AI-Native Engineering For Faster Time-To-Market

How Teamvoy delivered a production-ready mobile feature for a regulated FinTech platform using an AI-supported engineering delivery model, without expanding the team or adding a dedicated mobile specialist.

Book a Consultation

SERVICES

AI Consulting, Cost Optimization, System Integration

INDUSTRY

Banking

TECH STACK

Ruby, Ruby on Rails, PostgreSQL, React, NodeJS

REDUCTION IN TIME-TO-MARKET

30-40%

INCREASE IN ENGINEERING PRODUCTIVITY

20-25%

Summary

This case study shows how **Teamvoy** delivered a **production-ready mobile feature for a regulated FinTech platform** using an AI-supported engineering delivery model, without expanding the team or adding a dedicated mobile specialist.
A **Teamvoy backend engineer** implemented a complex mobile feature end-to-end, supported by **AI-assisted requirements analysis, UI state modeling, business-logic validation, and edge-case reasoning**.
AI was used as a **support system**, not as a decision-maker.

Key outcomes:

- 30–40% reduction in time-to-market
- 20–25% increase in engineering productivity
- Zero backend API changes
- Full compliance with regulatory, UX, and business constraints

Teamvoy's AI-supported engineering model enables smaller teams to deliver complex, regulated FinTech mobile features faster, while maintaining delivery quality, compliance, and engineering ownership.

01. Our Client

Our client is a **FinTech platform operating in the digital payments and crypto-finance domain**. The platform provides digital payments solutions for cryptocurrency and digital asset users, enabling crypto-to-fiat conversion and everyday payments through virtual cards compatible with Apple Pay, Google Pay, Garmin Pay, and Samsung Pay.

The company operates in **full regulatory compliance** and holds licenses for providing digital assets-related financial services, ensuring regulatory alignment across operational markets.

The company works with partners such as Binance Pay, enabling issuance and top-up of virtual crypto cards and secure payments across global payment networks. The platform maintains high standards of security, compliance, and reliability, forming a stable digital payments infrastructure for regulated financial environments.

This product ecosystem supports scalable product development, API-driven integrations, and controlled platform evolution within a regulated FinTech context, forming a foundation for structured delivery model software development in financial products.

02. Challenge

This project was designed as a controlled ai assisted product development experiment and a structured product experimentation initiative. The goal of the experiment was to validate a new delivery model software development approach based on cross-role execution and AI support.

Experiment Objectives:

Cross-Functional Delivery Validation	Role Dependency Reduction	AI Effectiveness Assessment
Validate the hypothesis that a backend engineer can implement a mobile feature end-to-end, proving feasibility of cross-role execution in production conditions and establishing a fintech delivery model.	Reduce dependency on narrowly specialized roles by introducing a cross-functional engineering model that enables highly adaptable and flexible teams and scalable delivery without role bottlenecks.	Evaluate AI across the full delivery cycle: requirements analysis, UI/UX logic modeling, state management, business rule interpretation, and resolving blockers in an unfamiliar mobile platform.

This project represents a real-world ai in mobile product delivery and ai assisted product development scenario in a regulated FinTech environment.

The main complexity came from the extensive scope of requirements, which combined complex UI logic such as gradients, progress bars, and animations with strict business rules around limits, resets, and percentage-based calculations. The feature also required full localization support, correct handling of time zones, strict UX consistency with the existing design system, and stable API integration without any backend contract changes.

This created a structured fintech app feature implementation case combining business logic, UI engineering, platform constraints, and regulatory requirements.

Main Goals

- Validate feasibility of **mobile feature development** by a backend engineer
- Ensure full compliance with acceptance criteria and regulatory constraints
- Integrate APIs through stable **API integration** without backend changes
- Maintain UX consistency and design system alignment
- Evaluate influence on **time-to-market** in a real production scenario
- Measure the real impact of **AI assisted development** on:

– engineering productivity,

– delivery quality,

– delivery confidence

03. Cooperation

The work was carried out through a structured ai native engineering model and cross functional engineering setup:

PM / QA	Requirements definition and validation (AI-assisted)
Backend Developer	Full-cycle mobile feature implementation
AI Assistant	Support with: • specification analysis • business logic validation • calculation modeling • UI state modeling • edge-case analysis • technical problem-solving

This cooperation model enabled ai assisted feature development without expanding the team composition or introducing new specialized roles and established a repeatable ai native engineering model for cross-disciplinary execution.

Stages:

Requirements Definition

A detailed specification was created by PM/QA with the help of AI, including acceptance criteria, edge cases, and business rules.

Analysis and Decomposition

The backend developer analyzed the requirements and decomposed the feature into logical blocks: API integration, UI, state management, business logic, and edge cases.

UI and Logic Implementation

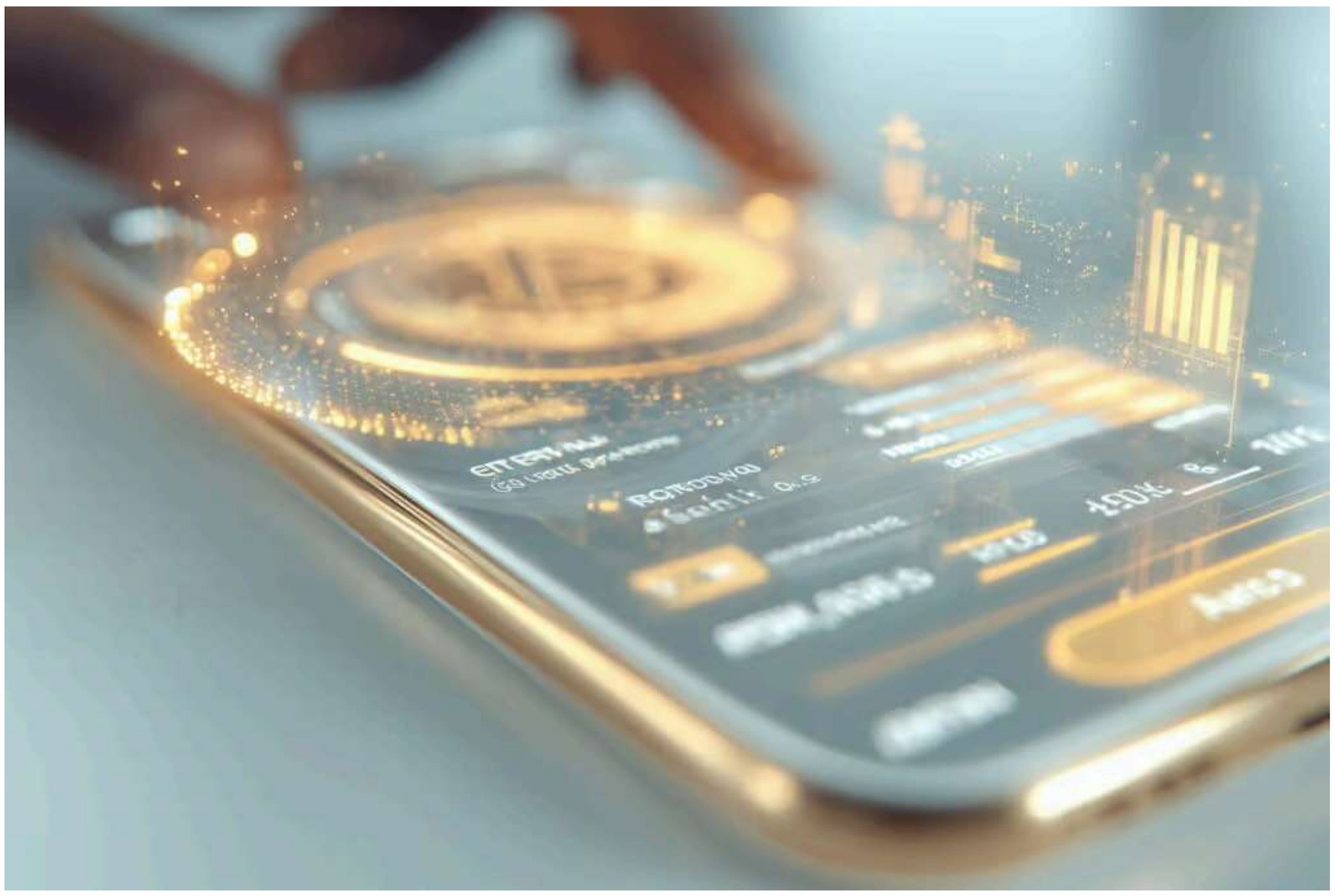
Development of the limits section, progress bars, color indicators, animations, and UI logic aligned with a structured Product Design approach.

API Integration

Integration of the getPerCardLimit endpoint and data refresh after top-up operations, ensuring stable API integration without backend changes.

Testing and Polishing

Minor issues were resolved with AI support, followed by final verification against acceptance criteria within a controlled ai assisted development workflow.



04. Solution

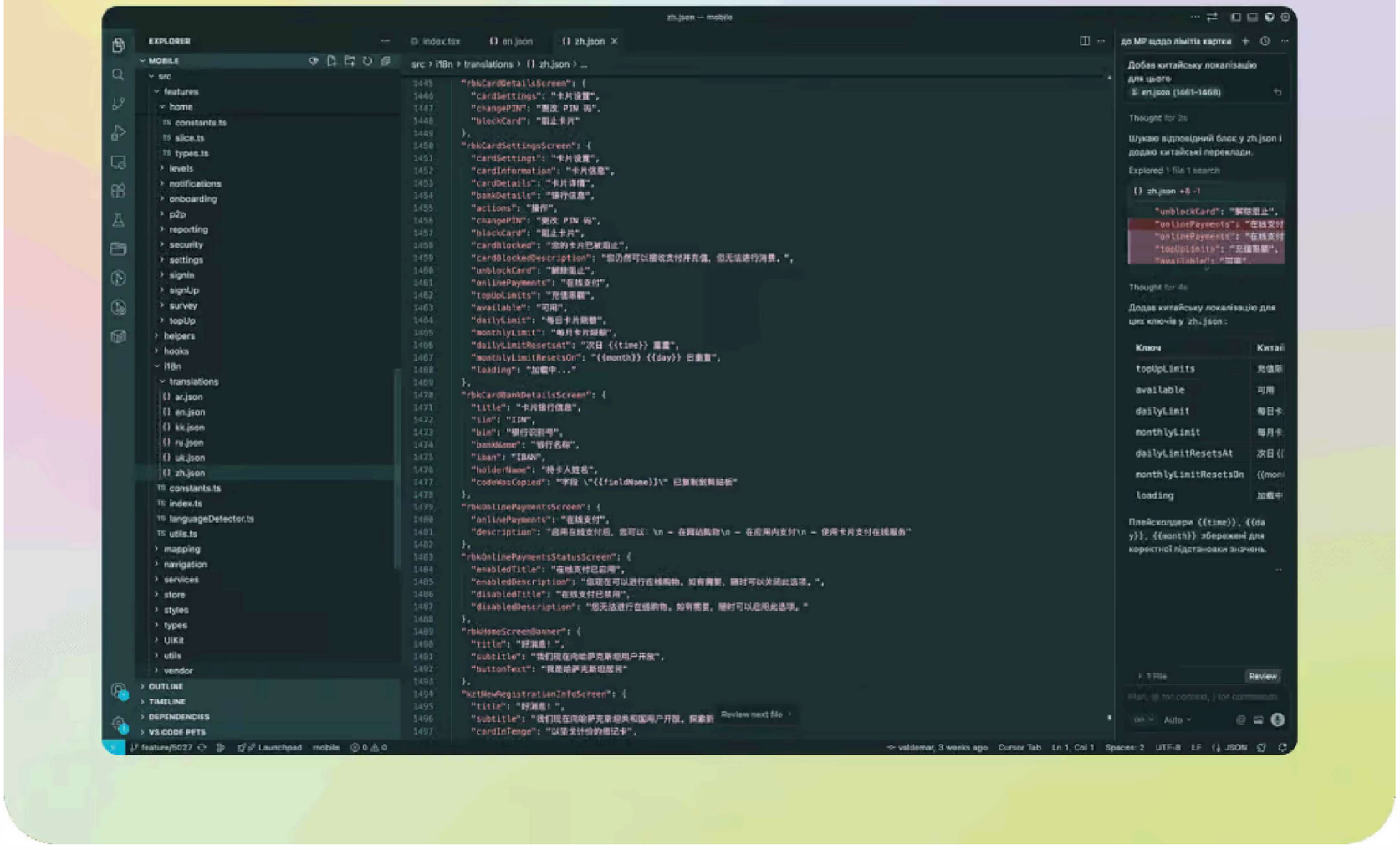
The “Card Top-Up Limits” mobile feature was successfully implemented on the “My Card” screen with full compliance to business, UX, and regulatory requirements, forming a complete fintech app feature implementation case.

Key Features:

- Display of daily and monthly limits
- Dynamic gradient-based progress bars
- Deterministic 1% step-based fill calculation logic
- Automatic refresh after top-up operations
- Full localization support
- Time zone–aware calculations

Key Engineering Decisions

- Deterministic formula-based logic for progress calculation
- Minimal deviation from the existing application architecture
- Caching and reuse of API data
- Full handling of edge cases, including 0%, 100%, and limit overflow
- AI used as a support system within an AI assisted development workflow, not as a decision-maker



05. Results

This project validates a real product delivery case and a practical AI native engineering model.

The experiment confirmed that a backend developer can deliver a production-ready mobile feature when supported by structured specifications, clearly defined business rules, controlled state management, and a structured ai assisted development workflow. This validates an ai supported engineering model focused on productivity and delivery reliability.

Business Results

- Feature delivered without involving a dedicated mobile specialist
- Reduced time-to-market by approximately **30–40%**
- Lower dependency on narrowly specialized roles
- Increased delivery flexibility and organizational resilience
- Measurable growth in engineering productivity by **20–25%** through AI-assisted workflows

The company gained not only a completed feature, but also a validated delivery model for AI assisted product development in regulated FinTech environments.