Pagelove Investor Memo: v2

The web's drift from simplicity is holding back **the next computing era**

The web began as a simple design: a document-sharing protocol for anyone to build the tools they need.

Over time, that simplicity has been lost. Simple, linkable HTML pages are now buried inside proprietary walled gardens, atop a stack of complicated systems and technologies: databases, frameworks, APIs, serverless functions, WebSockets, and more.

This creates three growing issues:

1) **Technical complexity robs developers' attention from work that matters**

   The ability to develop software (especially as a team) is directly linked to the ease with which you can read and understand it. Abstraction is one thing – obfuscation is another. The size and complexity of modern codebases puts a cognitive tax on every piece of work. And it's the biggest limiting factor on the success of AI coding agents.

2) **Greater complexity exposes more surfaces for bugs, vulnerabilities and exploits.**

   There's no such thing as perfect security. But the fastest way to undermine it is to add new elements, relationships and systems. Every one becomes a mouth to feed, with dire consequences of failure and huge cybersecurity bills in an attempt to mitigate the risks. We are on the brink of this multiplying drastically as LLMs add new attack vectors.

3) **For AI Agents, complexity deeply limits their power.**

   If humans can no longer make sense of a website with "View Source", Large Language Models have no chance. They can't show brands' products in Search, they can't operate your mission critical software – in short, they can't deliver the automated future that model companies have promised. Our greatest minds are trying to solve this with systems like Model Context Protocol, or computer vision, but these only add *another* token-expensive technology to the stack. They try to fix the symptom, instead of curing the cause.

If you designed the web today, it wouldn't look like this.

But it might look like Pagelove.

Pagelove reunites HTML with its data, giving any web page the behaviours of a full web application, while eliminating the rest of the stack.

"I gave the world wide web away for free because I thought that it would only work, if it worked for everyone."
– Tim Berners-Lee, The Guardian

Instead of databases and API calls, a Pagelove app stores its own contents, becomes its own API – and is immediately, radically easy to read for humans and AI alike.

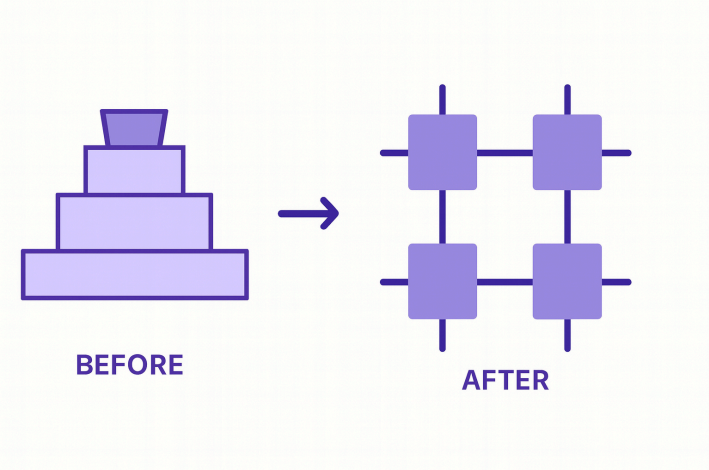
We do this by adding a simple, backwards-compatible enhancement to the HTTP standards to make surgical, element-level updates.

All this runs on a new, proprietary server design, building the first Content Delivery Network and Edge Computing Platform for Pagelove apps.

In Pagelove, a to do list becomes as simple as:

```
<ul>
      <li>Get milk</li>
      <li>Make pancakes</li>
</ul>
```

This opens new possibilities across the web – **not by adding to the stack, but collapsing it back to its core.**

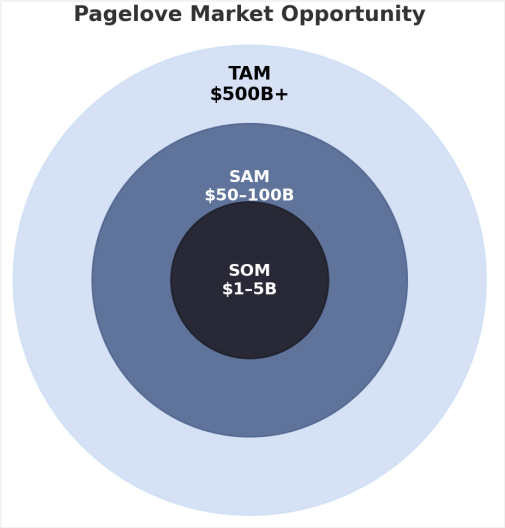


When the page and API are one, interoperability becomes a native characteristic. All your software can work together, without bespoke integrations. Every product page becomes accessible and linked immediately for potential buyers, and crawlable in LLMs' index. The web is reconnected by default.

For AI, LLMs will use a fraction of today's tokens to build comprehensive web apps with superior security and stability. Then AI Agents can use any Pagelove app just by reading its HTML. From productivity apps, to ordering train tickets, to managing your calendar and more – the AI native web is now here.

Pagelove will capture a multi-billion dollar market by pushing compute to the real network edge – **the browser**

Pagelove sits at the intersection of three fast-growing markets — CDN, Edge Compute, and Serverless — **worth more than $500B combined.**



By removing databases and serverless layers, Pagelove captures the economics of edge delivery and the adoption curve of AI-native applications.

- **TAM ($500B+)** – Combined global markets for CDN, edge compute, and serverless infrastructure.
- **SAM ($50–100B)** – Workloads needing simpler, cheaper, AI-ready web infrastructure.
- **SOM ($1–5B)** – Near-term opportunity through developer adoption and enterprise contracts.

The evolution of the edge ends in the browser

1998: Akamai created the first Content Delivery Network (CDN) by placing racks of servers with cheap magnetic disks in data centres around the world. They reduced latency by caching content closer to users.

2011: Fastly innovated on that model. By using smaller, faster servers equipped with high-speed SSDs and an open-source stack they cut costs and improved performance. Their hardware footprint was lighter and more efficient.

2017/2019: CDNs like Cloudflare and Fastly began bringing computation to the edge as demand for dynamic content closer to the user grew. Cloudflare Workers & Compute@Edge expanded their respective capability to execute business logic and manage state in their Points-of-Presence. Capability improved, but complexity increased, as did the hardware footprint and cost.

2025: Pagelove's advantage is structural. We push computation to the real edge: the user's browser. The web server becomes primarily a storage, retrieval, and distribution system. Mimicking Fastly's original strategy, our edge hardware can be lean — fast SSDs with a small footprint, distributed globally. But with Pagelove's innovation, the architectural pressure that has forced others to increase cost and deploy more hardware at the edge is avoided.

This design gives Pagelove a lasting advantage. It allows us to deliver applications that are cheaper to host, easier to scale, and naturally AI-ready. While incumbents continue to build heavier infrastructure to support computation at the edge, Pagelove's infrastructure footprint can remain lean.

Guiding principle for H2 2025 – H1 2026: Seed the network

Before launch, we will focus on building momentum among the world's best developers and partners.

- **Work directly with top developers** to build and host Pagelove apps using our SDKs and open-source tools.
- **Host events and cultivate community** to grow the early builder network.
- **Submit Pagelove's draft RFC to the IETF** for standardisation while going operational with proprietary platform capabilities.
- **Partner across the AI model and developer ecosystem** so Pagelove becomes the default substrate for agentic web apps (target partners include Anthropic, OpenAI, Cursor, Cline, Aider, Lovable, Bolt, Perplexity, and Dia).
- **Leverage our server IP advantage** to remain the only viable host for this class of application for as long as possible.
- **Deliver an on-premise enterprise product** where cloud-heavy incumbents will slow to follow.

Like Vercel and Cloudflare, Pagelove follows a proven model: freemium entry, usage-based pricing, and enterprise contracts. Our pricing is as simple as our architecture — request-based, lower cost, and higher margin. From day one, we are enterprise-ready with both cloud and on-premise delivery, building a durable moat through protocol innovation, defensible IP, and a developer-first culture.

However the web was meant to be built from documents, not frameworks. Where other web infrastructure delivers HTML, Pagelove makes the document active: data, schema, and behaviours travel with the page.

The result is a new kind of web application—globally scalable, document-centric, and intelligible to AI. **Pagelove is where the CDN, the edge, and the AI-native web converge.**

The concept is proven, **the roadmap is clear**

Phase 1: Build (2025–26): We are raising *≈£6M seed* to turn Pagelove from prototype to platform *(People, Servers, SDKs, early community etc…)*

This raise will fund the next 18 months of growth and position us for a Series A.

- **Build the platform** – expand our engineering team to complete the Pagelove runtime, CLI, and developer console.
- **Grow the network** – run open-source launches, developer events, and integrations with AI-ecosystem partners.
- **Land startup or enterprise pilots** – deliver on-premise deployments and prove commercial scalability.

The goal: demonstrate developer traction, ecosystem adoption, and enterprise readiness — three proof points that will unlock our Series A.

Complete:

✅ Proof-of-concept servers running Pagelove apps with selector-range updates

✅ DOM changes streamed automatically over websockets

✅ OIDC authentication & OAuth permissions

✅ Internal demos (collaborative to-do list, shared notes, e-commerce catalogue)

✅ AI agent demos reading and updating Pagelove apps directly, without MCP layers

In progress:

Core Runtime (Rust + WASM)

- Patentable server technology
- DOM-based, graph-aware file system that stores state directly in HTML documents
- WebSocket streaming of changes, OAuth-based permissions, and schema-driven validation baked in

Developer Experience (Pagelove Console)

- Management console built with Pagelove itself, dogfooding the technology
- Instant deploys to a global CDN: shipping an app is the same as serving a document
- Schema registry and validators to ensure apps are interoperable and AI-ready out of the box

Up next:

Phase 2 – Launch (2026): *Open-source release, showcase apps, hackathons, first enterprise pilots.*

Targeting startups with low switching costs and enterprise CTOs under pressure to cope with the move to AEO.

Phase 3 – Scale (Late 2026 →): ***Pagelove Cloud, enterprise contracts, standards work, Series A raise.***

A team with experience, network, and vision needed to deliver

Pagelove's team brings deep expertise in web infrastructure, product design, and developer growth — the combination needed to rebuild the web's foundations.

- **James Duncan (CEO)**
  5x founder, 2x exits. Invented core web infra (PaaS, containerisation, edge).
  - Commercial experience: >250k/year air miles selling to enterprises; 9 years of management consultancy;
  - Software engineer with 25x years working on cutting edge web tech.

- **Benji Portwin (CPO/COO)**
  Scaled and led product at Spotify, Monzo, AccuRx & New Look. Founded product-led startups/agencies.

- **Sam Johnston (CTO)**
  Software Engineer**,** technology executive and serial entrepreneur with over 20 years of experience founding and advising startups, and in leadership roles at top global technology companies including Google, Equinix, Citrix, and DXC Technology.

- **Max Tatton-Brown (CMO)**
  Agentic startup founding team, B2B Agency founder.
  - Clients inc GoCardless, Unbabel, Pleo, Pusher, Ably. VC-side with Creandum, Notion Capital, Forward Partners (now Molten)
  - Led multi-thousands member communities, written for Wired, Guardian, TechCrunch

Advisors: Nathan Torkington (O'Reilly/OSCON), Alistair Croll (Just Evil Enough, Strata & StartupFest) and others.

Appendix i. Frequently Asked Questions

Why hasn't this been done before?

Nobody has discovered the part of the web specification that we use for this purpose.

Is it enterprise-ready/ What about security?

Pagelove has a reduced attack surface compared to traditional web apps. Simpler code and fewer frameworks mean fewer vulnerabilities.

OIDC handles authentication across applications. Since data lives in HTML, standard web security models apply. It scales horizontally at the edge, and supports on-prem deployments.

What is defensible about this business?

There is a considerable amount of protectable intellectual property required to deliver the platform at scale. Pagelove will build a defensive IP portfolio while the system is developed, to ensure that our lead in the market can be protected.

At the same time, we will be providing a reference implementation of the core technique and submitting it for standardisation as part of the normal IETF RFC process.

What about developer fatigue/ resistance to something new?

- Developers only need to understand HTML.
- No need to replatform: they can use this for one feature with no impact on their existing stack.

How does this affect other companies in the market?

- **Vercel:** we are immediate competitors at a time where their reputation has created vulnerability. Except with higher margins.

- **Vibe-coding tools:** it takes far fewer tokens to write a single page of HTML, with no need for database partnerships, or complex architectural components. The outcome is a much simpler codebase for ongoing edits and development.

- **AI Model Providers:** Chat experiences can build independent working web apps right in the stream, no longer requiring their own environments. Chats can be natively collaborative between humans or multiple AI/ Agents.

- **HTMX/ Phoenix LiveView/ Astro.build:** these add a new framework to the stack, with the complexity and learning that involves. We are using the web's fundamental protocol to radically¸ re-empower the core of the web.

How big is the developer opportunity, really?

There are around five million active web developers using frameworks such as React, Astro, and HTMX.

If just 1 percent of them (50,000) spend $20 per month, and another 5 percent (250,000) spend $5 per month, that represents **over $30 million in annual recurring revenue before enterprise expansion.**

This bottom-up adoption reflects how developer ecosystems grow in practice — a small group of high-engagement early users, followed by a larger long tail of casual or part-time adopters.