

Mark Park

La Habra, CA | (310) 617-6737 | mark619park@gmail.com | [linkedin.com/in/markhpark](https://www.linkedin.com/in/markhpark) | mark-park.com

PROFESSIONAL SUMMARY

Software Development Lead (Backend) with 5+ years designing and shipping enterprise financial microservices at Franklin Templeton — a domain entered with no prior financial-systems background and self-taught (banking, settlement, KYC/OFAC compliance). Took three products — a transaction processing system, a rules-driven forms platform, and a client onboarding platform — from proof of concept through MVP to production. Promoted from contract engineer to Software Development Lead of a 5-engineer team. Early adopter of AI-assisted development, from the firm's Jan 2026 Copilot rollout through independently adopting Claude Code, integrating prompt-driven workflows to significantly increase delivery throughput while retaining ownership of architecture and code quality. Deep expertise in Java, Spring Boot, distributed systems, concurrency, and business-rules engines (Drools, Camunda), with cloud experience across Azure and AWS.

TECHNICAL SKILLS

Languages: Java 8/17/21 (led multi-version migrations), Python, C++, SQL, JavaScript (ES6), TypeScript

Backend & Frameworks: Spring Boot, Spring MVC, Spring Security, Spring Cloud (Eureka, Feign Client, API Gateway), Hibernate/JPA, REST APIs, Microservices, Drools (BRMS), Camunda, SAGA Choreography, Event-Driven Architecture

Cloud & Infrastructure: Azure (Service Bus, Event Grid, Cosmos DB, File Directory System), AWS (EC2, S3, ECS, Elastic Beanstalk), Docker, RabbitMQ, Redis

Databases: MSSQL, MySQL, MongoDB, Azure Cosmos DB

AI-Assisted Development: GitHub Copilot (IntelliJ, premium tier / multi-model), Claude Code (independent adoption, parallel git-worktree workflows), prompt-driven development, AI-augmented code review

Concurrency: Multithreading, custom thread pool executors, optimistic concurrency control (OCC), retryable services, deadlock resolution

Testing & Tools: JUnit, Mockito, JaCoCo, Postman, Maven, Git/GitHub, Jira, Agile/Scrum, Kanban

PROFESSIONAL EXPERIENCE

Brillio — Client: Franklin Templeton (Remote – New York, NY)

Contract Software Engineer → Software Development Lead (Backend) | May 2021 – Present

Contracted to Franklin Templeton; performance led to a two-year contract extension, conversion to full-time Brillio employee, and promotion to Software Development Lead. Worked full Eastern-time hours from the West Coast (6 AM PT start) for 5+ years.

Software Development Lead (Backend) — Account Opening Platform 2.0 (Ultra-High-Net-Worth Client Onboarding) 2025 – Present

- Promoted to backend lead on the recommendation of senior architects and management; led a team of up to 5 engineers rebuilding a heavily criticized legacy client-facing platform from POC to MVP to production, serving 500–1,000 daily active users.
- Led the platform's Java 17→21 migration ahead of the MVP release, completing the upgrade with minimal disruption to the delivery timeline.
- Sole backend engineer on-call during live-market (production) hours — monitored and maintained services in real time while concurrently developing features for upcoming releases.
- Architected the platform's two core microservices along clear domain boundaries: a Drools-based rules engine governing all UI field behavior, and a secure persistence service acting as the sole authorized gateway to internal Franklin Templeton systems.
- Adopted AI-assisted development (GitHub Copilot in IntelliJ, premium tier across all models) as it rolled out at the firm from Jan 2026, integrating prompt-driven workflows into daily engineering to significantly increase throughput while retaining ownership of architecture, review, and correctness; ranked in the firm's top 5% on AI cost-efficiency (prompt spend vs. code accepted) throughout.

- Extended beyond company-provided tooling by independently adopting Claude Code, running parallel instances via git worktrees to advance multiple workstreams concurrently, and using AI assistance to confidently extend delivery beyond backend into front-end and DevOps tasks.
- Automated over 60% of previously manual onboarding and KYC logic and improved responsiveness by 60–80% over the legacy platform.
- Built a generic recursive Cosmos DB save utility on the Patch API supporting arbitrary object structures at 5–10 ms response times, working around a known SDK upsert limitation.
- Designed a workflow system pairing database-controlled state with a dynamic rules layer; validated as a POC and shipped to production.
- Mentored and managed a summer intern new to enterprise Java, guiding him to ship several production APIs for database persistence and workflow logic.

Backend Engineer — SWP Forms (Systematic Withdrawal Plan)2024 – 2025

- Hand-picked by the lead architect to build a standalone Drools rules-engine microservice; cut processing time for 500+ business rules from several seconds to 50–100 ms using fact injection, agenda groups, and salience-based prioritization.
- Delivered the base architecture and MVP in 3 months before handoff; later extended to power a second production application.

Backend Developer — CashApp (Financial Transaction System)2021 – 2024

- Mentored by a 20-year architect on a 3-person team; became a primary contributor on core transaction-processing design across an 18-month POC-to-MVP build and 2 years of enhancements.
- Selected to assist the CashApp team's Java 8→21 upgrade, applying migration experience from Account Opening Platform 2.0 to help modernize a long-lived production codebase across multiple major language versions.
- Became the team's primary production maintainer after two years on the system, personally overseeing live market-hours processing of time- and data-sensitive transactions collectively worth tens of millions of dollars daily; the pipeline's strict timing windows meant any missed cutoff or incomplete data forced a restart or rolled settlement to the next business day — an outcome never allowed to occur on my watch, often requiring real-time coordination with teams across the firm.
- Drove major post-MVP enhancements: bulk transaction processing, deprecation of legacy processing in favor of FIS Global Plus posting — a multi-currency portfolio accounting and trust custody platform used to enter and settle trades, cash receipts, and disbursements — and integration of Azure-based check-reading AI.
- Owned the end-to-end financial processing lifecycle for \$10M+ in daily transactions (\$106B+ lifetime): bank confirmations, OFAC compliance scans, and government audit-trail processing.
- Designed an event-driven architecture on Azure Event Grid with message queues powering 20+ heavy scheduled jobs — several on 5-minute cycles, including OFAC scanning and posting transactions to banks.
- Eliminated production deadlock failures — from 2–5 incidents per day to zero — by refactoring the application workflow with an optimistic concurrency control layer, retryable services, and custom thread pool executors.
- Implemented SAGA choreography over RabbitMQ for cross-service transactions, later migrating messaging to Azure Service Bus.
- Led technical design discussions across cross-functional groups of 20–30 with product owners and business analysts.

EDUCATION

San Diego State University — Bachelor of Science, Computer Science, 2021