Ref. Ares(2025)11598907 - 26/12/2025

D4.4
Testing Report

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/

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	T4.4	
Related deliverables	D4.4	
Type		DEM - Demonstrator, Pilot, Prototype
		DMP - Data Management Plan
	✓	R - Document, Report
Status	Draft	
Version	1.0	
Due date	31/12/2025	
Submission date	26/12/2025	
Lead partner	Software Imagination & Vision SRL (SMV)	
Contributing partners	CYENS, KT, SC, IDL, UPC, CLK	
Reviewers	Fotis Liarokapis, Maria Poveda	
Keywords	Test case	

Dissemination level	
✓	PU: Public
	SEN: Sensitive

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

Revision History

Document History			
Version	Date	Change editors	Changes
0.1	10/2025	Andreea Paunescu, Simona Bica (SMV)	First Draft of deliverable
0.2	10/2025	Simona Bica, Daniel Ghergiceanu, Mihaela Caraveteanu (SMV)	First completions in Section 1, Section 2, Section 3, Section 4, Section 5 and Section 6
0.3	11/2025	Simona Bica, Andreea Paunescu (SMV), Neophytos Vlotomas, Haris Zaharatos, Maria Basta (CLK)	Updates in Section 1, Section 3, Section 4, Section 5 and Section 6
0.4	12/2025	Nuria Pelechano (UPC)	Contribution for Section 5
0.5	12/2025	Simona Bica, Andreea Paunescu, Marius Jianu (SMV)	Final version of deliverable ready for the internal review
1.0	26/12/2025	Fotis Liarokapis	Final version

Quality control		
Role	Partner (short name)	Approval date
Deliverable leader	SMV	26/12/2025
Quality manager	CYENS	26/12/2025
Project coordinator	CYENS	26/12/2025

Table of Contents

XR4ED Project Profile 2
Partners 3
Document information 4
Revision History 6
Table of Contents 8
List of Figures 10
List of Tables 10
List of Acronyms 11
1 Executive summary 12
2 Introduction 13
2.1 Purpose of the document 13
2.2 Structure of the document 13
3 System Architecture Overview 14
4 Testing methodology 15
5 Technical verification and testing by component 19
5.1 XR Authoring Tool 19
5.2 XR Marketplace 21
5.3 XR Avatars and Asset Library 24
5.4 XR DevOps Environment 27
5.5 Asset Tools 29
5.6 Interfaces & UX 36
5.7 Ethics & Privacy Observatory 38
6 Integration testing 40

XR-INT-001: XR Authoring Tool ↔ Marketplace 40
XR-INT-002: Marketplace ↔ Licensing 42
XR-INT-003: DevOps ↔ Marketplace 43
XR-INT-004: Avatars ↔ Authoring Tool 44
XR-INT-005: Marketplace ↔ Ethics & Observatory 45
XR-INT-006: Full Workflow 46
7 Conclusions 49
8 References 50

List of Figures

Figure 1 - Architectural design of the XR4ED solution 14

List of Tables

Table 1 - Test Case design 17
Table 2 – Load and Render 3D World 20
Table 3 – Verify the object interaction persistence 21
Table 4 – Asset upload metadata validation 22
Table 5 – Validate the license enforcement 22
Table 6 – Role-based Access to Private / Restricted Assets 23
Table 7 - Avatar Skeleton Calibration and Scaling 24
Table 8 - Verify Full-Body IK Solve Stability 25
Table 9 - Motion Matching Locomotion Integration 26
Table 10 – Verify the avatar latency performance 27
Table 11 – CI/CD Deployment 27
Table 12 – Verify the automatic rollback on failed deployment 28
Table 13 – Create 3D Object 30
Table 14 – Load & Upload 3D Object 31
Table 15 - Text-to-Voice Quality Verification 32
Table 16 - Text-to-Image Generation Accuracy 33
Table 17 – Publish the asset to Marketplace 34
Table 18 – Orchestrator (Backend of the Asset Tools) verification 35
Table 19 – XR Immersive Interaction Validation 36
Table 20 - Lite XR Web Responsiveness 37
Table 21 – Collaborative classroom interaction 38
Table 22 – GDPR consent logging 39
Table 23 - Data Subject Access and Export 39
Table 24 - Integration flow between XR Authoring Tool and Marketplace 41
Table 25 - Validating the licence issuance after a purchase in Marketplace 42
Table 26 - Integration flow between DevOps and Marketplace 43
Table 27 - Integration flow between Authoring Tool and Avatars 44
Table 28 - Integration flow between Marketplace and Ethics & Observatory 45

Table 29 - Verify full end-to-end workflow.....46

List of Acronyms

Acronym	Description
AI	Artificial Intelligence
API	Application Programming Interface
CI/CD	Continuous integration and continuous delivery/deployment
CFG	Classifier-Free Guidance
DoA	Description of Action
FPS	Frames Per Second
GA	Grant Agreement
GDPR	General Data Protection Regulation
ID	Identifier
KPI	Key Performance Indicator
PBR	Physically Based Rendering
UX	User Experience
TC	Test Case
UI	User Interface
WP	Work Package
XR	Extended Reality
XR4ED	Extended Reality for Education

1 Executive summary

The current deliverable corresponding to *Task 4.4 - Technical Verification and Integration Testing* addresses the most relevant aspects related to the testing and verification work in XR4ED.

The task intends to ensure that the XR4ED platform with the major components – XR Authoring Tool, Marketplace, Avatars and Asset Library, DevOps Environment, Ethics and Privacy Observatory, Asset Tools and Interfaces & UX – is verified against functional, technical and ethical requirements derived from earlier analysis and architectural design (D3.3, D4.1, D4.2 and D4.3).

The document consolidates all testing activities from component-level validation to full platform integration, ensuring that the XR4ED platform performs according to Description of Action (DoA), project objectives and user requirements.

The testing activities followed the recognized IEEE 829 and ISO/IEC/IEEE 29119 standards for software / systems testing. They comprise structured Test Cases (TC) addressing component functionality, usability, interoperability and compliance with ethical and accessibility principles. The proposed Testing plan, focused on integration, formalizes the testing methodology, defines the integration environment and pathway, and establishes a traceable relationship between system requirements, test cases and KPIs.

The integration testing phase confirmed the successful interaction among components, ensuring stable communication across APIs, services and user interfaces, and validating functional coherence, data consistency and end-to-end workflow reliability.

The results of integration testing demonstrate that XR4ED platform operates as a coherent, mature and stable ecosystem, capable of supporting immersive, accessible, and ethically compliant learning experiences.

2 Introduction

2.1 Purpose of the document

The deliverable documents the testing and validation activities conducted within *Task 4.4 - Technical Verification and Integration Testing*, consolidating results from all components developed under WP4. It depicts the verification process ensuring functional accuracy, integration robustness and ethical compliance of the system.

The report consolidates the methodology, test cases definitions and integration validation outcomes that collectively confirm the readiness of XR4ED for pilot deployment and further educational validation in subsequent project phases.

2.2 Structure of the document

The document is structured as follows:

Section 1. Executive summary

Section 2. Introduction: Presents a summary about the scope of the deliverable and its structure.

Section 3. System Architecture Overview: Presents significant aspects concerning the architectural design of the XR4ED platform, being more focused on the integration approach and implementation.

Section 4. Testing methodology: Depicts the methodological approach of testing applied in XR4ED.

Section 5. Technical verification and testing by component: Presents how the testing activities are performed for the components of XR4ED platform.

Section 6. Integration testing: Presents the integration testing approach specific to XR4ED.

Section 7. Conclusions: Presents the conclusions about the technical verification and testing activities applied in XR4ED.

3 System Architecture Overview

This section provides relevant aspects about the architectural design of the XR4ED platform, reflecting the integration approach and implementation pathway.

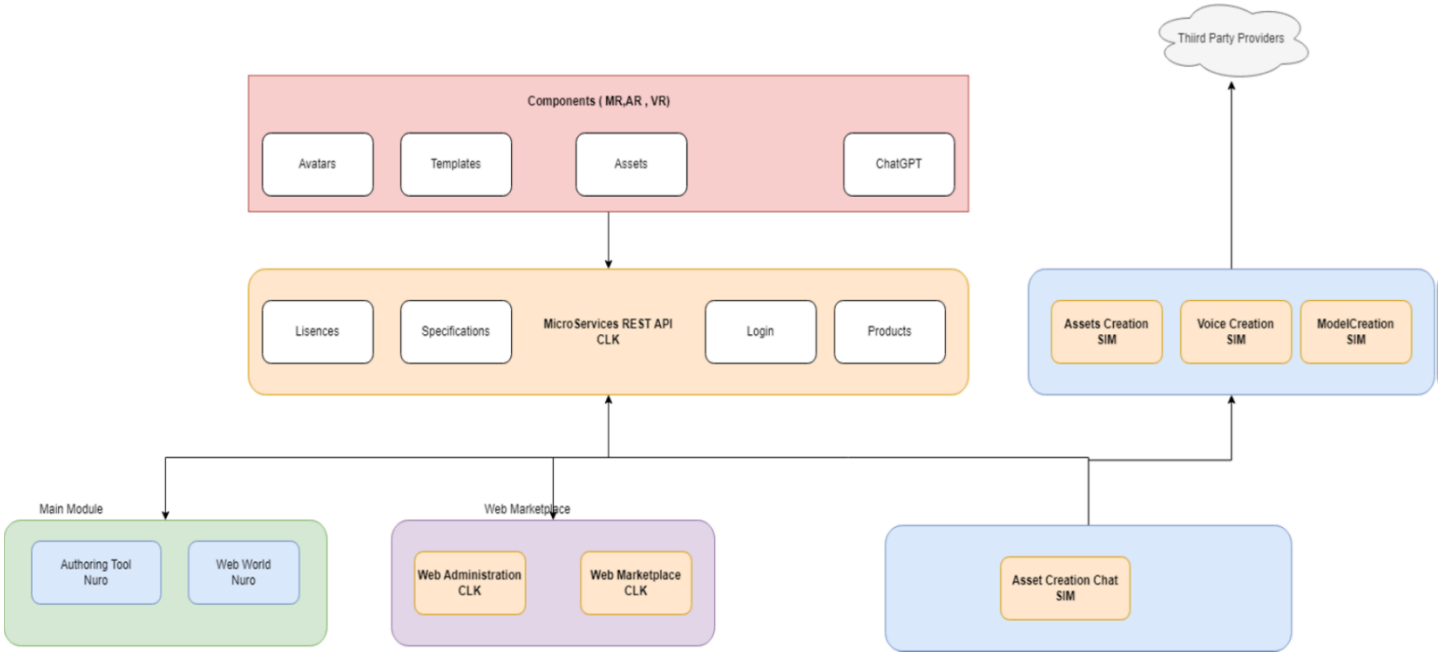

Figure 1 - Architectural design of the XR4ED solution

As described in Figure 1, the XR4ED solution has been conceived as a modular, service-oriented architecture designed to integrate all core functional and technical components necessary for immersive, ethical, and adaptive learning experiences.

From an operational perspective, the XR4ED architecture enables a complete learning workflow—from content creation in the XR Authoring Tool, to ethical validation via the Ethics and Privacy Observatory, to publication and reuse through the Marketplace, and finally to immersive delivery via the Interfaces & UX modules. The XR Authoring Tool provides trainers with intuitive authoring capabilities to design interactive XR lessons and environments, while the Marketplace acts as a centralized hub for asset sharing, licensing, and discovery. The Avatars and Asset Library ensure visual personalization and reusable digital resources that enhance engagement.

The architectural solution forms the backbone for integration testing described in Section 6, ensuring consistent performance, data integrity and compliance across the system.

Data flows within XR4ED are managed through a unified data model built around asset metadata, user profiles and licensing information. Key aspects are presented in the following:

- Metadata standardization: Adoption of a common XR metadata schemas ensures that assets remain interoperable between the Authoring Tool, Marketplace and Avatars.
- Event handling: Asynchronous messaging enables notifications and state synchronization (e.g. avatar updates, licence status).
- Storage and caching: Distributed storage supports low-latency retrieval of 3D models and AI-generated content, optimized through caching layers.
- Audit and traceability: Every data transaction (upload, purchase, consent) is logged within the Ethics & Privacy Observatory for verification and transparency.

The architectural design has been validated through iterative testing cycles under WP4, ensuring both functional and integration stability. The deployment model facilitates integration with external systems and potential third-party educational services, aligning with open standards for interoperability. The XR4ED architectural solution establishes a robust, flexible, secure, scalable and ethically compliant foundation capable of supporting large-scale educational deployments.

4 Testing methodology

This section addresses the relevant aspects concerning the testing stage in XR4ED: *testing approach*, *testing objectives*, *applicable standards*, *testing and verification methodology*, and *test cases (TC) design and specifications*. The major components of the *testing plan* are also depicted, addressing the testing pathway in XR4ED.

The well-defined testing process has different objectives during the various levels of the development of software or product, namely:

- Find defects or bugs at the developed code, either at unit or integration level
- Provide information concerning the level of quality of the developed software or product
- Ensure that the final result meets the defined business / project requirements
- Ensure that the final result satisfies the system specifications
- Provide a quality product to the end-user.

The most significant aspects of the verification approach are related to the testing methodology in line with the available standards. The testing process proposed for XR4ED, based on Agile methodology, and specifically the feature driven testing, considers both the methodological requirements of the project and the conceptual architectural design of the XR4ED platform.

The testing methodology combines internationally recognized standards IEEE 829 and ISO/IEC/IEEE 29119 with practical software quality assurance workflows tailored to the XR4ED architecture. The process has been structured into iterative stages: test planning, design, execution, validation and reporting. Each phase ensured traceability from system requirements and design specifications to measurable test outcomes. During the planning phase, test objectives were defined per component (XR Authoring Tool, XR Marketplace, XR Avatars and Asset Library, XR DevOps Environment, Asset Tools, Interfaces & UX, Ethics & Privacy Observatory), identifying success criteria and acceptance thresholds.

The design phase involved the creation of structured Test Cases that specify preconditions, detailed procedural steps, expected results and pass/fail criteria. These cases have been developed in compliance with IEEE 829 Test Case Specification templates to ensure consistency and reproducibility.

The methodology has also addressed interoperability, performance, security and ethical compliance testing. Usability and accessibility testing followed human-centred design principles to validate inclusiveness and learner experience quality.

Test design specification addresses the features to be tested. Using a standardized form in designing the test cases defines the basis of a common terminology and a consistent definition of processes, actions, requirements and results.

The proposed form for the Test Case, compliant with the above-mentioned standards and methodology, is presented in Table 1.

TC	***<Title of the test case>***
Test Case ID	
Objective	
Related requirement / deliverable	*<Related functional requirements or deliverables where relevant features are addressed>*
Preconditions	*<The setup needed for a test case to be executed successfully>*
Test environment (optional)	*<Hardware, software and network configuration needed>*

Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. ***<steps to be executed>*** 2. ***…*** 3. ***…***	*<Expected results>*	*<pass>, <fail>* *E.g.* *Pass if: Voice tone matches selected parameters.*

Table 1 - Test Case design

This form is used for both, the technical verification and testing by component and integration testing, as presented in the following.

Testing plan

The testing plan provides a structured and standards-aligned approach for verifying functionality, interoperability, performance and reliability across the integrated XR4ED platform.

The XR4ED platform integrates multiple modules – XR Authoring Tool, XR Marketplace, XR Avatars and Asset Library, XR DevOps Environment, Asset Tools, Interfaces & UX, Ethics & Privacy Observatory – each of them validated through component-level testing. The purpose of testing plan is to consolidate these verification activities into a uniform, coherent integration framework that ensures:

- Consistency with IEEE 829 and ISO/IEC/IEEE 29119 testing standards
- Full alignment with project KPIs and validation criteria defined in WP4
- Continuous improvement through integration of automated and manual tests within the DevOps pipeline.

The testing plan serves as the operational backbone for quality assurance during technical integration and validation of the XR4ED platform.

Objectives

The plan addresses cross-component interactions with the following objectives:

- Demonstrate seamless integration of the XR4ED components in end-to-end scenarios
- Detect versioning issues early in the development cycle
- Confirm that system KPIs are consistently achieved

- Establish traceability between component, integration, and validation tests.

Integration strategy

The integration strategy combines multiple complementary approaches:

- Contract Testing: Validates API schemas, authorization and error handling between connected services
- End-to-End Scenario Testing: Executes realistic user workflows
- Data-Flow Verification: Confirms metadata propagation and GDPR consent linkage through Ethics & Privacy logs
- Performance and Load Testing: Benchmarks system responsiveness under concurrent user conditions.

Test design and techniques

- Use-Case-Driven scenarios: Derived from project workflows and user stories
- Boundary and equivalence testing: Validates limits on asset size, metadata fields and concurrent sessions
- Security and privacy testing: Authorization checks, data minimization, consent enforcement
- Compatibility testing: Multi-device and browser verification for Web Interface & XR Client.

Defect management

Defining the severity levels:

- P1 – Blocking
- P2 – Major / critical deviation
- P3 – Minor / non-critical
- P4 – Cosmetic / UI only.

Define the entry and exit criteria for integration tests

Entry criteria:

- All dependent components deployed and reachable
- Test data defined, contracts published
- No open critical (P1) defects from component testing.

Exit criteria:

- All integration tests executed and passed (XR-INT-001, … 006)
- KPIs achieved:
  - Load time < 10 s (XR-FUNC-001)
  - Avatar latency < 100 ms (XR-AVT-004)
  - Licence validation 100 % success (XR-MKT-002)
- No unresolved P1/P2 defects.

Metrics and reporting

Define a specific pathway for metrics and reporting during the integration testing, as follows:

Metric	Description	Frequency
Pass Rate	% of passed integration tests	Each build
KPI Compliance	Load < 10 s; latency < 100 ms	Each build
Defect Density	Open defects / test cases	Weekly
Mean Time to Detect	Time from failure to detection	Continuous
GDPR Traceability	% of flows with valid consent log	Weekly

5 Technical verification and testing by component

This section presents the relevant testing and validation activities and specific Test Cases for each component of the XR4ED platform.

5.1 XR Authoring Tool

Responsible Partner: **CELLOCK**

This section presents the relevant aspects concerning the verification and validation activities reflected by appropriate Test cases corresponding to the XR Authoring Tool.

XR-FUNC-001: Load and Render 3D World

TC	*Verify the loading and rendering of 3D world*	
Test Case ID	XR-FUNC-001	
Objective	Validate the XR environment loads within 10 seconds with accurate textures.	
Related requirement / deliverable	D3.3	
Preconditions	XR World stored locally, VR device connected.	
Test environment (optional)	Integrated XR4ED platform (XR Authoring Tool, Marketplace, DevOps, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. Launch the XR Module. ***2. Select XR World.*** ***3. Verify the loading time.***	Load time < 10s; no texture errors.	Pass if environment loads fully and stably.

Table 2 – Load and Render 3D World

XR-FUNC-002: Object Interaction Persistence

TC	*Object interaction persistence*
Test Case ID	XR-FUNC-002
Objective	Verify object interactions persist after saving in World Builder.
Related requirement / deliverable	D3.3
Preconditions	Project opened in World Builder.

Test environment (optional)	Integrated XR4ED platform (XR Authoring Tool, Marketplace, DevOps, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. Move object. ***2. Save project.*** ***3. Reload.***	Position saved and restored.	Pass if saved configuration persists.

Table 3 – Verify the object interaction persistence

5.2 XR Marketplace

Responsible Partner: **CELLOCK**

This section presents the relevant aspects concerning the verification and validation activities reflected by appropriate Test cases corresponding to the XR Marketplace.

XR-MKT-001: Asset Upload Metadata Validation

TC	***Asset Upload Metadata Validation***
Test Case ID	XR-MKT-001
Objective	Ensure uploaded assets appear correctly in Marketplace.
Related requirement / deliverable	D4.2
Preconditions	User logged in; asset file ready.
Test environment (optional)	Integrated XR4ED platform (XR Authoring Tool, Marketplace, DevOps, Avatars, Ethics APIs)
Comments / Notes	

TC Steps (actions)	Obtained result	Pass / Fail Criteria
1. Upload asset. ***2. Add metadata.*** ***3. Verify listing in Marketplace.***	Metadata and previews displayed correctly.	Pass if upload successful.

Table 4 – Asset upload metadata validation

XR-MKT-002: License Enforcement

TC	*Validate the license enforcement*	
Test Case ID	XR-MKT-002	
Objective	Verify license key issued and validated after purchase.	
Related requirement / deliverable	D4.2	
Preconditions	Marketplace operational; Licensing API active.	
Test environment (optional)	Integrated XR4ED platform (XR Authoring Tool, Marketplace, DevOps, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. Purchase asset. ***2. Retrieve license.*** ***3. Validate use.***	License applied; unauthorized use blocked.	Pass if license validated.

Table 5 – Validate the license enforcement

XR-MKT-003: Role-based Access to Private / Restricted Assets

TC	*Role-based Access to Private / Restricted Assets*	
Test Case ID	XR-MKT-003	
Objective	Ensure that only authorised users can view/access restricted Marketplace assets.	
Related requirement / deliverable	D4.2	
Preconditions	Restricted asset exists; one authorised and one non-authorised user account.	
Test environment (optional)	Integrated XR4ED platform (XR Authoring Tool, Marketplace, DevOps, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. ***Log in as non-authorised user.*** 2. ***Search for restricted asset.*** 3. ***Attempt direct link access.*** 4. ***Log in as authorised user.*** 5. ***Search and open asset.*** 6. ***Attempt download / use.***	Non-authorised user cannot view or access; authorised user can.	Pass if access strictly enforced.

Table 6 – Role-based Access to Private / Restricted Assets

5.3 XR Avatars and Asset Library

Responsible Partner: **UPC**

This section presents the relevant aspects concerning the verification and validation activities reflected by appropriate Test cases corresponding to the XR Avatars and Asset Library.

XR-AVT-001: Avatar Skeleton Calibration and Scaling

TC	*Avatar Skeleton Calibration and Scaling*	
Test Case ID	XR-AVT-001	
Objective	Verify that the avatar's skeletal structure correctly resizes and aligns with the user's physical dimensions (height and arm span) to prevent "floating hands" or mismatched knees.	
Related requirement / deliverable		
Preconditions	User is wearing HMD and holding controllers; System is in "Calibration Mode".	
Test environment (optional)	VR Headset (HTC Vive/Oculus)	
Comments / Notes	*Based on the "SolveRoot" and calibration framework from the IK.*	
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. User stands in T-Pose. ***2. User initiates calibration sequence.*** ***3. System measures HMD height and Controller span.*** ***4. Avatar is generated.***	Avatar measures match user height; feet touch the ground; hands align with controllers.	Pass if: - Avatar height = User height (+/- 2cm). - No "floating feet" or bent knees when user is standing straight.

Table 7 - Avatar Skeleton Calibration and Scaling

XR-AVT-002: Verify Full-Body IK Solve Stability

TC	*Full-Body IK Solve Stability*	
Test Case ID	XR-AVT-002	
Objective	Validate that the IK solver naturally adjusts the spine and limbs without visual artifacts (twisting) when reaching for targets.	
Related requirement / deliverable		
Preconditions	Avatar successfully calibrated (XR-AVT-001 passed).	
Test environment (optional)		
Comments / Notes	*Tests the "Spine Twist" and "Analytic Solver" of the IK.*	
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. ***User reaches across their chest (left hand to right side).*** 2. ***User reaches high above head.*** 3. ***User rotates head 90 degrees while keeping body forward.***	Spine twists naturally toward the reach; Elbows do not invert; Head rotation does not instantly snap the torso orientation.	Pass if: - End-effectors (hands) remain attached to controllers. - No physical implausible poses (e.g., broken elbows).

Table 8 - Verify Full-Body IK Solve Stability

XR-AVT-003: Motion Matching Locomotion Integration

TC	*Motion Matching Locomotion Integration*
Test Case ID	XR-AVT-003
Objective	Verify that the avatar generates smooth, continuous locomotion animations (walking, turning, stopping) based on the user's trajectory derived from 3-point tracking.
Related requirement / deliverable	

Preconditions	User is calibrated; Room-scale VR setup active.	
Test environment (optional)		
Comments / Notes	*Validates the general responsiveness and quality of the Motion Matching system.*	
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. Walk: User walks forward at varying speeds. ***2. Turn: User changes walking direction and performs in-place turns.*** ***3. Stop: User comes to a sudden halt.***	The avatar's feet move naturally without significant sliding; transitions between walking, turning, and idling are smooth and blended.	Pass if: - Avatar trajectory matches user trajectory. - No "popping" or jerky animation artifacts during transitions.

Table 9 - Motion Matching Locomotion Integration

XR-AVT-004: Avatar Latency Performance

TC	***Avatar Latency Performance***	
Test Case ID	XR-AVT-004	
Objective	Verify that the end-to-end latency of the avatar movement is within acceptable limits to prevent motion sickness.	
Related requirement / deliverable		
Preconditions	Profiling tools enabled.	
Test environment (optional)		
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. Perform rapid hand movements.	Movement is responsive; no desynchronization.	Pass if: - Latency < 100 ms. - Frame rate maintains 90Hz (or target device native Hz).

2. *Measure photon-to-motion latency (or system processing time).*		

Table 10 – Verify the avatar latency performance

5.4 XR DevOps Environment

Responsible Partner: **CELLOCK**

This section presents the relevant aspects concerning the verification and validation activities reflected by appropriate Test cases corresponding to the XR DevOps Environment.

XR-DEV-001: CI/CD Deployment

TC	*CI/CD Deployment*	
Test Case ID	XR-DEV-001	
Objective	Validate CI/CD pipeline deploys new XR components automatically.	
Related requirement / deliverable	D3.3, D4.3	
Preconditions	Repo linked; CI/CD pipeline configured.	
Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. *Commit new build.* 2. *Monitor the pipeline.* 3. *Confirm Marketplace deployment.*	Build auto-deployed; status Ok.	Pass if build deployed successfully.

Table 11 – CI/CD Deployment

XR-DEV-002: Automatic Rollback on Failed Deployment

TC	*Automatic rollback on failed deployment*	
Test Case ID	XR-DEV-002	
Objective	Verify that the CI/CD pipeline detects deployment failures and automatically rolls back to the last stable XR4ED release.	
Related requirement / deliverable	D3.3, D4.3	
Preconditions	CI/CD configured; stable release available; rollback strategy implemented.	
Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. ***Prepare a faulty build.*** 2. ***Commit and push the change.*** 3. ***Monitor build, tests, deployment and health checks.*** 4. ***Observe reaction to failure.*** 5. ***Verify automatic rollback.*** 6. ***Check stability of system and logs.***	Pipeline triggers rollback; system returns to stable state with little / no downtime.	Pass if rollback is automatic and system stays stable.

Table 12 – Verify the automatic rollback on failed deployment

5.5 Asset Tools

Responsible Partner: **SMV**

This section presents the relevant aspects concerning the verification and validation activities reflected by appropriate Test cases corresponding to the Asset Tools.

XR-AST-001: Create 3D Object

TC	***Create 3D Object – UI workflow and Asset integrity***	
Test Case ID	XR-AST-001	
Objective	Validate that a user can create a 3D object via the Asset Tools and that geometry/ integrity is preserved in preview and export	
Related requirement / deliverable	D4.1, FR-Create 3D Object	
Preconditions	Authenticated user with Asset Tool access; Stable network connection, browser versions supported by Interfaces & UX layer; Sample template(s) available for 3D object creation	
Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. ***Open Asset Tools UI and navigate to 'Create 3D Object'.*** 2. ***Enter name, category and tags; select a creation template (if applicable).*** 3. ***Adjust parameters (dimensions, poly count target, material presets).***	- All UI inputs are validated (required fields, numeric ranges). - Progress indicator shows running/completed; no unhandled errors. - Preview renders without missing meshes or materials. - Export includes model + referenced textures; metadata (name, category, tags) is embedded or accompanies the package.	Pass if preview and export match on geometry/materials and UI validation behaves correctly.

4. *Click 'Generate/Build'; observe progress indicator and any logs.* 5. *When complete, open the 3D preview; rotate/zoom and inspect material channels.* 6. *Export the object as package (e.g., GLB/FBX + textures) and save locally.*		

Table 13 – Create 3D Object

XR-AST-002: Load & Upload 3D Object

TC	***Load & Upload 3D Object – Validation, metadata and error handling***	
Test Case ID	XR-AST-002	
Objective	Verify the UI supports loading an existing 3D object, validates dependencies, captures metadata and uploads successfully with robust error handling	
Related requirement / deliverable	D4.1, FR- Load & Upload 3D Object	
Preconditions	User authenticated, upload permission granted; Sample GLB/FBX with PBR textures; one variant missing intentionally a normal map for negative test.	
Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. *Open Asset Tools UI and navigate to*	- Validator reports dependencies; thumbnails generated; metadata fields pre-filled where possible.	Pass if valid asset uploads cleanly (no wrong metadata, no failures, no

'Load/Upload 3D Object'. 2. *Drag & drop the valid package; review auto-detected metadata and thumbnails.* 3. *Edit license, category and educational tags; save.* 4. *Upload and observe progress, retries and success confirmation.* 5. *Repeat with the package missing a normal map; observe validator warnings.* 6. *Try to upload a non-supported file to validate error messaging.*	- Successful uploads show a clear confirmation with asset ID/reference. - Unsupported file shows user-friendly message error with guidance; no crashes or dead ends. - Missing dependency triggers clear warning and allows fix-or-continue per policy.	required remediation) and invalid cases show correct warnings / errors without system instability.

Table 14 – Load & Upload 3D Object

XR-AST-003: Text-to-Voice Quality Verification

TC	*Text-to-Voice Quality Verification*	
Test Case ID	XR-AST-003	
Objective	Confirm that generated voice output aligns with text meaning and accuracy	
Related requirement / deliverable	D4.1, FR-Create voice asset	
Preconditions	Audio output enabled; Text input under 500 characters	
Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. Input text. ***2. Select voice profile and clarity parameters.*** ***3. Generate and playback voice sample.***	Voice output clear, natural, matches text meaning and tone settings.	Pass if expected behaviour observed without critical errors.

Table 15 - Text-to-Voice Quality Verification

XR-AST-004: Text-to-Image Generation Accuracy

TC	*Text-to-Image Generation Accuracy*	
Test Case ID	XR-AST-004	
Objective	Ensure that AI-generated images accurately represent input text prompts and style parameters	
Related requirement / deliverable	D4.1, FR-Create image asset	
Preconditions	User logged into Asset Tools; Stable internet connection.	
Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**

1. *Enter text prompt.* 2. *Adjust CFG scale and style parameters.* 3. *Generate and review image output.*	Generated image visually aligns with prompt description; Metadata reflects chosen parameters.	Pass if expected behaviour observed without critical errors.

Table 16 - Text-to-Image Generation Accuracy

XR-AST-005: Publish the asset to Marketplace

TC	*Publish to Marketplace – Listing, Licensing and Preview validation*	
Test Case ID	XR-AST-005	
Objective	Ensure assets uploaded from the Asset Tools are correctly published to the Marketplace with complete metadata, licensing and consistent previews	
Related requirement / deliverable	D4.1	
Preconditions	User logged into Asset Tools; At least one successfully uploaded asset into the platform Licensing API reachable; Marketplace account linked.	
Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. *From Asset Tools choose 'Publish to Marketplace' for the uploaded asset.* 2. *Select license type, price (if applicable), and*	- Marketplace listing contains accurate title, description, tags, license, and thumbnails. - Search index on Marketplace reflects the new listing; ranking/filters work on the new asset. - Preview in Marketplace visually matches Asset Tools preview.	Pass if listing accuracy and license settings match and the asset is discoverable with correct preview.

visibility (public/private). *3. Confirm publish and note the Marketplace listing URL/ID.* *4. Open Marketplace and search for the asset; verify listing and preview.* *5. Attempt to purchase/license (test account) to verify flow availability (do not finalize payment if not required).*		

Table 17 – Publish the asset to Marketplace

XR-AST-006: Orchestrator (Backend) of the Asset Tools Verification

TC	*Orchestrator verification – Metadata, Status & Audit Trail*
Test Case ID	XR-AST-006
Objective	Validate that published assets appear in the Orchestrator backend with correct status, metadata, and an auditable trail linking creation → upload → publish
Related requirement / deliverable	D4.1
Preconditions	Orchestrator access available; Asset published Audit logging enabled across components.

Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. ***Open Orchestrator backend and locate the asset by ID or title.*** 2. ***Verify lifecycle states (Created / Uploaded / Published) and timestamps.*** 3. ***Inspect metadata (author, license, size, hash, categories) and compare with Asset Tools/Marketplace.*** 4. ***Open audit trail/logs; confirm cross-component references (asset ID, user, operations).***	- Asset record exists with correct lifecycle states and timestamps. - Metadata fields match Asset Tools and Marketplace values. - Audit trail links user, operations, and component sources with unique IDs.	Pass if records are consistent and traceable across systems.

Table 18 – Orchestrator (Backend of the Asset Tools) verification

5.6 Interfaces & UX

Responsible Partner: **CELLOCK**

This section presents the relevant aspects concerning the verification and validation activities reflected by appropriate Test cases corresponding to the Interfaces & UX.

XR-UX-001: Heavy XR Immersive Interaction

TC	*XR Immersive Interaction Validation*	
Test Case ID	XR-UX-001	
Objective	Verify the immersive XR environments provide full motion tracking and stable feedback	
Related requirement / deliverable	D4.1	
Preconditions	Magic Leap or equivalent device calibrated.	
Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. Launch immersive XR classroom. ***2. Interact via hand gestures.*** ***3. Check synchronization with virtual objects.***	Motion tracking consistent; no desynchronization or lag observed.	Pass if expected behaviour observed without critical errors.

Table 19 – XR Immersive Interaction Validation

XR-UX-002: Lite XR Web Responsiveness

TC	*Lite XR Web Responsiveness*	
Test Case ID	XR-UX-002	
Objective	Validate the responsiveness and performance of the Lite XR (web) interface across devices	
Related requirement / deliverable	D4.1	
Preconditions	Lite XR environment active; browser cache cleared.	
Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. *Open Lite XR interface in desktop browser.* 2. *Interact with 3D scene.* 3. *Measure frame rate and input delay.*	Smooth interaction at >30 FPS, latency <100ms.	Pass if expected behaviour observed without critical errors.

Table 20 - Lite XR Web Responsiveness

XR-UX-003: Collaborative Classroom Interaction

TC	*Collaborative Classroom Interaction*
Test Case ID	XR-UX-003
Objective	Ensure multiple users can interact seamlessly in shared XR learning spaces
Related requirement / deliverable	D4.1
Preconditions	At least two active users connected; voice chat enabled.
Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)

Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. Join same XR classroom. ***2. Share 3D models and communicate.*** ***3. Observe latency and consistency.***	Synchronization <200ms; audio and gesture updates shared accurately.	Pass if expected behaviour observed without critical errors.

Table 21 – Collaborative classroom interaction

5.7 Ethics & Privacy Observatory

Responsible Partner: **CELLOCK**

This section presents the relevant aspects concerning the verification and validation activities reflected by appropriate Test cases corresponding to the Ethics & Privacy Observatory.

XR-ETH-001: GDPR Consent Logging

TC	***GDPR Consent Logging***	
Test Case ID	XR-ETH-001	
Objective	Ensure consent captured and timestamped correctly	
Related requirement / deliverable	D3.3	
Preconditions	User registers and accepts consent.	
Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. Create user. ***2. Accept consent.***	Consent entry with timestamp found.	Pass if entry exists.

3. Query log.		

Table 22 – GDPR consent logging

XR-ETH-002: Data Subject Access and Export

TC	*Data Subject Access and Export*	
Test Case ID	XR-ETH-002	
Objective	Verify that a user can request and receive a complete export of their personal data in line with GDPR	
Related requirement / deliverable	D3.3	
Preconditions	Test user with activity logs; export request mechanism available.	
Test environment (optional)	Integrated XR4ED platform (DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. Perform actions to generate personal data. *2. Submit data access / export request.* *3. Admin processes request.* *4. Generate data export package.* *5. Deliver export to user* *6. User verifies completeness and correctness.*	User receives complete, correct, readable export; logs show processing details.	Pass if export is correct, complete and secure.

Table 23 - Data Subject Access and Export

6 Integration testing

This section provides a structured set of integration test scenarios for the XR4ED solution. It focuses on verifying interactions and data exchanges between the major functional components of the platform.

Integration testing within XR4ED has been designed as an essential phase to validate the interoperability, functional coherence, performance, and stability of the platform as a unified system. Following the principles established in Section 4. Testing Methodology, integration testing ensured that the various software components - XR Authoring Tool, XR Marketplace, XR Avatars and Asset Library, XR DevOps Environment, Asset Tools, Interfaces & UX, Ethics & Privacy Observatory, work seamlessly together across shared interfaces and communication layers.

The testing process followed an incremental approach, beginning with interface validation between individual pairs of modules and progressing toward the full workflow verification. The tests built upon the component-level verification and focused on data exchange, API consistency and end-to-end user flows.

Each integration scenario has been designed according to IEEE 829 and ISO/IEC/IEEE 29119 standards, including defined preconditions, controlled input data, expected outputs, and clear pass/fail criteria.

Performance testing verified that module interaction remained stable under concurrent usage conditions. These activities collectively confirmed that the XR4ED components are technically and functionally integrated, and the whole platform is operational for use in the educational field.

The integration phase covers six test scenarios (XR-INT-001 … XR-INT-006) described in the following.

XR-INT-001: XR Authoring Tool ↔ Marketplace

TC	***Integration flow between XR Authoring Tool and Marketplace***	
Test Case ID	XR-INT-001	
Objective	Ensure the XR Authoring Tool correctly transfers metadata and files to the Marketplace API for public access.	
Related requirement / deliverable	D3.3, D4.2	
Preconditions	User logged in; XR World created in XR Authoring Tool; Marketplace API reachable	
Test environment (optional)	Integrated XR4ED platform (XR Authoring Tool, DevOps, Marketplace, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**

1. *From XR Module select "Publish to Marketplace".* 2. *Enter metadata (title, category, tags).* 3. *Confirm upload* 4. *Verify listing on Marketplace.*	The XR World is listed in Marketplace with correct metadata, preview image and download link.	Pass if the Marketplace reflects new asset; upload <10s; no data mismatch.

Table 24 - Integration flow between XR Authoring Tool and Marketplace

XR-INT-002: Marketplace ↔ Licensing

TC	*Validating the licence issuance after a purchase in Marketplace*	
Test Case ID	XR-INT-002	
Objective	Ensure that when a user purchases an XR component, a valid license is issued and validated at runtime.	
Related requirement / deliverable	D4.2	
Preconditions	Marketplace running; Licensing microservice deployed; payment simulation enabled.	
Test environment (optional)	Integrated XR4ED platform (XR Authoring Tool, Marketplace, DevOps, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. ***Purchase XR component.*** 2. ***System requests license from Licensing API.*** 3. ***Download and launch XR component.*** 4. ***Attempt launch without valid license (negative test).***	License is generated and validated. Unlicensed attempt returns access denied.	Pass if the License key is generated, stored and verified successfully.

Table 25 - Validating the licence issuance after a purchase in Marketplace

XR-INT-003: DevOps ↔ Marketplace

TC	*Integration flow between DevOps and Marketplace*	
Test Case ID	XR-INT-003	
Objective	Verify automated deployment of new XR component through DevOps CI/CD pipeline.	
Related requirement / deliverable	D3.3	
Preconditions	GitHub repository linked; CI/CD configured in DevOps; valid API tokens.	
Test environment (optional)	Integrated XR4ED platform (XR Authoring Tool, Marketplace, DevOps, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. ***Commit new XR component code.*** 2. ***Trigger DevOps build.*** 3. ***Verify build and artifact.*** 4. ***Confirm upload to Marketplace staging.***	Build pipeline executes successfully; component visible in Marketplace.	Pass if the build <5min.; no failed dependencies; API returns 200 “OK”.

Table 26 - Integration flow between DevOps and Marketplace

XR-INT-004: Avatars ↔ Authoring Tool

TC	*Integration flow between Authoring Tool and Avatars*	
Test Case ID	XR-INT-004	
Objective	Validate avatar synchronization across multiple users in a shared XR World.	
Related requirement / deliverable	D3.3	
Preconditions	More than 2 users connected via Portal Hopper in Authoring Tool; Avatars API enabled.	
Test environment (optional)	Integrated XR4ED platform (XR Authoring Tool, Marketplace, DevOps, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. *Users join session.* 2. *Perform gestures and movement.* 3. *Observe sync latency.* 4. *Validate audio sync.*	Avatars sync <100ms latency; audio synchronized; no desync or crash.	Pass if 95% packet sync rate; latency<200 ms.

Table 27 - Integration flow between Authoring Tool and Avatars

XR-INT-005: Marketplace ↔ Ethics & Observatory

TC	*Integration flow between Marketplace and Ethics & Observatory*	
Test Case ID	XR-INT-005	
Objective	Confirm user consent and privacy preferences flow from Marketplace to Ethics & Observatory.	
Related requirement / deliverable	D4.2	
Preconditions	User account created; Privacy API active.	
Test environment (optional)	Integrated XR4ED platform (XR Authoring Tool, Marketplace, DevOps, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
1. ***Register on Marketplace.*** 2. ***Accept data-sharing policy.*** 3. ***Privacy service logs consent.*** 4. ***Request access from Ethics & Observatory .***	Consent timestamp recorded; user can view, modify or revoke consent; audit log generated.	Pass if GDPR compliance is validated; consent retrievable and editable.

Table 28 - Integration flow between Marketplace and Ethics & Observatory

XR-INT-006: Full Workflow

TC	***Verify full end-to-end workflow from creation to deployment and user consumption***	
Test Case ID	XR-INT-006	
Objective	Ensure seamless interoperability across all major XR4ED components.	
Related requirement / deliverable	D3.3, D4.2	
Preconditions	All modules deployed (DevOps, XR Module, Marketplace, Licensing, Avatars).	
Test environment (optional)	Integrated XR4ED platform (XR Authoring Tool, Marketplace, DevOps, Avatars, Ethics APIs)	
Comments / Notes		
TC Steps (actions)	**Obtained result**	**Pass / Fail Criteria**
7. Developer builds XR World in XR Module. ***8. Publishes via DevOps.*** ***9. User purchases via Marketplace.*** ***10. Launches session via Portal Hopper.*** ***11. Verify avatars, privacy, performance.***	Full workflow executed without manual intervention; no errors or missing assets.	Pass if no API errors; latency < thresholds; GDPR compliance validated.

Table 29 - Verify full end-to-end workflow

Lessons learned

The integration process revealed several key insights that guided the system refinement and collaboration across partners:

- Continuous integration from early phases reduced the time to detect and solve interface mismatches
- Schema validation and semantic versioning between Marketplace and Authoring Tool avoided compatibility issues
- Synthetic test data improved GDPR assurance
- Cross-team coordination (weekly verification and defect-review sessions) enhanced quality outcomes by rapid feedback loop and full traceability.

Technical KPIs derived from the XR4ED implementation related to the Integration test scenarios

KPI Reference	Description	Target	Associated Integration Test(s)	Verification Method	Status
KPI-1	System responsiveness – XR environment load time	≤ 10 s	XR-INT-001, XR-INT-006	Load time measured during end-to-end workflow execution; benchmark compared with baseline	✅ Achieved
KPI-2	Avatar synchronization latency	≤ 100 ms	XR-INT-004, XR-INT-006	Real-time telemetry comparison of avatar movements across clients	✅ Achieved

Funded by the European Union

KPI-3	Marketplace transaction reliability	≥ 99% successful transactions	XR-INT-001, XR-INT-002, XR-INT-006	Purchase, license issuance, and validation flow tested for 100+ transactions	✅ Achieved
KPI-4	Data privacy compliance (GDPR)	100% of user actions logged with consent	XR-INT-005, XR-INT-006	Verification of consent entries in Ethics & Privacy logs	✅ Achieved
KPI-5	Deployment automation success rate (CI/CD)	≥ 95% successful builds	XR-INT-003	CI/CD logs reviewed per pipeline execution failed deployment auto-rolled back	✅ Achieved
KPI-6	Cross-platform UX stability	≥ 30 FPS sustained	XR-INT-006	FPS and latency metrics recorded during browser and device tests	✅ Achieved

Each KPI is covered by at least one dedicated integration test scenario and validated under operational conditions. XR-INT-006 (End-to-End Workflow) serves as the global validation, confirming multi-component compliance across all KPIs.

7 Conclusions

This deliverable provides a comprehensive verification of the XR4ED platform from a technical perspective. The integration of components has been successfully achieved, with functional, performance, and security criteria met according to project objectives. It is recommended that future iterations maintain continuous testing cycles and monitoring to ensure system resilience and scalability as new modules are introduced.

8 References

Deliverable D3.3 – XR4ED Architecture and XR Modules specification

Deliverable D4.1 – Interfaces / Components and UX of XR4ED platform

Deliverable D4.2 – Marketplace - 1

Deliverable D4.3 – XR4ED platform prototypes

DoA – Description of Action of XR4ED

IEEE 829-2008 - IEEE Standard for Software and System Test Documentation, https://standards.ieee.org/standard/829-2008.html

ISO/IEC/IEEE 29119-5:2024 - ISO/IEC/IEEE International Standard - Software and systems engineering - Software testing - Part 5: Keyword-Driven Testing, https://www.iso.org/standard/87233.html