



Metaverse
STANDARDS FORUM™

Open Standards for the Metaverse

The Case for Developing a Native Metaverse Browser

This document was developed by RP1, the creators of the first working metaverse browser prototype and the primary architects and maintainers of the Open Metaverse Browser Initiative (OMBI). RP1's hands-on experience designing, building, and operating a global metaverse browser prototype provides the technical foundation for the concepts, architecture, and standards described herein.

The OMBI is an open project under the Metaverse Standards Forum. Its purpose is to inspire and coordinate the development of needed open standards, and to build an open-source implementation of a metaverse browser. The goal is to enable any organization to use or build compatible metaverse browsers to deploy spatial services on infrastructure they own and control. The initiative is open to all contributors and is not owned by any single company.

2026.Q1

Executive Summary

The Moment We're In

The computing industry is approaching a fundamental transition as significant as the shift from text-based terminals to graphical user interfaces, or from desktop computing to mobile devices. As augmented reality (AR) glasses and mixed reality headsets become commercially viable consumer products, the infrastructure that delivers applications and services to these devices will need to evolve dramatically.

Just as the web browser unlocked the World Wide Web by providing universal, standards-based access to information, the metaverse browser will unlock the metaverse. This is not speculation — it's the natural evolution of how we interact with digital content. The decisions made today will determine whether it becomes an open ecosystem or fragments into proprietary silos.

What is the Metaverse?

The metaverse is an evolution of the World Wide Web. Where the web provides a portal to access internet content on 2D devices, the metaverse provides a portal to the spatial internet, solving for proximity-based content delivery across augmented reality and virtual environments.

A metaverse browser serves the same fundamental purpose for the metaverse that a web browser serves for the World Wide Web: it provides universal, standards-based access to services without requiring users to install separate applications for each service they wish to use. Just as web browsers eliminated the need to install different software for email, banking, shopping, and entertainment, metaverse browsers will enable users to access spatial services on-demand across any compatible device.

Unlike the web, where you manually navigate to one website at a time, the metaverse is proximity-based, meaning that you see the content that is within proximity to your current location. Your view incorporates services from potentially hundreds of independent providers simultaneously, with content appearing and disappearing automatically based on location and context.

This means that for AR glasses, the metaverse browser will connect you to location-based content and services owned by physical locations near you. When you approach an airport, you instantly connect to its spatial fabric and interact with real-time services — AI assistants, IoT devices, wayfinding, operational tools. The same locations can be accessed remotely through desktop or VR, enabling anyone to join and interact with real-world spaces from anywhere.

Why Open Standards Are Non-Negotiable

The XR industry today resembles online services in the early 1990s — fragmented into incompatible proprietary platforms. We are in the AOL moment. Just as walled gardens gave way to the open web when standards emerged, today's proprietary XR platforms cannot compete with an open metaverse built on universal standards.

Enterprises cannot deploy critical infrastructure on proprietary platforms. Recent failures — Microsoft discontinuing HoloLens, 8th Wall being deprecated — demonstrate the concrete risks. When a hospital integrates its spatial computing into surgery systems, or a warehouse bases its logistics on AR-guided operations, platform discontinuation means operational disruption, compliance issues, and safety concerns.

Five non-negotiable requirements define an open metaverse:

- 1. Interoperability:** Services built to standards work in any compliant metaverse browser. Content created in standard formats is universally usable without conversion.
- 2. Device Independence:** AR glasses, VR headsets, desktop, and mobile — all access the same spatial services. No device lock-in, no manufacturer monopolies.
- 3. Ownership:** Users own their data, identity, and content. Service providers own customer relationships and revenue streams without platform tolls.
- 4. Frictionless Access:** Browse-to-enter model eliminates app installation. Services stream on-demand, just like websites.
- 5. Open Tooling:** Anyone can create spatial content using standard tools and deploy without permission from platform operators.

Unlike proprietary platforms that lock users into a single vendor's ecosystem, an open metaverse — built on standardized protocols — would allow any developer to create services, any company to deploy infrastructure, and any user to access spatial experiences across devices from different manufacturers.

Real-World Applications Across Industries

Today, businesses run almost entirely on web infrastructure — communications, human resources, logistics, inventory, collaboration, presentation, documentation, sales, and artificial intelligence operate through web-based systems accessible via web browsers. Tomorrow, as spatial computing devices proliferate, these business functions will need to operate in three-dimensional, spatially-aware metaverse infrastructure delivered through the metaverse browser.

Enterprises: Factory workers see real-time inventory overlaid on shelves, assembly instructions on components, safety alerts spatially positioned. Warehouse workers follow optimized picking routes displayed as floor markers. Maintenance technicians access repair procedures overlaid on equipment with remote expert support.

Healthcare: Surgeons access patient records, drug interactions, and AI guidance spatially positioned during procedures. Medical training happens in photorealistic simulations. Remote specialists collaborate in shared virtual operating rooms.

Retail: Customers see product information, reviews, and availability overlaid on physical items. Virtual try-on shows furniture in homes or clothing fit. Checkout happens through spatial interfaces without physical lines.

Transportation: Airport passengers access check-in, security navigation, gate finding, and concierge services spatially. Staff coordinate baggage, food service, and operations through integrated AR systems.

Education: Universities create campus digital twins for remote tours. Students conduct experiments in virtual laboratories. Distributed classrooms enable spatial presence and 3D content collaboration.

These are not speculative scenarios. Major enterprises want to deploy XR technology across operations but cannot because appropriate open infrastructure does not exist. They remain stuck in proof-of-concept purgatory, unable to justify dependence on proprietary platforms they don't control.

The Technical Foundation

Metaverse browsers are universal clients that manage spatial presence while connecting users to multiple concurrent spatial services securely and performantly. They implement and utilize spatial standards (e.g., Scene Object Model, glTF, SMAP protocols, OpenXR) to render experiences from any spatial fabric — just as web browsers implement web standards to render pages from any web server.

Spatial fabrics are the server-side infrastructure defining 3D coordinate spaces and managing user presence within them. They provide map services, presence tracking, identity management, avatars, communication, and domain-specific services. Like web servers, anyone can operate spatial fabrics.

The architecture enables composition. A city operates a primary spatial fabric providing geography and presence management. Individual businesses operate secondary fabrics providing specific services, attaching at their physical locations.

The Path Forward

The web succeeded because it was built on open standards (HTML, HTTP, CSS, JavaScript) that anyone could implement without permission or licensing fees. This openness created the largest software ecosystem in history. The metaverse requires the same approach: open standards that enable competition, innovation, and user choice while preventing fragmentation into incompatible walled gardens.

Standardization is happening now across multiple standards organizations, including: Khronos Group (glTF, OpenXR, ANARI), W3C (decentralized identifiers, accessibility), OGC (GeoPose for spatial positioning), with many others. The Metaverse Standards Forum provides a neutral venue for multiple organizations and the industry to coordinate and align. The collaboration by the Metaverse Standards Forum and RPI represents the community-driven effort required to succeed.

How to Participate

Join the Metaverse Standards Forum: Participate in the Open Metaverse Browser Initiative (OMBI) open-source implementation project to help develop the spatial standards of the metaverse. [Click here for information on joining the Metaverse Standards Forum.](#)

Support through sponsorship: Building a native metaverse browser requires significant funding to build a qualified team to complete this project. Organizations interested in sponsoring this critical open source project can [contact the Metaverse Standards Forum.](#)

Keeping the Metaverse Open

The metaverse will be built. The question is whether it becomes an open ecosystem enabling unlimited innovation or fragments into proprietary silos limiting participation. The web succeeded because it was open — that openness resulted from deliberate choices by people who valued openness over control.

We are making that same choice now. The invitation is open to everyone who recognizes the importance of this work and wants to contribute to building infrastructure that belongs to the world, not to any single company.

This document examines the architectural principles and standardization efforts necessary to build an open, interoperable metaverse. It establishes the conceptual foundation by defining what the metaverse is, why it requires fundamentally different architecture than the current web, what problems currently prevent its emergence, and how the technical primitives of web infrastructure map to their spatial equivalents.

Table of Contents

Executive Summary	i
Table of Contents	v
1. The Metaverse	1
2. Defining the Metaverse	5
3. Current Problems: Where the Industry Is Today	7
4. Web Infrastructure to Metaverse Infrastructure	10
5. Five Non-Negotiables for an Open Metaverse	14
6. Introduction to the Metaverse Browser	20
7. Spatial Fabrics and Metaverse Servers	24
8. Use Cases: Cross-Industry Applications	27
9. Join the Effort: Building Open Standards for the Metaverse	32

1. The Metaverse

What is the Metaverse

Unlike the World Wide Web, which organizes information as pages accessed by typed URLs, the metaverse organizes services and content in three-dimensional coordinate spaces that can represent real-world locations, fictional environments, or abstract spatial structures.

A user accessing the metaverse does not manually navigate to services by typing addresses. Instead, services become available automatically based on proximity — both physical proximity in the real world (determined by GPS coordinates or visual positioning systems) and virtual proximity in digital spaces (determined by a user's position within a 3D coordinate system).

Why We Need the Metaverse

AR glasses and mixed reality devices cannot effectively use web infrastructure designed for 2D screens. Web browsers were architected for a model where users view one website at a time in a rectangular window. Users manually navigate between sites, each site loads in isolation, and interaction occurs through 2D input devices like keyboards and mice.

Spatial computing devices operate differently. A user wearing AR glasses in a physical space might simultaneously interact with dozens of services: navigation overlays, product information displays, communications interfaces, AI assistants, task management tools, and context-specific applications — all visible at once and organized spatially around the user. This is not a use case the web browser was designed to handle.

What the Metaverse Will Do

The open metaverse will need to provide a way for businesses to deploy the same types of services they currently run on the web, but delivered in spatially-aware, three-dimensional formats optimized for AR and VR devices. Case studies help illustrate this:

Retail (Stores)

- Employees would use spatial systems for shipping/receiving, warehousing, inventory management, training, product demonstrations, checkout, timekeeping, and HR functions
- Customers would interact with AI concierges, product locators, product information systems, specials displays, checkout interfaces, food court menus, and could be joined remotely by other people

- Corporate would operate virtual campuses with security check-in, conferences, training, remote work capabilities, guest services, all-hands meetings, and HR systems

Transportation (Airports)

- Terminal staff would access AI training, ticketing systems, baggage handling, TSA interfaces, cleaning coordination, food management, back office systems, and HR
- Travelers would use parking/transportation coordination, check-in, security navigation, gate finders, boarding systems, shopping/food interfaces, and AI concierges
- Tarmac operations would coordinate baggage handling, food restocking, fuel supply, ground traffic, air traffic control, lighting, security, and maintenance

Healthcare (Hospitals)

- Doctors and nurses would access scheduling, patient records, drug interaction databases, patient monitoring, surgery assistance, AI guidance, fall prevention, and virtual consultation
- Patients and visitors would use room finders, admissions, information systems, appointment scheduling, multimedia, room controls, and remote visitation
- Administration would manage time clocks, food restocking, cleaning compliance, records, security, employee scheduling, conference rooms, and maintenance

These are not speculative scenarios. Major enterprises want to deploy XR technology across their operations but are currently unable to do so because appropriate infrastructure does not exist.

Why the Metaverse Needs Open Standards

Without open standards, the metaverse will fragment into incompatible walled gardens controlled by individual platform vendors. This fragmentation creates several critical problems:

Security and Compliance: Enterprises cannot deploy their infrastructure on platforms they don't control because they cannot secure customer data, protect intellectual property, or comply with international, federal, state, and local regulations.

Vendor Dependency: Companies cannot build businesses dependent on platforms that may have conflicting brands, different customer experiences, device restrictions, or that may cease operations. The failures of platforms like HoloLens and 8th Wall demonstrate this risk (discussed in Section 4b).

Platform Restrictions: Closed platforms impose restrictions on sales transactions, take percentage cuts of revenue, limit device compatibility, and control what applications can be deployed.

Interoperability: Users need to access services across different devices — desktop computers, mobile phones, AR glasses, VR headsets — at varying levels of immersion. This requires services to be built once and accessible everywhere, not reimplemented for each proprietary platform.

The World Wide Web succeeded because no single company controlled it. Any developer could create a website, any company could run web servers, and any user could access web content through standards-compliant browsers from different vendors. The metaverse requires the same openness to achieve comparable scale and utility.

The Metaverse as a Collection of Services

A critical distinction between the web and the metaverse is scale and simultaneity. The web is fundamentally serial — you browse one website at a time. Even when you have multiple browser tabs or apps open, each one shows content from a single origin, and switching between them is a deliberate action.

Whereas the metaverse is fundamentally parallel and proximity-based. A user's view incorporates services from potentially hundreds of independent providers simultaneously, with services appearing and disappearing automatically based on location and context.

Consider a user walking through a shopping district wearing AR glasses. Their view might include:

- A navigation service showing directions to their destination
- A weather service displaying current conditions
- Multiple retail stores, each providing their own product information and promotions
- Restaurant services showing menus and wait times
- A payment service for transactions
- A messaging service for communications
- An AI concierge answering questions
- A calendar service showing appointments
- News services displaying updates
- Games or entertainment services

Each of these is an independent service, potentially operated by different companies, running on different infrastructure. Yet they all need to appear integrated in the user's spatial view, with clear security boundaries preventing one service from accessing another's data without permission.

The metaverse must support millions of such services — from global providers like payment processors and AI assistants, to local services like individual store inventories, to personal services like custom productivity tools. The total number of spatial services will eventually exceed the number of websites on the World Wide Web, because the metaverse includes not just information

services but also services representing physical objects, locations, and real-time interactions with the physical world.

This scale is why open standards are non-negotiable. No single company can build, maintain, and operate millions of diverse services. The ecosystem must be open for the same reason the web is open: to enable unlimited participation by developers, businesses, and creators worldwide.

2. Defining the Metaverse

The term "metaverse" requires precise definition because it has been used to describe everything from video games to virtual worlds to blockchain schemes. In the context of this document and the broader metaverse standardization effort, the metaverse refers specifically to shared, persistent, three-dimensional spaces accessed through metaverse browsers that support both AR and VR experiences.

A "spatial fabric" is the core building block of the metaverse in the same way that a website is the core building block of the World Wide Web. Metaverse browsers connect to spatial fabrics that provide a suite of services for a particular location, including a map of the objects and services within that location, and potentially other services that allow people and avatars to communicate with one another.

A "metaverse server" is analogous to a web server. The term server generically refers to a hardware computer system used to host services that can be connected to over a network. A metaverse server, in particular, is a computer system — both hardware and software — that stores and delivers metaverse content over a network using the protocols of the metaverse. Just like web servers, metaverse servers that host spatial fabrics can be operated by anyone.

AR Glasses with Six Degrees of Freedom (6DOF)

Six degrees of freedom means tracking both rotational movement (pitch, yaw, roll) and translational movement (forward/back, left/right, up/down) in three-dimensional space. This is the fundamental requirement for convincing augmented reality: the system must know not just where you're looking but where you are and how you're moving through space.

Modern AR glasses use visual positioning systems (VPS) that combine camera inputs, depth sensors, and sometimes GPS to determine the user's position in the world with centimeter-level precision. This positioning enables digital content to be "anchored" to specific locations in physical space — a product information display that appears to sit on a physical table, a navigation arrow that appears to be painted on the actual floor, a remote participant whose avatar appears to stand in a specific location in the room.

6DOF tracking is what distinguishes true spatial computing from earlier attempts at augmented reality. When a digital object is anchored in space, it stays in that location as you move around it, walk closer to inspect it, or turn your head to look at it from different angles. This spatial persistence is essential for productive work applications, not just entertainment.

Virtual Spaces

While AR glasses overlay digital content on the physical world, VR headsets create entirely virtual environments. The metaverse encompasses both. Virtual spaces in the metaverse are not fundamentally different from augmented views of real spaces — both are three-dimensional coordinate systems where users have spatial presence and can interact with services and content.

A virtual office space, for example, might contain the same productivity services as a physical office: document collaboration tools, communication systems, presentation displays, scheduling interfaces, and AI assistants. The difference is that the coordinate system represents a fictional space rather than GPS coordinates on Earth.

Virtual spaces are particularly important for:

- **Remote Collaboration:** Allowing distributed teams to meet in shared virtual environments
- **Training and Simulation:** Creating safe, repeatable environments for learning dangerous or expensive procedures
- **Entertainment:** Building fictional worlds for games, social experiences, and creative expression
- **Design and Prototyping:** Reviewing 3D models and designs at full scale before physical production

AR and VR Together: Unified Experiences

The critical insight is that AR and VR are not separate technologies requiring separate infrastructure — they are different interfaces to the same metaverse. A metaverse browser must support both seamlessly.

Consider a design review meeting. Some participants might be physically present in a conference room, wearing AR glasses that display a 3D prototype floating above the conference table. Other participants might join remotely through VR headsets, appearing as avatars standing around the same virtual prototype in the same virtual conference room. The VR participants see avatars of the AR participants; the AR participants see avatars of the VR participants. Everyone manipulates the same shared 3D model. The underlying infrastructure is identical for both AR and VR participants.

This is fundamentally different from how phone calls, video conferencing, and chat rooms work on the web. For example, video calls require separate software (FaceTime, Zoom, Teams, Meet). In the metaverse, presence and communication are foundational services that any spatial experience can incorporate through standard protocols.

3. Current Problems: Where the Industry Is Today

The Walled Garden Trap

The XR industry today resembles the online services industry of the early 1990s: fragmented into incompatible proprietary platforms, each requiring separate applications and preventing interoperability. Meta's Horizon platform, Apple's visionOS ecosystem, and various enterprise platforms each function as closed systems. Applications built for one platform don't run on others. Content created for one system can't be easily moved to another. Users can't seamlessly interact across platforms.

This fragmentation has predictable consequences:

Development Costs: Developers must build the same application multiple times for different platforms, dramatically increasing costs and reducing the pool of available applications.

Limited Reach: Each platform has a relatively small user base, making it difficult for developers to achieve scale or for users to find critical mass in social experiences.

Platform Risk: Businesses that build on a proprietary platform are entirely dependent on that platform's continued operation and favorable terms.

Device Lock-in: Platforms require specific hardware to access their service, making it impossible to interact with users tied in to another hardware platform.

User Lock-in: Users who invest in content, connections, or configurations on one platform face high switching costs, reducing competition and choice.

These are not new problems. The computing industry has solved them before.

The AOL Story: Walled Gardens Give Way to Open Standards

America Online (AOL) was one of the dominant online services in the early 1990s. Along with CompuServe, Prodigy, and others, AOL provided a curated environment with email, chat rooms, news, weather, shopping, and entertainment. Everything worked smoothly — within AOL's walled garden. Users paid hourly fees for access, and content providers paid AOL for inclusion.

The services were not interoperable. You couldn't send email from AOL to CompuServe directly. You couldn't access AOL's chat rooms from a Prodigy account. Each service had its own client software, its own protocols, its own content library.

Then the web browser arrived.

With a web browser, users could access any website on the internet without needing permission from a platform operator. Websites didn't need to negotiate with AOL or CompuServe to reach users. Developers could build websites using open standards (HTML, HTTP) that worked in any standards-compliant browser. Email using SMTP worked across all systems. The cost to users dropped to a flat monthly fee for internet access rather than expensive hourly rates.

The walled gardens couldn't compete. Users abandoned proprietary online services for the open web because the open web offered more content, more functionality, lower costs, and better interoperability. By the late 1990s, AOL itself had pivoted to become an internet service provider and web portal rather than a closed platform.

The lesson is clear: **when open standards exist that provide equivalent functionality to proprietary platforms, the open ecosystem wins.** It wins because it enables competition, accelerates innovation, reduces costs, and gives users more choice and control.

The XR industry is following the same path. Current proprietary platforms are the AOL of spatial computing. The metaverse browser is now at the same inflection point as the launch of the first web browser — the point where an open standard client enables universal access to services built by anyone, running on infrastructure operated by anyone, accessible through devices from any manufacturer.

Platform Dependency Risk: HoloLens and 8th Wall

Two recent examples illustrate the concrete risks companies face when building on proprietary platforms:

Microsoft HoloLens: Microsoft invested heavily in HoloLens, an enterprise-focused AR headset. Numerous companies developed applications for the platform, trained employees on it, and integrated it into their workflows. In 2024, Microsoft discontinued the HoloLens product line. Companies that had built their spatial computing infrastructure around HoloLens faced a difficult choice: abandon their investment entirely or attempt to port applications to a different platform with incompatible APIs and capabilities. Many simply stopped pursuing AR/VR initiatives altogether after the experience.

8th Wall: 8th Wall was a web-based AR platform that allowed developers to create AR experiences accessible through mobile web browsers without requiring users to install apps. It gained significant adoption because it solved a real problem: reducing friction for AR experiences. However, 8th Wall was acquired and its technology was subsequently deprecated in favor of the acquirer's different technical approach. Developers who had built businesses on 8th Wall's platform faced the same dilemma: rebuild on a different platform or abandon their work.

These failures demonstrate why enterprises cannot deploy critical infrastructure on proprietary platforms. The risk is not merely financial. When a hospital integrates spatial computing into surgery assistance systems, or when a warehouse bases its entire logistics operation on AR-guided picking, or when an airline deploys AR maintenance procedures across its fleet, platform discontinuation doesn't just mean lost development investment — it can mean operational disruption, regulatory compliance issues, and safety concerns.

This is why virtually every Fortune 500 company that expresses interest in deploying XR technology at scale currently remains stuck in "proof-of-concept purgatory". They understand the potential value but cannot justify the risk of dependence on proprietary platforms they don't control. The only use case they can currently justify is training, because training systems can be isolated and replaced if necessary without disrupting core operations.

The solution is the same solution that enabled enterprise adoption of web technology: open standards that allow companies to deploy infrastructure they control, using commodity hardware they can source from multiple vendors, running software they can modify or replace without vendor lock-in.

Other prominent closures: AltspaceVR, a once thriving virtual world community, was purchased by Microsoft in 2017 and shut down in 2023. Mozilla Hubs ended support for their production 2024. And in 2026, Netflix purchased Ready Player Me, turning it into an internal product, shutting out a very large community of users who relied on their avatars. Most recently, Meta has discontinued both Horizon Workrooms and Horizon Managed Services, as well as restructured Horizon Worlds away from VR towards mobile, leaving their remaining user base to wonder if they will be the next domino to fall.

While none of these closures had a significant impact on enterprise-level companies, they each have caused a significant amount of lost effort and revenues for myriads of independent developers and creators who have poured significant amounts of time into supporting these platforms – not to mention the users who frequented them and formed connections with others who shared the same experiences.

4. Web Infrastructure to Metaverse Infrastructure

Understanding how metaverse infrastructure relates to web infrastructure requires mapping familiar concepts to their spatial equivalents. This mapping helps clarify what changes (and what stays the same) in the transition from the World Wide Web to the metaverse.

Core Architectural Components

Web Browser	Metaverse Browser
Displays one web page at a time from a single origin in a 2D window	Displays a 3D spatial scene with content from multiple origins simultaneously
User manually navigates to websites by typing URLs or following hyperlinks	Content automatically appears based on user's location in physical or virtual space
DOM (Document Object Model) organizes page as tree of 2D elements	SOM (Scene Object Model) organizes scene as tree of 3D objects
Typically stateless HTTP requests for loading pages	Typically stateful real-time connections for continuous spatial synchronization
Web pages are self-contained documents	Spatial scenes are composed of objects from many independent spatial fabrics and services
JavaScript provides interactivity within a page	JavaScript provides interactivity within and between spatial services
Target devices: desktop monitors, phone screens	Target devices: AR glasses, VR headsets, desktop (for development), mobile (for limited AR)

Network and Communication

Web: HTTP and HTTPS provide stateless request-response communication. A web browser requests a page, the server sends it, the transaction completes. WebSocket provides persistent TCP (Transmission Control Protocol) connections when needed (chat, real-time updates), but this is not the default mode. Only WebRTC has access to UDP (User Datagram Protocol) connections.

Metaverse: Real-time bidirectional communication is the default because spatial presence requires continuous synchronization. When a user moves their head, their view changes. When another user in the same space moves, all participants need to see that movement immediately. UDP is necessary for ephemeral data like avatar positions where packet loss is acceptable but latency must be minimal. TCP is used for critical state updates that must arrive reliably.

What Stays the Same: The underlying internet protocols (TCP/IP) remain the same. TLS provides encryption. DNS provides name resolution. CDNs can distribute content. The difference is what protocols get used most frequently and how connections are managed.

Service Discovery and Access

Web: Users discover websites through search engines, bookmarks, links, and direct URL entry. Each website is isolated — visiting `nytimes.com` doesn't automatically reveal `washingtonpost.com`.

Metaverse: Services are discovered by proximity. If a user is standing in a location where a service is available, that service becomes accessible to their metaverse browser. This is analogous to how radio stations work: you don't "install" a radio station, you tune to it when you're in range. Spatial services can be global (available everywhere, like a weather service), regional (available in specific geographic areas, like a city's public transit information), or local (available at specific locations, like a store's product catalog).

What Stays the Same: DNS and similar registries can still be used to map content and service identifiers to network locations. Search functionality is still valuable for discovering content and services before encountering them spatially.

Content Formats

Web: HTML for structure, CSS for styling, images (JPEG, PNG, WebP), video (MP4, WebM), vector graphics (SVG).

Metaverse: glTF (GL Transmission Format) for 3D models and scenes. This is a JSON-based open standard developed by Khronos Group, analogous to how JPEG is a standard for images. glTF (or a .glb binary glTF file) can include geometry, PBR materials, textures, animations, and scene structure. glTF extensions for streaming, complex scenes, interactivity, avatars, spatial audio, volumetric media, and Gaussian splats are in development.

What Stays the Same: Images, video, and audio formats used on the web are also used in spatial contexts. A 3D scene might include a 2D image displayed on a virtual screen or a video playing on a surface. Note also that web pages can currently make use of 3D objects in WebXR.

Rendering

Web: The web browser renders the DOM to pixels on a 2D screen using the browser's rendering engine (Blink, WebKit, Gecko).

Metaverse: The metaverse browser renders the SOM to a 3D view using a rendering engine that can be implemented using a variety of low-level APIs such as Vulkan, OpenGL, DirectX, or Metal depending on the platform. The key difference is that rendering must handle spatial transformations, stereoscopic displays (different images for each eye in VR/AR), and continuous frame updates at high rates (90Hz or higher for comfortable VR). The metaverse browser will use the Khronos ANARI API to provide an abstracted API to portably access any rendering engine, independently of its implementation and capabilities.

What Stays the Same: The GPU does the actual rendering. Shading, texturing, lighting calculations are fundamentally the same. The browser is still converting a data structure into pixels, just in 3D instead of 2D.

Security Model

Web: Each web page runs in an isolated context. Scripts from one page cannot access another page's data without permission (same-origin policy). iframes allow embedding content from other origins with controlled interaction.

Metaverse: Security becomes more complex because multiple services must be composed visually in the same 3D space while remaining isolated in terms of data access and capabilities. A shopping service should not be able to read data from a payment service just because they're both visible in the user's view. A malicious service should not be able to create objects that impersonate objects from a trusted service.

This requires new security models, currently in development, that extend concepts like same-origin policy and capability grants to three-dimensional spatial contexts. For example, a service might be granted permission to display content within a specific spatial volume (a "bounding box") but not outside it, preventing visual interference with other services.

What Stays the Same: The fundamental security principles of isolation, least privilege, and explicit permission apply equally to spatial contexts.

Identity and Authentication

Web: Various systems exist (OAuth, SAML, username/password) with inconsistent implementation across sites. "Sign in with Google" or similar social login has become common but is still centralized around major platforms.

Metaverse: Decentralized identifiers (DIDs) are being standardized to provide user-controlled identity that works across all spatial services without requiring a central authority. This enables a user to authenticate once to their metaverse browser and have that identity recognized across independent spatial fabrics and services without each service needing to maintain separate accounts.

What Stays the Same: The cryptographic principles underlying authentication (public/private key pairs, certificates, signatures) are identical.

Infrastructure Deployment

Web: Companies deploy web servers (Apache, Nginx, IIS) that serve content over HTTP/HTTPS. Infrastructure can be self-hosted, cloud-hosted (AWS, Azure, Google Cloud), or CDN-distributed.

Metaverse: Companies deploy metaverse servers that manage 3D coordinate spaces, user presence, and real-time synchronization of services. Just as with web infrastructure, this can be self-hosted (on-premises servers), cloud-hosted, or distributed. The key difference is that spatial fabrics almost always maintain continuous stateful connections rather than handling stateless requests.

What Stays the Same: Standard datacenter infrastructure — servers, networking, storage, load balancing — works for metaverse services just as it works for web services. The difference is what software runs on that infrastructure.

5. Five Non-Negotiables for an Open Metaverse

For the metaverse to succeed as a universal ecosystem on par with the World Wide Web, certain fundamental requirements are non-negotiable. These requirements emerged from decades of experience with web infrastructure and from understanding why proprietary platforms fail to achieve the scale and longevity that open ecosystems provide. Each requirement addresses a specific failure mode that would fragment the metaverse into incompatible silos or create dependencies that prevent sustainable deployment.

5.1. Interoperability: No Software Lock-In

Software lock-in occurs when a service or application can only run on a specific platform's infrastructure, preventing migration to alternative providers or preventing use across different metaverse browser implementations. This is functionally equivalent to a website that only works in Safari — it undermines the fundamental value proposition of open standards.

The metaverse must support interoperability at multiple levels:

Service-to-Browser Interoperability: Any service built to standards should work in any standards-compliant metaverse browser. This is analogous to how a properly-coded website works across Chrome, Firefox, Safari, and Edge without browser-specific modifications.

Service-to-Fabric Interoperability: A service should be able to operate within any spatial fabric that implements standard protocols, just as a video element should be able to play within any web page and on any web browser.

Content Portability: 3D assets, scenes, and spatial content created in standard formats (primarily glTF) should be usable across different metaverse browsers and spatial fabrics without conversion or modification. This mirrors how images and videos created for one website work on any other website.

The technical foundation for this interoperability is standardized APIs and protocols. Just as the web relies on standardized HTTP, HTML, CSS, and JavaScript, the metaverse requires standardized protocols for networked service communication, such as the proposed SMAP Service Model Access Protocol (previously called NSO), a standardized scene representation (Scene Object Model), and standardized asset formats (glTF with appropriate extensions).

Without interoperability guarantees, developers face the same fragmentation that continues to plague app development: building the same application three times for different platforms, with different codebases to maintain and different feature sets across platforms.

5.2. No Device Lock-In

Device lock-in occurs when content or services only work on specific hardware from specific manufacturers. This creates the same problems as software lock-in but at the hardware level: users are forced to purchase specific devices to access specific services, manufacturers can extract monopoly rents, and innovation is suppressed because there's no competitive pressure to improve.

The metaverse must be accessible across:

AR Glasses: Different manufacturers' AR glasses — whether from Meta, Apple, Google, Microsoft, or emerging vendors — should all be able to access the same spatial services using their metaverse browsers. Device-specific features can exist (just as phone cameras have different capabilities), but core functionality must work across devices.

VR Headsets: Similarly, VR headsets from different manufacturers should provide access to the same spatial environments and services. A virtual meeting space should not require everyone to own the same brand of headset.

Desktop and Mobile: While desktop computers and mobile phones are not optimal for immersive spatial experiences, they should still be able to access spatial content at reduced levels of immersion. This mirrors how responsive websites work across desktop and mobile despite different screen sizes and interaction models.

The mechanism for achieving device independence is the same mechanism that works for the web: open standards that define functionality independent of implementation. OpenXR already provides this for XR device APIs — a metaverse browser can use OpenXR to interact with any compatible XR device without knowing the manufacturer.

Device lock-in is particularly dangerous because hardware manufacturers have strong incentives to create walled gardens where their devices only work with their platforms. Mobile app stores demonstrate both the business incentives and the costs to users and developers. The metaverse must provide an alternative to this pattern by making device independence a foundational requirement rather than an afterthought.

5.3. Ownership: Data, Content, Identity, and Revenue

Ownership in the context of the metaverse means that users and service providers retain control over their data, content, identity, and business relationships rather than ceding control to platform operators.

Data Ownership: Users must own their personal data. A spatial service that tracks user behavior, preferences, or interactions must store that data in ways the user controls. This might mean local storage, user-selected cloud storage, or encrypted storage with user-held keys. Platform operators should not have access to user data from services running on their infrastructure unless users explicitly grant permission. This is analogous to how web users can use browser extensions to block tracking and can choose whether to accept cookies, but extends to all spatial data including position, gaze tracking, hand movements, and voice.

Content Ownership: Creators must own the content they create. A 3D model, spatial environment, or interactive experience created by a developer belongs to that developer, not to the platform where it's published. The creator should be able to move content between platforms, modify it without platform permission, and control its distribution. This mirrors how a photographer owns the rights to photos they take regardless of what website displays them.

Identity Ownership: Users must own their identity across the metaverse. Rather than having separate accounts for each spatial fabric or service (analogous to having separate accounts for every website), users should have portable identities implemented through standards like Decentralized Identifiers (DIDs). When a user moves between spatial environments from different providers, their identity, reputation, social connections, and preferences should move with them — controlled by the user, not by any platform operator.

Revenue Ownership: Service providers and creators must retain ownership of their business relationships and revenue streams. If a retailer operates a spatial store, they should be able to conduct transactions directly with customers without mandatory platform intermediaries taking percentage cuts. This is fundamentally different from app stores that require all transactions to flow through the platform operator. On the web, if you buy something from a retailer's website, the platform operator (your ISP, your browser vendor) doesn't take a cut of the transaction. The same principle must apply to the metaverse.

These ownership requirements are not just philosophical preferences — they are practical necessities for enterprise adoption. Companies will not deploy critical infrastructure on platforms where they don't control their data, can't move their content, don't own customer relationships, and must pay tolls to platform operators for every transaction.

5.4. Frictionless Access and Streamability: Browse-to-Enter

One of the web browser's most important characteristics is that users can access any website immediately without installation. You type a URL or click a link, and within milliseconds the website loads and becomes interactive. This "browse-to-enter" model is fundamental to the web's success — it eliminates friction, enables spontaneous discovery, and allows users to try services without commitment.

The metaverse must provide the same capability. Users should be able to enter spatial environments and access spatial services without installing applications. When a user encounters a spatial service — whether through proximity in AR or through navigation in VR — that service should stream to the metaverse browser and become usable within seconds.

This is fundamentally different from current VR/AR platforms where each experience requires a separate application installation. Installing applications creates multiple friction points:

- **Discovery Friction:** Users must find the application in an app store before they can use it
- **Download Friction:** Installation requires downloading potentially gigabytes of data
- **Storage Friction:** Devices have limited storage; users must manage which applications to keep installed
- **Update Friction:** Applications require periodic updates that users must manage
- **Trust Friction:** Installation requires trusting the application with device-level permissions

Browse-to-enter eliminates these friction points by streaming service code and content on-demand, executing in a sandboxed environment within the metaverse browser, and automatically managing resources. When the user leaves a spatial environment, the browser can discard temporary data and free resources for other services.

The technical requirement for streamability is careful design of services as modular, progressively-loadable components rather than monolithic applications. A spatial service should load an initial minimal set of assets to become interactive quickly, then stream additional content as needed. This is exactly how modern websites work with progressive web app techniques and code splitting.

Streamability also means that services can update transparently. Rather than requiring users to download and install updates, services simply stream the latest version when accessed. This ensures all users are always using current versions, eliminating the fragmentation and security issues caused by users running outdated application versions.

5.5. Open Tooling: Creator and Developer Ecosystem

An open metaverse requires open tooling that allows anyone to create spatial content and services without requiring permission from or payment to platform operators. This mirrors how the web succeeded: anyone can learn HTML, CSS, and JavaScript using free resources and free tools, build a website, and publish it without asking permission from browser vendors.

Open tooling encompasses several layers:

Content Creation Tools: Artists and designers must be able to create 3D assets, spatial environments, and interactive experiences using industry-standard tools (Blender, Maya, Unity,

Unreal Engine) that export to standard formats (glTF). These tools should be freely available or competitively priced, not controlled by a single platform operator charging monopoly prices or limiting access. The USD to glTF conversion pipeline, for example, is open source and allows content created in Pixar's Universal Scene Description format to be exported for metaverse browser deployment.

Development Frameworks: Developers must be able to build spatial services using standard programming languages and frameworks. JavaScript/TypeScript provides the web parallel — widely known, well documented, with extensive libraries and community support. WebAssembly extends this to high-performance code in other languages. The same accessibility must apply to spatial service development. Documentation, tutorials, and examples should be freely available and community-maintained, not locked behind proprietary platforms.

Testing and Debugging: Developers need tools to test and debug spatial services across different devices and contexts. This includes browser developer tools equivalent to Chrome DevTools or Firefox Developer Edition, performance profiling tools, and compatibility testing frameworks. These tools should be open source and extensible, allowing the community to build additional debugging and analysis capabilities.

Publishing and Distribution: The path from "I built something" to "users can access it" must be open and straightforward. On the web, you can deploy a website to any server you control or to numerous hosting providers without requiring permission from browser vendors. The metaverse requires the same: developers should be able to deploy services to spatial fabrics they operate or to commercial spatial fabric hosting providers, making those services available to any user with a compatible metaverse browser.

Open tooling is what enables innovation at the edges. When anyone can build and deploy, innovation comes from unexpected sources — individual developers, small companies, academic researchers, hobbyists. The web's explosion of creativity came from lowering barriers to participation. The metaverse requires the same approach.

6. Introduction to the Metaverse Browser

The metaverse browser is a universal client application for accessing spatial services, analogous to how a web browser provides universal access to websites. Just as a web browser implements web standards (HTML, CSS, JavaScript, HTTP) to render web pages from any server, a metaverse browser uses spatial standards (Scene Object Model, SMAP protocols, glTF assets, OpenXR API) to render spatial experiences from any spatial fabric.

The core function of a metaverse browser is to manage the user's spatial presence — their position, orientation, and interaction capabilities — while connecting them to multiple concurrent spatial services in a secure, performant manner. When a user navigates through a spatial environment, the browser continuously determines which content and services are relevant based on proximity, streams the necessary code and assets, manages real-time synchronization of shared state, renders the combined 3D scene from multiple sources, and enforces security boundaries between services.

A metaverse browser runs on XR devices (AR glasses, VR headsets) as well as desktop and mobile platforms, though the experience quality varies by device capability. On AR glasses, the browser overlays spatial content on the physical world. On VR headsets, it renders immersive virtual environments. On desktop and mobile, it provides a windowed 3D view. The same underlying browser architecture supports all these contexts, adapting the presentation to device capabilities while maintaining access to the same spatial services.

The relationship between metaverse browsers and spatial services mirrors the relationship between web browsers and websites. Multiple browser implementations can exist (just as Chrome, Firefox, Safari, and Edge are different implementations of web standards), and users can choose which browser to use based on features, performance, privacy policies, or other preferences. Services work across all standards-compliant browsers without requiring separate implementations for each browser.

Why Not the Web Browser?

The question naturally arises: can web browsers be extended to support spatial experiences rather than building an entirely new class of browser? Several projects have attempted this approach, either by adding spatial capabilities through browser APIs (WebXR) or by building spatial frameworks that run within web browsers.

These approaches face fundamental architectural mismatches that prevent them from delivering the full functionality required for the metaverse:

Proximity-Based Service Discovery: Web browsers are designed around manual navigation. Users type URLs, click links, or use bookmarks to intentionally navigate to specific websites. The browser loads one website at a time (or multiple websites in separate tabs), with each website operating independently.

Spatial experiences require automatic service discovery based on proximity. When a user wearing AR glasses walks down a street, services from dozens of businesses, municipal systems, navigation providers, and social applications should become available automatically without any manual action. The browser must continuously evaluate the user's position against available services, connect to newly-relevant services, and disconnect from services that are no longer proximate. This is not a navigation model web browsers were designed to support.

Multi-Service Composition in Shared 3D Space: Web browsers can display content from multiple origins using iframes, but each iframe renders into a separate 2D rectangle. The website in one iframe cannot place objects into another iframe's display space. This isolation works for 2D pages but breaks for spatial composition.

In a spatial environment, multiple services must be able to display 3D objects in the same shared coordinate space while remaining isolated in terms of data access and capabilities. A retail service displays products on shelves, a payment service displays a checkout interface, a navigation service displays directional arrows on the floor, and an AI assistant displays an avatar – all visible simultaneously in the user's view, all from different service providers, all operating in the same 3D scene. Yet the retail service should not be able to read the payment service's transaction data, and the navigation service should not be able to impersonate the AI assistant.

This requires a 3D composition and security model that web browsers were not designed to provide. The Document Object Model (DOM) organizes web pages as hierarchical trees of 2D elements. A Scene Object Model (SOM) organizes spatial scenes as hierarchical trees of 3D objects with spatial transformations, but also needs cross-origin security boundaries at the object level, not just the document level.

Real-Time Stateful Synchronization: Web browsers were optimized for stateless HTTP request-response patterns. You request a page, the server sends it, the transaction completes. While WebSocket and WebRTC add persistent connections for real-time features like chat or video calls, these are additions to the core architecture, not the foundation.

Spatial presence is inherently real-time and stateful. Your position, orientation, movements, and interactions must be continuously synchronized with spatial services at rates high enough to feel responsive (typically 90+ frames per second for VR, 60+ for AR). Other users' positions and actions must be synchronized to you with minimal latency. This continuous bidirectional state synchronization is the default mode for spatial services, not an optional enhancement.

Web browsers could theoretically be extended to handle this through aggressive use of WebSocket and WebRTC, but the architecture would be fighting against the browser's core design rather than working with it. The networking stack, resource management, and rendering pipeline would need fundamental restructuring.

UDP for Ephemeral Data: Web browsers restrict network communication to TCP-based protocols (HTTP, WebSocket) and WebRTC's highly constrained peer-to-peer model. UDP — User Datagram Protocol, which allows sending data without guaranteed delivery — is not directly accessible to web applications for security reasons.

Spatial applications need UDP for ephemeral data where the latest state matters more than guaranteed delivery. Avatar positions, for example, change every frame. If a position update packet is lost, it's better to skip it and use the next update than to delay rendering while waiting for retransmission. Sending position updates over TCP would introduce latency and jitter that degrades user experience. VoIP protocols use UDP for the same reason — a dropped syllable is less disruptive than delayed speech.

A metaverse browser can provide native UDP communication for appropriate use cases while maintaining security through different mechanisms than web browsers use.

Performance and Resource Access: Web browsers operate in a restricted sandbox with limited access to device capabilities for security reasons. This includes restrictions on memory allocation, thread creation, GPU access patterns, and system resource usage. These restrictions make sense for untrusted websites but create performance limitations for spatial applications.

A metaverse browser can provide more direct access to device capabilities (GPU compute, spatial audio hardware, XR sensors) because spatial services run in a different security context than arbitrary websites. The browser maintains security through explicit capability grants scoped to spatial boundaries rather than through universal restrictions that limit all code equally.

The fundamental issue is that web browsers were not designed for the metaverse's requirements. Extensions and workarounds can provide limited spatial functionality — enough for simple demos and proof-of-concept applications — but cannot deliver the full performance, feature set, and user experience that the metaverse requires. This is similar to how text-mode terminals were once extended to display “windows” through [extended control characters](#), but graphical user interfaces required fundamentally different architecture.

Building metaverse browsers as distinct software from web browsers allows the architecture to be purpose-built for spatial requirements while learning from decades of web browser development experience. Many components can be reused: JavaScript/TypeScript runtimes, WebAssembly support, rendering engines, TLS/security infrastructure, and network stacks. What changes is how these components are organized and what capabilities they're optimized for.

7. Spatial Fabrics and Metaverse Servers

To understand metaverse infrastructure, it helps to map spatial concepts to their web equivalents. On the web, websites are hosted on web servers. Anyone can run a web server — large companies operate massive server farms, small businesses rent servers from hosting providers, enthusiasts run servers at home. The web server (Apache, Nginx, IIS) is the software that speaks HTTP and serves web pages. The website is the collection of HTML, CSS, JavaScript, images, and other resources that the server provides.

In the metaverse, spatial fabrics are hosted on metaverse servers. Anyone can run a metaverse server — enterprises can operate on-premises infrastructure, spatial service providers can offer commercial hosting, institutions can run their own servers. The metaverse server is the software stack that implements spatial protocols. The spatial fabric is the collection of services, 3D environments, and real-time functionality that the server provides.

What is a Spatial Fabric?

A spatial fabric is the server-side infrastructure that defines a three-dimensional coordinate space and manages the presence of users and services within that space. The term "fabric" captures the idea of a continuous, connective substrate — just as fabric weaves threads together into cloth, a spatial fabric weaves together multiple services, users, and spatial content into a coherent shared experience.

At its core, a spatial fabric provides several foundational capabilities:

Map Service: The map defines the coordinate space of the spatial fabric. It stores information about terrain, static geometry, content, services, and attachment points for subordinate spatial fabrics. It organizes where everything is located and how spaces connect.

When a metaverse browser connects to a spatial fabric, it queries the map service in order to determine what's in the user's vicinity. As the user moves, the browser continuously requests updated information about newly-proximate areas and releases information about areas the user has left.

Avatar Service: Avatars are the visual representation of users in spatial environments. The avatar service tracks avatars as they move around the spatial fabric. This enables shared experiences — multiple users can see and hear each other's avatars, observe their actions, and interact in the same space. This service is also responsible for real-time spatial audio such that voices are perceived to come from their avatar location. Presence synchronization must happen in real-time with low latency to avoid disorienting mismatches between what different users see.

Persona Service: The persona service manages publicly visible names, information, appearance, and customization of users' avatars, ensuring that each user's chosen representation appears consistently to other users.

Communication Services: Long distance voice communication, text chat, and other forms of user-to-user communication are fundamental to shared experiences..

Additional Services: Spatial fabrics can provide numerous other services depending on their purpose — inventory systems, payment processing, AI assistants, analytics, content management, access control, and more.

Primary vs. Secondary Spatial Fabrics

The architecture supports composition of multiple nested and overlapping spatial fabrics. The primary spatial fabric defines the coordinate space, provides content and services via its map, along with user presence and communication services. Secondary spatial fabrics provide additional content and services that exist within or alongside the primary fabric's space.

For example, a city might operate a spatial fabric that provides a digital twin of the city's geography. This spatial fabric can act either as a primary spatial fabric if directly invoked by a metaverse browser, or as a secondary spatial fabric if it is attached to a larger spatial fabric. Individual homes and businesses would operate secondary spatial fabrics that provide their specific content and services — a restaurant's menu and reservation system, a retailer's product catalog and checkout, a transit authority's schedule and routing, a home's utilities and entertainment. These secondary fabrics attach to the city's fabric at the specific coordinates of their physical location and become available when users are proximate. Users can navigate to anywhere within the city's fabric and all of its secondary fabrics.

Self-Hosting and Commercial Hosting

Just as with websites, spatial fabrics can be self-hosted or commercially hosted.

Self-Hosted: An organization runs metaverse server software on infrastructure they control — on-premises servers or cloud virtual machines they manage. This provides maximum control over data, compliance with regulations, customization of functionality, and independence from hosting provider business decisions. Enterprise users particularly value self-hosting for the same reasons they self-host web infrastructure: security, compliance, control, and avoiding vendor lock-in.

Commercial Hosting: Organizations deploy spatial fabrics to hosting providers who operate metaverse server infrastructure as a service, similar to web hosting providers or cloud platforms. This reduces operational burden, provides scalability, and allows organizations to focus on

content and services rather than infrastructure management. The hosting provider handles server maintenance, updates, security patches, and resource scaling.

The critical requirement is that spatial fabric hosting remains competitive with multiple providers offering compatible services. On the web, you can move a website from one hosting provider to another because they all speak standard protocols (HTTP, MySQL, etc.). The same must be true for spatial fabrics — standard protocols ensure that services aren't locked to specific hosting providers.

Similar to Web Infrastructure

Metaverse infrastructure is not fundamentally different from web infrastructure — it's a different application of similar architectural principles to different requirements. The shift from stateless documents to stateful spatial presence, from manual navigation to proximity-based discovery, and from 2D to 3D composition necessitates different protocols and different browser architecture, but the overall model of open standards enabling anyone to provide services and anyone to access them remains the same.

8. Use Cases: Cross-Industry Applications

The metaverse is not a narrow technology for specific applications — it is a general-purpose infrastructure that will transform how digital services are delivered across every industry. Understanding use cases helps clarify why open standards are necessary and what functionality metaverse infrastructure must provide.

Enterprise Operations

Large enterprises currently run extensive operations on web infrastructure. Everything from internal communications (email, chat, video conferencing) to supply chain management, inventory tracking, HR systems, and customer relationship management operates through web applications. These systems will need spatial equivalents optimized for AR and VR interfaces.

Manufacturing and Logistics: Factory floor workers wearing AR glasses can see real-time inventory data overlaid on physical shelves, step-by-step assembly instructions displayed on the components they're working with, and safety alerts that appear when approaching hazardous areas. Warehouse workers see optimized picking routes displayed as lines on the floor and product information displayed when looking at items. Maintenance technicians see repair procedures overlaid on equipment and can call remote experts who see what the technician sees and can annotate the view.

The web analogy: these are location-aware web applications that provide context-specific information and interactive functionality. The difference is that spatial positioning makes the applications fundamentally more productive — instructions appear where they're needed, information is attached to physical objects, and spatial relationships are preserved.

Collaborative Design and Review: Teams distributed across multiple locations can meet in shared virtual spaces to review 3D designs at full scale. Architects walk through building models with clients who might be physically on another continent. Automotive engineers examine vehicle designs and make annotations that appear to everyone simultaneously. Product designers iterate on prototypes in real-time collaborative sessions.

The web analogy: this is video conferencing combined with collaborative document editing (Google Docs, Figma), but for three-dimensional content where spatial relationships are essential. Existing web tools can handle 2D design collaboration; spatial tools are necessary for 3D.

Training and Simulation: Rather than watching training videos or reading manuals, employees practice procedures in simulated environments that exactly replicate real conditions. Medical

residents practice surgical procedures on virtual patients. Pilots train in flight simulators. Heavy equipment operators learn on virtual machinery before touching real equipment. Hazardous materials handlers practice emergency response in virtual scenarios.

The web analogy: interactive training modules and certification systems already exist on the web. Spatial versions add the critical element of realistic physical interaction and spatial awareness that is essential for procedures that involve physical manipulation, navigation, or coordination.

Why open standards matter: Enterprises will not deploy critical operational infrastructure on proprietary platforms. Manufacturing processes, logistics systems, and training programs have decades-long lifespans. The platforms they're built on must be under enterprise control or must be based on open standards that ensure longevity and portability. A factory cannot afford to have its AR-guided assembly process become unusable because a platform vendor changed terms or discontinued a product.

Universities and Education

Educational institutions face the same requirements as enterprises: they need control over their infrastructure, data ownership, compliance with regulations (FERPA in the US, GDPR in Europe), and longevity. A university's investment in metaverse infrastructure should last decades, not be dependent on a consumer platform's business model.

Campus Digital Twins: Universities can create spatial digital twins of their physical campuses. Prospective students can tour campus remotely through VR, experiencing residence halls, classrooms, and facilities as if physically present. Current students use AR to see building information, classroom locations, and event notices overlaid on the physical campus. Campus services — dining hall menus, library book availability, equipment reservation — become spatially accessible.

The web analogy: this is analogous to a university website that provides information and services, but organized spatially around the physical campus rather than as hierarchical web pages. Navigation is proximity-based — walk near the library and library services appear — rather than menu-based.

Virtual Laboratories and Field Work: Science education often requires expensive laboratory equipment or travel to field locations. Spatial environments enable virtual laboratories where students conduct experiments that would be too expensive, too dangerous, or physically impossible in real laboratories. Geology students explore virtual field sites that would require expensive travel to visit physically. Chemistry students manipulate molecular models at scales that clarify structure-function relationships.

The web analogy: interactive simulations and virtual labs already exist as web applications. Spatial versions add immersion, spatial manipulation, and the ability to collaborate with other students in shared virtual spaces.

Distance Learning in Shared Spaces: Rather than video conference grids where students watch the instructor through 2D windows, spatial platforms enable shared virtual classrooms where instructor and students are spatially present. The instructor can demonstrate concepts using 3D models that all students can examine from their own perspectives. Students can work together on shared projects in virtual group spaces.

The web analogy: video conferencing (Zoom, Teams) and learning management systems (Canvas, Blackboard) provide the web version. Spatial versions add presence, spatial interaction, and the ability to work with 3D content collaboratively.

Why open standards matter: Universities need to own their educational content and infrastructure. Course materials developed in proprietary platforms become unusable if the platform changes or disappears. Open standards ensure that educational investments remain viable long-term and that content can be migrated between systems as technology evolves.

Commerce and Retail

Retail is already undergoing digital transformation through e-commerce websites and mobile apps. Spatial commerce is the next evolution — retail experiences that bridge physical and digital, that provide information contextually, and that enable new forms of customer interaction.

Spatial Storefronts: Physical retail stores can provide AR-enhanced shopping experiences. Customers wearing AR glasses see product information, reviews, prices, and availability overlaid on physical items. Virtual try-on features show how furniture would look in customers' homes or how clothing would fit. Wayfinding shows routes to specific products within large stores. Checkout happens through spatial interfaces without waiting in physical lines.

Virtual Showrooms: Products can be displayed in virtual showrooms where customers browse in VR from anywhere. Automotive dealers provide virtual test drive experiences. Furniture retailers let customers arrange full room layouts virtually. Retailers can offer unlimited virtual shelf space without physical inventory costs.

Location-Aware Services: Services become available based on physical proximity. Walk past a restaurant and see today's specials and wait times. Enter a store and have access to that store's inventory, promotions, and customer service. These services are provided by individual businesses through their own spatial fabrics, not through a centralized platform.

The web analogy: this is e-commerce (Amazon, individual retailer websites) combined with local search (Google Maps showing nearby businesses). Spatial commerce adds the ability to interact with products in three dimensions, to overlay information on physical retail environments, and to conduct transactions through spatial interfaces.

Why open standards matter: Retailers need to own their customer relationships and transaction data. Open standards ensure that retail businesses can deploy spatial commerce infrastructure that integrates with their existing systems, works across different customer devices, and doesn't require paying tolls to platform operators for every transaction.

Infrastructure and Smart Cities

Municipal governments and infrastructure operators can provide public services through spatial interfaces that improve accessibility and efficiency.

Transportation: Transit systems provide real-time route information, delays, and alternatives overlaid on the physical environment. Passengers see which bus or train to board displayed above the vehicle. Navigation through complex transit stations shows clear directional cues. Accessibility information helps users with mobility limitations find elevators, ramps, and accessible routes.

Public Safety: Emergency information appears spatially — evacuation routes displayed on streets, hazard warnings at relevant locations, emergency service contact options contextually available. Public health information can be distributed spatially during emergencies.

Municipal Services: City services become spatially accessible — permit applications linked to specific properties, parking information at relevant locations, public facility information (libraries, parks, community centers) available based on proximity.

The web analogy: municipal websites provide this information today through web pages. Spatial versions make the information contextually available based on where the user is, reducing friction and improving accessibility.

Why open standards matter: Municipal infrastructure must be accessible to all residents regardless of what device they own. Open standards ensure that public services work on any compatible device and that cities maintain control over public infrastructure rather than being dependent on private platform operators.

Recurring Themes

Several requirements appear across all these use cases:

Multi-Device Access: Different users have different devices. Within a single organization, some employees might use AR glasses, others VR headsets, others desktop computers. Services must work across devices.

Longevity: Infrastructure investments span decades. The systems deployed today must remain viable for 10-20+ years, which requires open standards that outlast any particular vendor's product lifecycles.

Compliance: Different industries have different regulatory requirements (HIPAA for healthcare, FERPA for education, GDPR for EU users, industry-specific regulations). Self-hosted spatial fabrics allow organizations to maintain compliance just as they do with current web infrastructure.

Integration: Spatial services must integrate with existing enterprise systems — databases, authentication systems, payment processors, analytics platforms. Open standards enable integration through standard APIs and protocols.

These requirements reinforce why the metaverse must be built on open standards rather than proprietary platforms. No single use case can justify the investment in open standards, but the aggregation of requirements across industries makes the path forward clear.

9. Join the Effort: Building Open Standards for the Metaverse

The metaverse will not be built by any single organization. Like the World Wide Web, it requires collaborative standardization efforts, open-source implementations, and broad industry participation. The work is happening now across multiple standards organizations and open-source projects.

Current Standardization Efforts

Several standards development organizations (SDOs) are working on different pieces of the metaverse infrastructure:

Khronos Group: Developing standards for 3D asset formats (glTF and extensions for spatial use cases), rendering APIs (ANARI for abstracted rendering), and XR device interfaces (OpenXR). Khronos operates under a royalty-free IP framework that ensures standards remain open.

W3C (World Wide Web Consortium): Working on web-related spatial standards including decentralized identifiers (DIDs) for portable identity, accessibility standards for spatial experiences, and potential future spatial browser APIs.

OGC (Open Geospatial Consortium): Developing GeoPose and related standards for spatial positioning and coordinate systems that bridge physical and virtual spaces.

Metaverse Standards Forum: A coordination body that brings together these SDOs, companies, and individuals to ensure alignment across different standardization efforts. The Forum provides a venue for discussing architectural questions, mapping dependencies between standards, and building consensus on approaches.

To Be Determined: The proposed SMAP (Service Model Access Protocol) standard provides the protocol and API layer for services to communicate with metaverse browsers. It is being developed with plans to formalize through appropriate SDOs once the initial prototype demonstrates viability.

Open Source Implementation

Standardization works best when paired with open-source implementations that prove the standards are implementable and that surface issues early. Several open-source efforts are underway or planned:

Open-Source Metaverse Browser: Serves as a testbed for implementing emerging standards and serves as a usable exemplar metaverse browser implementation. This project follows established open-source governance models with technical steering committees, maintainer roles, and community contribution processes.

Spatial Fabric Server: Open-source server implementations that provide basic spatial fabric functionality, allowing developers to run their own metaverse infrastructure and experiment with service development.

Developer Tools (planned): Debugging tools, performance profilers, asset conversion utilities, and other developer tooling to make spatial service development accessible.

Conformance Testing (planned): Test suites that validate whether implementations correctly follow standards, ensuring interoperability between different browsers and fabrics.

These efforts are intended to operate under permissive open-source licenses (Apache 2.0 is commonly used) that allow commercial use, modification, and redistribution without restrictive obligations.

How to Participate

Here are the two most impactful ways to get involved:

Join the Metaverse Standards Forum: Participate in the Open Metaverse Browser Initiative (OMBI) open-source implementation project to collaborate with the Standards Organizations developing the spatial standards of the metaverse. Review draft specifications, contribute use cases and requirements, implement experimental prototypes, and provide feedback on what works and what needs improvement. [Click here for information on joining the Metaverse Standards Forum.](#)

Become a Financial Sponsor: Building a native metaverse browser requires significant funding to build a qualified team to complete this project. Organizations interested in sponsoring this critical open source project can [contact the Metaverse Standards Forum.](#)

Here are some additional ways to contribute:

Open Source Development: Contribute to open-source metaverse browser and spatial fabric projects through core functionality development, bug fixes, documentation, testing, or developer tools. Projects welcome contributions at all skill levels.

Content Creation: Develop 3D assets, spatial environments, and example services that demonstrate capabilities and surface requirements. Share these publicly to help others learn and validate that tools and standards meet real-world use cases.

Service Development: Build spatial services using emerging standards and provide feedback on developer experience. What's intuitive? What's challenging? What's missing? Service developers are essential for validating that standards serve actual needs.

Research and Analysis: Conduct academic or independent research on spatial computing architectures, security models, performance characteristics, and user experience to inform better standards and reveal important tradeoffs.

Advocacy and Education: Help explain why open standards matter for the metaverse, what problems they solve, and what opportunities they create. Counter misinformation and clarify what the metaverse means in this context.

The Path Forward

Building the metaverse is a multi-year effort requiring sustained collaboration. The foundational work happening now — defining core standards, building implementations, establishing governance structures, and building community — will determine whether the metaverse becomes an open ecosystem or fragments into proprietary silos.

The experience of the World Wide Web provides both inspiration and warning. The web succeeded because it was open, but that openness was not inevitable — it resulted from deliberate choices by individuals and organizations who valued openness over control. The metaverse faces the same choice. Current proprietary platforms have substantial resources and market presence. They will not voluntarily adopt open standards if they believe closed ecosystems serve their interests better.

Open standards succeed when they provide better outcomes for the broader ecosystem — more innovation, larger markets, faster advancement, greater reliability. Demonstrating these benefits requires building working systems that prove the model works. This is why exemplar implementations, early adopters, and public deployments are critical. They make the abstract concrete and show that open metaverse infrastructure can deliver real value.

The invitation is open to anyone who recognizes the importance of this work and wants to contribute. The metaverse will be built — the question is whether it will be built on open standards that enable broad participation or on proprietary platforms that create dependencies and limitations. The choice is being made now through the work being done today.