SOLUTION BRIEF

Prerender with Dynamic Attributes

Fast Pages, Live Data—No Trade-offs Required

In digital commerce, speed drives growth. Faster pages improve user experience, boost conversions, and lift SEO—but dynamic content often gets in the way.

When prices, inventory, or promos are always changing, pre-rendering can seem out of reach. Most tools treat dynamic data as a blocker. Harper makes it a feature.

The Shortcomings of Traditional Approaches

Most modern web stacks force a choice between speed and flexibility.

Frameworks like Next.js attempt to bridge the gap with features like Incremental Static Regeneration (ISR), but they still rely on external APIs and separate data layers that introduce latency and complexity. Every dynamic update requires cache revalidation, regeneration logic, and additional infrastructure to scale cleanly.

Meanwhile, traditional CDNs offer excellent performance for static assets, but treat caching as an all-or-nothing operation. You can cache the whole page or not at all. That binary model makes serving real-time data messy, brittle, and expensive.

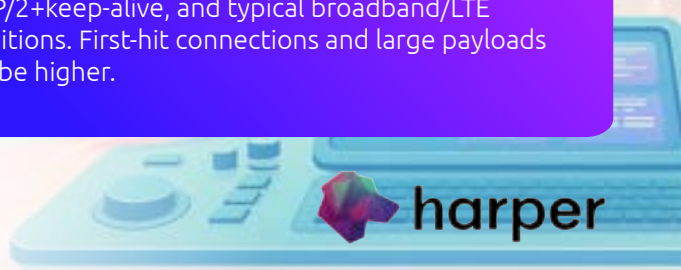
At scale, even small performance penalties compound, adding milliseconds for every round-trip between origin, application, and data layers.

The Harper Solution

Harper makes pre-rendering viable for dynamic, data-rich experiences. By unifying the database, cache, messaging, and app layer into a single distributed platform, Harper delivers the speed of static rendering with the flexibility of live data, no rewrites required.

For 95% of users, Harper delivers full page load in under 600 ms.

Assumes in-region PoPs, pre-rendered HTML with dynamic values computed in ~200 ms server time, HTTP/2+keep-alive, and typical broadband/LTE conditions. First-hit connections and large payloads may be higher.



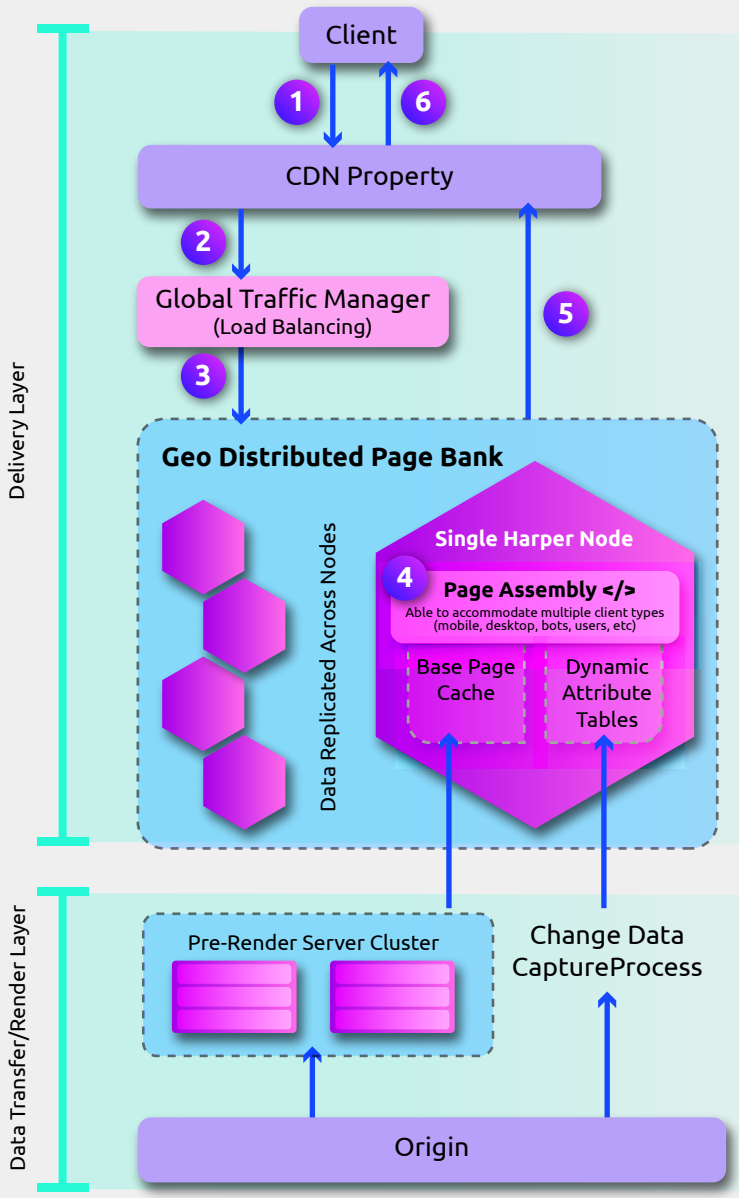
How it works:

You pre-render the core layout and content of a page—everything that rarely changes—and cache it globally. The fast-changing elements, like price or inventory, are stored in a lightweight attributes table directly within Harper's runtime. When a request comes in, those values are injected on the fly with nearly no detectable latency penalty, typically in just 1 or 2 milliseconds.

Unlike ISR or API-bound solutions, Harper eliminates the need to regenerate entire pages or manage complicated revalidation logic. Your frontend doesn't change. Your stack doesn't have to move. You just layer Harper in front and start shipping faster experiences.

And because Harper's architecture is distributed by default, both content and data live closer to every user, delivering consistently fast performance no matter where in the world your customer are.

Pre-Render w/ Dynamic Attributes



Conclusion

Pre-rendering doesn't have to come at the cost of freshness, and real-time data doesn't have to slow you down.

Harper's dynamic attribute prerendering unlocks a new model for digital commerce performance: fast, flexible, and fully future-ready. Whether you're optimizing for search bots or real buyers, the results speak for themselves.
Ready to move faster? Let's talk.

Contact Sales at hello@harperdb.io.

harper