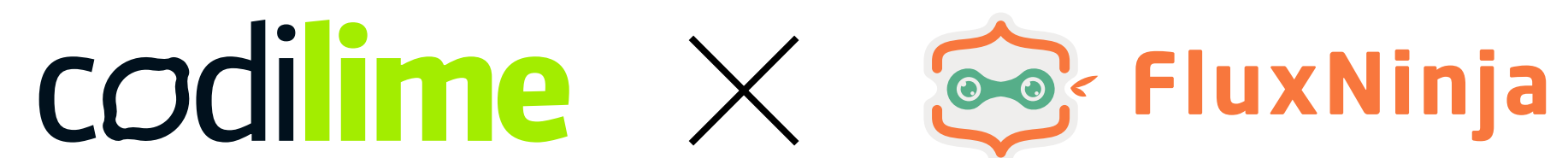


# FluxNinja

## case study



### Development of Intelligent Load Management for Cloud-Native Apps

FluxNinja is a cloud-native application reliability platform that provides a unified control layer for observability and control of microservices-based applications. The platform is designed to help platform and site reliability teams boost their productivity by automating reliability processes such as prioritized load shedding, intelligent autoscaling, client-side rate limiting and distributed rate limiting. FluxNinja achieves this through its open-source project, Aperture, which uses a declarative policy language expressed as a signal processing graph to go from telemetry to actions within minutes.



# Challenge



The FluxNinja development team aimed to create a modern, cutting-edge architecture for the Aperture open-source project. They wanted to build a platform that met the demands of a contemporary microservices-based application while incorporating advanced technical solutions, thus adding complexity to the project.

1. **Modern architecture:** The developers sought a scalable, modular, and maintainable design, requiring research and implementation of the latest industry patterns and best practices.
2. **Advanced technical solutions:** The team integrated innovative solutions to optimize performance, efficiency, and adaptability, exploring new algorithms, data structures, and technologies to keep Aperture at the forefront of application reliability solutions.



# Challenge



Looking at the more technical details: building this intelligent load management platform was based on policies which use fully **dynamic and configurable pipelines for metrics ingestion** to enable the use of any label or property of a request.

One of the most crucial parts of the project was the creation of policies. Policies are small "**reliability applications**" designed to provide a way to constantly observe and control network traffic and react to various scenarios; for example, load shedding or concurrency limiting. Policies can also be dynamically configured. However, for a policy to be able to make a decision, the requests need to be carefully classified, labeled and processed. To achieve this result our team decided to use OpenTelemetry Collector.



# Challenge



Moreover, each request coming into the system has to be processed and analyzed by many Aperture components without a noticeable increase in CPU usage, and without overloading the system and becoming a hindrance, which was highly challenging. By preparing a configurable list of OpenTelemetry receivers, processors and exporters we were able to quickly digest a great amount of traffic and **feed the policy with the data** it needed to properly actuate.

By overcoming these challenges and embracing modern architecture and advanced technical solutions, the Aperture developers built a robust load management platform. FluxNinja's state-of-the-art product effectively handles today's microservices-based applications, maintaining optimal performance and reliability.



# Services we provided

Frontend development



Backend development



DevOps

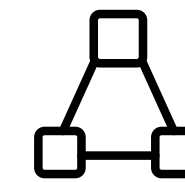


# Results & benefits

Broad deployment options: Aperture can be deployed on almost any infrastructure, enabling seamless integration into existing setups and ensuring compatibility with various environments.



Customizable policies: Users can choose from predefined policy blueprints or create custom policies based on specific metrics, catering to diverse application reliability requirements.



Multi-language support: Designed for a global audience, Aperture appeals to a wide customer base by offering support for multiple languages.



# The scope of the project



The project scope reflects the ambitions of our partner towards their flagship product. FluxNinja's primary goals were:

1. Creation of **Aperture open-source software** that enables:

- classifying traffic by extracting labels from requests
- observability of labeled traffic
- assigning policies to perform an action when violation of SLO occurs

2. Creation of **FluxNinja ARC** – a paid system that enables users to view dashboards related to their services and applied policies.



# The scope of the project



The CodiLime team successfully contributed to these ambitious and wide-ranging plans by delivering:

- Optimized OpenTelemetry pipelines that can process logs, metrics and traces.
- Classifiers that operate on default request metadata and use the OPA Rego language.
- Different methods of using Aperture:
  - as an Envoy filter in service mesh deployment,
  - as an SDK in multiple languages - Golang, Java, JavaScript/TypeScript, Python,
  - as a sidecar in virtual machine/local deployment.
- UI design and development for FluxNinja ARC.





# Client's testimonial



“In the startup world, founders’ primary focus is on gathering requirements from users and researching the latest technologies for architecture, etc. However, you need a trusted and skilled engineering team to get all these things implemented. The FluxNinja team has partnered with CodiLime to hire that engineering talent. We have been super impressed with the quality of the work and the dedication of the CodiLime team. It has been a very cost-effective process and helped us scale our team to over 15 team members in 3 months.

Thank you, team CodiLime!”

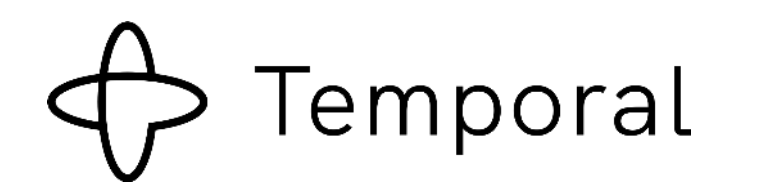
**Jai Desai**  
Co-founder & COO



# Technologies and tools



TypeScript



# About CodiLime

**codilime**

Since 2011, CodiLime has been the engineering partner of choice for semiconductor companies, networking services, telecom services, and software solution providers.

We have come a long way – from a startup to a company that hires more than 350 top-notch software developers, network engineers, DevOps experts, and solution architects. **We focus on five N.E.E.D.S. - Networks, Equipment, Environment, Data and Security.**

We aim to link engineering talent with business domain expertise. Everything to provide our clients with delivery excellence and custom-tailored solutions.

Check out what our partners have said about us and how they evaluated our cooperation.

[Go to about us page](#)

