

An Open, Cross-Platform, Web-Based Metaverse Using WebXR and A-Frame

Giuseppe Macario  

Universitas Mercatorum
Piazza Mattei 10, Rome, Italy

Ministry of Defense
Via XX Settembre 8, Rome, Italy

August 27, 2024

Abstract

The metaverse has received much attention in the literature and industry in the last few years, but the lack of an open and cross-platform architecture has led to many distinct metaverses that cannot communicate with each other. This work proposes a WebXR-based cross-platform architecture for developing spatial web apps using the A-Frame and Networked-Aframe frameworks with a view to an open and interoperable metaverse, accessible from both the web and extended reality devices. A prototype was implemented and evaluated, supporting the capability of the technology stack to enable immersive experiences across different platforms and devices. Positive feedback on ease of use of the immersive environment further corroborates the proposed approach, underscoring its effectiveness in facilitating engaging and interactive virtual spaces. By adhering to principles of interoperability and inclusivity, it lives up to Tim Berners-Lee’s vision of the World Wide Web as an open platform that transcends geographical and technical boundaries.

Keywords: Metaverse, Virtual Worlds, WebXR, Spatial Computing, Extended Reality, Open Standards, World Wide Web, Browsers.

1 Introduction

The advent of extended reality (XR)—namely augmented reality (AR), mixed reality (MR), and virtual reality (VR)—together with a renewed interest in

the metaverse after Facebook Inc. changed its name to Meta Platforms Inc., introduces not only unprecedented opportunities for the digital landscape but also significant challenges due to the considerable variation in capabilities between XR devices and conventional computing platforms such as smartphones, tablets, and desktop computers. Where once the primary considerations were screen size and input method (e.g. touch versus mouse and keyboard), developers now must navigate a complex array of sensory inputs, immersive environments and spatial interactions. Therefore, while the web expands to encompass these new immersive experiences, it has become necessary to find the right balance between embracing the unique features of each platform and ensuring universal accessibility. Not only would this allow the web to harness the full potential of XR technologies, but creators could also deliver rich and immersive experiences while preserving the user’s freedom to choose their preferred platform and interface.

Tim Berners-Lee, inventor of the World Wide Web, wanted his creature to be an “open platform that allows anyone to share information, access opportunities and collaborate across geographical boundaries” (Solon, 2017). To achieve this result, CERN (2024) states that the web should be based on open standards, avoiding proprietary systems. The web indeed owes much of its success to the adoption of standards that allow content to be enjoyed by a wide audience, regardless of the browser or the operating system. Due to this inclusive and flexible nature, the web

has already become a major platform for consuming two-dimensional content.

On the other hand, the journey of the web towards a universally accessible platform has not been without its challenges. Instances where proprietary technologies were widely adopted, included but not limited to the ones developed by Microsoft in the 1990s (e.g. ActiveX), exemplify the potential pitfalls of deviating from open standards. This approach led to a significant portion of web content being optimized exclusively for specific browsers, notably Internet Explorer for Windows, thereby limiting accessibility across different platforms. Similarly, the advent of mobile computing devices, including smartphones and tablets, initially posed significant obstacles for web usability due to the vast differences in screen sizes and interaction models compared to traditional desktop computers. However, the web community’s commitment to inclusivity and universal access spurred the evolution of standards to overcome these challenges: for example, CSS emerged as a fundamental solution, enabling web content to dynamically adapt to a highly diverse array of devices.

In the growing area of virtual reality and digital interconnectedness, the vision of an open metaverse—a vast, shared, and interoperable virtual space—has captivated the imagination of technologists, creators, and users alike. This work sets out to propose a cross-platform architecture designed to build such an open metaverse, delivered through the World Wide Web. Central to this exploration is the following research question:

RQ. *How can a WebXR-based cross-platform architecture facilitate the development of an open, accessible, and interoperable metaverse, and what are the implications for user engagement within this digital ecosystem?*

The inquiry probes into the viability of leveraging WebXR technology, alongside the integration of the A-Frame and Networked-Aframe frameworks, which enhance the WebXR ecosystem by simplifying the creation of interactive 3D content and enabling real-time, multiplayer experiences across various devices.

2 Background

2.1 Metaverse

When the American writer Neal Stephenson first described a virtual environment named Metaverse in a 1990s sci-fi novel, little did he know that the product of his imagination would soon become reality: Active Worlds (1995) and Second Life (2003) are the first two popular actual forms of Stephenson’s Metaverse.¹ Since then, the general public has familiarized with the concept of metaverse in information technology, and several computer scientists have given various definitions, for instance Ritterbusch and Teichmann (2023):

Definition 1 (Metaverse). *A (decentralized) three-dimensional online environment that is persistent and immersive, in which users represented by avatars can participate socially and economically with each other in a creative and collaborative manner, in virtual spaces decoupled from the real physical world.*

Updating a prior definition of virtual world by Bell (2008), the metaverse can also be defined more concisely as

Definition 2 (Metaverse). *A persistent synchronous virtual environment, shared by people represented as avatars, facilitated by networked devices.*

These definitions do not explicitly mention XR/VR/MR/AR² nor do they require interoperability between different metaverses, which makes sense because most implementations, especially the older ones (including the aforementioned Active Worlds and Second Life), do not have XR capabilities and are incompatible with each other. On the other hand, after 30 years since the inception of the metaverse, technology is mature enough to handle these new demands: this is where WebXR and A-Frame come in.

¹The Metaverse (with an initial capital letter) is the proper noun of the environment described by Neal Stephenson; in this paper, the metaverse (common noun in lower case) will denote any actual implementation in computer science, distinct from Stephenson’s fictional Metaverse.

²From now on, XR/VR/MR/AR will be abbreviated to XR.

2.2 WebXR

The W3C Immersive Web Working Group has been established to address the challenges of crafting a standard named WebXR Device API; its stated mission is “to help bring high-performance virtual reality (VR) and augmented reality (AR), collectively known as extended reality (XR), via APIs to interact with XR devices and sensors *in browsers*” (W3C, 2024). As a matter of fact, the centrality of browsers as gateways to digital information has remained a constant since the inception of the World Wide Web: recognizing this key role, the World Wide Web Consortium (W3C) has made the strategic decision to leverage the browser as a foundational platform for spatial web apps (see section 4) that render immersive environments. WebXR spatial web apps can run on any device equipped with a modern browser, enabling a wide audience to engage with 3D environments without necessarily resorting to specialized software or hardware.

The WebXR Device API is designed to offer a unified, platform-agnostic abstraction layer that grants access through the web to the fundamental features shared by all XR devices. By abstracting the complexities of underlying hardware, the WebXR Device API enable web developers to access a broad spectrum of interaction controllers through a streamlined interface while simplifying the real-time rendering of immersive environments.

The foundational layer for graphical rendering in WebXR applications is WebGL, i.e. a JavaScript API that leverages the capabilities of OpenGL for Embedded Systems (OpenGL ES) to render interactive 2D and 3D graphics directly within web browsers. WebGL provides a direct access to the GPU through a low-level interface, enabling high-performance rendering of 3D graphics and animations. In order to abstract away the complexities of direct WebGL programming, WebXR applications typically use a higher-level open-source framework or library, to choose from

Babylon.js Open-source framework (Apache license) by Microsoft and other contributors,

Three.js Open-source framework (MIT license) by Ricardo Cabello and other contributors,

A-Frame Open-source framework (MIT license) by Supermedium (formerly Mozilla VR) and other

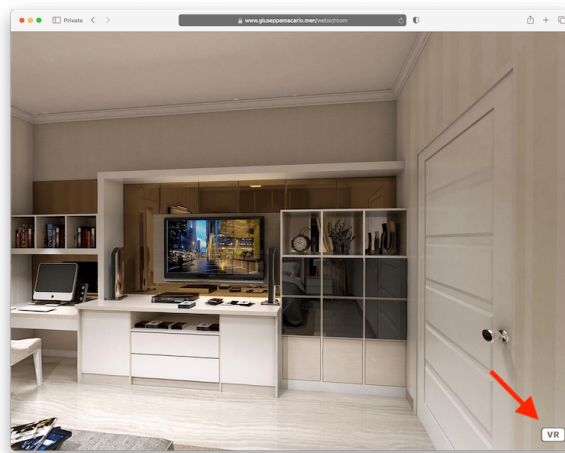


Figure 1: VR button in the lower right corner of an A-Frame scene

contributors.

A constantly updated list of browsers supporting WebXR is available online³ and, at the time of writing (2024), it shows that the latest versions of Chrome, Edge, Opera, and Samsung Internet, fully support the API. Firefox and Safari also support the API—not yet on iOS⁴—although it may have to be manually enabled by the user. Consequently, all the most popular XR headsets also support WebXR, through their built-in browsers: Meta Quest, Microsoft HoloLens, Apple Vision Pro, HTC Vive, Samsung Gear, Google Cardboard and others. Also, desktop and mobile browsers support WebXR; however, since desktops, laptops, smartphones and smart TVs are not specifically designed for XR, the browser is forced to graphically simulate the immersive features of WebXR, depending on the screen and the platform. This is why A-Frame⁵ apps can show a button in the lower right corner (fig. 1) that enables users to transition from a traditional 2D web environment into an immersive 3D experience.⁶ This critical interface element, which

³<https://caniuse.com/webxr>

⁴In the meantime, WebXR is supported by other iOS apps such as Mozilla’s WebXR Viewer.

⁵See section 2.3.

⁶In fig. 1, the button is labeled as “VR”, but other imple-

provides users with a more engaging and interactive experience, can be displayed on any browser supporting WebXR. Depending on the needs of the developer, its behavior can be customized; in the case of desktop browsers, it usually toggles full screen mode.

2.3 A-Frame

Three.js is an open-source WebGL-based cross-browser JavaScript library and API used for 3D graphics within web browsers. By abstracting the Three.js syntax into an HTML-like syntax through the entity-component-system (ECS) architectural pattern (Wang et al., 2020), the A-Frame framework makes WebGL/Three.js applications much easier and faster to code. The ECS is a popular and desirable pattern in 3D and game development, which follows the composition over inheritance and hierarchy principle (Supermedium, 2024).

A-Frame provides the entire WebXR Device API, and is therefore suitable for XR devices. Through a web browser, the resulting applications run on common XR systems as well as conventional computers and mobile devices, thus reaching the largest possible number of users without the need for additional hardware or software.

In accordance with the ECS pattern, an A-Frame component is a modular and reusable data container that adds appearance and/or behavior to an entity. Therefore, an A-Frame-based virtual world is basically a combination of A-Frame entities and components, which can optionally load 3D models and multimedia assets such as images, sounds and videos.

2.4 Networked-Aframe

Since A-Frame is primarily geared towards single-user experiences, the Networked-Aframe framework builds upon A-Frame by introducing networked components that enable real-time, multi-user experiences within the same virtual space, mirroring the dynamics of physical users’ interactions. Networked-Aframe communicates via WebRTC, a peer-to-peer API, or alternatively via WebSockets, a client-server API.

mentations could use different labels.

3 Previous work

According to Lombardi and Lombardi (2010), the metaverse should

1. be entirely open and non-proprietary,
2. include a virtual-world browser similar to a web browser,
3. provide hyperlinks for traversing the virtual world.

The authors implemented these recommendations in Open Cobalt, an open-source distributed platform designed for creating decentralized 3D virtual worlds and written in the Squeak programming language (a derivative of Smalltalk). However, Open Cobalt has not been actively maintained in recent years, and its usage has declined as other platforms and technologies for virtual reality and 3D interaction, like Unity or Unreal Engine, have become more popular and supported.

Kim et al. (2022) observed that the accessibility of the metaverse is hindered by the requirement to install specific software or to use dedicated devices. Therefore, they developed a web-based metaverse that users can access simply by typing a URL into the address bar of their browser. The graphics and physics of their metaverse are powered by the Unity game engine, which is proprietary software.

Havele et al. (2022) argue that a truly comprehensive Metaverse should extend beyond isolated “micro-verses” or restrictive “walled gardens”. This paradigm shift underscores a move from earlier, more fragmented virtual experiences to a more integrated and accessible virtual ecosystem. A “Unified Metaverse”, they contend, would facilitate smooth transitions between different virtual environments without exiting a three-dimensional state, supported by robust interoperability standards. The authors propose that such a metaverse would naturally evolve from the current two-dimensional web, suggesting that it could emerge through enhanced web connectivity and the implementation of open web standards. They highlight X3D as a royalty-free ISO/IEC open standard for 3D computer graphics and interactive applications on the web, developed and maintained by the Web3D Consortium (2024). It serves as a successor to the Virtual Reality Modeling Language (VRML) and provides a

framework for representing a wide range of 3D graphics elements such as rendering, animation, and user interaction. X3D applets run within web browsers, using OpenGL technology to display 3D content across different browsers and operating systems. However, X3D has not achieved widespread acceptance, which may be due to the learning curve associated with its implementation and use.

Novel contribution This paper introduces an approach that aligns with the foundational principles set forth by Lombardi and Lombardi (2010) for an open, non-proprietary virtual world, enhanced with a modern browser capable of navigating through interconnected spaces via hyperlinks. The proposed solution leverages current web technologies to ensure widespread compatibility and usability across most contemporary browsers and devices. By addressing the challenges of accessibility and interoperability highlighted in Kim et al. (2022) and Havele et al. (2022), this approach seeks to foster an open and integrated metaverse environment.

4 A “Hello world” spatial web app

The academic introduction of the term spatial computing is attributed to Greenwold (2003):

Definition 3 (Spatial computing). *Human interaction with a machine in which the machine retains and manipulates referents to real objects and spaces.*

According to Apple (2023), “Featuring visionOS, the world’s first spatial operating system, Vision Pro lets users interact with digital content in a way that feels like it is physically present in their space” and “With more than 2,000 spatial apps designed for Apple Vision Pro [...] visionOS 2 enables developers to take further advantage of spatial computing” Apple (2024b): in other words, spatial computing and spatial apps need a spatial operating system (visionOS, in this case).

On the other hand, in section 2.2, we introduced the concept of “spatial web apps”. By spatial web



Figure 2: “Hello world” in visionOS and WebXR

apps, we mean spatial apps that do not necessarily have to run on a spatial operating system such as visionOS. To run a spatial web app, users can simply open a URL in any browser; therefore, unlike Apple’s spatial apps, spatial web apps can run everywhere.

Although no installation is required, a spatial web app can still be installed using features such as Progressive Web Apps (PWA) and “Add to Home Screen”, which allows mobile websites to open like native apps, bypassing the Apple or Google App Stores. An app delivered as a regular web page will never need to be manually updated by the user because it is sufficient for the developer to update the web page. Moreover, the app automatically benefits from browser features such as history and tabs, which are already well known to the general public.

Apple (2024a) provides an example of “Hello world” in visionOS, i.e. a spatial app that shows a floating globe (fig. 2); this downloadable Xcode project contains 180 files, for a total of about 250 MB of source code. However, Apple Vision Pro also supports WebXR through Safari for visionOS, which allows the developer to create a spatial web app without using Xcode and avoiding Apple’s review process. Instead of compiling a project of 180 files, WebXR does the same thing in very few lines of A-Frame code—even just one line—within one HTML document:

```
<body>
  <a—scene>
```

```

<a-sphere src="url(https://example.com/texture.jpg)"
  position="0 1.5 -5" radius="1" rotation="0 0 -30"
  animation="property: rotation; to: 0 360 -30;
            loop: true; dur: 10000; easing: linear">
</a-sphere>
</a-scene>
</body>

```

In general, a spatial web app consists of at least an HTML document with JavaScript, possibly including textures and 3D models. A minimal valid HTML page with the necessary code is available online,⁷ where devices can also test the app.

Overall, this example demonstrates some of the advantages of a WebXR spatial web app over a native visionOS spatial app. Firstly, the app can run on any device, not just on Apple Vision Pro. Secondly, the source code of an HTML document is extremely light compared to a large Xcode project and does not need to be compiled. Furthermore, the spatial web app does not have to be distributed through an app store. These advantages also apply to other non-Apple headsets.

5 Conceptual architecture

The conceptual architecture, as illustrated in fig. 3, represents a layered approach to developing spatial web applications. Each layer builds upon the foundational technologies:

JavaScript Serves as the primary programming language.

WebGL (OpenGL ES) Provides a low-level graphics API for rendering 2D and 3D graphics in the browser.

WebXR Device API Facilitates direct interaction with XR hardware.

Three.js Acts as a lightweight 3D library that builds on top of WebGL, simplifying 3D content creation.

A-Frame Provides a web framework for building VR experiences, making Three.js accessible to web developers without deep knowledge of WebGL.

Networked-Aframe Extends A-Frame for multi-player experiences, enabling the development of a shared virtual environment.

⁷<https://www.giuseppemacario.men/webxr/hello-world>

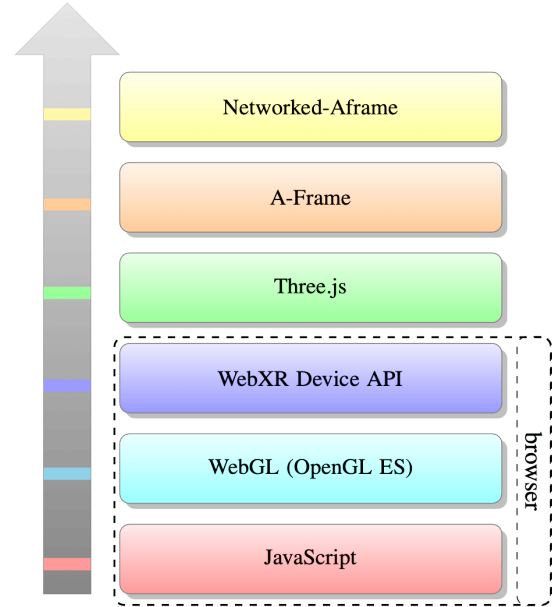


Figure 3: Technology stack of a WebXR/A-frame spatial web app (abstraction layers in ascending order)

As seen in section 2.2, the first three components are natively supported by web browsers; therefore, in fig. 3, they are encapsulated within a “browser” environment, indicating the stack’s capability to deliver immersive experiences directly through a web browser without the need for external applications.

6 Comparison between Second Life and the proposed open metaverse

Existing metaverse platforms can be represented by Second Life, which remains the oldest and most popular virtual world directly inspired by Neal Stephenson’s concept of the Metaverse. It shares several technical features with the envisioned metaverse, such as customizable avatars (Linden Research, 2024a; Stephenson, 1992) and regions sized as multiples of 256, specifically $256\text{ m} \times 256\text{ m} = 65,536\text{ m}^2$ each (Linden Research, 2024c; Stephenson, 1992). The pio-

neering role of Second Life in virtual worlds is unquestionable, but it exhibits several limitations when compared to the open metaverse proposed in this paper. The following points highlight several key areas where the new metaverse offers significant improvements; Second Life is used as a benchmark, but each point is applicable to other major virtual worlds as well.

Access to other virtual worlds Second Life confines users within its closed ecosystem, offering no direct interaction with other virtual worlds. In contrast, the proposed metaverse promises potentially unlimited access to a diverse array of virtual environments in a very simple way, namely through HTML links. To connect to a different virtual environment, one simply needs to click on a hyperlink, and the new environment will load in the current browser window or in a new window.⁸

Supported platforms Second Life clients (also known as *viewers*) run on Windows, macOS and Linux, while the proposed metaverse can be accessed with any web browser, significantly expanding the range of supported devices, including mobile ones.

In-world electronic currency Second Life operates with a proprietary in-world currency, which is limited to transactions within its platform. The new metaverse model, however, supports the use of various electronic currencies, including both fiat and cryptocurrencies, thereby significantly broadening the economic flexibility for users.

Protocol and server Second Life uses a proprietary protocol designed by Linden Lab; only a portion of it has been made public (Linden Research, 2024b), although a reasonably compatible open-source protocol named libOpenMetaverse has been made available by reverse-engineering Linden Lab’s protocol (Open Metaverse Foundation, 2024). The primary server process in Second Life is called a *simulator*, which simulates one region. Each CPU core is dedicated to

one server process; therefore, a host⁹ can execute four or more server processes at once (Linden Research, 2024d) depending on the number of CPU cores it has. On the other hand, the proposed metaverse is designed to function on any web server using HTTPS, which simplifies deployment and reduces costs for developers who wish to host and manage their virtual spaces.

Costs for virtual land/space Owning virtual land in Second Life involves subscription fees set by Linden Lab, while the open architecture of the proposed metaverse allows any website to host a virtual space, enabling users to create and own virtual spaces either for free or at significantly reduced costs.

Encryption The security measures in Second Life are relatively limited, particularly concerning data encryption. The client encrypts user authentication (username and password) (Linden Research, 2024e) but everything else is transmitted over unencrypted UDP packets. On the other hand, the proposed metaverse uses HTTPS for all data exchanges. This standard practice for Internet safety greatly enhances the protection of user information and interactions, offering a more secure and trustworthy virtual environment.

7 Evaluation

7.1 Method

Performance tests were conducted using Cloudflare’s infrastructure, which provides detailed logs and analytics to trace each request and response through the network. These tests were complemented by Google PageSpeed Insights¹⁰. Static assets were distributed through Cloudflare’s Content Delivery Network (CDN), reducing load times and latency by caching assets at edge servers. This strategy minimizes latency by conducting computations geographically closer to the user, a critical factor for the immersive experiences offered by the virtual environment.

⁸Depending on the presence of the attribute `target="_blank"`.

⁹By metonymy, the term *server* can also refer to a host.

¹⁰<https://pagespeed.web.dev>

To ensure the reproducibility of the tests, all experiments were performed using the default settings of Cloudflare’s free plan, without any modifications. The complete source code of the spatial web apps and the deployment scripts are provided in the Italian Ministry of Defense’s repository¹¹, accessible through a personal account available for free upon request.

7.2 User experience survey

A user experience survey was conducted with 100 volunteer participants from the Multipurpose Research Center of the Ministry of Defense to gather qualitative feedback on the platform. The survey consisted of 10 questions designed to evaluate various aspects of the platform’s usability and effectiveness. Responses were collected using a Likert scale from 1 to 5, with additional space for general feedback. Analyzing these responses helped refine the platform’s design, focusing on user-driven improvements.

7.2.1 Survey questions

1. How would you rate the ease of navigation within the virtual environment? (1 Very difficult, 5 Very easy)
2. How would you rate the visual quality of the virtual environment? (1 Very poor, 5 Excellent)
3. How satisfied are you with the response time of the virtual environment? (1 Very unsatisfied, 5 Very satisfied)
4. How easy was it to access the virtual environment from different devices? (1 Very difficult, 5 Very easy)
5. How responsive did you find the multiplayer interactions? (1 Very unresponsive, 5 Very responsive)
6. How well do you think the virtual environment supports collaboration and interaction between users? (1 Not well, 5 Very well)
7. How secure did you feel using the virtual environment? (1 Not secure, 5 Very secure)
8. How satisfied are you with the overall user experience provided by the virtual environment? (1 Very unsatisfied, 5 Very satisfied)

9. How likely are you to recommend the virtual environment to others? (1 Not likely, 5 Highly likely)
10. Do you think this virtual environment is better or worse than proprietary virtual environments you already know? (1 Much worse, 5 Much better)
11. Optional: Please provide any additional feedback or suggestions.

7.3 Results

7.3.1 Performance metrics

The following metrics provide a quantitative foundation for assessing the prototype’s performance, highlighting its strengths in handling complex scenes and real-time interactions, albeit with noted areas for improvement.

Load time Load times were measured across various devices, including five desktop computers (both high-performance and standard models) and five mobile devices (from high-end to mid-tier smartphones), from the initiation of the application to the full rendering of the 3D scene. The Largest Contentful Paint (LCP), a crucial user-centric metric for perceived loading speed, was evaluated using Google PageSpeed Insights. On average, the LCP consistently remained below 2.5 seconds for both desktop and mobile platforms, ranking above the 75th percentile. This performance level indicates superior efficiency in content delivery and optimization, well-aligned with current web standards.

Frame rate The frame rate was monitored using Chrome DevTools on a suite of five desktops and five mobile devices to ensure a smooth user experience across various user densities and scene complexities. The application consistently maintained a frame rate of over 60 FPS on both desktop and mobile platforms, often reaching up to 120 FPS. Despite this high performance, noticeable dips occurred when the user count exceeded 20, indicating areas where further optimization is necessary.

¹¹<https://virtual.difesa.it>

Network latency (ping) Network latency was assessed to evaluate the responsiveness of multiplayer interactions. The delay revealed an average latency of 120 milliseconds, which is below the threshold that might disrupt real-time interactions. However, occasional spikes up to 300 milliseconds were observed during peak server load times.

7.3.2 Survey results

Ease of navigation 85% of participants rated navigation as easy or very easy.

Visual quality 88% rated the visual quality as high or very high.

Response time 79% are satisfied or very satisfied with the response time.

Cross-device accessibility 91% consider that accessing the platform from different devices was easy or very easy.

Multiplayer responsiveness 82% found the multiplayer interactions to be responsive or very responsive.

Collaboration and interaction between users 77% found the virtual environment to be collaborative.

Security 83% found the virtual environment to be secure or very secure.

Overall satisfaction 87% are satisfied or very satisfied with the user experience.

Recommendation 85% are likely or very likely to recommend the platform to others.

Comparison with proprietary virtual worlds 92% found this virtual environment to be better or much better than proprietary virtual environments they know.

Additional feedback Common suggestions included enhancing avatar customization options and improving performance on smart TVs.

8 Discussion

The evaluation showed that the prototype is robust, supporting immersive virtual spaces across multiple devices while maintaining a high quality of user experience. This underscores the practical viability of the

conceptual architecture and highlights its potential as a foundational framework for developing open metaverse environments. Technical performance was generally strong, though there is room for improvement in handling complex scenes with high user density. Users gave positive feedback about the ease of use and the immersive quality of the environment.

While the prototype showcased the architecture’s capabilities, several limitations emerged. Insights into cross-platform accessibility indicate a need for further refinement to optimize the experience on slower devices (smart TVs). Performance issues under high-density conditions highlighted the need for optimization, particularly in terms of resource management and scene complexity. Network latency, although acceptable, remains a concern for real-time interactions, suggesting that further improvements in synchronization and data transmission are necessary.

The insights gained from this research suggest various directions for future work:

Optimization Dynamic content loading and rendering optimizations could enhance performance, especially on constrained hardware. These strategies may involve more efficient use of resources and improved algorithms for real-time processing.

User interface (UI) and User experience (UX) design Exploring innovative UI/UX designs could further improve usability and accessibility, making virtual spaces more intuitive and engaging for a broader audience. This could include the integration of adaptive interfaces that adjust to user preferences and needs.

Multiuser interactions Investigating advanced techniques for networked interactions and data synchronization could enhance the realism and fluidity of multiplayer experiences. This might involve the development of new protocols or the refinement of existing ones to reduce latency and improve the consistency of shared virtual spaces.

Scalability Examining architectural modifications or extensions to support larger, more complex environ-

ments without compromising performance would be valuable. Potential approaches could include the use of distributed computing architectures or cloud-based rendering solutions.

Future features of WebXR Looking ahead, the WebXR Device API is going to expand its functionality to include advanced features that will further enrich immersive web experiences. These anticipated enhancements include the implementation of world anchors, which would allow digital content to be consistently positioned within the physical world, and hit-testing capabilities, enabling interaction between virtual objects and the real-world environment detected by platform sensors. Additionally, the API is expected to facilitate the exposure of world structure, offering developers a deeper understanding of the physical space around the user through data from platform sensors.

9 Conclusion

This research aimed to address the persistent fragmentation in the metaverse by proposing a WebXR-based cross-platform architecture using the A-Frame and Networked-Aframe frameworks. The development and evaluation of a prototype demonstrated this architecture’s ability to facilitate engaging and interactive virtual spaces, available across various platforms and devices, reflecting a significant step toward an open, interoperable metaverse. The architecture’s reliance on open web standards supports Tim Berners-Lee’s vision for an accessible and universal web, and ensures that the metaverse remains an inclusive digital ecosystem, free from the constraints of proprietary technologies.

The prototype was thoroughly evaluated, receiving positive feedback particularly for its ease of use and the immersive quality of the environment. This feedback underscores the practical viability of the proposed solution for creating a user-friendly metaverse. Performance tests further validated the robustness of the architecture, with data from Cloudflare’s infrastructure and Google PageSpeed Insights contributing to a better understanding of network efficiency and

responsiveness. The application maintained excellent load times and frame rates, important metrics for immersive experiences. Additionally, the implementation of the HTTPS protocol for security aligns with best practices for protecting user data, reinforcing the architecture’s suitability for widespread adoption.

Looking forward, further research will be essential to explore advanced features of WebXR and other emerging web technologies that could enrich the metaverse experience. Implementing such technologies could potentially expand the scope of interactive and collaborative possibilities within the metaverse, which moves us closer to a truly interconnected digital world that interfaces with the physical world through extended reality.

References

- Apple. 2023. Introducing Apple Vision Pro: Apple’s first spatial computer. <https://www.apple.com/newsroom/2023/06/introducing-apple-vision-pro/>
- Apple. 2024a. Hello World: Use windows, volumes, and immersive spaces to teach people about the Earth. <https://developer.apple.com/documentation/visionos/world>
- Apple. 2024b. visionOS 2 brings new spatial computing experiences to Apple Vision Pro. <https://www.apple.com/jo/newsroom/2024/06/visionos-2-brings-new-spatial-computing-experiences-to-apple-vision-pro/>
- Mark W. Bell. 2008. Toward a Definition of Virtual Worlds. *Journal of Virtual Worlds Research* 1, 1 (2008). <https://jvwr-ojs-utexas.tdl.org/jvwr/article/view/283/237>
- CERN. 2024. A short history of the Web. <https://www.home.cern/science/computing/birth-web/short-history-web>
- Simon Greenwold. 2003. *In Form*. Master’s thesis. MIT. <https://www.media.mit.edu/publications/in-form/>
- Anita Havele, Nicholas Polys, William Benman, and Donald Brutzman. 2022. The Keys to an Open,

- Interoperable Metaverse. In *Proceedings of the 27th International Conference on 3D Web Technology*. ACM, Article 9, 7 pages. <https://doi.org/10.1145/3564533.3564575>
- Junmyeong Kim, Changhyeon Jeong, and Kanghyun Jo. 2022. Development of Web-based Metaverse Platform. In *International Workshop on Intelligent Systems*. <https://doi.org/10.1109/IWIS56333.2022.9920930>
- Linden Research. 2024a. Controlling your avatar’s appearance. <https://community.secondlife.com/knowledgebase/english/controlling-your-avatars-appearance-r216/>
- Linden Research. 2024b. Protocol. <https://wiki.secondlife.com/wiki/Protocol>
- Linden Research. 2024c. Region. <https://wiki.secondlife.com/wiki/Land#Region>
- Linden Research. 2024d. Server architecture. https://wiki.secondlife.com/wiki/Server_architecture
- Linden Research. 2024e. Technical overview of Second Life security. https://wiki.secondlife.com/wiki/Linden_Lab_Official:Technical_overview_of_Second_Life_security
- Julian Lombardi and Marilyn Lombardi. 2010. *Opening the Metaverse*. Springer, 111–122. https://doi.org/10.1007/978-1-84882-825-4_9
- Open Metaverse Foundation. 2024. libOpenMetaverse Library Quick Start. <https://github.com/openmetaversefoundation/libopenmetaverse>
- Georg David Ritterbusch and Malte Rolf Teichmann. 2023. Defining the Metaverse: A Systematic Literature Review. *IEEE Access* 11 (2023), 12368–12377. <https://doi.org/10.1109/ACCESS.2023.3241809>
- Olivia Solon. 2017. Tim Berners-Lee on the future of the web: “The system is failing”. <https://www.theguardian.com/technology/2017/nov/15/tim-berners-lee-world-wide-web-net-neutrality>
- Neil Stephenson. 1992. *Snow Crash*. Bantam Books.
- Supermedium. 2024. Entity-Component-System. <https://aframe.io/docs/1.6.0/introduction/entity-component-system.html>
- W3C. 2024. Immersive Web Working Group Charter. <https://w3c.github.io/immersive-web-wg-charter/immersive-web-wg-charter>
- Sheng-Ming Wang, Yu-Chen Wang, Wei-Jie Pan, and Cheng-Yen Lin. 2020. Developing Innovative Cross-platform Web-based Virtual Reality Services. In *International Conference on Consumer Electronics (ICCE)*. IEEE. <https://doi.org/10.1109/ICCE-Taiwan49838.2020.9258343>
- Web3D Consortium. 2024. Recommended Standards. <https://www.web3d.org/standards>