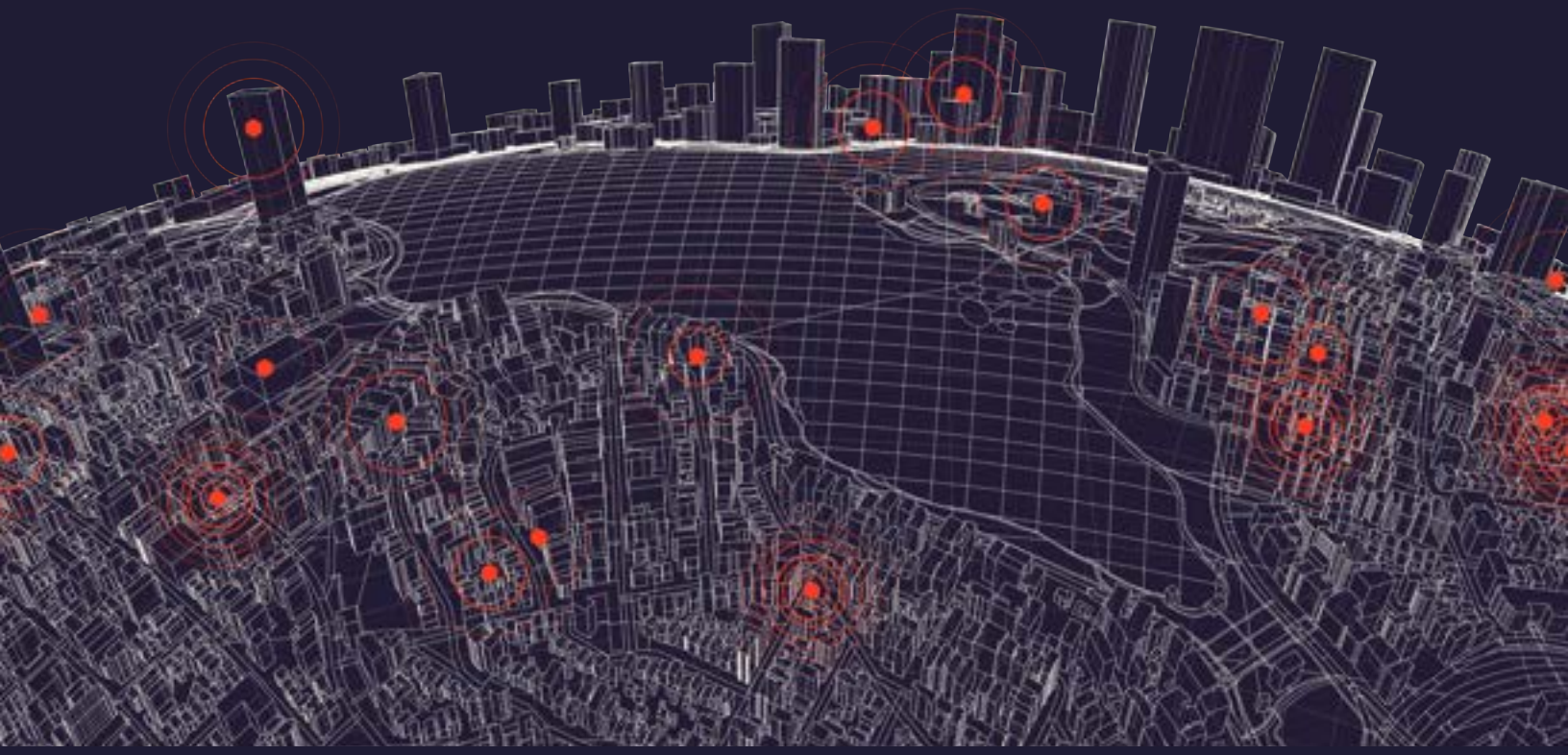
NSTREAM

WHITE PAPER

How Web Agents + Streaming Applications

Unlock the Full Value of Streaming Data



Introduction

Enterprises today must maintain "always-on" situational awareness and operational responsiveness to capitalize on the right opportunities during the brief windows they are available. Understanding streaming data in the moment — and in context — is the key to unlocking its full potential.

Yet many organizations struggle to obtain meaningful, contextual insights from their streaming data in a timely manner. This is because traditional approaches (often involving stateless, database-backed applications) involve too much latency, making it impossible to deliver real-time responses. Additionally, they lack the end-to-end architecture required for analysis, learning, and prediction.

Enter: Web Agents, a stateful microservices architecture designed to derive insights from data in motion. Web Agents provide an elegant solution for achieving real-time situational awareness and decision automation that modern enterprises require. They allow businesses to organize streaming data on an entity-specific level, execute business logic, and process in real time without breaking streaming context.

Together with streaming APIs and real-time user interfaces, Web Agents enable businesses to build streaming applications that deliver real-time observability, live modeling of complex business operations, and responsive decision automation at scale.

Continue reading this guide to learn:

- What it really means to unlock the full value of streaming data
- How Web Agents solve challenges to real-time data processing
- The composition of real-time streaming applications
- The best way to develop real-time streaming applications

Contents

SECTION 01

What does it mean to unlock the full value of streaming data?

When thinking about the best way to unlock the value of streaming data, it's important to keep in mind that data is the means, not the end.

The whole point of acquiring data is to do something with it that positively impacts the business.

The motivation for adopting streaming data is not just to be able to inform smarter decisions, but to inform smarter decisions as quickly as possible. Each incoming message represents new information that needs to be accounted for as quickly and efficiently as possible. Businesses must be able to use their streaming data to answer essential questions such as:

- What is going on across the business?
- What does this information mean?
- How should we act in response to this meaning?
- How can we program autonomous action to remediate common problems?

SECTION 02

Challenges to real-time data processing

To answer the above questions, businesses must instantly analyze their streaming data in context.

However, traditional stream processing architectures require the repeated querying of one or more databases for context before the message can be processed. The resulting database load equals the number of streaming events per second times the number of context queries needed to enrich each event. This approach is neither scalable nor cost effective — plus, it is unfeasible to achieve the desired latency (within a few hundred milliseconds) for "real-time" insights.

Additionally, as difficult and expensive as it is to combine a single streaming data source with one or more static data sources, combining multiple streaming data sources together is effectively impossible to do in-stream. You can't "query" another streaming data source for context when processing an event. The only way to splice multiple streams together is to write one or both of the streams to a database — but doing so breaks the flow of data and increases latency exponentially.

SECTION 03

How Web Agents solve challenges to real-time data processing

A Web Agent is a URI-addressed distributed object that encapsulates business logic for individual entities.

Web Agents are stateful — they preserve their data and compute context locally between operations. Statefulness enables ultra-low processing latency, increases system throughput, and improves resilience through self-sufficient operation.

Web Agents keep everything they need to do their jobs as state, including data such as recent history, real-time status of related entities, and partial computation products to facilitate fast re-evaluation of analytics. By optimizing for data locality, Web Agents spend more time computing, and less time waiting on database round-trips.

On top of that, Web Agents are able to dynamically connect to each other based on real world relationships among the sources they represent, such as containment or proximity. Web Agents make (and break) their links based on changes in the real world as they process events. The magic of linking is that it allows a Web Agent to compute — concurrently and in real time — using both its own state and the state of other agents to which it is linked, enabling granular contextual analysis, learning and prediction, and even automated response.

SECTION 04

Web Agents and real-time streaming applications

Web Agents are a crucial component in the development of real-time streaming applications**.** Streaming applications go beyond stream processing and streaming analytics because they not only surface valuable insights, they perform automation, provide visualization, and enable action at the earliest moment.

Moreover, there is a fundamental difference between legacy streaming applications and real-time streaming applications. With real-time streaming applications, the entity model updates in real time, business logic executes on real-time state instead of queried state (thanks to Web Agents), APIs are streaming instead of polled, and user interfaces are universally event-driven instead of being half-driven by user events and half-driven by polled backend queries.

Streaming APIs

Streaming APIs are the key to the dynamic connectivity of Web Agents mentioned above. These APIs continuously push information out to recipients without those recipients having to ask for new data. Without streaming APIs, critical business logic and end user observability are left out of the stream, forced to live minutes to hours behind the wavefront of data.

Streaming APIs power agent-to-agent and agent-to-UI communication while handling backpressure on both ends and across all streams leaving or arriving at an agent. They enable real-time streaming application composition and an ecosystem of streaming applications with complete location transparency.

Real-time user interfaces

In addition to assisting users in gleaning new insights from streaming data, real-time UIs are the best way to verify that large-scale automated systems are working as intended. Humans are good pattern recognizers, but they can't recognize patterns unless the underlying signal is shared with their senses in an intuitive fashion.

Visualizing raw data is not very useful. Key performance indicators derived from raw data tell a more meaningful story to people. Aggregated and reduced KPIs are even more valuable to humans, since they paint a high-level picture of what's going on to better guide the user's attention. Real-time UIs consume aggregated and reduced KPIs, as well as insights generated by Web Agents, and present them in a way that is easy to understand, similar to a business intelligence dashboard. Here is an example of what a real-time UI looks like for a cellular network simulator.

SECTION 05

Using Nstream to develop real-time streaming applications

Nstream is the only streaming application platform that allows businesses to execute business logic at the speed of change. All other application platforms are driven by request, instead of by the arrival of new data. With existing stateless architectures, it is impossible to perform arbitrary computations as soon as new data arrives.

The basics of real-time streaming application implementation

Real-time streaming application development is straightforward with Nstream. Every application entity is represented by a Web Agent. All Web Agents retain their state between operations in observable properties called Lanes, which come in several varieties that correspond to common data structures and access patterns.

Cache-coherent distributed pointers, called Links, form streaming relationships between Web Agents. Application code has memory-latency access to the complete state of a Web Agent's Lanes and Links. Nstream continuously synchronizes linked states in the background via multiplexed streaming APIs. Scoped callback functions notify observers of precise state changes with half-ping latency.

The basics of real-time streaming application architecture

External data is ingested via connectors that subscribe to broker topics, receive database transaction log updates, make periodic queries, and forward updates to relevant Web Agents based on embedded information. These connectors translate ingestion events into ordinary state changes, decoupling Web Agents from their underlying data sources.

Application code responds to state changes by updating derivative computations to keep them consistent with observations. Each external event triggers a cascade of updates that propagate across the Web of Agents as fast as the network allows.

(CONT.)
USING NSTREAM TO DEVELOP REAL-TIME STREAMING APPLICATIONS

An example of real-time streaming applications in action

Let's say there is a company that wants to develop a streaming application for real-time traffic and infrastructure monitoring. They are looking for a solution that is cost effective, delivers real-time performance, and can be scaled across distributed infrastructure. In previous efforts to use traditional stream processing for its needs, the company found that the latency incurred was too high, reducing the effectiveness and value of the data.

Realizing that legacy architectures are not the right fit, the company turns to Nstream to help them unlock the desired streaming application for their use case. First, the company configures Web Agents to replicate traffic intersections — one per intersection — for the cities in which they operate. For example, they convert voltage readings from traffic lights so that the application can understand green, yellow, and red states. They also include a data source that tracks passing vehicles at each intersection and counts the vehicles that pass every day, along with the detection of each pedestrian that presses a crosswalk button.

Latest state is processed and updated locally by the Web Agent as the data is generated and received. By maintaining the latest state and only transmitting updates to data subscribers, Nstream drastically reduces the volume of data that sensors or other systems pass over the communication network. This delivery of streaming data without unnecessary latency unlocks the real-time monitoring solution the company was looking for.

What's more, Nstream leverages available compute on the existing infrastructure of each city to run the streaming application. A simple software update is all that's required to automatically create Web Agents for new intersections. Nstream is the only application platform designed to scale this way on distributed compute, helping the company avoid adding expensive infrastructure for each city.

SECTION 06

Conclusion

Real-time streaming applications built using Nstream benefit from Web Agents' stateful, concurrent computations that may be distributed over many runtime instances, covering many availability zones or even hybrid enterprise/cloud resources. They are the most efficient way to unlock the full potential of streaming data to drive business value.

For technical details on how to build real-time streaming applications using Web Agents, please refer to our documentation for developers. If you're ready to get started with an open source implementation of Web Agents and streaming applications, check out the project on GitHub.

About Nstream

Founded in 2015, Nstream (formerly Swim, Inc.) enables organizations to build open-source, full-stack applications directly on top of streaming data, providing real-time insights that organizations can take action on instantly. Nstream's customers use Nstream's unique streaming APIs and stateful objects to build applications with business logic to gain more and better insights. As a result, data-driven decisions can be made within minutes to reduce operational costs, improve productivity and agility, enhance the customer experience, and outpace competitors – rapidly. Operate with true real-time data and take more intelligent action now.