DevOps practices and compliance with data protection, intellectual property and ethical requirements. Together, these perspectives demonstrate that the Marketplace has evolved from a technical prototype (D4.2) into a service that can be safely and reliably used by schools, teachers, learners and third-party developers across Europe.

6.1 User experience and accessibility as implemented

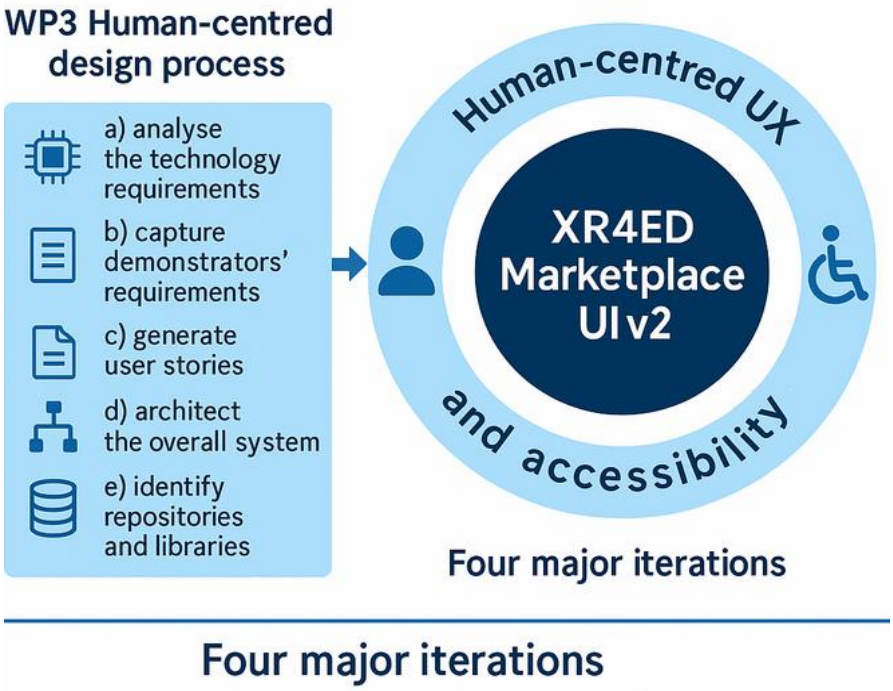


Figure 19: User experience & Accessibility implemented (WP2/WP3)

The XR4ED Marketplace user experience is grounded in the human-centred design work of WP2 and WP3, which elicited requirements from educational stakeholders and defined a common design system for the XR4ED platform. The Version 2 Marketplace UI builds on these foundations and on the UX concepts already introduced in D4.1 and D4.2, but is now fully implemented and integrated with the final platform backbone.

At a high level, the Marketplace UX is structured around three primary user journeys:

- **Discover and select XR solutions and assets** – the journey followed mainly by teachers, school staff and other facilitators who browse the catalogue, compare offerings and decide what to adopt in their context.
- **Publish and maintain XR offerings** – the journey of XR developers, FSTP teams and consortium partners who onboard solutions, update metadata and monitor usage.
- **Administer and curate the catalogue** – the journey of platform administrators who oversee content quality, taxonomy and policy compliance.

These journeys are supported by a modern, responsive web interface (as already illustrated in Figure 8: *XR4ED Marketplace UX* in Section 4) that adapts to desktops, laptops and mobile browsers and is optimised for launching XR content on head-mounted displays and AR-capable devices.

Key UX characteristics of Version 2

In Version 2, several UX characteristics have been consolidated and refined:

- **Clear information architecture**
  - The catalogue is organised into meaningful categories (e.g. STEM, history, arts, awareness topics) and collections (e.g. “virtual field trips”, “training of young professionals”), mirroring the XR4ED thematic pillars and open call focus areas.
  - Each solution or asset page follows a consistent template: summary, intended users and contexts, devices and platforms supported, pedagogical notes, technical requirements and links to related assets or documentation.
- **Guided user flows**
  - For educators, the UI emphasises *task-oriented* flows such as “find content for my subject and age group”, “filter by device I have available” and “see only free or open-source options”.
  - For providers and developers, the submission wizard guides users’ step by step through entering metadata, uploading assets, selecting licences and confirming platform compatibility, aligning with the onboarding lifecycle described in Section 4.2.

- **Consistent visual language and design system**
  - The visual design follows the XR4ED design system developed in WP3: shared typography, colour palette and layout patterns across the platform, with reusable components such as cards, tabs and filters.
  - This consistency makes it easier for users moving between the Marketplace, DevOps area, authoring tools and other XR4ED components.

Accessibility and inclusiveness

The Marketplace UX is aligned with XR4ED's ambition to build an "ecosystem of trust" and to support accessibility and inclusion as core principles, not afterthoughts. Concretely, Version 2 incorporates:

- **Accessible web front-end practices**, including:
  - A semantic HTML structure compatible with assistive technologies.
  - A focus on sufficient colour contrast for text and key UI elements, following WCAG 2.1 AA targets.
  - Forms and filters designed with clear labels, error messages and keyboard navigation support.
- **Inclusive content presentation**, for example:
  - Metadata fields that explicitly capture language, reading level and target group, helping teachers match solutions to their learners' needs.
  - Support for tagging solutions related to accessibility and inclusion use cases (e.g. content that supports students with disabilities, or scenarios focused on inclusive education).
- **Multi-role, family-friendly positioning**
  - Although the primary focus is on educators and learners, the Marketplace interface accommodates other roles mentioned in the proposal (such as parents and school staff) by providing explanatory text, tooltips and help content that do not assume technical expertise in XR development.

In short, the implemented UX combines a coherent design system, guided flows for each core role and a baseline adherence to accessibility principles, laying the foundation for further refinement based on pilot feedback and analytics, as described later in the validation chapter.

6.2 DevOps, environments and operational processes

Beyond UX, the Marketplace as delivered in Version 2 must operate as a reliable service integrated into the XR4ED platform backbone. This relies on cloud-based hosting, containerisation, a microservices architecture and a DevOps workflow that supports continuous evolution without disrupting users. These aspects extend and concretise the deployment and CI/CD principles described in D4.2 and in Section 3 of this deliverable.

Multi-environment deployment and integration with XR4ED

The Marketplace is deployed across **multiple environments** that mirror the stages of the software lifecycle:

- A **development environment** used by the core technical team for feature implementation and internal testing.
- One or more **staging or pre-production environments** where integrated builds are tested alongside other XR4ED components and where FSTP teams can trial their integrations without impacting real users.
- A **production environment**, exposed under the XR4ED domains, that hosts the live Marketplace available to teachers, learners and third-party providers.

All environments are based on the same architecture: containerised services (frontend, API and microservices) orchestrated by Kubernetes, with PostgreSQL as the primary database, and with storage and networking provided by the CELLOCK cloud infrastructure. This consistency ensures that behaviour tested in staging is representative of the behaviour observed in production.

DevOps pipeline and release process

A **continuous integration / continuous deployment (CI/CD)** pipeline automates the path from source code to running services:

- Code changes are committed to a shared repository and automatically subjected to unit, integration and end-to-end tests.

- On successful tests, container images are built and tagged, then deployed first to staging and finally to production through controlled release steps.
- Configuration is managed as code, enabling repeatable deployments and reducing configuration drift between environments.

This DevOps workflow supports the iterative evolution of the Marketplace during the project, including the integration of new XR services, APIs and UI features from other work packages and from third-party projects.

Monitoring, performance and resilience

Operational monitoring is essential to ensure that the Marketplace remains responsive and available as usage grows. Building on the principles set out in D4.2, the Version 2 deployment includes:

- **Application and infrastructure monitoring**, tracking response times, error rates, resource utilisation and background job health.
- **Structured logging**, used both for debugging and for security monitoring, with particular attention to sensitive operations such as authentication, content publication and role changes.
- **Performance testing and optimisation**, where tools such as Lighthouse and JMeter have been used to identify client-side and server-side bottlenecks, leading to optimisations in API calls, database queries and content delivery (e.g. via CDNs).

At infrastructure level, the Marketplace benefits from:

- **Auto-scaling and load balancing**, allowing the system to scale horizontally (by adding more container instances) during peak usage, for example when multiple schools access XR content simultaneously.
- **Redundant storage and backup mechanisms**, with regular backups stored in separate locations and a disaster recovery plan that supports rapid restoration in case of failure.

Together, these practices ensure that the Marketplace can support the project's pilots, open-call experiments and early adopters while remaining ready to scale to larger user communities.

Operational processes

Finally, the DevOps approach is complemented by **operational processes** that govern:

- **Release planning**, aligning feature deployments with WP4 milestones, WP5 open call timelines and feedback from pilots.
- **Change and incident handling**, where any critical issue in production triggers a documented response workflow and, if needed, a rollback to a previous stable version.
- **Configuration and access management**, ensuring that only authorised personnel can modify production settings or secrets.

These processes formalise the transition from a research prototype to a service that can be reliably offered to schools and external developers.

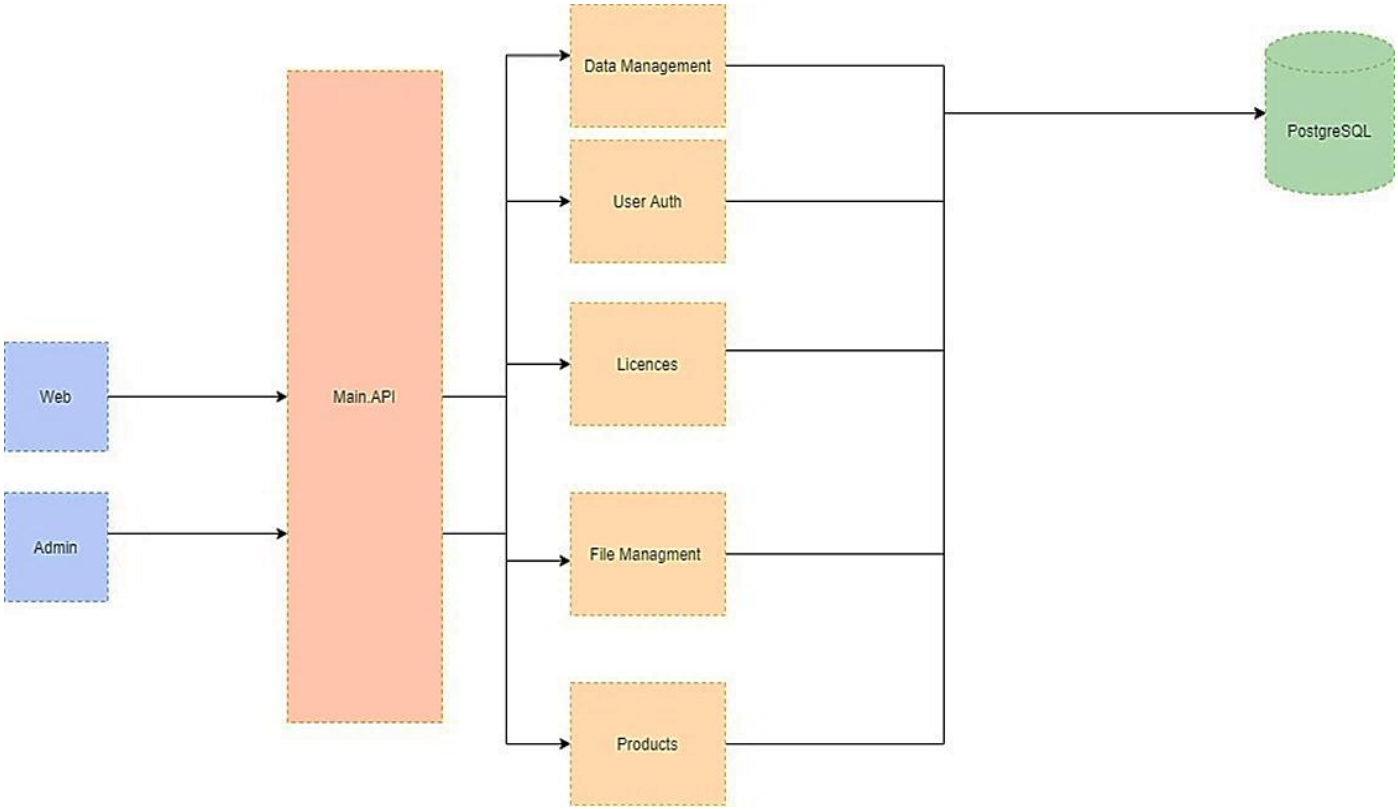


Figure 20: XR4ED 3-tier architecture underpinning the Marketplace deployment

If you want to emphasise microservices, you can also reuse one of the slides titled **"Microservices – User Authentication"** or **"Microservices – Licenses"** as an optional figure in an annex, and cross-reference from the text that discusses microservice-based deployment.

6.3 Compliance, privacy and content licensing in the Marketplace

The XR4ED Marketplace operates within a strict regulatory and ethical framework, reflecting both the legal obligations of an EU-funded digital service and the project's ambition to create an ecosystem of trust for XR in education. This subsection summarises how Version 2 of the Marketplace addresses data protection, intellectual property, terms of use and broader ethical considerations, building on the principles already outlined in D4.2 and in the XR4ED proposal.

6.3.1 Data protection and privacy

Data protection is anchored in the General Data Protection Regulation (GDPR) and related national frameworks. From a Marketplace perspective:

- The platform minimises the personal data it collects, focusing on what is necessary for account management, solution publication and analytics (e.g. contact details for providers, basic profile information for educators, aggregated usage data).
- All traffic between clients and the platform is protected via HTTPS/TLS, and sensitive data (such as passwords and selected personal details) is stored encrypted or hashed and salted, as appropriate.
- Role-based access control ensures that only authorised users can view or modify sensitive information (e.g. administrative actions, provider billing data).
- Users are informed, through privacy notices and terms of use, about what data is collected, for which purposes and how they can exercise their rights (access, rectification, erasure, restriction and portability).
- Data protection impact assessment (DPIA) and ethics work in WP2 provide the overarching framework for assessing risks related to XR usage in education, including the handling of data collected by XR experiences accessed via the Marketplace.

6.3.2 Intellectual Property Rights and content licensing

The Marketplace has been designed to respect and protect intellectual property rights, while also enabling reuse and interoperability in line with XR4ED's open standards approach:

- Providers submitting XR solutions and assets must confirm that they hold the rights to publish the content or that it is distributed under suitable licences.

- The onboarding workflow includes fields to specify the **licence type** (e.g. proprietary, Creative Commons variants, open-source licences) for both software and media assets, making rights and reuse conditions explicit on product pages.
- A takedown procedure is foreseen to handle suspected IP infringements or other policy violations, including a channel for rights holders to report issues and for administrators to remove or unlist contested content while a case is investigated.
- For solutions developed under XR4ED open calls, contractual conditions ensure that outcomes are compatible with the requirement to publish in the European XR4ED Solutions Catalogue and to use open standards such as WebXR and OpenXR.

This model supports a mixed ecosystem where commercial products, open educational resources and research prototypes can coexist, while maintaining clarity on what users are allowed to do with each item.

6.3.3 Terms of use and ethical framework

Compliance is not limited to GDPR and IPR. The Marketplace is also framed by:

- **Terms of use and user agreements**, which define acceptable behaviour, responsibilities of users and providers, and conditions under which accounts or content may be suspended or removed. These terms are aligned with the compliance and legal considerations already discussed in D4.2.
- **Alignment with the XR4ED ethics and ecosystem of trust framework**, which addresses issues such as age-appropriate use of XR in schools, transparency about XR risks and benefits and sensitivity to vulnerable populations.
- **Awareness of broader regulatory domains**, such as consumer protection and e-commerce rules, particularly in scenarios where paid offerings or bundled services are made available through the Marketplace.

These elements ensure that the Marketplace not only complies with formal regulations but also reflects the project's commitment to responsible innovation in XR-based education.

7 Validation, lessons and roadmap – from prototype to final Marketplace

The first version of the XR4ED Marketplace, documented in D4.2, provided a blueprint for a one-stop shop that brings together XR solutions, tools and services for learning, training and education. D4.5 captures the evolution of this blueprint into a deployed, operational component of the XR4ED backbone, integrated with the platform architecture, DevOps environment and compliance framework described in the previous chapters.

This chapter consolidates the validation view of the Marketplace Version 2. It summarises how the platform has been technically verified and integrated within XR4ED, how feedback will be captured from developers, pilots and internal users, what lessons can be drawn from the development and integration work in WP3–WP5 and which next steps are foreseen for the Marketplace beyond the project.

7.1 Technical verification and integration status

Technical verification of the Marketplace Version 2 has focused on confirming that the implemented platform matches the architecture, technology stack and feature set defined in XR4ED requirements and in the first Marketplace deliverable. The verification work has been carried out primarily within WP4, with strong links to WP3 (requirements and architecture) and WP5 (open calls and pilots).

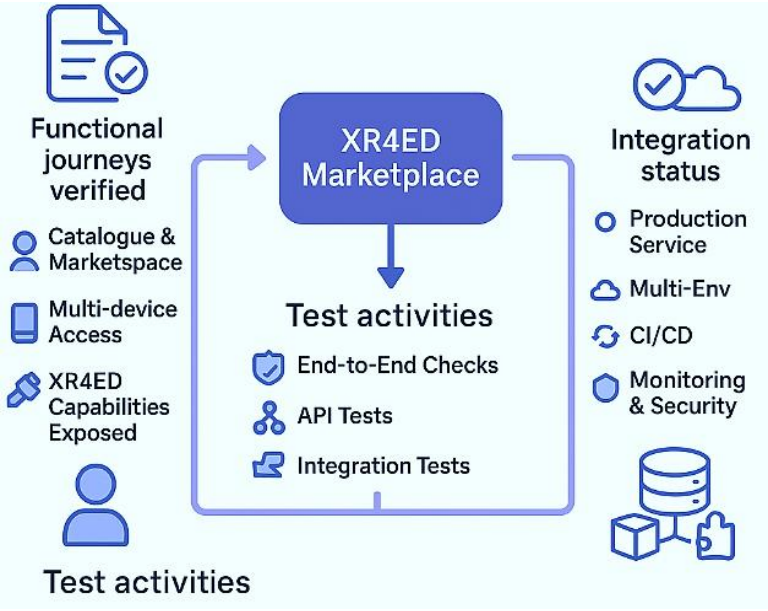


Figure 21: Technical Verification & Integration Status

From a **functional** perspective, the Marketplace has been verified against the core user journeys and capabilities described earlier in this deliverable:

- **Catalogue and Marketspace functions**
  - Registration and login for end-users and providers using the shared authentication and user management services.
  - Browsing and full-text search across XR solutions and assets, with filtering by subject, target group, XR technology and other metadata fields.
  - Creation, submission and update of solutions and assets by providers through the guided onboarding workflow.
  - Administrative review, approval and publication of submissions and the ability to curate catalogue entries and categories.
- **Multi-device access flows**
  - Access to content for VR, AR and mobile, including download links and QR-based flows where applicable, as described in the device-specific access patterns.
- **Exposure of XR4ED capabilities**
  - Representation of XR4ED consortium tools (authoring tools, DevOps environment, reusable XR components) as Marketplace entries, in line with the platform capabilities defined in the project description.

From a **technical and architectural** perspective, verification has addressed the alignment between the implemented system and the layered architecture described in Chapter 3:

- The **frontend** is implemented as a React/Next.js web application written in TypeScript, providing the public user interface and the provider/admin views.
- The **backend** is implemented as a Node.js/Express API layer, also written in TypeScript, exposing RESTful endpoints for all core Marketplace operations.
- The **database** is a PostgreSQL instance storing users, solutions, assets, metadata and licensing information.

- A **set of microservices** (authentication, user profile, products, file management, licensing, notifications and integration) provide specialised functionality behind the main API, following the logical component breakdown in Section 3.2.

These components have been deployed in a **containerised cloud environment**, orchestrated by Kubernetes and integrated into the XR4ED backbone as described in the deployment architecture. Multiple environments (development, staging/pre-production and production) are maintained, allowing features to be tested and validated before they reach end-users.

Technical verification has therefore concentrated on:

- End-to-end functional checks across the main user journeys (discovery, submission, administration), ensuring that front-end flows map correctly to backend services and database operations.
- API-level tests, confirming the behaviour of key endpoints (search, listing retrieval, submission, user and metadata management) and their adherence to the documented OpenAPI specification.
- Integration tests with shared XR4ED services (authentication, user management, DevOps and other backbone components) to ensure consistent identity handling and secure cross-component communication.

At the time of writing, the Marketplace has reached a **stable integration status** within the XR4ED platform backbone:

- The platform runs as a production service under the XR4ED domains, using the shared infrastructure provided by the cloud environment.
- The implemented features correspond to the scope outlined for Version 2 in this deliverable and in the original platform design.
- The technical foundations (architecture, CI/CD pipeline, monitoring and security mechanisms) are in place to support further population of the catalogue and Marketspace by XR4ED partners and third-party projects in the remaining project period.

7.2 Feedback from developers, pilots and internal users

Because the Marketplace is both a technical platform and a community-facing service, validation is not limited to internal testing. Feedback from developers, pilots and internal users is essential to refine user experience, content structures and integration workflows.

Within XR4ED, feedback is collected through several **complementary channels**:

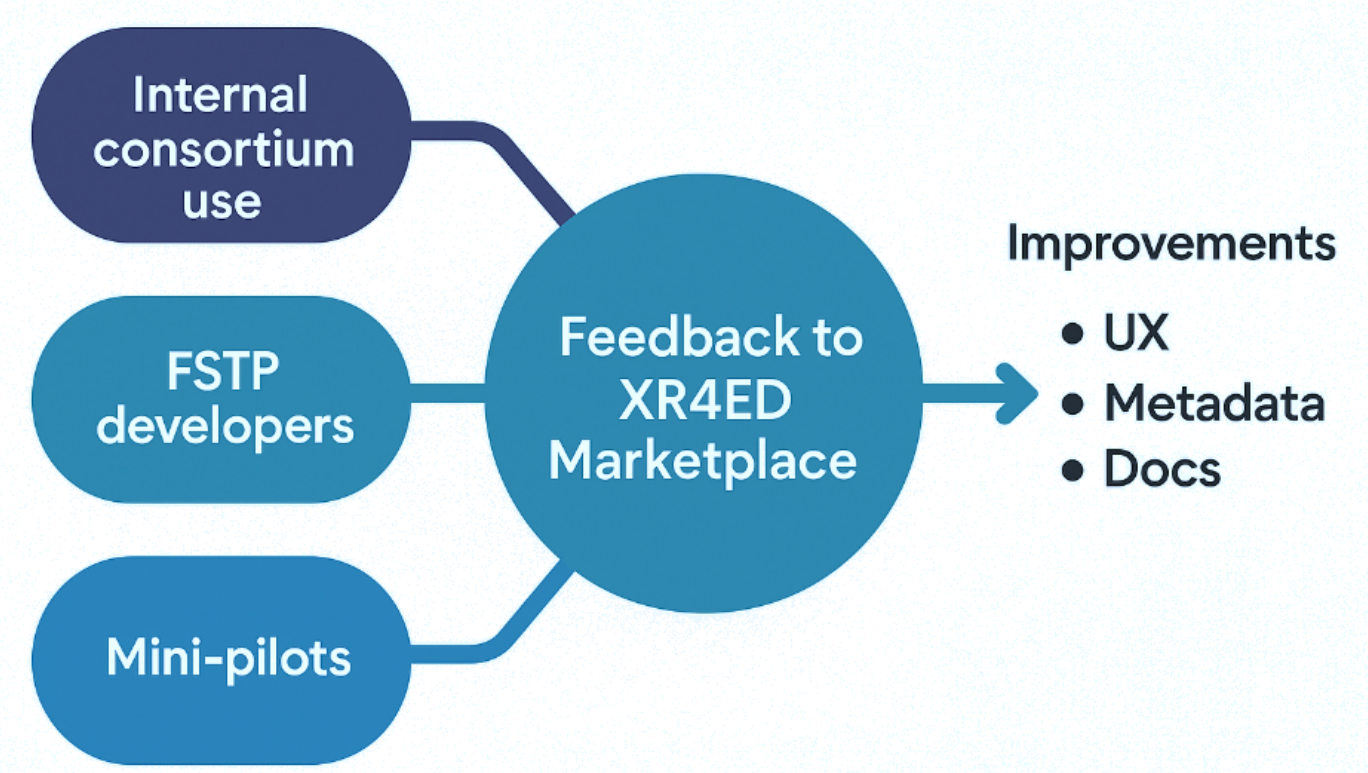
- **Internal use within the consortium**
  - WP3 and WP4 partners use the Marketplace in day-to-day development and integration activities (for example, publishing consortium tools and assets), providing direct feedback on UX, onboarding and administration flows.
  - Technical teams report issues and suggestions through the project's issue-tracking tools, which feed back into WP4 development iterations.
- **Open call developers and FSTP projects**
  - The open call Developers track requires funded projects to integrate their XR solutions with the XR4ED platform and to publish their outcomes in the Solutions Catalogue.
  - As these projects use the DevOps environment and Marketplace submission flows, their experiences and integration challenges will be used to fine-tune the API design, documentation and provider onboarding processes.
- **Mini-pilots and educational users**
  - The Pilots track provides evidence on how XR solutions accessed via the Marketplace perform in real educational environments. Pilot teams are expected to supply qualitative feedback (e.g. ease of discovering suitable solutions, clarity of product pages, practical issues in download and access) together with their pedagogical evaluation.
  - Experiences from pilots will inform improvements to metadata schemas, categorisation and guidance offered to educators in the Marketplace interface.

In addition to these project-internal mechanisms, the Marketplace has been designed to support **continuous feedback** over time:

- The platform can incorporate contact forms, support channels and lightweight feedback widgets on solution pages, allowing educators and providers to report issues or suggest enhancements.

- Usage analytics and search logs, collected under the monitoring framework, can highlight patterns such as common search terms with few results or high-traffic solution types, which in turn inform curation and UX adjustments.

At this stage, D4.5 focuses on describing the **feedback processes and integration points** rather than presenting exhaustive quantitative results, which will mature as open-call projects and pilots progress. Subsequent WP5 and WP6 deliverables will complement this deliverable with detailed usage and impact evidence.



7.3 Lessons learned and open points

The development and integration of the Marketplace across WP3, WP4 and WP5 have already yielded a number of technical and organisational lessons, even before full pilot and open-call results are available.

From a **technical and architectural** standpoint:

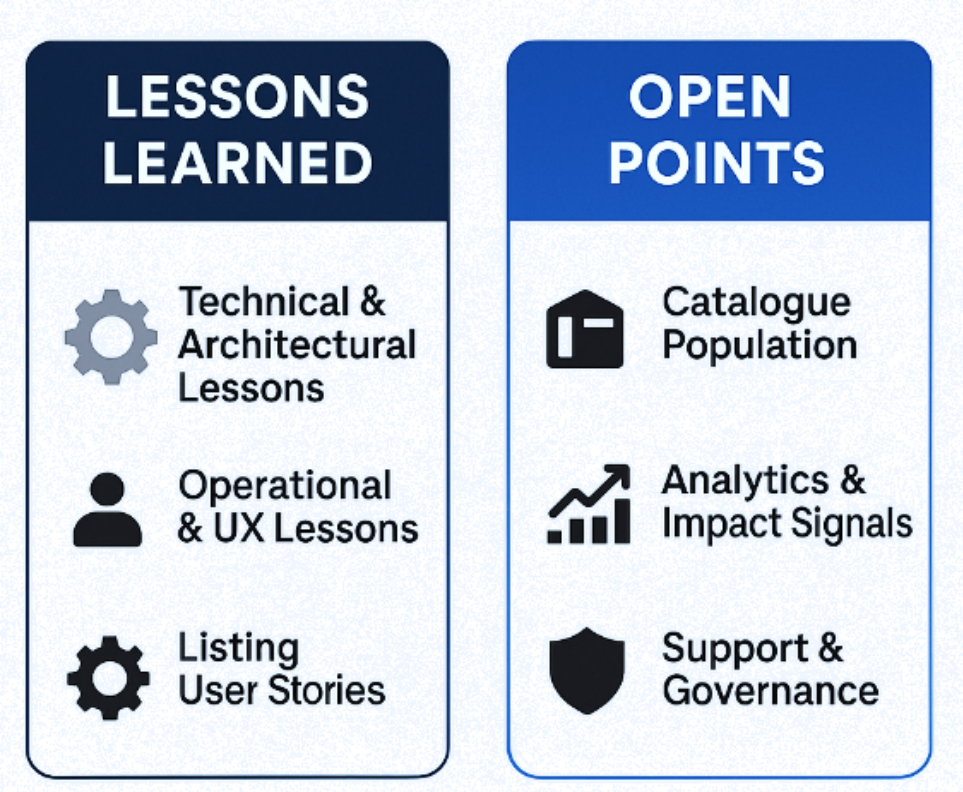
- The decision to adopt a **microservice-oriented architecture** has proved appropriate for a platform expected to evolve beyond the project. Separating authentication, products, file management, licensing, notifications and integration services has facilitated incremental development and reduced coupling between the Marketplace and other XR4ED components.
- A consistent **technology stack** (TypeScript, React/Next.js on the frontend and Node.js/Express with PostgreSQL on the backend) has simplified maintenance and improved type safety across the platform. This homogeneity has also supported smoother onboarding for developers.
- Investing early in **API-first design and OpenAPI documentation** has reduced friction when integrating internal XR4ED tools and when preparing for external developer use through the DevOps environment and open calls.

From an **operational and UX** perspective:

- The layered deployment architecture, with mirrored development, staging and production environments, has been important to manage change without disrupting users, especially as the Marketplace began to be populated with real entries.
- Aligning UX and accessibility decisions with the human-centred design work of WP2 and WP3 has helped ensure that new features added in WP4 remain consistent with the overall XR4ED design system and accessibility principles.

Despite the progress made, several **open points** remain that will be addressed in the remaining project period and, where necessary, beyond it:

- **Catalogue population and curation**
  - While the Marketplace structure, metadata schema and workflows are in place, continuous population and curation of entries is required to fully demonstrate its potential as a European reference platform. This includes integrating additional consortium tools, assets and open-call outcomes as they become available.
- **Analytics, impact and quality signals**
  - Basic analytics and monitoring infrastructure exist at platform level, but further work is needed to translate these into user-facing indicators (e.g. ratings, usage badges, featured items based on verified impact) in a way that respects privacy and avoids bias.
- **Support processes and governance**



- The technical building blocks for notifications, licensing and compliance are in place; however, governance processes (for example, who approves content, who handles takedown requests, which criteria define “featured” solutions) need to be finalised and documented in alignment with WP2 ethics and WP6 exploitation work.

Addressing these open points will further strengthen the Marketplace as a sustainable asset and will ensure that technical capabilities are matched by clear operational practices.

7.4 Roadmap and sustainability snapshot

The XR4ED proposal positions the Marketplace as one of the project’s main exploitable assets and as a cornerstone of a long-term, on-demand platform for XR in education. D4.5 provides the technical and architectural basis for this ambition by documenting the “as-built” state of the Marketplace towards the end of the development cycle.

A concise **roadmap for the remaining project period** can be outlined along the following lines:

➔ **Finalising population and integration**

- Continue onboarding XR4ED consortium tools, reusable components and assets into the catalogue and Marketspace, ensuring that each major XR4ED capability described in the project description is represented by one or more Marketplace entries.
- Support funded Developer and Pilot projects in publishing their solutions and, where applicable, linking them to pilot use cases and lessons learned.

➔ **Refining UX and workflows based on feedback**

- Apply incremental UX improvements informed by internal usage and early feedback from developers and educators, especially in the areas of solution discovery, submission forms and admin tools.

- Adjust metadata schemas and categorisation where needed to better reflect real-world usage and educational contexts.

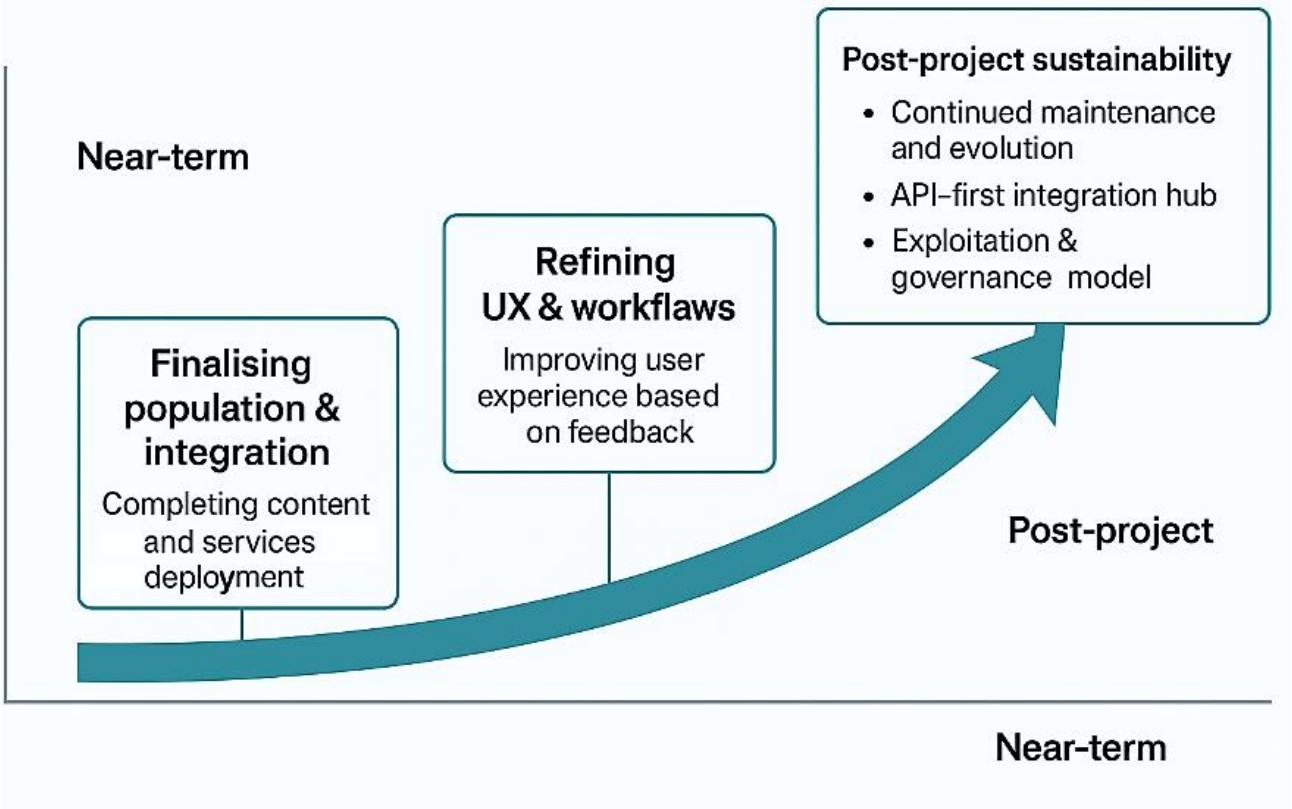


Figure 22: Roadmap and sustainability Snapshot

➔ **Consolidating compliance and governance**

- Finalise and document operational procedures for content moderation, IP and takedown processes, role management and handling of support requests, building on the compliance mechanisms described in Section 6.3.

Looking **beyond the end of the XR4ED project**, the Marketplace is intended to remain a visible and reusable asset that can support further innovation in XR for education:

- The cloud-native, microservice-based deployment architecture and CI/CD pipeline are designed to allow **continued maintenance and feature evolution** with manageable effort.

- The API-first design and clear separation between platform backbone and user-facing layers position the Marketplace as an integration point for future XR initiatives, allowing additional tools, repositories or national platforms to connect or federate catalogues.
- Exploitation and sustainability activities in WP6 will determine the long-term **hosting, ownership and governance model**, including how the Marketplace can be aligned with wider European EdTech ecosystems and how the community of providers and users engaged through the open calls and pilots can be maintained and expanded.

In summary, validation activities carried out to date confirm that the Marketplace has progressed from a forward-looking blueprint in D4.2 to a deployed and integrated platform component in D4.5. The remaining work focuses on densifying content, refining user experience and establishing governance arrangements that will allow the XR4ED Marketplace to continue operating as a European reference point for XR in education after the project's completion.

8 Conclusions

This deliverable has presented the XR4ED Marketplace – Version 2 – as a fully-fledged, deployed and integrated component of the XR4ED platform backbone. It builds directly on the conceptual design and first implementation captured in D4.2 and now documents the "as built" status of the Marketplace towards the end of the development cycle, when it can be operated as part of the wider XR4ED ecosystem rather than as an isolated prototype.

From a **conceptual and ecosystem perspective**, the Marketplace realises XR4ED's ambition to act as a European one-stop shop for XR in education. It consolidates the XR Solutions Catalogue and the Marketspace into a single-entry point through which teachers, learners, developers, intermediaries and other stakeholders can discover XR experiences, reusable components and related services. The catalogue dimension offers structured, searchable access to XR solutions and assets, while the Marketspace dimension enables offer–demand interactions around services such as content development, integration support and training. Both are tightly connected with XR4ED's cascade funding scheme: developers' projects and mini-pilots are required to publish their results in the Marketplace, thereby enriching the European XR4ED Solutions Catalogue with solutions tested in real educational settings. In this way, the Marketplace moves beyond a static list of applications to become an evolving, community-facing reference platform for XR in education.

At **architectural** level, D4.5 confirms the viability and value of the microservice-based, cloud-native approach defined in WP3 and implemented under T4.1. The final implementation uses a consistent technology stack – React/Next.js for the frontend, Node.js/Express with TypeScript for the backend and PostgreSQL as the core data store – deployed as containerised services orchestrated by Kubernetes. Around the main web and API layers, dedicated microservices provide authentication and user management, product and metadata handling, file and content management, licensing, notifications and integration capabilities. This modular decomposition supports clear separation of concerns, facilitates independent evolution of components and allows the Marketplace to scale as usage grows or as new XR4ED capabilities are brought online. It also anchors the Marketplace firmly in the XR4ED backbone, where it interoperates with the DevOps environment, authoring tools and other platform services through well-defined APIs and integration patterns.

In terms of **implemented features and user experience**, the Marketplace now offers complete end-to-end flows for the main user roles identified in the project. Educators and other end-users can register, browse the catalogue by subject, age group, XR technology or educational context, compare solutions and access XR experiences via device-specific flows covering VR headsets, AR-enabled devices and mobile/WebXR scenarios. Providers – including XR4ED partners and third-party developers – can onboard solutions and assets through a guided submission workflow that covers metadata, licensing and technical requirements, and can subsequently maintain and update their entries. Administrators have access to curation and moderation tools that allow them to review submissions, manage categories and collections, adjust visibility and manage user roles. These capabilities are delivered through a user interface that follows the common XR4ED design system and human-centred design principles developed in WP2 and WP3, with attention to consistency, clarity and baseline accessibility (for example semantic structure, colour contrast and keyboard navigation).

From an **operations and compliance** standpoint, the Marketplace has been engineered to function as a service that can be run under realistic conditions, not only within a research environment. Multi-environment deployment (development, staging and production), CI/CD pipelines, monitoring and logging are in place to manage releases and support incident response without compromising platform stability. Security and performance measures – including HTTPS/TLS, role-based access control, authentication and authorisation mechanisms, and performance monitoring – are applied consistently across environments. On the compliance side, the Marketplace aligns with GDPR principles by limiting the personal data it processes, securing data at rest and in transit and informing users about data use and their rights. Content licensing is made explicit through metadata fields in the onboarding workflow, enabling coexistence of proprietary, open-source and open educational resources while safeguarding

intellectual property. These technical and procedural elements contribute to XR4ED's broader "ecosystem of trust", in which privacy, ethics and inclusiveness are treated as core design constraints rather than optional add-ons.

The **validation and lessons-learned** view presented in Chapter 7 underlines how the Marketplace has evolved from a forward-looking design to an integrated platform component. Technical verification has checked alignment between the implemented system and the planned architecture, including functional tests of the key user journeys, API-level checks and integration tests with shared XR4ED services. In parallel, the project has put in place channels to capture qualitative feedback from consortium users, open-call developers and pilot teams, which will inform refinement of the UX, metadata schemas and onboarding processes as more solutions are published and piloted. Lessons learned so far highlight the benefits of the chosen microservice architecture and coherent technology stack, the importance of API-first design for internal and external integration and the value of maintaining mirrored environments to support safe evolution. Open points have also been identified, notably the need for ongoing catalogue curation, the translation of analytics into meaningful quality and impact signals and the finalisation of governance procedures for moderation, takedown and support.

Looking ahead, the **roadmap and sustainability snapshot** outlined in D4.5 indicates that the Marketplace is well positioned as a key exploitable result of XR4ED. In the remaining project period, emphasis will be placed on densifying the catalogue and Marketspace with additional consortium tools, assets and third-party solutions from the developers' and pilots' tracks, and on incrementally improving UX and workflows based on real-world feedback. Beyond the project, the cloud-native deployment model, microservice architecture and API-based integration patterns are intended to support continued maintenance and functional extension under an appropriate hosting and governance model to be defined in WP6. The Marketplace can thus act as a stable integration point for future XR initiatives and national or European platforms, as well as a reference catalogue that showcases XR solutions aligned with European values and standards for education.

In summary, this deliverable shows that the XR4ED Marketplace has successfully progressed from the **conceptual and early implementation stage in D4.2** to a **consolidated, operational component in D4.5**. It:

- Realises the vision of a **European one-stop-shop and reference platform** for XR in education, bringing together solutions, services and stakeholders in a single, trusted environment.

- Exposes XR4ED's own capabilities – including authoring tools, DevOps services and reusable XR components – as reusable entries, while providing clear pathways for third-party contributions via cascade funding and open calls.
- Provides a robust **technical, operational and compliance framework** that can support further growth in content, users and integrations and can be maintained beyond the end of the project.

As such, the Marketplace stands not only as a core asset of the XR4ED platform but also as a concrete contribution to the broader European effort to foster high-quality, trustworthy XR innovation in education, training and lifelong learning.

9 References

[1] XR4ED Consortium, *XR4ED – eXtended Reality for Education, Grant Agreement and Description of Action (Part B)*, Horizon Europe Project XR4ED, Grant Agreement No 101093159, 2022.

[2] XR4ED Consortium, *D4.1 – Platform Backbone and Integration – Initial Version*, Project Deliverable, Horizon Europe Project XR4ED, 2023.

[3] XR4ED Consortium, *D4.2 – XR4ED Marketplace – Version 1*, Project Deliverable, Horizon Europe Project XR4ED, 2024.

[4] XR4ED Consortium, *XR4ED Platform and Marketplace – Progress Presentation*, internal project presentation, Horizon Europe Project XR4ED, 2024.

[5] XR4ED Consortium, *XR4ED Project Website*, 2023–2025. Available at: https://xr4ed.eu/ (accessed November 2025).

[6] European Commission, *XR4ED – eXtended Reality for Education, CORDIS Fact Sheet*, CORDIS, 2023. Available at: https://cordis.europa.eu/project/id/101093159 (accessed November 2025).

[7] F. Liarokapis, V. Milata, J. L. Ponton, N. Pelechano and H. Zacharatos, "XR4ED: An Extended Reality Platform for Education," *IEEE Computer Graphics and Applications*, vol. 44, no. 4, pp. 79–88, July/Aug. 2024.

[8] European Parliament and Council of the European Union, *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation)*, Official Journal of the European Union, L119, 4 May 2016.

[9] World Wide Web Consortium (W3C), *Web Content Accessibility Guidelines (WCAG) 2.1*, W3C Recommendation, 5 June 2018 (and subsequent updates). Available at: https://www.w3.org/TR/WCAG21/ (accessed November 2025).

[10] World Wide Web Consortium (W3C), *WebXR Device API*, W3C Technical Report, 2025. Available at: https://www.w3.org/TR/webxr/ (accessed November 2025).

[11] Khronos Group, *OpenXR™ Specification*, Version 1.x, Khronos Registry, 2024. Available at: https://www.khronos.org/registry/OpenXR/ (accessed November 2025).

10 Annexes

This annex collects selected figures reused from the internal XR4ED Platform and Marketplace – Progress Presentation (2024). They provide additional visual context on the XR4ED architecture, the Marketplace roadmap and risks, and sample user interface views.

These materials are not required to understand the main body of the deliverable but may be useful for reviewers and technical stakeholders who wish to see the evolution from the early design work in WP3 to the final Marketplace implementation and to visualise concrete screens of the running system.

Figure 23: XR4ED Platform

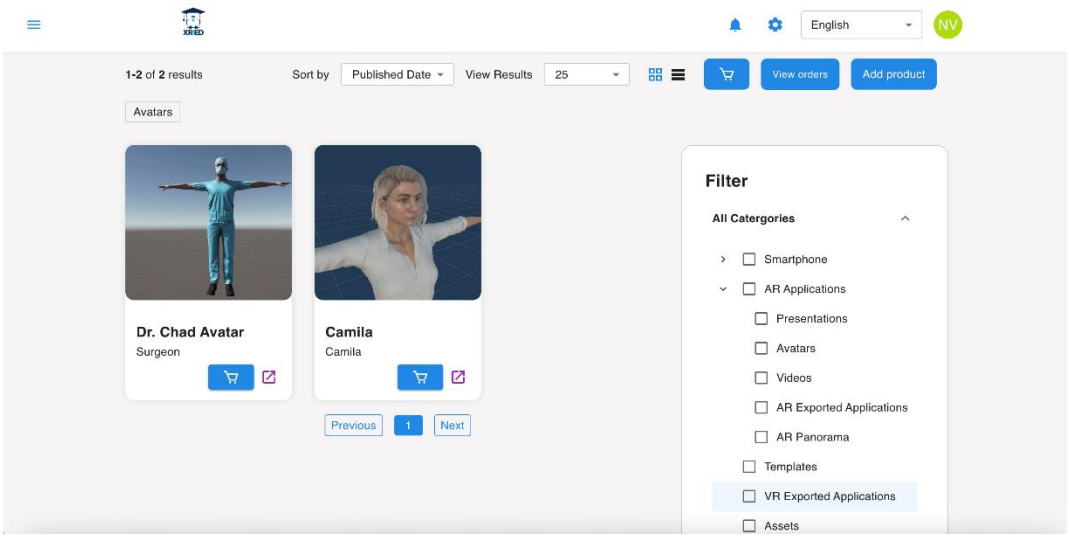


Figure 24: List of Products

Figure 26: Product Details

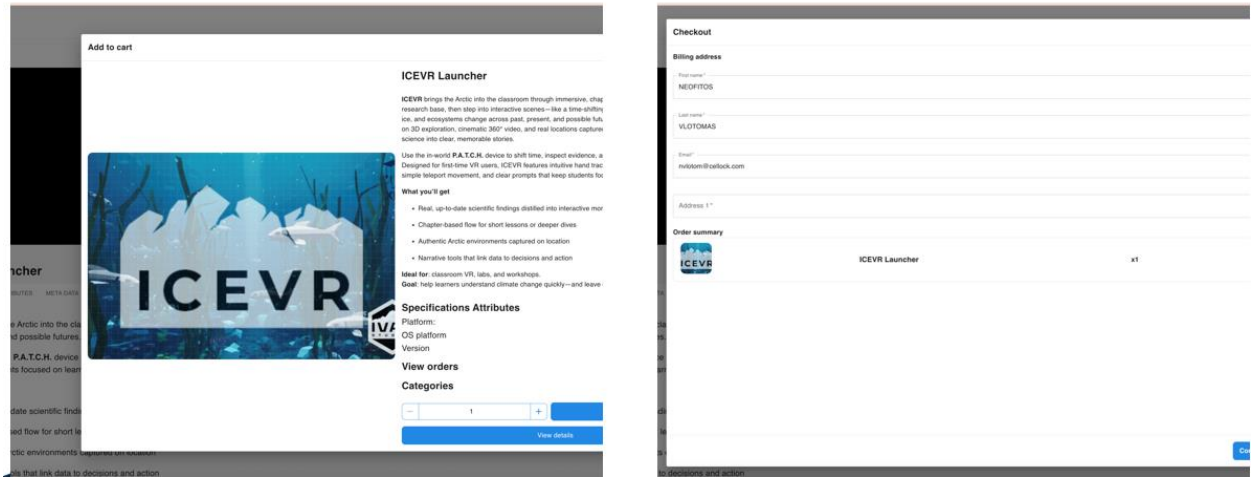


Figure 25: Check out Product

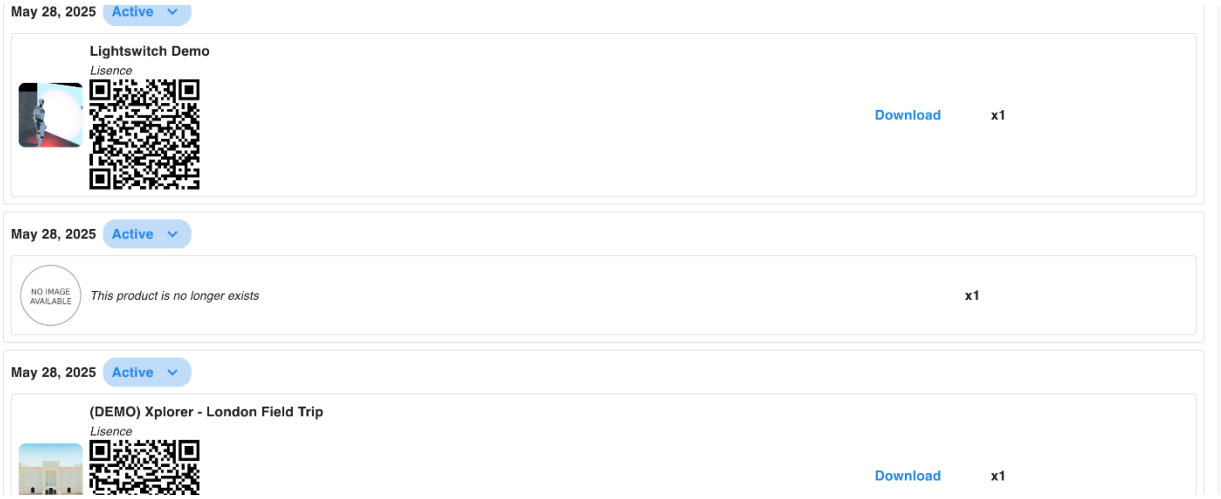


Figure 27: View my Orders

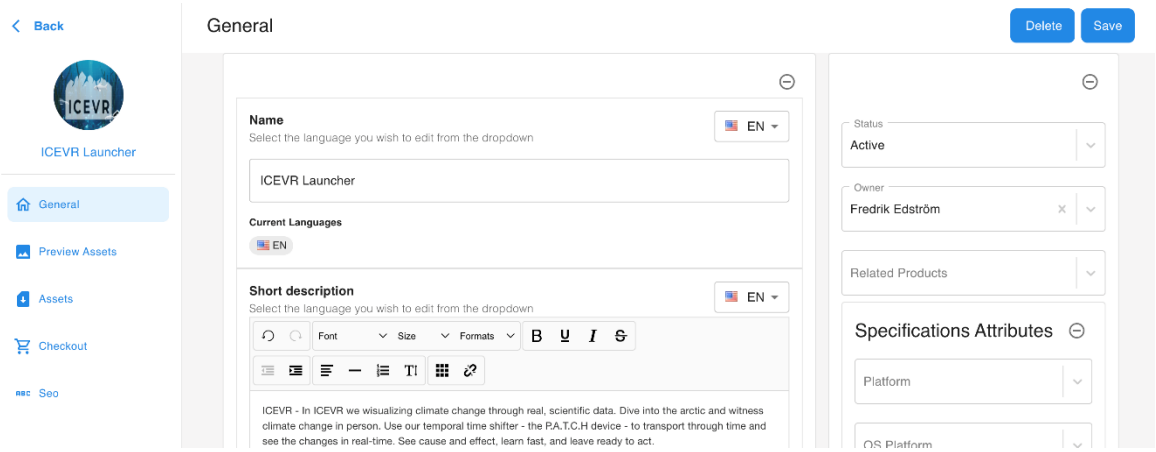


Figure 28: Edit Product_1

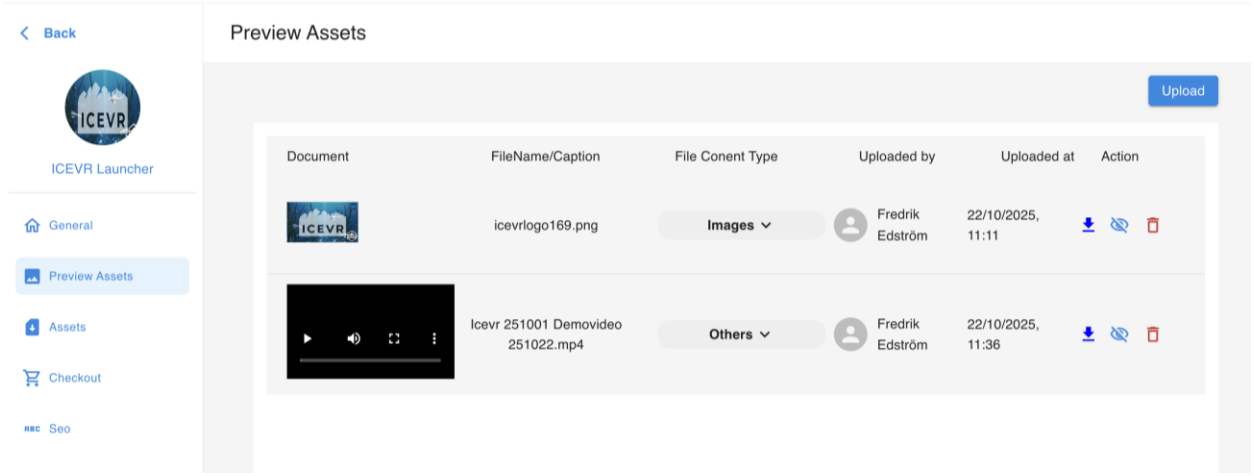


Figure 29: Edit Product_2

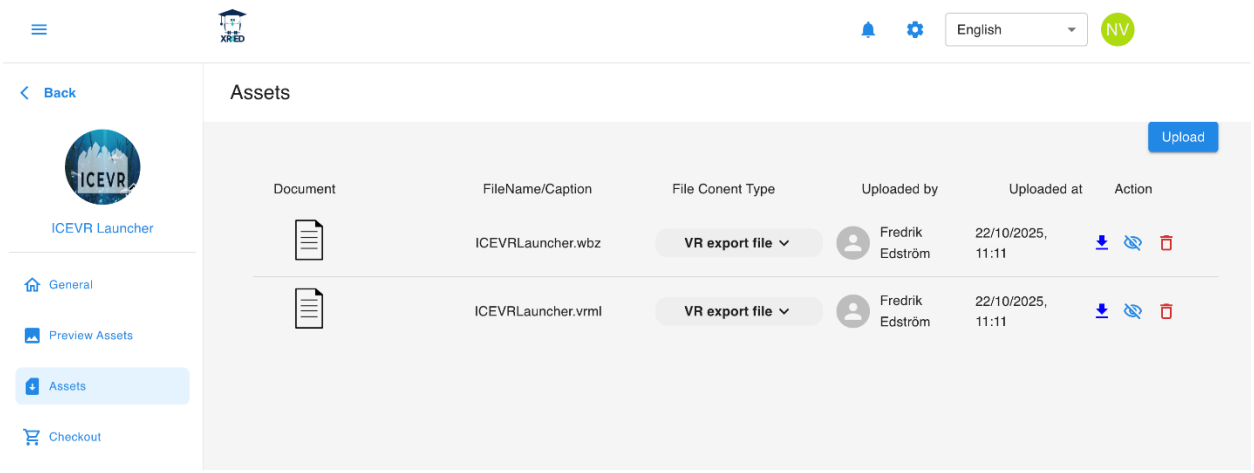


Figure 31: Edit Product_3

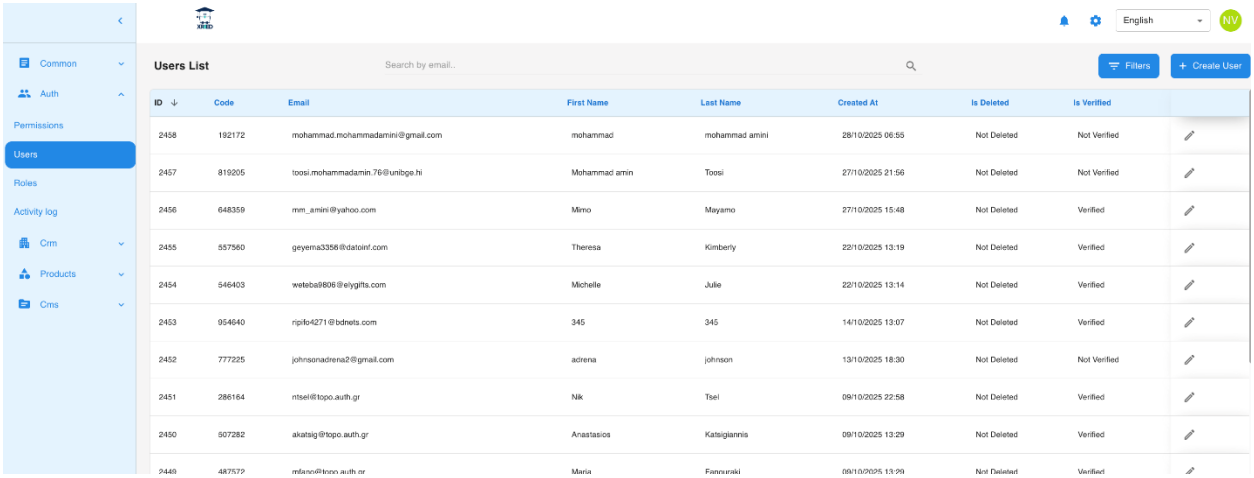


Figure 30: User Management

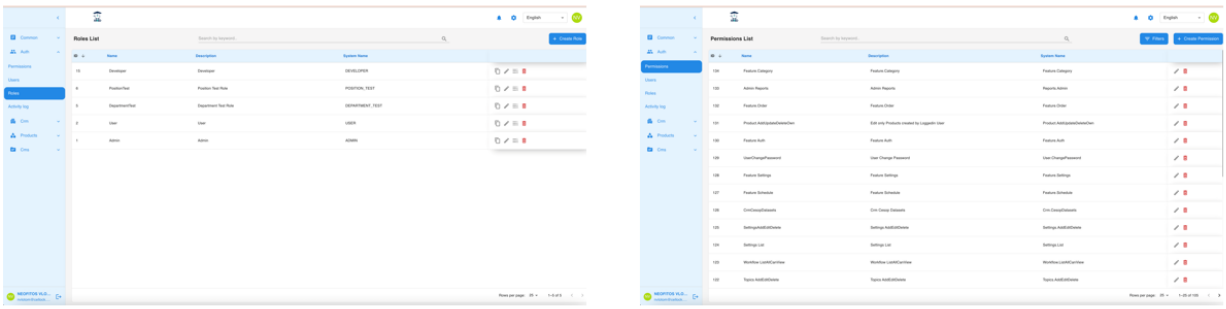


Figure 32: List of Roles and Permissions

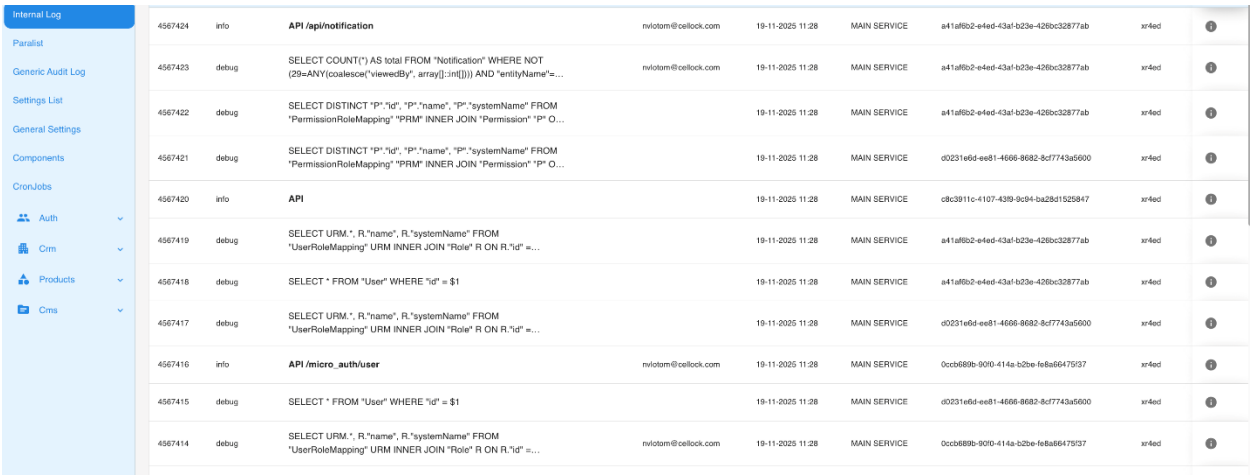


Figure 33: Logging