

WHITE PAPER

▶SEARCH▶TR/01▶03

Building Continuous, Event-Driven Applications with Apache Kafka and Swim



▶SEARCH▶TR/01▶03
▶RS:/0211-SEARCH...A01

▶RS:/011

▶RS:/0211TR / 0N

▶SEARCH▶TR/01▶03



▶SEARCH▶TR/01▶03

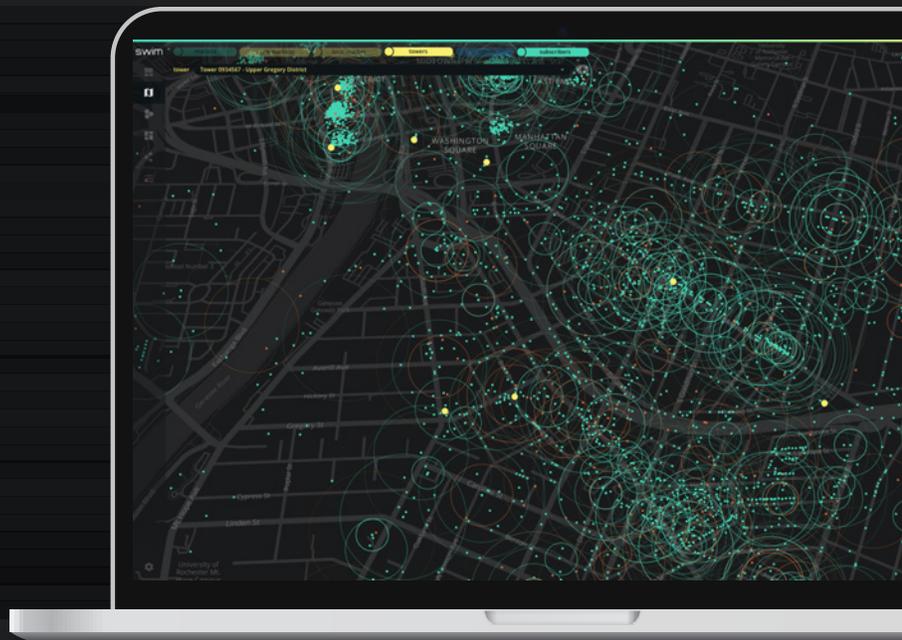
▶RS:/011

▶RS:/0211TR / 0N

Contents



Shifting Architectures Create New Opportunities	01
Considerations for Building Continuous Intelligence Applications	02
Introducing the Swim Solution for Continuous Intelligence	03
What does the Swim Platform actually do?	03
Continuous Intelligence Apps with Kafka and Swim Continuum	04
Swim Continuum Simplifies Continuous Intelligence App Development	05
Why Swim?	06
Where does Swim Continuum fit?	07
Where to Learn More about Swim	07



Shifting Architectures Create New Opportunities

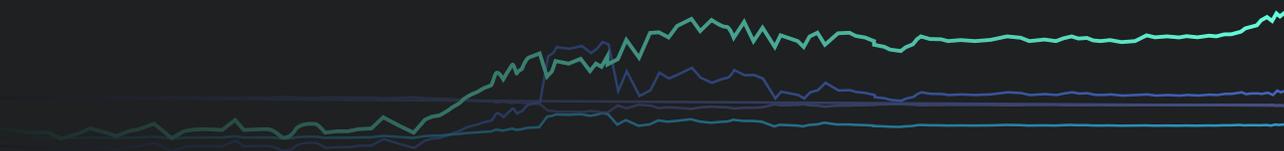
Today's digital businesses and those moving towards digitization are rapidly embracing event-driven architectures to augment, and in some cases replace, traditional batch-driven IT. Event-enabling digital enterprises introduces new capabilities, but also prepares for the inevitable exponential growth in volume of events to be handled. Many have turned to Apache Kafka to handle the growth in event volumes.

Adoption of Kafka has helped many organizations reduce the latency of key analytics and KPIs by speeding data delivery into data warehouses where they are subsequently analyzed. But even faster analytics can be achieved, allowing IT systems to stay in complete sync with the real world.

More recently, a number of companies have further reduced the latency of their analytical systems by analyzing events in-stream, without first storing events in an analytic repository. The resulting low latency analytics have become known as Continuous Intelligence – a new class of continuous, event-driven applications that deliver timely, accurate situational awareness directly from event streams as events occur.

Continuous Intelligence (CI) applications, in order to more effectively extract “the signal from the noise”, require two notable shifts in the application architecture. The first is the embrace of continuous event-driven architectures such as Apache Kafka. The second is a shift in the use of Kafka (or other event-driven architectures) from streaming data into data warehouses or data lakes prior to analysis, to continuous, in-stream analysis, learning and predicting directly from event streams. These two shifts deliver ‘always-on’, situational awareness and operational responsiveness directly to users and applications by slashing the delays that inevitably arise from batch-oriented data lakes and warehouses.

Once implemented, CI applications can accelerate digital transformation by bringing new possibilities within reach. Continuous Intelligence applications initially enable decisions to be made using accurate measures of current conditions. Some users are now building increasingly predictive intelligence by integrating machine learning into their CI applications. Still others are coupling CI applications with automation in order to take faster, more accurate action and achieve a reduction in human involvement.



Considerations for Building Continuous Intelligence Applications

Building CI applications is not without risks. Many of the risks arise from the selection of technologies and tools used or underestimation of the efforts needed to integrate them.

If you have been using Apache Kafka, your development teams are exposed to several options for the implementation of stream processing systems, ranging from directly-integrated Kafka libraries, to database semantics atop Kafka, and graphical composition tools reminiscent of data pipeline construction tools. Some tools are Kafka-specific, where others are infrastructure agnostic, providing additional freedoms. Tools that overlay database semantics can appear straightforward to use, but fall short of functional requirements and can require large infrastructure commitments to operate. Some tools use Kafka itself for rebuilding state following failures leading to highly-unpredictable behavior following multiple failures.

The greatest value of CI applications is achieved by businesses operating virtual or physical infrastructures that produce event streams so large, fast and complex as to be beyond human comprehension. Building CI applications using limited, or overly-simplified tooling can result in brittle apps, tedious management and burgeoning infrastructure costs. With this in mind, your challenge will be to select a CI platform architected from the start to deliver enterprise-level capability by being:

- **Scalable:** A platform for CI applications should scale easily and elastically to the scale needed by the business without reaching design limits.
- **Observable:** A CI platform should provide management and deployment tools to avoid becoming unwieldy to modify and improve, constraining design teams and driving costs up.
- **Durable:** Enterprise CI applications can only deliver value only if they can operate 24x7 and quickly recover from inevitable failures with minimum management attention.
- **Flexible:** Few large business infrastructures change predictably. Therefore, a CI platform must provide agility and tolerance of change in order to avoid becoming a barrier to progress.
- **Maintainable:** CI application platforms should allow for rapid improvement and provide a repeatable build and deployment process that fits into existing DevOps practices.
- **Efficient:** Operating CI applications at scale requires efficiency to ensure that resource consumption is affordable and predictable as application demands grow.

Selecting a platform for developing and operating CI applications atop Kafka is a complex task. This paper will help architects, developers and administrators understand how **Swim Continuum** helps to address business-critical needs by supporting effective CI applications atop Kafka or other streaming infrastructures.

Introducing the Swim Solution for Continuous Intelligence

Swim Continuum is an open, end-to-end platform for building and running large-scale Continuous Intelligence applications. CI applications built using Swim Continuum enable better and faster decisions by equipping people and applications with timely, usable assessments of the detailed and aggregate measures reflecting real-world conditions in the business.

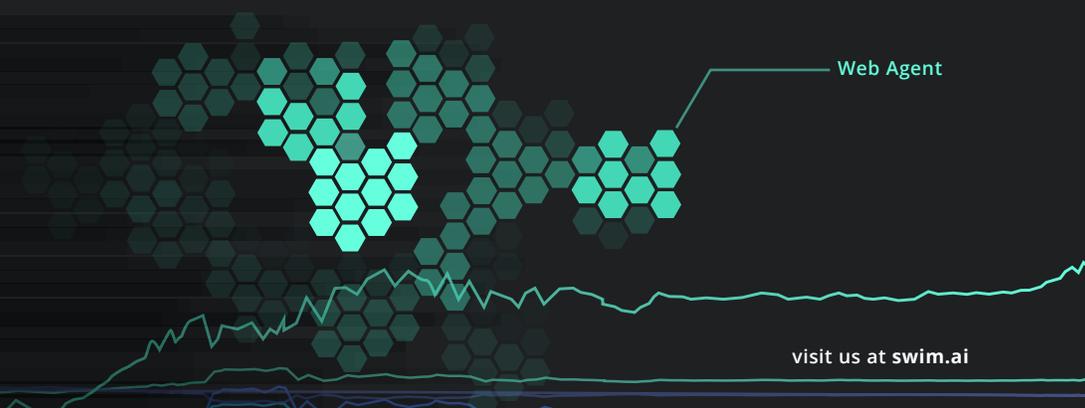
By enabling developers to easily build business logic, package it and distribute it flexibly on distributed infrastructures, Swim Continuum helps developers quickly build platform-agnostic CI applications that scale elastically, run efficiently and are both manageable and observable.

Swim Continuum simplifies and accelerates the development of CI applications by letting developers focus on business logic, rather than infrastructure and integration. Using Swim, developers build richly self-aware and stateful business logic in Java. Swim provides all the connectivity, external interfaces and state management tools to turn the Java logic into distributed CI applications.

What does the Swim Platform actually do?

To create Continuous Intelligence applications, developers build, deploy and run logic that efficiently extracts measures and KPIs from event streams. This business logic is typically implemented as a directed acyclic graph (DAG) – similar to what you recognize from Hadoop or Spark, but running on individual events with very low latency.

Each node of the DAG is built as a “Web Agent” - a well-defined, atomic, encapsulated, consistent logic element. **Web Agents** ingest events from Kafka and subsequently filter, enrich, correlate aggregate, transform and communicate processing results to other Web Agents and external applications and databases (see figure 1, next page).



Continuous Intelligence Apps with Kafka and Swim Continuum

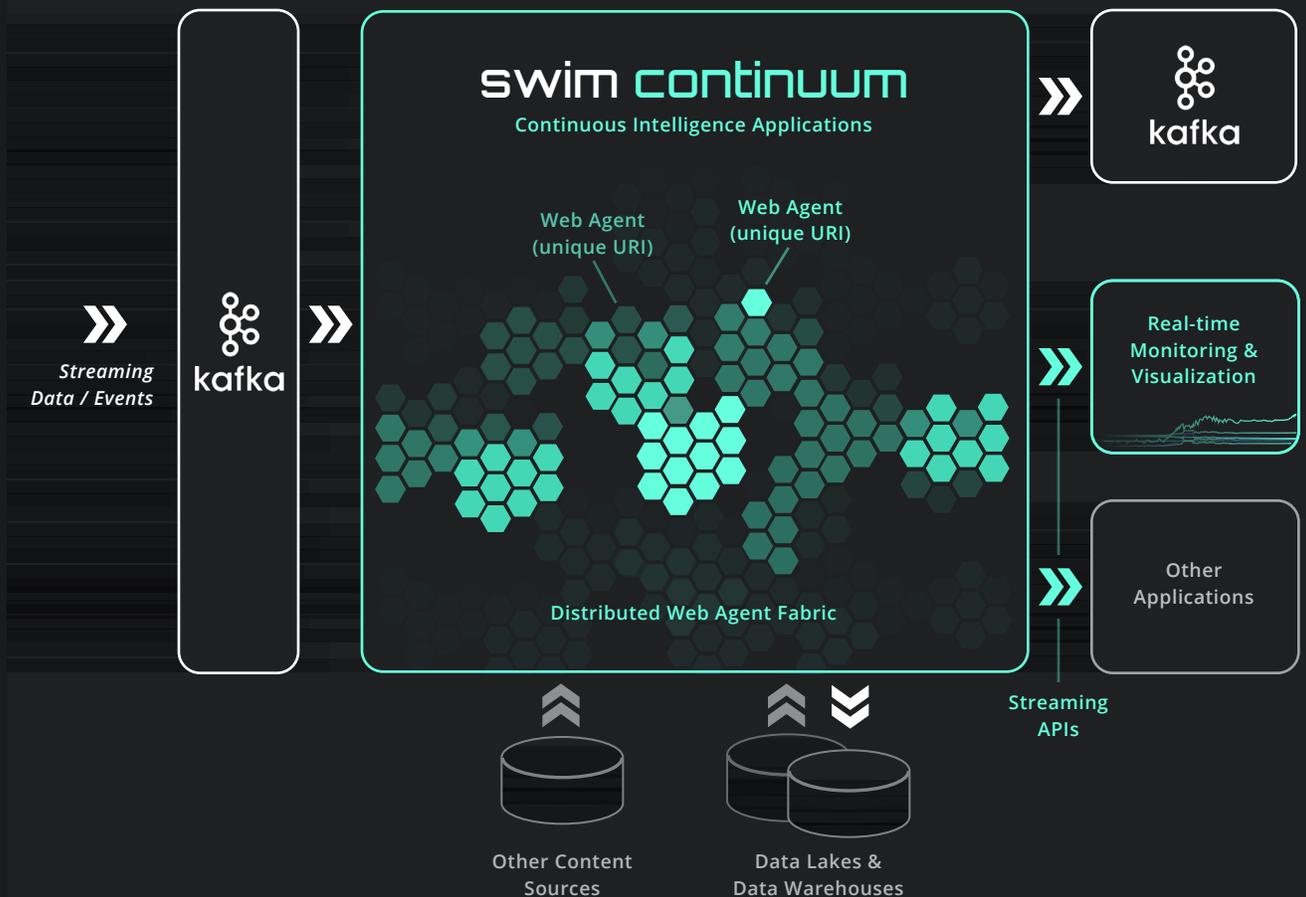


Figure 1: Building event-driven continuous intelligence applications with Apache Kafka and Swim

With Swim, **Web Agents** operate atomically and efficiently on in-memory state data that is automatically replicated to enable Web Agent failover. By processing in memory, efficiency and speed reach levels unattainable when using external storage such as a database. Developers can still interact with external applications and databases where needed.

Once built, Web Agents are dynamically deployed by Swim, atop a distributed architecture of Swim servers running in the cloud, on-premises and at the edge. The resulting architecture delivers efficiency, latency and reliability needed for large-scale Continuous Intelligence.

But can Continuous Intelligence applications be easy to build?

Swim Continuum Simplifies Continuous Intelligence App Development

With Swim, developers implement business logic using prebuilt services provided by Swim, including:

- **Prebuilt Connectors:** Swim provides connectors to Apache Kafka, Apache Pulsar and other streaming systems to minimize the coding burden and to ensure consistent, reliable consumption and production of events.
- **State Data Management:** Web Agents become self-aware and stateful through the use of pre-built, in-memory data structures called “Lanes”. Available in several forms, Lanes provide efficient, in-memory data persistence in a variety of structure types that are automatically replicated for rapid failover and provide the ability to join data from multiple sources.
- **Inter-Agent Communication:** Pre-built “Links” provide cache-coherent communications between Web Agents via universally-addressable data Lanes to share and join information between Web Agents for enrichment or correlation of event data.

Once the Web Agents are created, SwimOS, the open source core of the Swim Continuum, simplifies the build, deployment and administration processes:

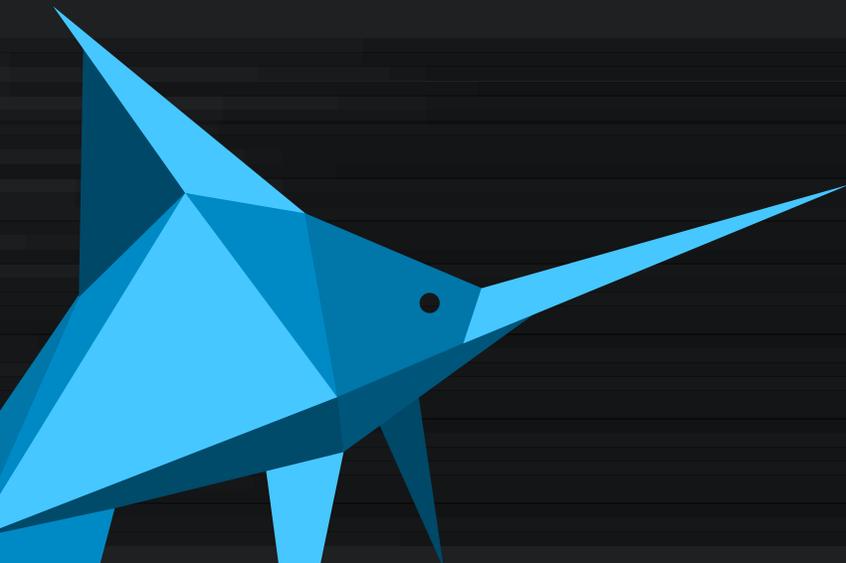
- **Build:** Swim builds Web Agents using Gradle, Maven or a similar Java build tool.
- **Deployment:** Swim dynamically distributes Web Agents and groups of Web Agents called “Planes” across the Swim server infrastructure, optimizing for Web Agent and data locality.
- **Universal Addressability:** All Web Agents and their data Lanes are directly addressable via URIs, simplifying communications and management.
- **Comprehensive UI:** Swim Continuum enables introspection and observation of Web Agents and their data Lanes via Universal Addressability to enable Swim’s web GUI or your own custom web UI to access the results of the CI application.



Why Swim?

Starting with an open, end-to-end and highly-scalable platform assures smooth growth of your CI applications from infancy through becoming a trusted enterprise asset. Swim's agnostic approach to platforms decouples CI applications from underlying infrastructure - Apache Kafka vs. others, native vs. Kubernetes deployment - on-premises, in the cloud and at the edge. Most importantly, Swim excels in the following attributes:

- ✓ **Trusted:** Swim Continuum uses Java as its core language, leveraging the long history IT teams have of building, deploying and managing reliable CI applications using DevOps tools and practices your teams are already using.
- ✓ **Capable:** Swim simplifies stateful, contextual analysis required to reduce vast streams of events into useful intelligence. The capabilities accessible to developers are only limited by Java and the many languages, tools and other platforms that can be integrated with it.
- ✓ **Scalable:** Starting out with an efficient but scalable platform avoids future disruptions to your development timeline that can occur with less scalable tools.
- ✓ **Performant:** Web Agents maintain state in-memory and communicate with each other using Links - fast, low-latency interconnects. They have no dependency on external systems or storage except where desired by the developer.
- ✓ **Interoperable:** Swim interoperates with Apache Kafka and other streaming sources easily, providing connections and schema translations out of the box, freeing you from dependency on a single streaming data infrastructure.
- ✓ **Durable:** Swim Continuum automatically manages state data in Lanes to provide the ability to fail over rapidly in response to node failures and provides efficient checkpoints that help avoid unpredictable event reprocessing delays seen in some other stream processing platforms.
- ✓ **Efficient:** Swim automates many of the mundane functions of distributed architectures, including basic event transformation, filtering and enrichment, inter-agent and external



Where does Swim Continuum fit?

The **Swim Continuum** platform is built to enable Continuous Intelligence and other stateful, event-driven applications at enterprise scale. Look to Swim Continuum when you anticipate needing to...



Easily analyze billions of events daily at lowest possible latency



Scale elastically and effectively through in-memory state sharing



Provide stateful, self-aware event analysis



Grow from measurements and KPIs toward predictive and automated responses



Deliver 24x7 availability by continuing to operate while recovering from failures

Where to Learn More about Swim

For more information about Swim Continuum, visit the Swim web site at www.swim.ai and follow us on Twitter @swim.

For technical professionals, details of the Swim platform are available in videos, white papers and blogs available on the Swim resource page at <https://www.swim.ai/resources/>. Interesting uses of Swim to solve customer challenges are detailed on the Swim customer page at <https://www.swim.ai/customers/>. Developers interested in contributing to SwimOS, or studying the development environment should have a look at the Swim developer site at <https://www.swimos.org>.

Copyright © 2021 by Swim.ai, Inc. All Rights Reserved. Swim.ai, SwimOS, Swim Continuum and the Swim logo are trademarks or registered trademarks of Swim.ai, Inc. and/or its affiliates in the U.S. and worldwide. All other product names, logos, and brands are the property of their respective owners.