Streaming, Messaging and Persistence for Personal.ai

with NATS.io

ABOUT PERSONAL AI

Location: California, U.S.A.

Employees: ~25

Industry: Software, Consumer Technology, Media

Tech Stack: Kubernetes, NATS+JetStream, MongoDB, Postgres, Elastic, TensorFlow, Istio, VueJS, AWS

Overview

Personal.ai is a revolutionary artificial intelligence platform that's trained on the memories provided by conversations, social media interactions and other forms of communication. The system builds an AI model for each user, with the mission of empowering every individual to own their intelligence and be their own thought leader. The AI is self-trained on data that an individual creates and captures as life happens. Personal.ai automatically categorizes all data and creates a structured view of an individual's life that enables many use cases including time capsules, AI-powered mentorship, always-on AMA and interactive NFTs.

Personal.ai uses NATS to connect its various microservices, reduce latency for job completion and reduce infrastructure costs. The backend of the application runs in Kubernetes clusters in Amazon Web Services (AWS) with NATS providing connectivity for streaming, messaging and data persistence with NATS JetStream.

Background

Personal.ai is attempting something that has never been done before – to literally create a personal artificial intelligence agent for creators and people to give them creative and intellectual superpowers. Personal.ai will also serve as a living memory construct that can either augment a user's own memory or talk to other people with conversations and responses that interpret how the user would respond and think. "I was having a conversation with my dad a while back. Basically, he was saying, 'Hey, I never see you. I don't really talk to you very much. Do you think there will be a day that I can talk to you like I talk to Alexa?,' " explains Personal.ai CTO Sharon Zhang. "That sounded like a really good problem to solve."

Personal.ai leverages various technologies, such as automated speech recognition, natural language processing and content collection to capture all media, content and spoken words generated by a user and turns it into structured data. Unlike most AI companies that rely on larger generalized data models, Personal.ai constructs a model for each user in about 20 minutes. The model self-trains and improves over time based on the content it ingests and processes. The user controls their personal data and content; all processing takes place in a cloud environment leveraging the Oasis blockchain technology to track changes and maintain a chain of custody. "It's really thinking about AI and how we can use blockchain to preserve privacy while solving a core human problem," says Zhang.

Application Architecture

The Personal.ai application incorporates many thousands (and eventually millions) of parallel streams of data at the same time. A user might require multiple streams simultaneously to capture voice data, online or web activity and content from other sources. Each message is critical because it consists of unique data. A persistence layer preserves stream data (audio, text, etc) until a worker is available for processing. Personal.ai opted for a unidirectional pub/sub model that pushed data labeled by subject to workers before sending it after processing into databases for storage. Personal.ai uses WebSockets to collect data from traditional HTTP sources and uses a VueJS front-end for audio capture. Audio streams are processed by Speechmatics, which provides automated speech recognition as a service. The data storage layer includes a Postgres database for storing relational information about users, a MongoDB object store as intermediate storage for fast writes and an Elastic storage layer for longer-term storage and quick queries. Personal.ai also stores some data in the AWS S3 Object Store.

PERSONAL.AI NEEDED

a messaging and data streaming system that provided:

- Fast speeds with low resource requirements
- A flexible persistence layer
- A broad array of client languages
- Rapid scalability
- Low management overhead
- Flexibility to follow streaming or messaging conventions
- Always-on reliability

Challenges: Resiliency, Reliability, Cost, Extensibility and Management Overhead

This unique application architecture, the strict performance requirements and the large number of components presented many DevOps challenges."We need to be able to manage these parallel streams at the same time," says Bala Sista, VP of Engineering at Personal.ai. "We have thousands upon thousands of streams going on at the same time, each going to their own unique data model." Additionally, Personal.ai has multiple microservices for other middleware and backend functions as well as external services for automated speech recognition. Personal.ai uses a hybrid on-chain/off-chain instance of the Oasis blockchain to enforce data permissions and verify and publish data changes to the distributed ledger. "There are a lot of moving parts in the system," says Sista.

Because the volume, number and type of data streams are unpredictable, Personal.ai required a fast-scaling system and an always-on messaging and streaming service. An individual user might have multiple streams running simultaneously, either for multiple audio streams or data ingested from social media and community forums, to name two examples. With highly sensitive data flowing through its system, Personal.ai wanted robust user segregation. The individual data models required to create each user's AI meant that Personal.ai would need to manage a large number of subjects through its messaging and streaming infrastructure. Sista wanted fast messaging with a small footprint to save costs.

Because Personal.ai used a wide variety of software languages and technologies, it required ready-made clients for easy integration into the messaging and streaming infrastructure. To hold onto user data until it was sure all necessary processing had taken place, Personal.ai required a flexible persistence and caching layer. The persistence layer would ideally be integrated with the message and streaming infrastructure. Lastly, the messaging and streaming system had to be reliable with high availability and able to operate in lower-resource environments like individual smartphones.

Sista initially built Personal.ai using Amazon Lambda serverless functions with a Simple Queuing Service (SQS) and Simple Messaging Service (SMS) to call up the workers and manage the pub/sub infrastructure. Personal.ai deployed a Redis key-value store for persistence. A bug in the Personal.ai health checks resulted in a massive AWS bill. Other concerns emerged when bottlenecks appeared in data pipelines and management of expensive GPU instances used for specific processes was imprecise. "We knew then we needed to move to something else and Kubernetes was the best option," says Sista.

In this new approach, each user and their associated streams and data became a Kubernetes micro-pod in Amazon's managed EKS Kubernetes service. Sista also wanted to move away from the SMS/SQS pairing to a more robust, reliable, flexible and easier to manage messaging and streaming infrastructure. Personal.ai ruled out Kafka from the start due to its heavy infrastructure requirements, scaling challenges and high management overhead.

Other messaging systems, such as RabbitMQ and MQTT, were also ruled out due to the inability to easily run active-active configurations at scale or to handle streaming data, plus considerable overhead and resource requirements.

Why Personal.ai Chose NATS with NATS JetStream

Sista had previously heard of NATS, a well-known and widely used open source connection fabric hosted by the Cloud Native Computing Foundation (CNCF). NATS provided both data streaming and traditional messaging capabilities. NATS had also just added a new persistence layer called NATS JetStream,which included a key-value store for caching and materialized views.

TECHNOLOGY BENEFITS

- Low latency, high performance
- Short learning curve
- Small footprint and resource requirements
- 45+ clients and pre-baked connectors
- Combines data streaming with persistence layer and KV store
- Can scale horizontally or vertically in seconds
- Works equally well as a data plane and as a management plane
- Extensible to SaaS++ model with Leaf Nodes

BUSINESS BENEFITS

- Low cost with high performance
- Open source and hosted by CNCF
- Simple and robust data security
- Reliable – never goes down and doesn't lose data
- Minimal management overhead – less than one FTE
- Covers use cases of multiple other solutions (messaging, streaming, caching, KV store)

NATS proved to be fast, flexible and easy to manage. NATS also has a extensibility feature called Leaf Nodes, which allows users to connect NATS to on-device or on-premise NATS instances. With Leaf Nodes, NATS can behave like a single distributed system that would enable the AI and data processing to run directly on edge devices like smartphones without requiring connectivity back to the cloud. The edge caching would also enable eventual data consistency and ensure that captured data could persist until all processing was finished – on cloud-based GPUs, for example. With 45 language clients and counting, Sista found that the NATS ecosystem covered everything

"We compared all the messaging and streaming technologies and it was clear right away that NATS checked all the boxes," says Sista. Personal.ai was able to get their first NATS test environment up in a matter of minutes.

"The YAML manifests are very clear and easy to use. The way NATS is designed makes startup quick and painless," says Sista, who notes that the community and the Synadia team were extremely helpful and responsive.

Personal.ai found that NATS installed on small public cloud instances was extremely cost effective compared to other messaging and streaming options like Kafka and RabbitMQ.

Despite the smaller compute capacity of these instances, NATS was still handling high volumes of streams and messages without any performance lags. "I can run NATS on my laptop and it always connects and works well," explains Zhang. "I could never do that with the other messaging and streaming technologies."

The wide number of clients mean that Sista could connect, observe and monitor all internal services using NATS. With JetStream, Personal.ai was able to eliminate a separate key-value store and simplify its architecture. Sista also configured NATS to manage all of Personal.ai's microservices and external services for ASR. Most importantly, NATS proved highly reliable and available, "It just doesn't go down," says Zhang, "We can always count on NATS."

With Synadia Platform, businesses and developers can focus on innovation rather than infrastructure, accelerating their journey to becoming truly data-driven, edge computing-enabled, AI-powered organizations.

To learn more about how Synadia can transform your approach to distributed applications, visit us for a free trial or contact our team for a personalized demonstration.

FREE TRIAL

CONTACT US

About Synadia
Synadia provides a secure, scalable, and high-performance data and communications platform designed for distributed systems. It empowers developers and enterprises to accelerate the delivery of distributed applications. Synadia leverages NATS, a connective technology, to enable real-time, secure communication across cloud, on-premises, edge, and IoT environments. NATS is an open-source platform powering thousands of applications globally. Founded in 2017 by the creator of NATS, Synadia is backed by leading VCs and strategic investors, including Forgepoint Capital, True Ventures, Bold Capital Partners, LDVP, Singtel, Accenture, and Samsung Next. Synadia's diverse customer base ranges from innovative startups to Global 500 enterprises in Finance, Retail, Automotive, and Industrial Manufacturing to innovative startups across FinTech, AI, Green Energy, and Gaming. Learn more at https://www.synadia.com/.

About NATS
NATS is a connective technology built for the ever increasingly hyper-connected world. It is a lightweight, low-latency technology that enables applications to securely communicate across any combination of cloud vendors, on-premises, edge, web and mobile, and devices. NATS consists of a family of open source products that are tightly integrated but can be deployed easily and independently. NATS is unique in its simplicity and performance, and as a result powers some of the largest production environments. NATS is being used globally by thousands of companies, spanning use-cases including microservices, edge computing, mobile, IoT and can be used to augment or replace traditional messaging.

011-CS-1-25-01