fauna

For more information, contact a Fauna expert at sales@fauna.com

OPTIMIZE OPERATIONAL USE CASES WITH TRUE SERVERLESS:

How Fauna Solves the Limitations of DynamoDB

Unlike traditional key-value stores like DynamoDB that require you to adapt your data model and application to their limitations, Fauna's fully serverless, document-relational approach lets you build dynamically as your application evolves and scales over time.

The DynamoDB Challenge: A Limited Purpose Key-Value Store

While DynamoDB offers a fairly serverless deployment experience, it still requires development teams to be infrastructure-aware and struggles to meet the demands of general purpose operational use cases. Designed for ultra-low latency and high concurrency, its partition-aligned architecture & limited data model forces teams to compromise by relying on complex single-table designs, rigid data models, and expensive add-ons for consistency and multi-region capabilities. As applications grow and evolve, the operational overhead and cost inefficiencies associated with DynamoDB become a significant burden. It's time for a new approach.

The Fauna Solution: A True Serverless Operational Database

Organizations familiar with DynamoDB love Fauna because it expands the serverless experience beyond deployment, while layering on a document-relational model. This seamlessly combines ACID transactions, flexible schema + full enforcement, and powerful query capabilities with JSON document flexibility—all with zero configuration. With a multi-active serverless engine, Fauna eliminates the operational overhead associated with DynamoDB's partition and scaling management, providing an out-of-the-box solution that supports dynamic workloads and evolving access patterns. With Fauna, teams can focus on innovation and delivering value, rather than managing infrastructure.

Data Model

Unlock Document-Relational Power & Flexibility

Combine the flexibility of document storage with dynamic and native relational capabilities. Gain first-class support for relational access patterns like joins and ACID transactions, all within a unified database solution. Unlike DynamoDB's rigid key-value structure, Fauna allows for dynamic schema flexibility and enforcement, enabling you to adapt your data model as your application evolves.

Development Agility & Velocity

Unleash Developers to Focus on what Powers your Business

With Fauna, developers can avoid the time-consuming tasks of rebuilding tables, modifying applications for performance issues, or fitting data into rigid models. Fauna increases productivity through its fully serverless platform combined with the flexibility of its dynamic document-relational model and expressive, TypeScript-inspired query language. Unlike DynamoDB's language and transaction limitations, Fauna's native strong consistency & support for relational access patterns simplifies developer reasoning & operability.

Performance & Scale

Low Latency & Zero-Ops Auto-Scaling

Fauna optimizes performance by reducing total net operation time, condensing multiple roundtrips into a single transaction with server-side functions & multi-document transactions. In contrast to DynamoDB's partition-aligned capacity planning, which can lead to inefficiencies & increased latency, Fauna is natively multi-region and multi-active, delivering effortless scalability and high availability.

Total Cost of Ownership

Simplify your Stack & Eliminate Overhead

DynamoDB operates more as a tool than a platform, often requiring a side-car relational database or additional services to meet common application needs. Conversely, Fauna is a fully featured platform that integrates advanced relational capabilities, eliminating supplementary tech debt by reducing infrastructure & database overhead like partition management, single-table-design complexities, and transactional workarounds.

Customer Success

Since Fauna documents do not have to follow a specific schema, we are able to save time and effort -- but we still can leverage Fauna's relational capabilities. It's the best of both worlds.

Artur Tomusiak,
Senior Software Engineer
HANNON HILL

Fauna and Fastly have been instrumental in our ability to test and release features quickly. This stack also enables us to deliver ultra-low latency because it's fully distributed and API-driven.

Isis Baulig, CTO
climatiq

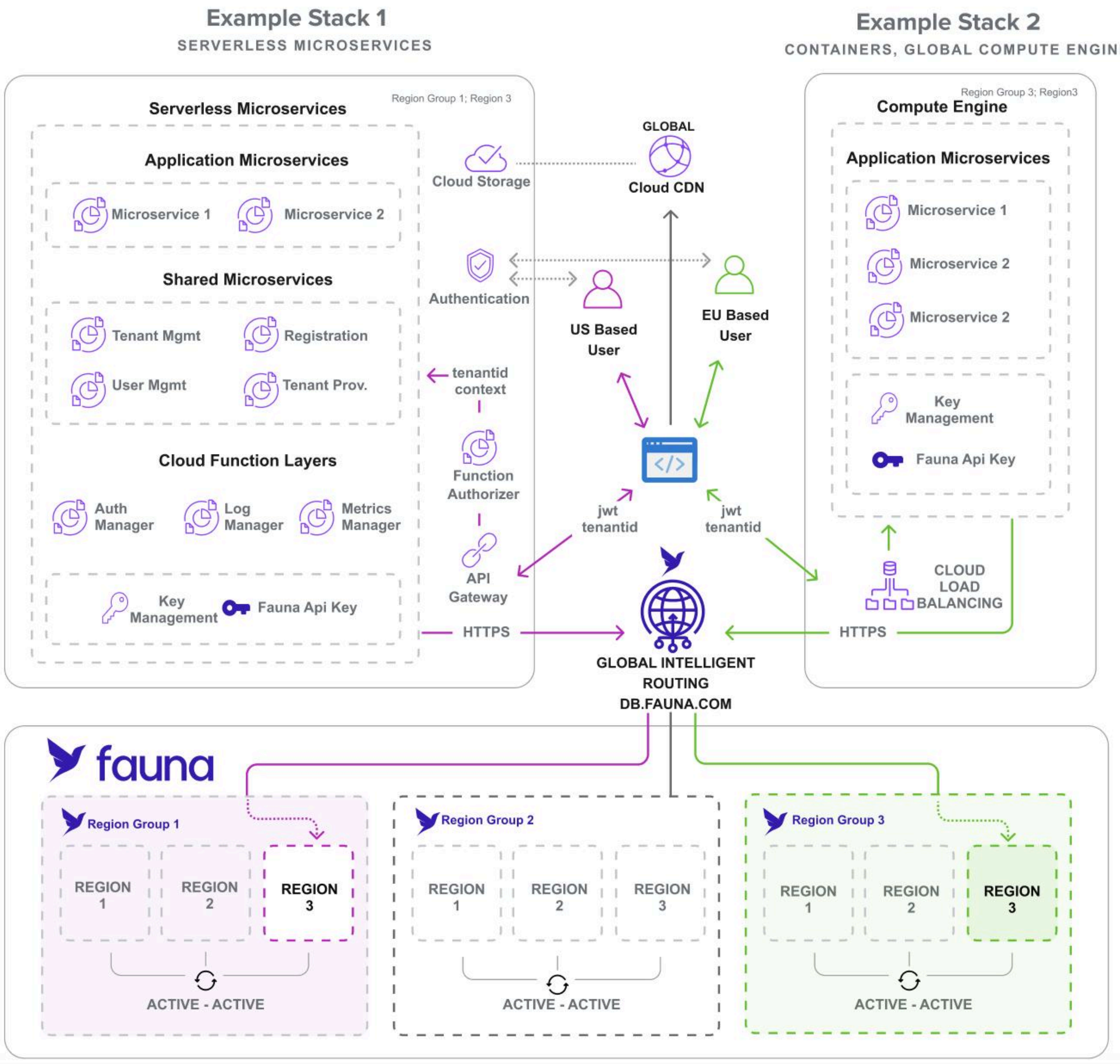
We needed a database that could support a distributed, multi-tenant architecture with robust ABAC and user controls. We looked at Dynamo and Mongo, but only Fauna delivered it all without extensive engineering.

Arjun Bhatnagar, CEO
cloaked

All of our web-based properties are built on Fauna and Cloudflare Workers, because they are both distributed-by-default and delivered as APIs, so they scale as we grow.

Cameron Bell, Software Manager
connexin

Modern Architecture



Fully Featured

Document-Relational Data Model

Combines document flexibility with relational querying, schema enforcement, and ACID transactions.

Multi-Active Serverless Engine

Auto-distribution, replication, multi-active reads & writes, & strong consistency. Eliminate server & cluster management, capacity planning, memory allocation, sharding, & more.

Cloud API Connectivity

Secure, modern HTTPS delivery with intelligent routing. Stateless, lightweight, and performant.

Schema as Code

Start schemaless and add structure that can be enforced over time with zero-downtime migrations.

Modern Security Model

Stateless, token-based authentication and dynamic ABAC offers identity and context-based access for more fine-grained and secure control

Native Database Multi-Tenancy

Provide a secure container for each business or consumer your application serves through a native parent-child database model

Real-time Event Streaming

Enable live state management and power secure, modern and scalable interactive user experiences without the overhead of polling

Industry Compliant

SOC2 Type 2, GDPR, PCI, PII, & HIPAA certified.

Common Use Cases

Multi-Region or Global Workloads

Dynamic + Relational Access Patterns

Multi-Tenant Apps

Distributed Edge Workloads

Serverless Architectures

80,000+ Active development teams

180 Countries with teams globally

1 million+ Databases created