

Nicholas Panella

Full-stack engineer

nickipanella@gmail.com · 856.404.7471 · Tampa, FL · linkedin.com/in/nicholasjp · github.com/nicholasipanella

About me

I've spent the last few years on the platform team at a crypto and ATM operations company, mostly working across Postgres, Go, Node, and Next.js. The work I gravitate towards is the cross-cutting stuff. Access control, audit tooling, design systems, payment integrations. I like being able to follow a feature from a database trigger up to the UI it powers.

Concretely, the wallet backend I own runs at about \$2M and 7,400 transactions a month — roughly 7x what it did when I took it over in early 2025 — and that growth is what pulled me deeper into the platform side. I tech-lead the engineers working across these services, and most of the bigger decisions get written up as design docs the team builds against. I'm now looking for staff or principal-level platform/full-stack roles where the work spans the stack and the systems have real user impact.

Tools I work with

Day to day I write Go and TypeScript. The backend services use Node and Express on one side and Go on the other, with Postgres and Liquibase underneath, wired together as event-driven microservices. The admin console is Next.js and React on a custom design system. On the integrations side I've worked a lot with Fireblocks, Styth, Veriff, Onramper, Moonpay, Hookdeck, AppsFlyer, and Twilio. Infrastructure runs on AWS and Kubernetes, packaged with Docker, Helm, and GitLab CI/CD, with observability tooling and on-call rotations on top.

Languages: Go, TypeScript, JavaScript, SQL

Backend: Node, Express, Go, REST, Socket.IO, microservices, event-driven

Frontend: React, Next.js, dnd-kit, custom design systems

Data: Postgres, Liquibase, materialized views

Infra: AWS, Kubernetes, Docker, Helm, GitLab CI/CD

Observability and on-call: production monitoring, alerting, incident response

Integrations: Fireblocks (custody + NCW), Styth, Veriff, Hookdeck, Onramper, Moonpay, AppsFlyer, Twilio

Things I'm most proud of

- A custom design system I built from scratch for the admin app. The two pieces I'm most proud of are the data grid (row grouping, bulk actions, expand and collapse) and a form field that handles cross-field dependencies cleanly. Sheet, modal, and frame components on top, plus a glass-morphism dark theme.
- Our non-custodial wallet backend, which I own end to end. The custom pieces are what I'm most proud of: the address-balance pipeline splits locked, pending, and available across Postgres materialized views; the onramp router fans multi-vendor purchase events out through Hookdeck with HMAC verification on each one; and the device and key-backup flow lets users restore on a new phone without us touching their keys.
- The identity and access stack across the admin platform. End-to-end Stytl magic-link auth with JWT decoding and a session-expiration modal in the UI. RBAC enforced twice — route-level on the server, and permission-aware grids and actions in the console. Veriff handles KYC, with an async session-void hook that fires when an identity is deactivated. The identity messaging system (SMS and email templates with lifecycle hooks) sits on the same model. It's the single most cross-cutting piece of the platform — two vendors, both runtimes, and every entity workflow in the app.

Experience

CoinFlip (formerly Olliv by CoinFlip)

Staff Full-Stack Engineer, tech lead for the wallet, ATM operations, and rewards services.
Tampa, FL. Aug 2022 to now.

I started as lead web developer on Olliv, the company's consumer-facing non-custodial crypto product (web app and Chrome extension), which was retired in 2023 as part of a strategic refocus on the core ATM business. I moved into the platform team after that and now own pieces of a few services. I lead the technical direction across them and mentor the engineers shipping on them.

Wallet backend (Node and TypeScript), Jan 2024 to now

- Designed the wallet's core data model from scratch. The account schema is multi-currency, and the asset catalog is country-scoped so what shows up on a given device matches local availability. Fireblocks sits underneath for custody, and the schema currently backs ~1,000 monthly active wallets.
- Owned the webhook router that fronts our purchase pipeline. It's an event-driven fan-out across multiple onramp vendors, with HMAC-per-event verification, idempotent handlers, and at-least-once delivery semantics through Hookdeck — currently routing ~7,400 purchase events per month at ~\$2M/mo in volume. Revenue posts back to AppsFlyer with per-OS payloads on the way out.
- Designed the address-balance pipeline. Postgres materialized views back it, with a locked, pending, and available breakdown and a refresh cadence tuned for the read

patterns above it. Quote and order endpoints sit on top, with a daily expiry cron, currently serving ~1,800 quote and order flows a week.

- Built the device and key-backup orchestration on top of the Fireblocks NCW SDK. Device registration, setup status, and recovery records all live in our schema so users can restore on a new phone without us ever touching their keys. The flow currently runs at a ~93% OTP-to-wallet completion rate.
- Added crypto-to-crypto swaps through our swap provider. Quote fetching, order create and submit, and status tracking, all surfaced back to the client with the same models as the rest of the wallet.
- Real-time updates go out over Socket.IO on per-user and per-device channels, so balance, transaction, and order changes show up in the client within seconds. The proxy also wires into our internal crypto, identity, and notification services for pricing, KYC, and push, email, and SMS delivery.
- On-call rotation for the wallet end-to-end. Dashboards and alerting on top of our observability stack, and the service runs on Kubernetes alongside the rest of the platform.

ATM operations platform (Go, Next.js, React, Postgres), Sep 2025 to now

This is where most of my current work lives, and where the design system above came from. Three repos ship together for it: a Go admin service, a Next.js and React admin console, and a shared Go and Postgres master service. Most of the bigger pieces below started as design docs I wrote for the team.

- The drag-and-drop workflow builder is end-to-end work. The UI sits on dnd-kit with tab-based sub-flows, drop zones, and a validation error log, plus an unsaved-changes guard that aggregates errors across dirty drafts. Underneath, the Postgres schema has categorized screen types and screen-linking constraints, so product can author flows without code changes.
- The transaction approval queue is a state machine I designed for approve, send, and remove transitions, with concurrency-safe handling and integrity checks before any deletion goes through.
- Built the identity messaging system. SMS and email templates, CRUD APIs, and lifecycle hooks for things like deactivation and async Veriff session voiding.
- Wired up RBAC across the whole app. Route-level gating on the server side, permission-aware grids and actions in the UI, and Stytc magic-link auth on top. The auth flow has JWT decoding and a session-expiration modal.

Rewards and referrals service (Go), Jun 2025 to now

- Designed the referral system on this service. Data model, CRUD layer, secure and collision-safe code generation, conflict handling on duplicate signups, and the public-facing APIs.

JPMorgan Chase & Co.

Associate Software Engineer (promoted from Software Engineer Analyst in 2019). Newark, DE. Feb 2017 to Aug 2022.

- Co-architected the CRM platform that won the *PWM Wealth Tech Award 2022* for best CRM applications. Used by hundreds of advisors across JPM's entire PWM front-office.
- Built reactive React UIs over Spring Boot microservices, plus dashboards with inline commenting on aggregated data.
- Wrote multi-functional NPM libraries that other internal front-end teams pulled into their apps.
- Owned the test automation and QA work for the apps I shipped on.
- Contributed to an action-based API framework that improved performance across internal apps.

Volunteering

Spent five years as part of JPMorgan's Force for Good and Code for Good programs. Managed a developer team building tech for a non-profit, and mentored college students at the annual hackathon.

Education

Bloomsburg University of Pennsylvania. B.S.B.A. in Information and Technology Management, December 2016. President of the Computer Science Club.