Ville Misaki 三﨑ヴィッレ

Senior full stack developer · Yokohama, Japan (permanent resident) · ville@misaki.fi · github.com/vlumi

Back end, front end and infrastructure alike, these days mostly Go services on AWS and Azure, React and TypeScript front ends, native iOS in Swift, and the build and release automation around them. Twice in the architect's chair: on a product customized for each customer, and across a credit card company's development teams. On the side I build things to understand them, lately with Claude Code, which I use daily at work as well. The latest is a map of the Japanese power market from public data, made over a long weekend.

SELECTED WORK

Retail app rebuilt from scratch, Go back end and Azure infrastructure — Reaktor client, 2026–
The app replaces the systems the client's previous vendors built, with the features its users actually want, done right. I joined for native iOS, but have spent most of the time on the Go back end and the Azure infrastructure (Terraform, Azure Pipelines). Everything that would otherwise be a manual step is automated, so a change goes from a merged pull request to production without anyone touching it. The rewrite also brings the app's first AI feature: a recommender that by regulation may not favor any vendor, so it recommends by product attributes, never by brand. I built the first prototype, and the team has built on it since.

Global retail back end in Go on AWS — Reaktor client, 2025–2026
A six-developer team took over, from the vendor that had built them, the API and the scheduled batch jobs behind the in-store app of one of the largest retailers in its field in Japan. The system runs in several regions worldwide. Feature work on the Go services, and on-call for the system with its monitoring and incident response. Taking it over took months of handover sessions with the previous vendor, often whole days, and a good while after that of finding out how things had been built. Along the way we cleaned up, automated and documented what we worked on, and handing the system on to the next vendor took a few weeks of short daily sessions.

I also built a tool that compares the regions' batch networks, every parameter and dependency, and reports only the differences that matter. It caught copy-paste errors and other mistakes that had passed manual checks and would have broken production.

PayPay Step renewal — PayPay, 2021
PayPay Step pays cashback at a rate set by how much a user used the app the previous month, by the count and the total of eligible purchases. Working out which tier a user was in, or heading for, had been a database-heavy query that would not scale. It was needed both by the cashback engine on its periodic fetch and by the app showing the user the month's progress. The renewal, which merged the campaign with other Yahoo! Japan services, needed the status read at over ten thousand requests a second. I designed the new architecture and its data flows: per-user aggregates in DynamoDB, updated from the transaction stream over Kafka as each purchase lands, so the tier became a fast read. A daily batch re-aggregation verifies the stream's results, repairs gaps, and recomputes under changed campaign rules. I presented it as "PayPay Step Merge Journey" (Speaker Deck, October 2021).

Batch performance in Rakuten Card's core system renewal — Rakuten Card, 2016–2017
The credit card company moved its core system from a COBOL mainframe to Java EE on Oracle Exadata, live in July 2017 for 14 million cardholders, in the same half-year it became Japan's largest card issuer by shopping volume. A vendor's tool translated the COBOL to Java, with the processing logic kept as it was. I was brought in to make the most critical batch jobs fast enough for the cutover. Parallelizing them with Apache Spark was part of it, but the jobs had been written for the mainframe's network database, which streamed fixed-length byte records in a fixed order, header rows mixed in, each job arranging the data for the next instead of querying anything. Rewriting them around proper SQL gave most of the speed. The cutover ran old and new systems side by side for a month, with night shifts comparing the results, and the jobs held. The company's announcement put batch processing at more than twice the old speed. I presented the work at Oracle OpenWorld and JavaOne in 2017 and at EclipseCon Europe in 2018 (slides on SlideShare).

Technology decisions on a product customized per customer — BlueCielo ECM Solutions, 2005–2014
Kronodoc is an engineering project document management system for industrial customers, many on instances customized for them as consulting work and maintained under contract. I built such customizations myself, and as product architect made the technology decisions on the company's only product and implemented them: Latin-1 to UTF-8 across live customer data, database transactions where there were none, and deployments from patching to snapshots with each instance's configuration kept apart, which also let us offer it as SaaS.

Maps built to understand a domain — fiftysixty.misaki.fi · nebulosa.misaki.fi · 2026, side projects
The Japanese power market: JEPX day-ahead prices, the transmission companies' half-hourly supply-demand records and OCCTO's interconnector forecasts on a map of the nine areas. A clock runs over the 48 slots: price by area, the supply stack against demand, solar curtailment, and the flows between areas against the line limits. React, TypeScript, MapLibre and deck.gl, with a fetch script on the host and static files, no backend.

Ground tracks of a SAR satellite constellation, the earlier of the two, built to understand orbits: satellite positions from public orbital elements, passes over a place, the radar's reach on the ground, a live clock. Two bugs found while building it were fixed upstream, in MapLibre and luma.gl, and its coordinate code became geo-coord on npm.

EXPERIENCE

Reaktor Japan K.K., Tokyo — Full Stack Developer, since 12/2021
Reaktor was one of the first agile software companies in Finland and still works the way it started: small cross-functional teams, something usable released early, then incremental releases steered by what the client learns. Client projects as a senior developer: back end (Go, Java), web (React/TypeScript), mobile (Swift), infrastructure (AWS, Azure, Terraform), from prototype and MVP to inherited legacy. Besides the two Go projects above: two years embedded with Furyu's engineers on PiCTLINK, in Japanese, for the iOS app, the AWS side and a rebuilt customer-facing gallery.

PayPay Corporation — Senior Software Engineer, 1/2021–11/2021
Back-end development and operations for the customer lifecycle services, cashback, campaigns and the points program, event-driven over Kafka at the scale of Japan's largest payment app: Kotlin and Scala on Spring Boot, AWS, Kubernetes. On-call, monitoring and incident response included.

Rakuten, Inc. / Rakuten Card, Fukuoka — Software Engineer to Lead Application Architect, 9/2014–1/2021
Full-stack development (Java EE, Spark, JS), the core-renewal batch work above, and as lead application architect the development process and the technology choices across the teams. Conference talks at Oracle OpenWorld, JavaOne and EclipseCon Europe.

BlueCielo ECM, Helsinki and Rijswijk — Software Engineer to Product Architect, 7/2005–8/2014
The document management product above (Perl, Java EE).

Earlier, Finland — 1999–2005
B2C and B2B2C services for SMS and web (Perl, Java), and before that Windows software in C++.

SKILLS

Go, TypeScript/React/Node.js, Swift; Java, Kotlin, Scala; SQL and NoSQL (PostgreSQL, Oracle, DynamoDB, CosmosDB) · REST APIs, microservices, event-driven systems (Kafka), batch and scheduled jobs · AWS, Azure, Kubernetes, Docker, Terraform, CI/CD (GitHub Actions, Azure Pipelines, Jenkins) · automated testing from unit to browser, fully automated releases · AI-assisted development daily (Claude Code) · English fluent, Finnish native, Japanese conversational

ALSO

Side projects, open source, free, no ads: Donpa Squad, Lattice Five and Puck Around (games, on the App Store); Yomidori (a Japanese reading companion, on the App Store); Nitpitch (a string-instrument tuner on the App Store, pitch detection in Swift, in daily use at home for the violin); Photo Diary (self-hosted gallery with statistics dashboards, and its iPhone companion on the App Store). BSc in Computer Science, University of Helsinki. Living in Yokohama with family.

Updated 2026-10-30