Synadia

MANIFESTO

Are Your Apps Truly Nomadic?

Breaking Free of Federated Architectures:
How Synadia is Redefining Distributed App Architectures for the Edge and AI

By Justyna Bak, Borja Etxebarria, Kevin Hoffman, Jean-Noél Moyne, Michael Röscheter, Byron Ruth

Are Your Apps Truly Nomadic?

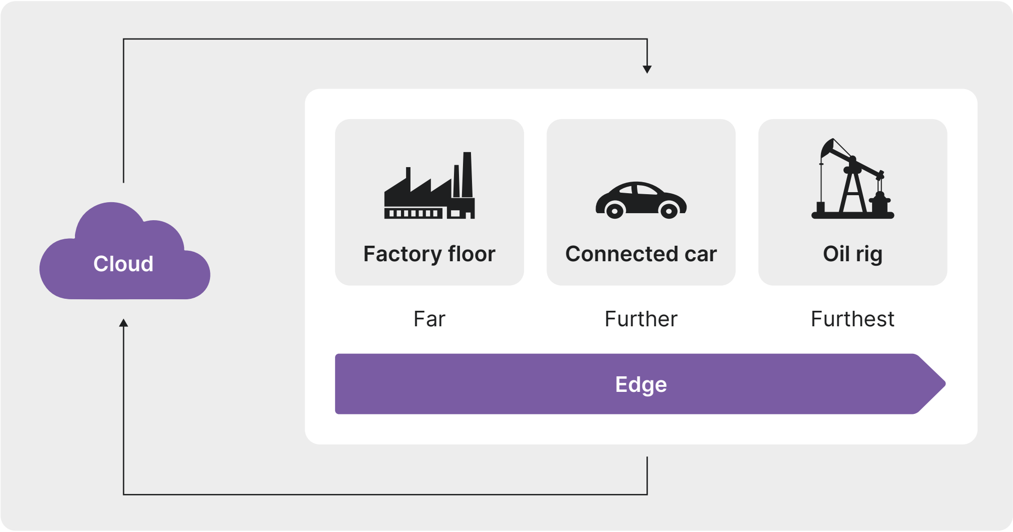
Truly nomadic applications are the next step beyond 12-factor apps, designed for edge computing, IoT, and AI. They are scalable, portable, and work reliably in challenging environments.

These apps stand out with their “batteries included” approach, enabling them to:

- **Work offline:** Stay functional without constant connectivity.
- **Handle spotty connections:** Operate smoothly even with unreliable networks.
- **Use resources efficiently:** Make the most of limited power, bandwidth, and hardware.
- **Adapt to changing conditions:** Adjust to shifts in environment or workload on the fly.

Nomadic apps can run anywhere – on a factory floor, in a connected car, or on a remote oil rig – no matter how far they are from the cloud or a data center.

Nomadic apps can move freely from cloud to edge



Nomadic Apps Manifesto

The cloud is the new mainframe. The edge is the next innovation frontier. NATS.io is the technology bridging the gap – powering real-time connectivity, data, and workloads across distributed application architectures, from cloud to edge.

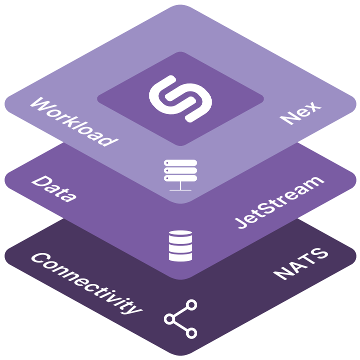
Distributed application architectures have evolved from Java monoliths to Kubernetes-managed microservices, and now to edge computing and multi-region, multi-cloud deployments. Each stage has brought significant advancements and new challenges, shaping the landscape of modern computing.

What's next?

Enter Synadia, founded by NATS inventor Derek Collison and home to its core maintainers. Synadia is revolutionizing distributed computing by:

1. Making apps truly location-independent and nomadic
2. Enhancing developer productivity and reducing operational overhead
3. Enabling decentralized, real-time data processing at the edge
4. Facilitating quick, localized decision-making with edge-deployed AI models

The Synadia Platform, built on the NATS foundation, is redefining how highly scalable, flexible **nomadic apps** are built, deployed, and managed for the edge.



The Synadia Edge-native Tech Stack:
All that apps need to thrive at the edge and become truly nomadic

Audience

- **Software developers** building cloud-native and edge-native applications
- **Software architects** designing scalable distributed architectures
- **Business leaders** looking to leverage technology for strategic advantage
- **AI specialists** deploying AI models at the edge
- **Industry analysts** analyzing technology trends

Summary

There's never been a better time to be an app developer

The evolution from Java monoliths to microservices has reshaped software development, emphasizing modularity, scalability, and resilience. Now, edge computing is emerging as the next frontier. Why? Edge computing is bringing data computation closer to data sources and end users, putting data where users need it most. By reducing latency and enhancing access to both data and services, it ensures faster, more efficient interactions for users in real time. The best news? Developers now have access to tools like NATS that simplify the building and management of complex, scalable applications for the edge-to-cloud or edge-to-datacenter continuum. This shift allows more **focus on innovation and less on infrastructure headaches.** With next-gen technology readily available, whether you're developing AI applications at the edge or enhancing enterprise systems, there's never been a better time to make a significant impact in the software world.

NATS: The high-speed highway for edge computing

With its low-latency and high-availability, NATS shines in edge computing scenarios, where speed and reliability are critical. As organizations rapidly embrace edge computing and AI, NATS offers a flexible and efficient system that integrates seamlessly with various platforms, making it the go-to choice for **simplifying the complex world of data communication.**

Cloud is the new mainframe; AI innovation moves to the edge

Edge computing is rapidly revolutionizing the AI economy by enabling real-time data processing and decision-making at the source. This shift reduces reliance on centralized cloud infrastructures, lowers latency, and optimizes bandwidth usage. NATS is a critical component in this new paradigm, providing the reliable, secure, and scalable **messaging capabilities essential for deploying AI models at the edge.**

Synadia: making your applications truly nomadic

Synadia, the force behind NATS, leads in advancing distributed application architectures. Synadia's core product, Synadia Platform, **enhances NATS with advanced management, monitoring, security, and scalability features,** making it ideal for developing and managing complex, resilient applications.

The platform builds on NATS's strengths–providing a lightweight, cloud-agnostic messaging system–with tools that streamline administration and improve security. This approach modernizes application development, much like how advanced frameworks enhance foundational technologies.

Synadia's open-source commitment fosters a collaborative ecosystem, driving innovation and ensuring NATS remains at the forefront of distributed computing. Together, they offer a versatile platform that meets the dynamic needs of today's digital environment.

01

Distributed Application Architectures and Messaging

"Enterprises are beginning to seek placement for workloads that have not migrated to the public cloud. This represents approximately 70% of all workloads."

Gartner October 2023 Press Release

Evolution of distributed application architecture

As we trace the history of distributed application architectures, we see a clear progression from monolithic systems to microservices and now to edge computing. Each stage has brought significant advancements and new challenges, shaping the landscape of modern computing.

Java Monoliths in the 2000s: Foundation of application portability

In the early 2000s, Java promised “write once, run anywhere” through the Java Virtual Machine (JVM). While enabling the first generation of cloud deployment, Java monoliths were often complex and hard to scale, with configuration management often hindering true portability.

Challenges Introduced: Configuration complexity, difficult scaling, and deployment challenges across varied environments.

Microservices with Kubernetes and Docker in the 2010s: Decentralizing architecture

The limitations of monoliths led to the microservices approach. Docker containerized these services and delivered on the “write once, run anywhere” promise, while Kubernetes managed their lifecycle, scaling, and fault tolerance. This greatly improved productivity, resilience, and scalability.

Challenges Introduced: Complexity in service discovery, networking, and data consistency across distributed services.

Nomadic apps supporting edge computing: The emerging frontier

Now, we are moving towards nomadic applications designed to operate at the edge–closer to data sources and end users. Edge computing reduces latency and bandwidth issues by processing data locally, requiring applications to be highly portable and adaptable.

Challenges Introduced: Managing connectivity, data synchronization, and dynamic scaling across diverse environments, including resource-constrained scenarios; inefficient use of computer resources.

2000	**Java Monoliths**
2010s	**Microservices with Kubernetes and Docker**
2020s	**Nomadic Apps at the Edge**

Evolution of messaging systems to support each emerging application architecture

As we trace the history of messaging systems, we see a progression from purpose-built solutions to much more capable, complex platforms, which provide a range of quality of service options from synchronous communication through fast streaming to HA-available asynchronous queues. Each stage has brought significant advancements and new challenges, shaping the landscape of modern distributed systems.

Pre-2000s	**RPC, TIBCO Rendezvous, IBM's MQSeries, JMS**
2000s	**SOAP**
2010s	**Web Sockets, Message Queues and Brokers like RabbitMQ, Streaming Platforms, NATS**
2017	**Synadia founded**
2024	**Synadia Platform Introduced**

Remote Procedure Calls (RPC): Early Synchronous Communication

In the early stages of distributed app architectures, RPC mechanisms like CORBA-, DCOM- and XML-RPC- enabled synchronous communication between distributed but tightly coupled systems. RPC allowed a program to execute code on a remote server as if it were local but still required a largely monolithic application design.

Old Problems Solved: RPCs simplified the development of distributed systems by abstracting the complexity of network communication, allowing developers to focus on business logic rather than network protocols.

New Challenges Introduced: Tight coupling between services, difficulties in scaling and integrating, and proprietary wire protocols and discovery mechanisms.

From TIBCO Rendezvous and IBM's MQSeries to JMS

As distributed systems grew in complexity, there was a shift towards messaging middleware like TIBCO's Rendezvous and IBM's MQSeries. These platforms facilitated more reliable communication through message queuing and provided support for various messaging patterns, laying the groundwork for the Java Message Service (JMS) standard, which further enhanced interoperability.

Old Problems Solved: These systems addressed the need for reliable message delivery and decoupling of application components, improving fault tolerance and load balancing.
Challenges Introduced: Complexity in implementation and management, dependency on specific vendor technologies, limited support for real-time processing, and active-passive fault-tolerance relying on expensive and specific hardware requirements.

SOAP in the 2000: Standardization with XML

The 2000s saw the emergence of SOAP (Simple Object Access Protocol), which standardized web services communication using XML. SOAP provided a more structured framework for message exchange, enabling interoperability across different systems and platforms.

Old Problems Solved: SOAP standardized the communication between disparate systems, integrating different technologies and platforms.
Challenges Introduced: Verbosity of XML leading to bandwidth and performance issues, complexity in implementation and maintenance, scalability challenges due to synchronous nature, and infrastructure requirements such as proxies and load balancers.

WebSockets in the 2010s: Real-Time Web Communication

WebSockets emerged in the early 2010s as a solution for real-time web communication, enabling full-duplex communication from Javascript in the web browser over a single HTTP connection. This technology allowed for live updates and interactive features in web applications, transforming user experiences.

Old Problems Solved: WebSockets addressed the limitations of HTTP's request-response model, enabling more interactive and dynamic web applications by supporting real-time data exchange.
Challenges Introduced: Scalability issues, limited message persistence, and complex server management.

Message Queues and Brokers: Reliability and Asynchronous Processing

Technologies such as RabbitMQ and ActiveMQ provided robust solutions for reliable message delivery and asynchronous processing. These systems decoupled message producers and consumers, allowing for better load balancing and fault tolerance.

Old Problems Solved: These platforms addressed the need for decoupling application components, enabling scalable and resilient system designs.
Challenges Introduced: Complexity in managing broker infrastructure, message persistence overhead, and potential latency issues.

Streaming Platforms: Handling High-Throughput Data

Apache Kafka revolutionized the way organizations handle high-throughput, low-latency data streams. Kafka's distributed, fault-tolerant architecture made it ideal for processing and analyzing large volumes of data in real time.

Old Problems Solved: Kafka provided a scalable solution for handling large volumes of streaming log data, enabling real-time analytics and processing.
Challenges Introduced: Complexity in managing Kafka clusters; ensuring data consistency; handling large-scale data ingestion; designed for streaming log data; fixing improper queuing, publish/subscribe or request/reply; and required partitioning in order to scale the consumers.

NATS and Lightweight Messaging: Simplicity and Performance

NATS is a proven, lightweight, high-performance messaging system designed for modern distributed applications. It offers low-latency, highly available messaging, queueing and request/reply with a minimal footprint, making it ideal for microservices and edge computing. With JetStream enabled, it enables persistence of real-time data streams, key-value buckets, and object stores.

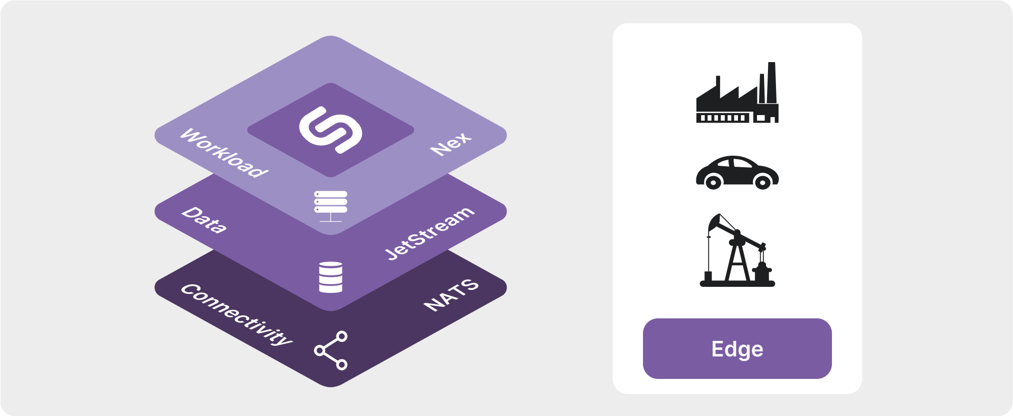
Old Problems Solved: NATS solved the problem of complex and heavyweight messaging and streaming systems by offering a simpler, smaller, more performant alternative, suitable for modern application architectures.
Challenges Introduced: Repurposing existing infrastructures for modern applications.

The Future: Edge Messaging and Beyond

With edge computing, messaging systems must evolve to handle decentralized, real-time data processing. NATS, with its lightweight design and efficient connectivity, is leading this transition, providing the backbone for nomadic applications that operate closer to data sources and end users.

This evolution towards edge messaging represents the next frontier, addressing the need for low-latency, high-throughput communication in increasingly decentralized systems.

The Synadia Edge-Native Tech Stack



02

Building Edge-Native and AI-Ready Apps

IDC's March 2024 forecast predicts global spending for edge computing will hit $232 billion in 2024 – a 15.4% jump from 2023 – and reach $350 billion by 2027.

IDC's March 2024 forecast

Getting apps ready for edge computing with NATS

The shift from centralized cloud applications to decentralized edge applications is driven primarily by the need to **process data closer to its source,** reducing latency, improving responsiveness, and lowering network costs. While it sounds like a no-brainer, processing data directly at the edge introduces its own set of unique challenges, such as intermittent power and network availability.

Building Cloud-Native Applications vs. Edge-Native Applications

BUILDING CLOUD-NATIVE APPS	PILLARS	BUILDING EDGE-NATIVE APPS
• Optimizing for high-performance computing • Implementing microservices architectures	Workload	• Designing for resource-constrained devices Ensuring resilience during network outages • Managing updates and versioning across distributed edge nodes
• Storing data with virtually unlimited cloud resources • Replicating and caching data globally • Ensuring high availability and protecting against data loss utilizing multiple data centers & regions	Data	• Storing data with limited local resources • Implementing local caching with eventual consistency • Ensuring data integrity in offline scenarios
• Managing inter-service communication • Load balancing across multiple instances and/or clouds • Managing API versioning and backward compatibility	Connectivity	• Optimizing for unreliable or intermittent network connections • Managing edge-to-cloud communication • Designing for offline functionality

What works for cloud applications does not translate to the edge

In the context of messaging solutions for edge and embedded systems, there's a crucial distinction between "what" we do in terms of data exchange and "how" edge systems are deployed and managed.

The "what" encompasses publish/subscribe messaging, service mesh, data streaming, and 'NoSQL' data storage. These components may look similar in an architecture diagram, suggesting a straightforward implementation.

But the "how" reveals significant differences in edge environments compared to traditional centralized systems. Unlike unitary systems with well-defined REST APIs, edge environments are characterized by:

- **Clustering and multi-clustering:** Edge systems often operate in a clustered configuration to ensure resilience, scalability, and distributed processing capabilities.
- **'Leaf Node' extension points:** In computer science, a "leaf node" refers to the final node in a hierarchical structure, like a tree, that does not branch further. In the context of edge computing, leaf nodes represent devices or systems located at the farthest edge of the network, often at the site of data generation. These nodes are integral for extending the network's reach and processing capabilities, enabling localized data handling and decision-making.
- **Built-in reliable store and forward:** Data mirroring and sourcing between clusters and leaf nodes ensure data integrity and availability, even in the face of network interruptions or failures.
- **Request/Reply mechanisms using subjects**: Unlike traditional systems that rely on HTTP endpoints, edge systems may use subject-based addressing for request/reply interactions, enhancing flexibility and reducing dependency on network infrastructure like proxies, load-balancers, and dynamic DNS service.

Edge solutions evolve gradually over time, often incorporating existing hardware and software that cannot be easily exchanged or upgraded without significant cost. The deployment and management of edge systems are complex, taking into account the constraints of already deployed components and the need for seamless integration with legacy systems.

"Nothing but NATS": A new way to building for the edge

Utilizing NATS and its execution engine (Nex) for your tech stack allows for greater flexibility and resilience in edge environments, where network conditions may be unreliable and system components need to operate with a degree of autonomy. The use of subjects for request/reply, rather than HTTP endpoints, further enhances the adaptability of these systems to varied and challenging edge conditions.

NATS is the most simple and versatile approach for tech stacks when building distributed applications in the era of edge computing, IoT, and AI. It simplifies the connectivity landscape, which traditional methods complicate with multiple protocols and manual configurations.

NATS has the potential to transform Edge Computing just as Kubernetes did for Microservices

From all the things that K8s brings into microservices, NATS can easily bring equivalent features to edge apps, such as service discovery, load balancing, self-healing setups, network security, observability, and multi-tenancy. Additionally, with little effort, it can help with config and resource management or rolling updates. If we consider that we can now deploy and manage workloads in NATS via Nex, we get automated deployment, scaling, and infrastructure-agnostic abstraction.

These comprehensive features of NATS not only enhance system robustness but also significantly reduce the communication overhead that developers often face. Developers reportedly spend 80% of their time managing communication overhead, leaving only 20% for building app logic. The Synadia Platform, leveraging NATS and Nex, aims to reverse this ratio, allowing developers to focus more on app logic and less on communication management.

NATS Lets Developers Spend More Time Building Applications

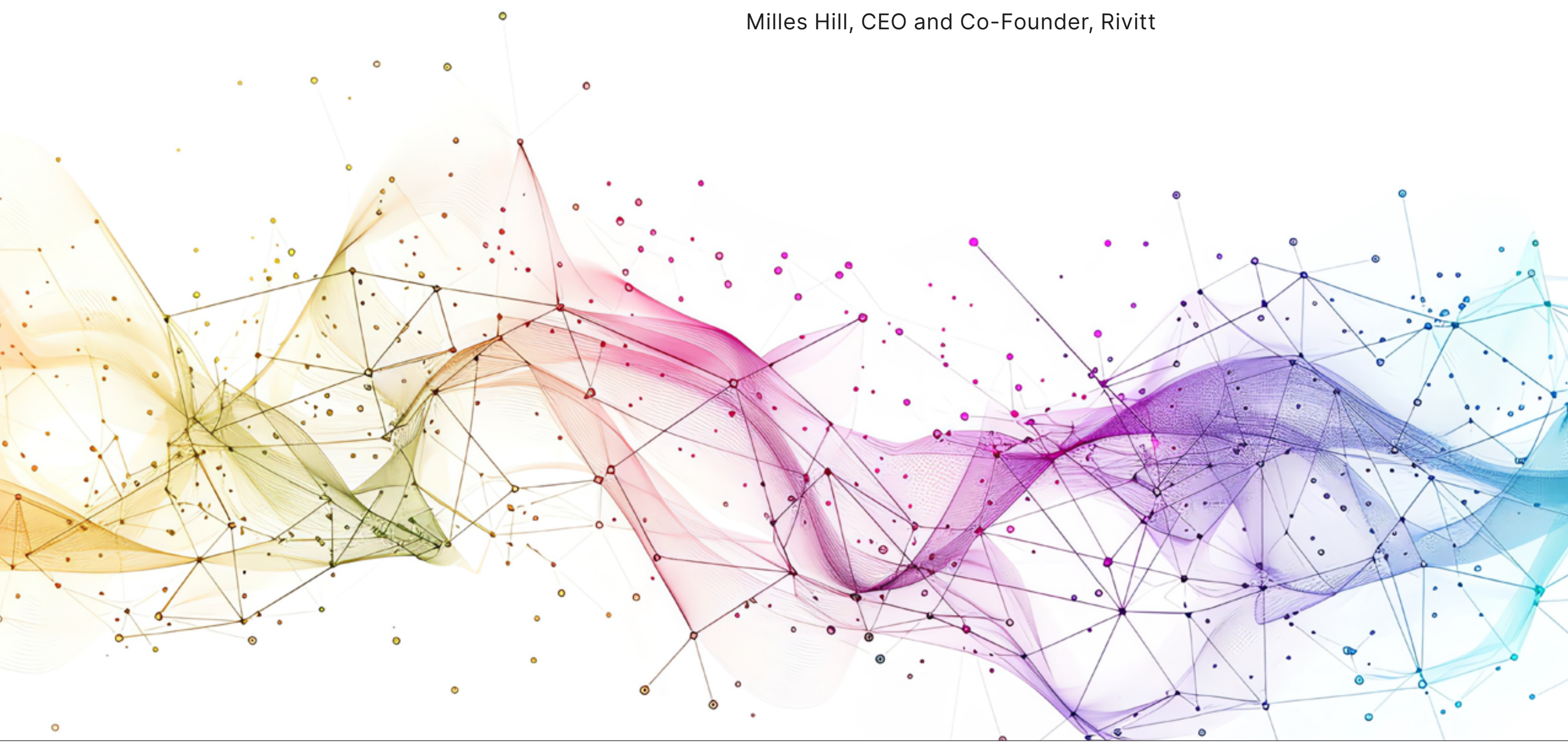
	BEFORE NATS	WITH NATS
Managing Apps	80%	20%
Building Apps	20%	80%

NATS advantages include:

- **Simplified Connectivity:** NATS offers a unified system supporting various messaging models, reducing latency and complexity.
- **Integrated Data Management:** With JetStream, NATS provides a unified data layer for streaming, key-value stores, and object storage.
- **Reduced footprint:** A single binary of less than 20MB in size provides the following functionalities:
  - Messaging and queueing (including MQTT server functionality)
  - Streaming (including reliable store and forward data transmission between the cloud and the edges)
  - Key/Value and Object Store
  - Service mesh
- **Efficient Compute Solutions**: Nex simplifies deployment and workload management, supporting true application nomadism.

In the evolving landscape of application architectures, NATS is a cornerstone for the next generation of distributed systems, especially those leveraging edge computing.

> ***"Synadia, with its NATS foundation, has enriched our solution for oil and gas companies by enabling real-time access to rapidly changing data at the very edge of harsh field operations–and then moving the data in milliseconds anywhere the customer wants for rapid decision-making and remediation."***
>
> Milles Hill, CEO and Co-Founder, Rivitt



Why NATS is the perfect tech stack for edge applications

Edge applications offer significant advantages over cloud applications in terms of scalability, flexibility, and component management.

Edge-Native Applications Offer Significant Advantages over Cloud-Native Applications

CATEGORY	CLOUD APPLICATION	EDGE APPLICATION	TECHNICAL BENEFITS	BUSINESS BENEFITS
SCALABILITY	Limited and complex scaling due to cloud infrastructure and costs. Increases network costs on the cloud	Scales up or down independently for each part of the model building process. Processes the input data locally and scales as the number of edges increases and does not require all of the data to be sent from the edge to the cloud	Handles varying workloads and data volumes with ease. Reliable and secure data storage and transmission to and from the edge. Focus only on business logic	Efficiently accommodates growth and changing demands. Improves developer efficiency and development time
FLEXIBILITY	Limited by centralized cloud architecture for data storage and event collection	Flexible data storage and event collection	Supports various storage solutions and data sources	Adapts to different business needs and operational scenarios
COMPONENT SUBSTITUTION AND UPGRADES	Difficult to update individual components without affecting the whole system	Easy to substitute, scale, and upgrade individual components	Minimizes disruption and risk during updates	Reduces downtime and enhances system reliability and performance
GEOGRAPHICAL PLACEMENT	Centralized cloud components face latency and redundancy issues due to location constraints	Components can be geographically distributed as needed	Improves data sovereignty, latency, and redundancy	Enhances performance and compliance with regional regulations

Example: a truly nomadic app built with nothing but NATS

One of Synadia's developers has built a satellite tracking application, to demonstrate a truly nomadic app that is able to run efficiently and consistently across diverse environments without reconfiguration or code changes. It is a powerful, real-time visualization tool built entirely on NATS.

This single-binary app tracks thousands of active satellites orbiting Earth, updating their positions in real time at 60-90 frames per second. It demonstrates NATS' capability to handle high-volume, low-latency data processing without relying on external databases.

The app includes features like:
1. Live satellite tracking visualization
2. On-demand processing of gigabytes of video data
3. Image comparison and analysis tools

What makes this app truly remarkable is its nomadic nature. It can run autonomously on anything from a Raspberry Pi to a large cluster, seamlessly scaling from embedded local operations to distributed cloud deployments without code changes. This flexibility showcases the power of NATS in creating efficient, portable, and scalable applications for complex data processing and visualization tasks. Check out the NATS attributes for nomadic apps like this satellite tracking application.

- **Single binary deployment:** The entire application, consisting of dozens of services, can be deployed as a single binary. This allows it to run autonomously on any device, from a Raspberry Pi to a large cluster.
- **Flexible communication:** All services communicate with each other over NATS, regardless of whether they are running in the same binary or in case they were distributed across containers/machines.
- **Environment-agnosticism:** The same code can run embedded locally or connect to a global NATS supercluster, with only environment variables changing.
- **No external dependencies:** The app uses NATS for all communication, data storage, and processing, eliminating the need for external databases or services.
- **Scalability:** It can easily scale from running all services in one binary to distributing them across thousands of containers without code changes.
- **Offline capability:** The app can function completely offline when needed, then sync data when connectivity is available.
- **Consistent performance:** The app handles real-time data for thousands of satellites and processing gigabytes of video data smoothly on a single machine.
- **Cross-platform:** The demo worked seamlessly across browsers, phones, and different devices.
- **Edge-ready:** The app can easily switch between running locally and in any environment, including edge devices, by simply changing connection details.
- **Nomadic by design:** The development appproach prioritizes location-independence, allowing services to run anywhere without code modifications.

Use Case

Syandia's customer, a manufacturer of specialized medical devices, requires its devices to be operated by qualified and trained professionals for consistency and safety.

Challenges

As a result, the customer's staff depends on executing local product training at their facilities worldwide as part of the overall service model. Each local training event needs to be captured and all documentation stored as part of the certification process. The captured information covers a broad range of formats: video recordings, product training results, device logs, personal notes, and training plans.

Sending all traffic back to the central cloud was cost-prohibitive and sometimes challenging for remote sessions. At the same time, the customer wanted to explore AI-based data analytics for a range of use cases: for example, online, in-training feedback, offline analysis, and AI-based training.

NATS Solution

The NATS native edge capabilities allowed the customer to create a simpler on-demand data delivery platform. Data is stored locally, is immediately available, and is only transferred where and when required.

Results

- User interactions with the devices during the training stream back to an AI engine in real time, where networks allow it. Immediate feedback is provided when common user error or bad practices are being detected.
- Training data for AI models is transferred asynchronously after the event. The NATS capability to buffer and seamlessly restore transfers through "leaf nodes" makes data collection simple, even on poor networks or when devices are switched off after business hours.

The role of edge in the AI economy

Edge computing is a significant trend impacting many applications, but it is especially transformative for AI. Applying AI at the edge means using existing data in entirely new ways. Information initially intended for quality assurance and documentation can now be used for dynamic decision-making in ongoing business activities.

Edge computing has two primary roles to play in the AI economy:

- Collect, process, and distribute data from all points of generation, such as factory floors, retail POS locations, or connected vehicles. This data is then used for AI model training in the cloud.
- Perform AI model inferencing as close as possible to where the data is created: at the edge.

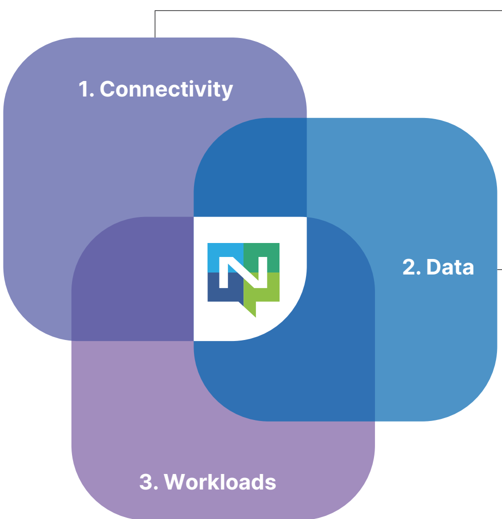
AI at the edge is not a replacement for AI in the cloud. Rather, the future of AI includes both large centralized and smaller, specialized AI models that will co-exist, while interacting to create the business process outcome or desired user experience.

Why NATS is the perfect tech stack for AI at the edge

NATS excels in simplifying communication, enhancing scalability, and improving efficiency for edge applications, including AI:

MODZY

- **Optimal for high-value / low-volume data:** Transmit generated data reliably and securely between the edge and the cloud. NATS ensures these high-value events, such as AI-identified threats and opportunities, are not lost. For example:
  - Processing video feeds to identify potential fires in arenas, factories, or warehouses.
  - Analyzing sensor data to detect excessive machine vibration, using AI models to predict and alert for maintenance needs before a breakdown occurs.
- **Optimal for low-value / high-volume data:** Deploy NATS at the edge to process data at the source. This involves high-volume, high-velocity, low-value events (e.g., video frames) distributed between edge applications. For instance:
  - Our customer Modzy processes sensor data at the edge and sends only inferred events to the cloud for alerts.
- **Bidirectional functionality:** NATS facilitates reliable data distribution both from the edge to the cloud and vice versa. This is crucial for updating AI models, distributing binaries, or any time-sensitive data changes to and from multiple edge locations.
- **Ready-to-use solution:** NATS and JetStream eliminate the need for extensive development and debugging of infrastructure code, allowing you to focus on application logic without reinventing the communication layer.



NATS provides a flexible, globally distributable connectivity layer that allows multiple applications, services, and organizations (such as different business units or companies) to interact securely across any environment.

For different patterns, NATS offers different data quality of service guarantees. Without the persistence layer turned on, NATS offers “at most once” delivery of data. With persistence turned on, NATS offers exactly once and “at least once” quality of service semantics. JetStream, the NATS persistence layer, enables these last two quality of service modes as well as data streaming and key-value and object storage building blocks. The data in these primitives is persisted, served, replicated, and moved between any points of presence, from edge to cloud (even with limited connectivity).

NATS Execution Engine (Nex) allows applications and processes that use NATS to be managed, deployed, and executed across any nodes in the system with zero re-platforming. For example, applications from the cloud to the factory floor to improve latency or break a single centralized application into more localized units of compute that run closer to the end user.

NATS-based Tech Stack is Ideal for Building Edge AI Applications

CATEGORY	EDGE APPLICATION WITHOUT NATS	EDGE APPLICATION WITH NATS	NATS TECHNICAL BENEFITS	NATS BUSINESS BENEFITS
AI MODEL CONSUMER MANAGEMENT	AI model consumers must register and deregister for updates	AI model consumers subscribe to updates on well-known subjects	Eliminates the need for managing consumer registrations	Reduces administrative over-head and simplifies maintenance
AI TRAINING DATA DISTRIBUTION	Data collectors need to discover and connect to sample generators	Data collectors publish data to well-known subjects	Simplifies data distribution and removes discovery protocols	Speeds up data integration and reduces complexity
AI MODEL INFERENCE SCALING	Scaling requires reconfiguration and manual load balancing	Inference processes are automatically balanced using round-robin	Enables seamless and immediate scaling	Improves scalability and responsiveness to workload changes
AI MODEL STORAGE AND RETRIEVAL	Model generation processes store and distribute models to consumers	Models are stored in NATS persistent storage and accessed as needed	Provides on-demand access, reducing network load and latency	Ensures timely access to the latest models
AI MODEL UPDATE NOTIFICATIONS	Consumers require custom logic for reconnecting or polling	Consumers receive real-time notifications of new models	Supports efficient on-demand generation and notifications	Increases efficiency and keeps consumers up-to-date

Why is the event-based communication paradigm superior for edge AI

AI is an evolving and inhomogeneous use case, flexible data access is required for both AI training and executing AI queries. The size of AI models as well as their deployment location are equally important. The largest and most capable models will stay centralized, while specialized solutions are appearing in medical devices, integrated into industrial QA and consumer electronics.

NATS enables applications with deployment models to communicate transparently and efficiently, whether those are medical scanners or large compute farms for AI and data aggregation. By moving the service endpoint to a NATS subject-based addressing, rather than HTTP load balancers and DNS server names, services can be redeployed and re-architected without breaking the solution or touching the configuration. NATS seamlessly supports asynchronous (streaming) as well as synchronous (service) use cases in the same architecture.

Examples of real AI applications implemented with NATS

AI is impacting every industry at an increasing velocity. The NATS community and ecosystem are building applications and services across various sectors, including:

- **Industrial Manufacturing & IoT:** Predictive maintenance, process optimization, and digital twins.
- **Autonomous Vehicles:** Optimized routes and safety.
- **Green Energy:** Energy optimization.
- **Retail:** Personalized shopping experiences and real-time inventory management.
- **FinTech:** Regulatory compliance and algorithmic trading.
- **Government:** DoD/IC communities, e.g., drone CUAS.

03

Synadia Platform

“You cannot simply forklift apps from the cloud to the edge.”

Derek Collison, CEO and Founder of Synadia, NATS designer

The ultimate solution for building distributed apps in the era of edge computing, IoT and AI

Synadia's tech stack, based on three key open source technologies – NATS, JetStream, and Nex – supports distributed systems architectures for edge computing, including AI at the edge. Here's how it compares to traditional cloud-native approaches.

Synadia's Edge-Native Tech Stack

CLOUD NATIVE APPS TECH STACK	PILLARS	BUILDING EDGE-NATIVE APPS WITH SYNADIA
• **Container orchestrators / runtimes** (e.g. Kubernetes, ECS, Fargate, GCP and Azure counterparts) • **Serverless runtime** (e.g. AWS Lambda, GCP/Azure counterparts)	**Workload**	**NEX**
• **Streaming** (e.g. Kafka, Pulsar, Red Panda, Kinesis, Azure Event Hub) • **Key value stores** (e.g. Redis, DynamoDB) • **Object stores** (e.g. Minio, S3, GCP/Azure counterparts)	**Data**	**JetStream**
• **Message queues** (e.g. RabbitMQ, SNS, SQS, GCP/Azure counterparts) • **IoT connectivity** (e.g. EMQX, other MQTT brokers) • **Service discovery** (e.g. Consul, Kubernetes services, service mesh) • **Micro-service connectivity** (e.g. RPC, API calls, OPC-UA) • **API gateways** (e.g. Kong, AWS API gateway) • **System-level observability** (e.g. Splunk, Datadog)	**Connectivity**	**NATS**

The Open Source advantage

One of the key strengths of NATS and the Synadia Platform lies in their open-source nature. Being open-source means that NATS benefits from a vibrant community of developers and contributors who continuously improve and expand its capabilities. This collective effort leads to rapid innovation, rigorous testing, and a wide array of features that are constantly being refined.

Open-source projects also provide transparency, giving users the confidence that the technology is secure and reliable. Moreover, the flexibility and adaptability of open-source software allow organizations to customize and extend NATS to meet specific needs, integrating it seamlessly with other technologies.

Synadia leverages this open-source foundation to offer robust enterprise-grade solutions, ensuring that businesses can scale and innovate without being locked into proprietary systems. This open-source advantage not only accelerates the adoption of new technologies but also fosters a collaborative ecosystem where users and developers can contribute to the ongoing evolution of NATS and its applications in distributed computing and edge AI.

Synadia Platform: NATS elevated to an enterprise solution

Synadia to NATS is like an Integrated Development Environment (IDE) to a programming language – while the language provides the core syntax and functionality, the IDE enhances the development process with tools for debugging, code completion, and project management, making it easier and more efficient to build complex applications.

Synadia Platform:
Everything developers love about NATS and more

Synadia Platform with NATS offers:

Reliability: The platform ensures 99.99% uptime, with 24-hour patching and zero downtime upgrades, keeping your systems consistently online.

High performance and security: Leveraging a hardened configuration on high-performance cloud infrastructure, the integrated NATS cluster is optimized for both security and efficiency.

Minimal management overhead: Simplify your operations with intuitive account and user management, comprehensive monitoring, and observability tools, facilitating easy management of data streams, key-value stores, and objects.

Advanced features:

- Benefit from the management HTTP API and OIDC integration for secure, streamlined user management.
- Prepare for future needs with Nex, which simplifies the deployment and integration of transient and long-lived NATS services.

Expert support: Access to best practices and expert guidance helps you build robust applications using the platform's services and NATS.

Multi-region readiness: Synadia's tech stack, built on NATS, enables consistent and rapid design of modern distributed systems. It is adaptable to single-cloud deployments and ensures future-proof security when integrating multiple cloud providers, regions, and edge locations.

AI at the Edge with Synadia

The Synadia Platform is particularly suited for edge computing environments, which require efficient handling of real-time data and robust connectivity. This makes it ideal for AI at the edge, where quick, localized decision-making is crucial. By processing data closer to its source, the platform reduces latency and bandwidth usage, enabling real-time analytics and decision-making.

- **Local processing:** Allows for immediate data processing and analysis at the edge, crucial for applications such as predictive maintenance, autonomous systems, and real-time monitoring.
- **Reduced latency:** Minimizes the time lag in data transmission, essential for time-sensitive AI applications.
- **Bandwidth efficiency:** Reduces the need to send large volumes of data to central servers, lowering costs and improving efficiency.
- **Scalability:** Easily scales with the addition of new edge devices, supporting the growing data and processing demands of AI applications.

The Synadia Platform is designed to simplify the complexities of distributed systems, making it an ideal choice for handling real-time data, deploying edge computing solutions, or managing large-scale IoT networks. Its focus on reliability, security, and ease of management allows developers to focus on building and optimizing applications without the usual operational burdens.

Looking ahead

The next big innovations in edge and connectivity that we at Synadia foresee in the coming years include:

- **Increased AI capabilities at the edge:** As processing power improves and AI solutions become more specialized and thereby smaller, more advanced AI and machine learning capabilities will move from the cloud to edge devices.
- **5G and edge computing convergence:** The continued rollout of 5G networks will transform daily activities both private and at work by providing sufficient bandwidth for mobile devices to integrate with AI service from anywhere.
- **Edge-to-cloud continuum:** We'll likely see a more seamless integration between edge and cloud computing, creating a continuum where data and workloads are dynamically deployed based on requirements.
- **Hyperlocal data centers:** There will be a proliferation of small, hyperlocal data centers to support edge computing needs. Companies like Akamai are already repurposing existing infrastructure to create networks of edge data centers.
- **Industry-specific edge solutions:** We'll see more tailored edge computing solutions for specific industries like healthcare, manufacturing, retail, and agriculture, addressing unique challenges and opportunities in each sector.

Conclusions

The evolution from monolithic applications to Kubernetes-managed microservices and now to edge-native nomadic apps represents a journey toward more dynamic, resilient, but often inefficient computing architectures.

AI at the edge demonstrates how NATS is pivotal for efficient deployment and communication in the edge computing era. NATS simplifies the connectivity, data management, and compute functionalities essential for today's distributed, edge-focused AI applications.

Just as Kubernetes was crucial for the initial microservices boom, NATS is essential for the successful deployment and management of AI-driven applications at the edge, ensuring they thrive in a decentralized computing environment.

Synadia's tech stack, based on NATS, allows consistent and rapid design of modern distributed systems that can adapt to single cloud deployments but, more importantly, remain future-proof and secure when multiple cloud providers, regions, and edge locations are added.

CONTACT US

To learn more about how Synadia can transform your approach to distributed applications, get in touch with our team.

Authors

Justyna Bak, VP of Marketing, Synadia

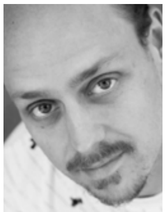
A full-stack marketer with expertise in Data Analytics, AI, AppDev, and Networking, and a recognized influencer, panelist, and presenter (Google, HBR)

/in/justynambak /justyna_bak

Borja Etxebarria, Solutions Engineering Director, Synadia

Field Engineer with long term experience on stream processing, messaging and distributed systems. Advised and supported companies deploying these technologies to successfully enhance their business. Previously: Confluent/Kafka, TIBCO

/in/borjaetxebarria



Kevin Hoffman, Director of Engineering, Cloud Services, Synadia

A startup co-founder, former CTO, and accomplished author. Expert in building distributed systems, His passion: bringing joy to development through amazing developer experiences

/in/kevinshoffman https://kevinhoffman.blog

Jean-Nõel Moyne, Field CTO, Synadia

A full-stack CTO with extensive experience in distributed systems, messaging, networking and data stores and products spanning all of these attributes. Previously: Lava.ai CTO,TIBCO Software Fellow and LBL Staff Scientist

/in/jean-noel-moyne /JohnnyXmas

Michael Röschter, Solution Architect EMEA, Synadia

Field Architect with deep history in distributed business systems. Experienced in finance, telco, automotive, logistics, manufacturing and embedded systems. Focused on designing the best platform to support the customer's business process. Previously TIBCO, Teradata, Denodo, Azul

/in/michaelroeschter

Byron Ruth, VP of Product and Engineering, Synadia

A leader, enabler, and engineer with expertise in distributed systems and a background of building large scale application platforms for pediatric research.

/in/byron-ruth

007-MF-2-24-11