How NATS is designed to deliver AI workloads at the edge

Byron Ruth & Jeremy Saenz | Synadia

Before we begin...

nats-kubecon.vercel.app

Who are we?

Jeremy Saenz

- **Engineering** at Synadia
- Long-time **gopher** (Martini, Negroni, CLI, and more!)
- Moved from Eng to Product, and back to Eng.

Byron Ruth

- **Developer Relations** at Synadia
- A NATS project **maintainer**
- **Release manager** for the NATS server
- Co-host of the **NATS.fm** podcast
- Previously 14 years in **pediatric biomedical research**

NATS Primer

Current **mainstream tech** used for building and operating distributed systems is **limiting** and overly **complex** for engineering **teams**.

A “modern” OSS stack

What components do we need?

- gRPC - 1:1 request-reply and streaming
- RabbitMQ - M:N messaging and queues
- Kafka - scalable data streaming
- Redis - key-value
- Minio - object storage
- Envoy - proxy, routing, load balancing
- Istio - security, policy, observability
- Consul - service discovery

What do we have?

Complexity stems from inherent limitations.

- ✕ Cumbersome discovery with HTTP/DNS
- ✕ Limited 1:1 communication patterns
- ✕ Perimeter-based security models
- ✕ Routing via gateways and load balancers
- ✕ Centralized and location dependent
- ✕ Architectural and operational complexity
- ✕ Multiple technologies to learn

What do we want?

Rethinking the fundamentals to simplify.

- ✔ Services that are implicitly discoverable
- ✔ Flexible M:N communication patterns
- ✔ Decentralized, zero trust security
- ✔ Intelligent routing without additional infra
- ✔ Localized data for decision making
- ✔ Single platform to architect and operate
- ✔ Single technology to learn

NATS simplifies **connectivity** and **communications** of application **services** and **data** spanning all **clouds**, **geographies**, and **edges**.

What is NATS?

Optimized for simplicity, adaptability, and community.

- Open sourced in 2011
- Incubating in CNCF
- 16MB static Go binary
- No external dependencies
- Client-server architecture
- 4 OSes and 7 arches
- 11 official client libraries

1000+

GitHub contributors

250M+

Docker pulls

30+

Community clients

7300+

Slack members

Teams ❤ NATS

Not convenient for only one role.

Developers

Build progressive distributed applications with location transparent **messaging**, **streaming**, **key-value**, and **object storage** APIs using a single client SDK, supported in all major languages.

Architects

Design and dynamically adapt topologies spanning **multiple clouds**, **geographies**, and extending to the **edge** without interrupting existing workloads.

Operators

Leverage built-in server **management**, **security**, and **monitoring** tooling enabling **complete visibility** and control over the entire system.

Year in review

What has been going on?

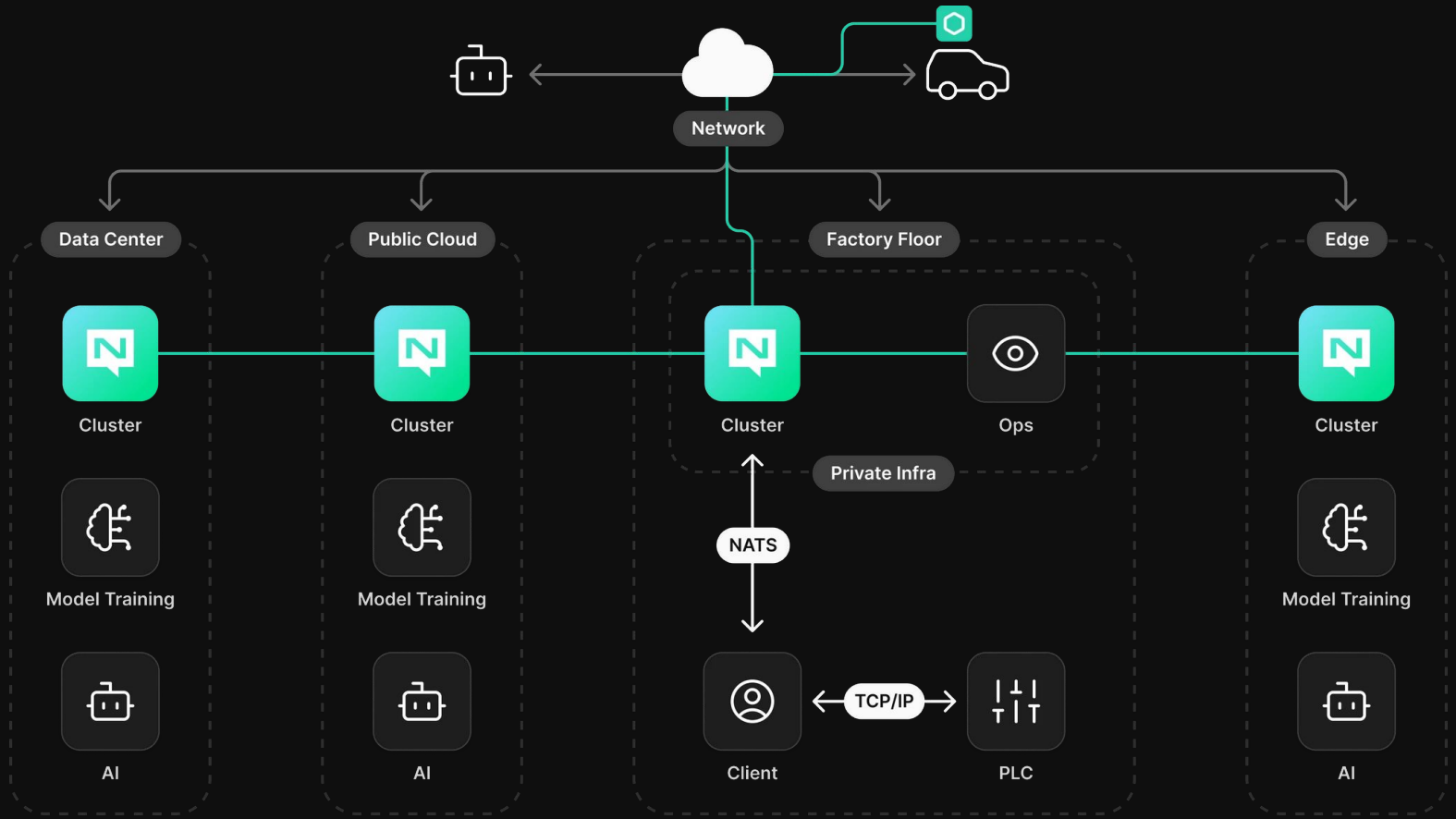
- NATS 2.10 release
  - Auth callout
  - v2 cluster networking
  - Multi-filter consumers
  - Stream subject transforms
  - Stream compression
  - Reduced recovery times
  - Opt-in TLS-first handshake
  - MQTT QoS2
- Simplified JetStream client API
- Services API
- Object Store API
- New .NET client
- New Helm chart
- New Spark connector

AI at the Edge

83% believe that **edge computing** will be **essential to remain competitive** in the future.

Ramalingam.R, Tung.Teresa. (2023). Leading with Edge Computing. Accenture.

*Survey included 2,100 C-suite executives across 18 industries in 16 countries.


Network
Data Center
Public Cloud
Factory Floor
Edge
Cluster
Cluster
Cluster
Ops
Cluster
Private Infra
NATS
Model Training
Model Training
Model Training
AI
AI
AI
Client
TCP/IP
PLC

Why NATS?

Adaptive and scalable connective technology.

- Topologies can span clouds and the edge(s)
- One consistent application connectivity layer
- One consistent security model
- Streams for storing and forwarding data
- Object store for storing and distributing models

Demo

nats-kubecon.vercel.app

Thank you

Resources and questions!

- Website - nats.io
- Slack - slack.nats.io
- Docs - docs.nats.io
- Examples - natsbyexample.com
- Podcast - nats.fm
- Newsletter - synadia.com/newsletter
- Screencast - synadia.com/screencast

Meet the team at booth N38!

Session Feedback