NSTREAM

Why Real-Time Streaming Data Isn't Delivering Real-Time Insights

Streaming (real-time) data has become the lifeblood of today's enterprises. They are collecting and processing data in real-time, but they still struggle to deliver valuable streaming data insights quickly enough for meaningful action.

The reason? Traditional data processing uses store-then-process architectures that compound cost and complexity when handling large amounts of data at low latency. In an attempt to deliver real-time to the business, large enterprises spend valuable money, time, and talent on data source technologies in an effort to reduce latency, when in reality the multiple systems compound latency. For other enterprises, such solutions are cost-prohibitive.

Streaming data applications built with Nstream are able to seamlessly combine streaming data and data at rest, and apply context and run business logic instantaneously in a process-then-store approach. These streaming data applications complete the data pipeline by running the entire application stack in stream. Here's how they work to overcome the challenges of streaming data analytics.

Stored data is growing at an exponential rate.

181 Zettabytes

in 2025, compared to 97 Zettabytes in 2022[1]

And organizations are generating massive volumes of streaming data.

>80% of Fortune 500 are processing data in real time[2]

Half of surveyed organizations generate **500,000 messages per second or more**[3]

1/3 of those companies peak **at over a million messages per second**[3]

They're desperate to make full use of it.

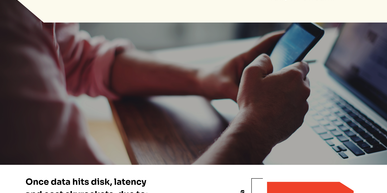
89% want automated action within seconds[4]

43% want **millisecond latency**[4]

70% of streaming data users develop **custom applications**[5]

But are they doing it well?

only 16% of applications run fully in-stream.[4]



Once data hits disk, latency and cost skyrockets, due to:

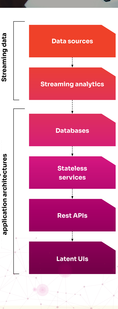
Stateless services
Typical applications "forget" information as soon as it's processed. So, to store the state of an entity, multiple round-trips to a database are required.

Storage
Storing this huge volume of data racks up massive costs.

REST APIs
APIs must be polled continuously — multiple times a minute — to keep the data fresh.

Latent UIs
Visualizations suffer lag, so users look at past events. Output cannot drive automation as UIs are static.

Compute power
The only way to reduce (but not eliminate) latency is by increasing costly compute infrastructure.

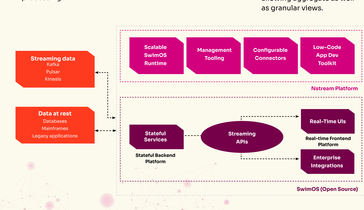


Streaming data applications eliminate latency and reduce cost, thanks to:

Stateful services
Applications "remember" the state of all data so there's no need to store before processing.

Streaming APIs
allow communication between services and the UI without polling.

Real-time UIs
show the state of the business at any moment and are interactive — allowing aggregate as well as granular views.



For Nstream's customers, streaming data applications lead to real-world results...

10 GB per second of streaming data from 10+ data sources

200M+ events per second, with millisecond latency

Real-time experience scoring of **100M+ users**

Real-time interactive map view of **20M+ IoT devices**

Extrapolating real-time status of **5M+ "dumb" legacy equipment**

Real-time interactive UI at **network-level latency**

Live heat-map view of **30M+ users** at street-level accuracy

...and quantifiable benefits.

10x **faster time to value**
Customers have deployed in production at scale (200M+ events per second) in just 6 weeks

70% **lower total cost of ownership (TCO)**
4x fewer engineering hours to design, build, test, and maintain

100x **lower latency,** as data is pushed instead of polled/queried

5x **fewer connections/integrations to maintain**

Contact Nstream to speak with our team and see a real-time application in action.

NSTREAM

Sources
[1]https://www.g2.com/articles/big-data-statistics
[2]Based on use of Apache Kafka, https://kafka.apache.org/
[3]Swim (Nstream), *The State of Streaming Data*, Q3 2021
[4]Ibid.
[5]Ibid.
[6]Ibid.
[7]Ibid.
[8]Ibid.