Kshitiz Agrawal

Go backend engineer with 2+ years building distributed systems handling 10M+ daily events at a fintech scale-up

Kshitiz1403 | kshitizagrawal | kshitizagrawal@outlook.com | +91-9829970771

Experience

FinBox Backend Engineer *Jan'24 - Present (Full Time), Oct '23 - Dec'23 (Intern) | Bengaluru, IN*

Workflow Orchestration Platform | *Temporal-Based Business Process Engine*

- Designed & implemented the platform: a Temporal-based execution engine that parses Serverless Workflow specs, plus a visual canvas where a dedicated team of application engineers compose and publish business workflows daily without writing code — runs 1M+ workflows/day.
- Core engineering constraint: any change to the engine must maintain Temporal's deterministic replay guarantees across all in-flight workflows, so features ship backward-compatible with running workflow histories. Enforced by an extensive test suite including replay tests.
- Designed for graceful degradation at scale: cooldown periods for trigger loops, byte-level size limits to catch bloated workflows at runtime, and cycle detection to terminate infinite loops — so a misconfigured workflow degrades itself rather than starving others.

DataDancer | *In-Memory Workflow Engine & JSON Transformer*

- Built a Go library implementing the Serverless Workflow Specification for short-lived workflows where Temporal's durability guarantees aren't needed — runs fully in-memory, no orchestration infra to provision per workflow. Runs 10M+ workflows daily across 4+ microservices.
- Each workflow defines its own activities, loggers, and OpenTelemetry instrumentation injected at runtime — same bring-your-own-function model as Temporal activities, without the durability machinery underneath.
- The most common activity type is config-driven JSON transformation — aggregating and remapping data across schemas, including calling out to other APIs mid-transform — which is what powers ETL across those 4+ microservices.

Octopus | *Internal HTTP Proxy & Control Plane*

- Built an HTTP proxy routing 500 RPS across arcorss 400 services at FinBox services, with all routing and vendor behavior driven by config — no redeploy needed to add or change a service.
- Built a circuit breaker for vendor selection: a periodic job scores each vendor on error rate, latency, and cost from Prometheus, drops a vendor from the ready pool on error, and rebalances the pool as scores update — so a vendor recovers automatically once it's healthy again.
- Built a plugin system so vendor-specific integration logic doesn't live in the Octopus binary — teams write a plugin for their vendor's quirks and load it without touching the proxy.

Pickle | *Multi-Tenant Configuration Management Service*

- Extracted config management out of the monolith into a standalone Go + gRPC service (200 RPS) — config logic was locked inside the monolith before this, so no other service could use it.
- Added multi-tenancy so each service manages its own config in isolation, with no cross-tenant access by default.
- Built user-level RBAC scoped per namespace — a user might get read-only on one namespace, read-write on another, plus separate permissions for creating or listing namespaces entirely.
- Logs every config change for traceability, useful for audits without being built against a specific compliance requirement.

Technical Skills and Interests

Languages: Go, TypeScript, JavaScript, Java, Python
Tools & Platforms: Kubernetes, Docker, Terraform, Helm, Temporal, Kafka, OpenTelemetry, GraphQL
Databases: PostgreSQL, MySQL, MongoDB, Redis
Cloud & Frameworks: AWS, Azure
Interests: Distributed Systems, Database Internals

Education

Vellore Institute of Technology, Bhopal *Bhopal, India*
B.Tech in Computer Science Engineering & Specialisation in AI & ML; CGPA: 8.5 Aug '20 - Jul '24