# The Web Already Has a Z-Axis:

## Hypermedia as Entity-Component-System for a Spatial Internet

The internet is already an Entity-Component-System. URLs are entities, media types are components, and protocols are systems. Every attempt to build a 'metaverse' has failed because it tried to replace this architecture instead of extending it. The web already has everything it needs except a z-axis.


| | |
|---|---|
| AUTHOR | Adam Paigge |
| ORGANISATION | Supernova Labs |
| LOCATION | Manchester, UK |
| CONTACT | adam@supernovalabs.co.uk |
| PUBLISHED | 2026 |
| SERIES | Foundation Paper 03 of 03 |

ABSTRACT

This paper argues that the spatial internet does not need to be invented — it needs to be recognised. The web is already an Entity-Component-System of unprecedented scale: URLs are entities, MIME types are components, browsers and renderers are systems. Every corporate attempt to build a metaverse has failed because it tried to replace this architecture with a new one, rather than extending it with a missing primitive. That primitive is the z-axis: a shared coordinate system that gives existing web entities a position in navigable three-dimensional space. This paper describes the architecture of that extension — the four concerns of the spatial system, the protocol stack beneath it, and the implementation path using components that already exist and work. Adding a z-axis to the web means recognising what is already there. No company owns this. No platform gates it. It is the web. It always was.

CONTENTS

01 OBSERVATION

# The Observation

> *"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." — Mark Weiser, 1991*

The internet is the most successful Entity-Component-System ever built. This is a structural observation about what the web already is, and it explains why every attempt to build a spatial internet from scratch has failed.

An Entity-Component-System is an architectural pattern from game development. Entities are identifiers. Components are data attached to entities. Systems are processes that operate on components. The pattern separates identity from data from behaviour, allowing any combination to be composed at runtime without inheritance hierarchies or tight coupling.

The web works the same way. A URL is an entity — a stable identifier that can be resolved to something. The media type of the response is its component — text/html, model/gltf+json, application/activity+json, video/mp4. The browser, the Three.js renderer, the ActivityPub server, the video player — these are all systems that know how to process specific component types. HTTP content negotiation is the mechanism by which an entity declares which components it can provide, and the client selects which system will process them.

This is the actual architecture. And it already works at a scale no purpose-built ECS has ever achieved: billions of entities, thousands of component types, decades of system evolution, federated across every device on earth.

02 HISTORY

# Why Every Metaverse Has Failed

Every corporate attempt at a spatial internet has made the same error: building a new ECS from scratch. A new entity model (user accounts on a proprietary platform), new component formats (proprietary asset types), new systems (proprietary renderers, proprietary social graphs, proprietary economies). Each one asks the entire internet to migrate into its container.

This is the equivalent of responding to the invention of cinema by declaring that all books must now be films. The correct response is to add cinema to the set of media types that can be linked, embedded, and composed with everything that already exists. You do not replace the library. You add a screening room.

The web survived the transition from text to images, from images to video, from video to interactive applications, from applications to real-time communication — all without replacing its fundamental architecture. Each transition added new component types and new systems while the entity model (URLs) and the linking model (hyperlinks) remained unchanged.

> *Spatial computing is not a special case. It is the next media type.*

03 ARCHITECTURE

# The ECS Made Explicit

### 3.1 ENTITIES ARE URLS

Every addressable thing on the internet is already an entity. A web page, an image, an API endpoint, a person's identity on a federated social network, a glTF mesh, a Gaussian splat environment, a sound effect, a live video stream. Each has a stable identifier — a URL — that can be resolved, linked to, embedded, cached, and composed with other entities.

The critical property is that URLs are universal and decentralised. Anyone can mint a URL by running a server. No central registry approves entity creation. No platform owns the namespace. The entity model is just the URI specification — a thirty-year-old standard that has never needed replacement.

### 3.2 COMPONENTS ARE MEDIA TYPES

A component, in ECS terms, is data attached to an entity. On the web, this is the resource that a URL resolves to, typed by its MIME type. The same entity can serve different components depending on what the client requests — this is HTTP content negotiation, already the mechanism by which the web composes heterogeneous data without coupling.

| Component (MIME type) | Data | Existing System |
|---|---|---|
| text/html | Document structure + content | Browser layout engine |
| model/gltf+json | 3D geometry + materials | Three.js, Babylon.js |
| application/activity+json | Social graph + activities | ActivityPub (Mastodon) |
| video/mp4 | Temporal visual sequence | Video decoder |
| application/wasm | Portable executable code | WASM runtime |
| application/x-spz | Gaussian splat point cloud | Spark renderer |
| audio/mpeg | Sound | Web Audio API |

Adding spatial computing to the web means registering new component types (splat environments, spatial presence, world state) and building new systems to process them. It does not mean replacing the entity model, the linking model, or any existing component type. An HTML page does not stop being an HTML page when it is rendered as a texture on a quad in a Three.js scene. It is the same entity, the same component, processed by an additional system that gives it spatial coordinates.

### 3.3 SYSTEMS ARE PROTOCOLS AND RENDERERS

A system, in ECS terms, is a process that queries for entities with specific components and operates on them. The browser layout engine processes text/html. Three.js processes model/gltf+json. An ActivityPub server processes application/activity+json. Systems compose without coordination — a web page can embed a video, which embeds a WebRTC stream, which carries spatial audio. No central orchestrator manages this composition. Each system processes the components it understands and ignores everything else. The spatial system is one more entry in this list.

04 SPATIAL SYSTEM

# The Spatial System

The web already has 3D rendering — WebGL since 2011, WebGPU since 2023. What it lacks is a spatial system: a coherent way to declare that a set of entities occupy a shared space, that this space is navigable, and that visitors to this space can perceive each other. The spatial system has four concerns:

### 4.1 WORLD GEOMETRY — THE BAKED STAGE

A world is a static environment that visitors move through. Gaussian splats, streamed via formats like .spz and rendered by systems like Spark, provide photorealistic navigable environments that scale from mobile phones to VR headsets through Level-of-Detail budgeting. The environment is baked: geometry, lighting, and atmosphere compressed into a streamable point cloud. Critically, the world is a component on an entity. It has a URL. It can be linked to, cached, served from a CDN or a peer mesh. It is a document.

### 4.2 DYNAMIC OBJECTS — THE SCENE GRAPH

Interactive objects are glTF meshes placed into the world's Three.js scene graph. They are separate entities from the world — each has its own URL, its own provenance, its own permissions. This separation is essential for federation. The world is hosted by one server. The objects in it may come from anywhere. A visitor can bring objects into a world the same way a user can paste a link into a chat — by providing a URL to an entity that the world's renderer knows how to process.

### 4.3 DOCUMENTS — HTML-IN-CANVAS

This is the key that closes the loop. If an HTML page can be rendered as a texture on a surface in the 3D scene, then the entire existing web becomes spatial content. Every website is already a spatial object waiting for a position. A Mastodon timeline on a wall. An orbital mechanics console on a table. A blog post pinned to a lectern. A video call on a screen. They are all just HTML surfaces with URLs, rendered by the browser's existing layout engine and sampled into the spatial system as textures.

### 4.4 PRESENCE — FEDERATED VISITORS

When two people visit the same world, they should see each other. Presence is a component on a visitor's identity entity: their position, orientation, avatar geometry, and interaction state. The transport is WebRTC for low-latency co-presence, with BPv7 delay-tolerant networking as a fallback for degraded or mesh-only links. Identity federation follows the same model as ActivityPub — your avatar and public key are components on your identity entity, hosted on your home server. The world verifies that your home server trusts you. That is sufficient.

05 PROTOCOL

# The Protocol Stack

This paper builds on the four-layer architecture described in the first Supernova Labs foundation paper, reframed here as systems in the web's ECS:

| Layer | Function | ECS Role | Implementation |
|---|---|---|---|
| Physical | Entry point: tap, scan, click | Entity creation | NFC card → URL → world loads |
| Mesh | Local-first delivery | Component transport | Meshtastic, WebTorrent, libp2p |
| Settlement | Provenance + ownership | Component provenance | Script chain (ledger-agnostic) |
| Experience | Navigable, composable, spatial | All systems composing | Three.js + Spark + WebXR |

The Physical layer creates entities. An NFC card is a physical URI — a tangible thing that resolves to a URL when tapped. That URL loads a world. No account creation, no app install, no platform. Tap and you are there.

The Mesh layer transports components. When the backbone is available, components travel over HTTPS. When it is not — at a festival, in a disaster zone, on a spacecraft — components travel over local mesh networks via Meshtastic radios, BPv7 bundles, or WebRTC peer connections. The component does not care how it arrived.

The Settlement layer provides provenance. The Script chain — ledger-agnostic by design — anchors facts of ownership without requiring any specific blockchain's consensus mechanism. It is infrastructure beneath the protocol, not the product.

The Experience layer composes all systems into a navigable whole. Three.js renders geometry. Spark streams splats. The browser renders HTML. WebRTC carries presence. Web Audio spatialises sound. WebXR provides immersion. These are all existing systems. The experience layer is their composition, given a shared coordinate system.

06 DEFINITION

# What This Is

It is a protocol. Anyone can host a world the same way anyone can host a website. The technology is web standards plus open-source renderers plus an open specification. There is no company that runs the spatial web. There is no account to create, no terms of service to accept, no API key to request.

It is the web, with a z-axis. Every existing website, every existing protocol, every existing standard continues to work exactly as it does today. The spatial system is additive. It gives existing entities an optional position in 3D space. Entities that do not want spatial positions are unaffected.

It runs on any device with a web browser. VR is one possible rendering target, the same way a phone screen or a laptop monitor are rendering targets. WebXR provides immersion for devices that support it. For everyone else, it is a navigable 3D scene in a browser tab. The experience degrades gracefully — the same world, less immersion — rather than excluding users without specific hardware.

> *No company owns this. No platform gates it. No app store approves it. It is the web. It always was.*

07 IMPLEMENTATION

# The Implementation Path

### AVAILABLE NOW

Gaussian splat rendering: Spark 2.0 (World Labs, open-source) streams 100M+ splats in-browser via Three.js/WebGL2, with Level-of-Detail budgeting for steady framerates on any device.

3D mesh rendering: Three.js with react-three-fibre provides a complete scene graph with physics (Rapier, Cannon), interaction, and XR support.

HTML-in-canvas: Same-origin content can be rendered to canvas textures today. The cross-origin path via foreignObject SVG and cooperative mesh proxying is an engineering task, not a research problem.

Identity federation: ActivityPub, the protocol behind Mastodon, already federates identity and social interaction across independently hosted servers.

Physical entry points: NFC cards resolve to URLs on any smartphone manufactured in the last decade. No app required.

Mesh networking: Meshtastic provides LoRa mesh networking on commodity ESP32 hardware at ranges up to 10km per hop.

### REQUIRED ENGINEERING

A scene manifest format declaring: here is a world (URL to splat), here are the objects in it (URLs to glTF entities), here are the document surfaces (URLs to HTML pages with spatial positions), and here is the presence endpoint.

A spatial presence protocol extending ActivityPub or building alongside it, carrying position, orientation, and avatar state between federated servers.

A permission and sandboxing model for visitor-brought entities entering a world, analogous to the browser's same-origin policy but for spatial interactions.

A cooperative content proxy for html-in-canvas across origins, running as a mesh node that serves content within the same trust domain.

These are integration and specification tasks, using technologies that independently exist and work. The spatial web is waiting for someone to compose the pieces.

08 CONCLUSION

# The Argument

The internet is already a spatial hypermedia system. It just does not know it yet.

Every URL is an entity. Every media type is a component. Every protocol is a system. The architecture that serves a web page to a browser is the same architecture that can serve a world to a renderer, an object to a physics engine, a document to a canvas texture, and a person's presence to everyone else in the room.

Adding a z-axis to the web means recognising what is already there and adding the one system that is missing: the system that gives entities a position in shared, navigable, three-dimensional space.

---

The web page becomes a panel. The link becomes a portal. The URL becomes a place. The browser tab becomes a world. And the person at the keyboard, or holding the NFC card, or wearing the headset, becomes a visitor in a space where every document, every object, every other person is there — present, addressable, composable, and free.

Supernova Labs Ltd. · supernovalabs.co.uk · adam@supernovalabs.co.uk · © 2026