

# Iman Tumorang

+65 87434146 /+372 56132890 | [iman@tumorang.com](mailto:iman@tumorang.com) | Tallinn, Estonia  
[linkedin.com/in/imantumorang/](https://www.linkedin.com/in/imantumorang/) | [github.com/bxcodec](https://github.com/bxcodec) | [imantumorang.com](https://imantumorang.com)

## SKILLS

**Languages:** Go (Golang), TypeScript, JavaScript

**Platforms:** Docker, Kubernetes, MongoDB, PostgreSQL, MySQL, OpenSearch/Elasticsearch, Redis, Kong, RabbitMQ, SQS/SNS,

**Development Tools:** OpenAPI 3, Swagger, BuddyWorks, CircleCI

**Area of Interest:** Software Architecture, Microservices Design, Open Source, Blogging, Public Speaking, Site Reliability, and Performance

## EXPERIENCE

### Staff Software Engineer (Product Platform)

April 2024 - Present

**Veriff** (YC W18) – Leading Identity Verification Platform in EMEA and LATAM.

*Tallinn, Estonia – Hybrid*

- Led the domain separation of the Client Configuration Monolith Service across all teams. Investigated, planned, and created a proof of concept (demonstrating how it's done for one domain). Started by decoupling the domain, then migrated from the old large shared database to a global and regional database tailored to each team's domain.
- Establishing Notification Platform V2 with a clear domain separation between the product and the notification platform domain.
- Created and owned the Architecture Northstar for the Platform team to address long-term engineering challenges. Collaborated closely with Principal/Staff Engineers, Director of Engineering, Head of Engineering, and Engineering Managers to align the quarterly roadmap with the Northstar and product requests.
- Leading the server-driven UI platform development that reduces the go-to-market time of new product features. Unblocking new market opportunities related to risk-based verification flow.
- Establishing Platformization of API Gateway, migrating monolith API to Kong Gateway, and decentralized domain-centric logic to each domain service.
- Implemented observability libraries (Open telemetry, logging, metrics, and tracing) for P0 services, Public API, Webhook, Mobile API, Auth, and Config service.

### Fractional Software Architect, [SoftwareArchitect.ID](https://SoftwareArchitect.ID)

Jan 2022 - Present

*Anywhere, Remote*

- Started my own business as a Fractional Software Architect for various companies, from small-sized to mid-sized companies across Southeast Asia
  - Giving consultations and feedback about their system so they can scale their business.
  - Led and hired the team members and set up the example/foundations for the teams to follow.

### Staff Software Engineer / Software Architect (Product Platform + DevExp) **Xendit** (YC S15) – Leading Payment Gateway in Southeast Asia

Nov 2020 - Jan 2024

*Singapore, Singapore – Hybrid*

- **Created and Owned the Architecture Northstar to set the teams' architecture direction and vision and aligned it with the annual company strategy**, especially for all these teams below.
  - Merchant Financial Group (MFG), (2023-2024)
    - Top Up and Withdrawal (2023-2024)
    - Unified Transaction View (2023-2024)
    - Reporting Platform (2023-2024)
    - Ledger / Transaction Platform as Service (2023-2024)
    - Billing Platform as a Service (2022-2024)
  - Authorization and Authentication Platform as a Service (2022-2023)
  - Notification Platform as a Service (2021-2022)
  - File Manager Platform as a Service (2020-2021)
  - XenShield (Fraud Prevention and Risk Management) for payment. (2020-2021)
- Led the Report Platform re-architect initiative focused on decoupling Ledger and Report for domain boundary separation and better performances. Move the reporting to Datalake using DatabrickSQL. **Reduced critical issues: regular DB restart (from 4+ times a day) to zero(none)** due to heavy queries for report export on Ledger read replicas.
- Moved the UI data source from Postgres read replica to Elasticsearch/OpenSearch. **Reduced average response times from p95 3s into p95 300ms** for transactions and balance history endpoints used in

Dashboard and public API.

- Led the Ledger / Transaction Platform re-architect initiative, which focused on improving ledger performances by introducing the journal entry layer and eventual consistency for the ledger line creation. **Improved ledger writing performance, increased the Request Per Second (RPS), and reduced the average Response time in Money-Out flow by at least 250% and Money-In flow by at least 450%** from the existing system during the burst request.
- Led the Unified Transaction View (UTV) project to consolidate all transaction activities for merchants in the Xendit Dashboard to improve the user experience for viewing their transactions, reconciliations, and exporting the reports. Co-led a mini team of 6 people, including myself: 1 backend, 1 frontend, 2 data engineers, and 1 QA engineer.
- Led the Billing Platform re-architect initiative that later contributed to reducing **AR days from >=180 days to 50 days on average**—reduced toils related to invoicing reconciliation. And help the product monetize its service easily with a configurable billing platform.
- Led Auth Platform re-architect and led a mini independent team to re-architect the Authorization and Authentication systems to be centralized. Reduced toils to revoke fraudulent merchants' access **usually need 3-5 days to be entirely revoked** from the entire system (multiple Dashboards, multiple API Gateways, Mobile). **Now, it can be done instantly.**
- Led the notification re-architecture to omnichannel notification v2 for better maintenance. **Reduced toil to maintenance notification services from 60 queues into only N channels (1 notification channel = 1 queue).**
- Led an RFC implementation for the consolidated queue-service library that impacts all engineering teams at Xendit that use Message Broker. This will later help **improve reliability and scalability for all microservices** that use the queue/message broker platform in Xendit (e.g., RabbitMQ, SNS + SQS) and **remove the single point of failure** across the Xendit organization regarding **RabbitMQ failure**.
- Maintained the Engineering blog by encouraging the engineers to write a blog, creating a reward system for every blog posted, and reviewing all drafts by the engineers) that **later helped to build the Xendit brand in engineering communities and increase the hiring rate.**
- Coordinated the development of the Xendit core libraries (Go Logger, Queue Service, Go Open API, Go Tracing, Go Circuit Breaker, Go Transactional Outbox, and Go Mod Proxy), which will help reduce developer toil **on standard tooling.**

**Senior Software Engineer (Core Payment) [Xendit](#) (YC S15) –**  
**Leading Payment Gateway in Southeast Asia**

**July 2019 - Oct 2020**

*Jakarta, Remote*

- Designed the new architecture of the e-wallet payment service to support multi-country payments in the SEA region. Later helped to **increase the Total Payment Value (TPV) up to 1250%** in 6 months during the pandemic.
- Designed the complete architecture of Direct Debit Payment and developed it using Golang and NodeJs, **making Xendit the first payment gateway that supports direct debit in the SEA region.**
- Created and proposed an RFC document for a new GitHub flow and new deployment flow for Xendit-wide organization to **simplify the deployment flow from 5 steps to only 3 steps.**
- Coordinated the re-architecture plan of the new Virtual Account Service, designed the new architecture, planned the execution timeline, and built it with the team to simplify the complex flow of the virtual account product using Golang, Node JS, AWS SNS+SQS, and Kubernetes on EKS. That later helped to **reduce customer issues on production by 50%** after migrating to new architecture \*based on Zendesk ticket count.

**Software Engineer, [Kurio](#) – Indonesian News Aggregator**

**May 2017 - July 2019**

*Jakarta, Indonesia – Onsite*

- Coordinated the development of backend API and all related microservices for Kurio V4, designed the architecture, and developed the API with the team.
- Built the new personalized feed service for Kurio's readers using Golang, Aerospike DB, and Redis, which contributed to **reducing the average feed API response time from 200ms to 160ms.**
- Built targeted push notifications for Kurio's readers, **increasing user retention by 5%.**
- Migrated the core service of Kurio from PHP to Go (Golang). Ported from Monolith Laravel to Microservice Golang, **reducing the average API response time from 700ms to 200ms.**

**Junior Software Engineer, [Bornevia](#) – Zendesk but for Indonesia**

**2016 - 2017**

*Jakarta, Indonesia – Onsite*

## COMMUNITIES and CONTRIBUTIONS

- Open Sources (maintaining): [dbresolver](#) | [faker \(golang\)](#) | [go-clean-arch](#) | [github-readme-medium-recent-article](#)

- Talks: [Go Singapore Talk](#) | [Go Jakarta Talk](#)
- Blog Post: [Medium](#) | [Hackernoon](#) | [imantumorang.com](#) | [Notes.SoftwareArchitect.ID](#)
- Side Hustle: [Easyread](#) | [SoftwareArchitect.ID](#)