

166 Commits to a Company

A 96-hour development sprint that bridged the gap between engineering features and driving user behaviors.

166 COMMITS / 4 DAYS

Milestones
Shipped: 3

Pull Requests
Merged: 17

Date Window:
Days 20-23

The Forcing Function of Solo Pull Requests

At high velocity, structure is survival. Writing formal PRs for a solo project isn't bureaucracy—it is an architectural discipline to document intent and force clarity before merging.

Summary Box: Documentation of Intent
(Why did I change this 3 weeks later?)

Here is a street spaces to code connectator output
change's error: for manuarly: writing reading the
reposiciry documentation.

This is no commistion to Solo authenttiity.

Changes Table:
Diffable Change History

File name	Change Type	Line count diff
file/docs.ml	ADDED	27 → 18
file/docs.ml	MODIFIED	4 → 0
filenamev.ml	DELETED	-10 → -13

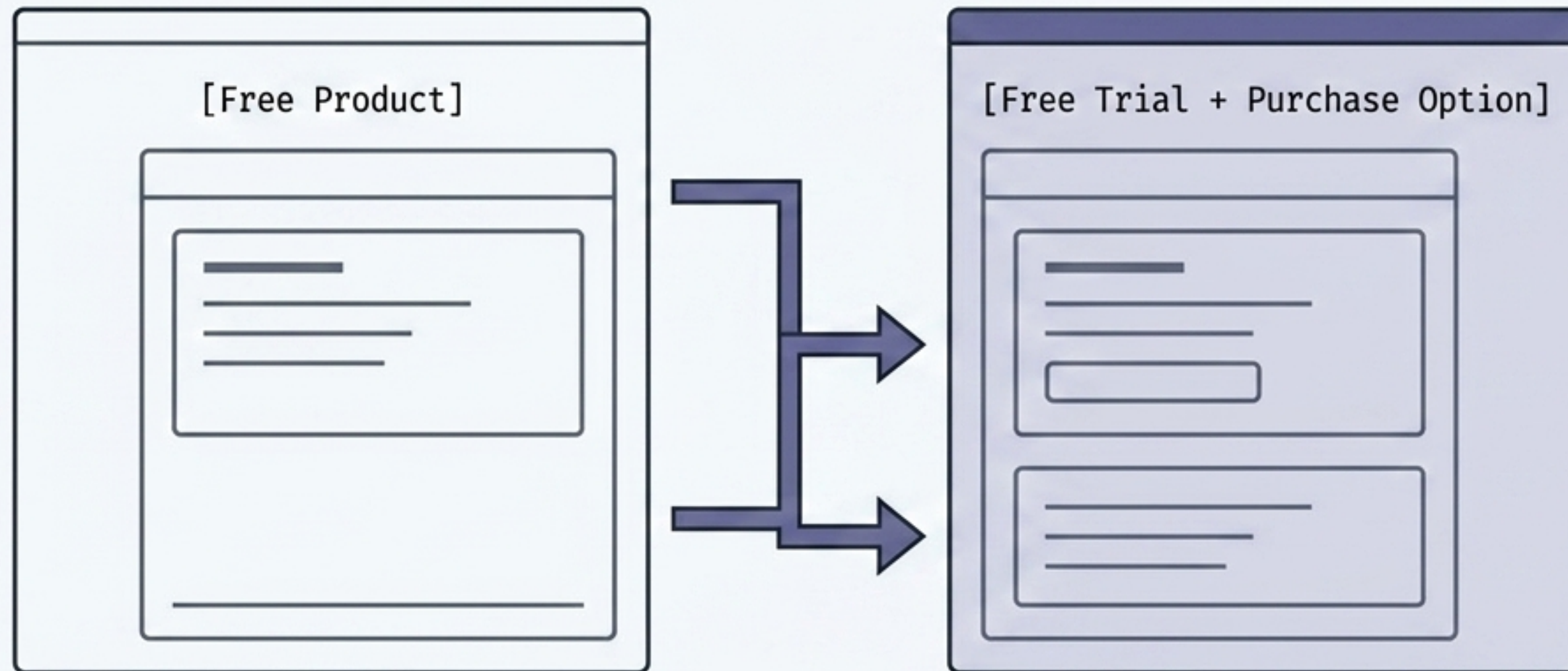
Test Plan: The Forcing Function
(Articulating exactly what is shipping)

- ☒ Unit Tests Passed
- ☐ Integration Tests Verified
- ☐ Performance Check Complete

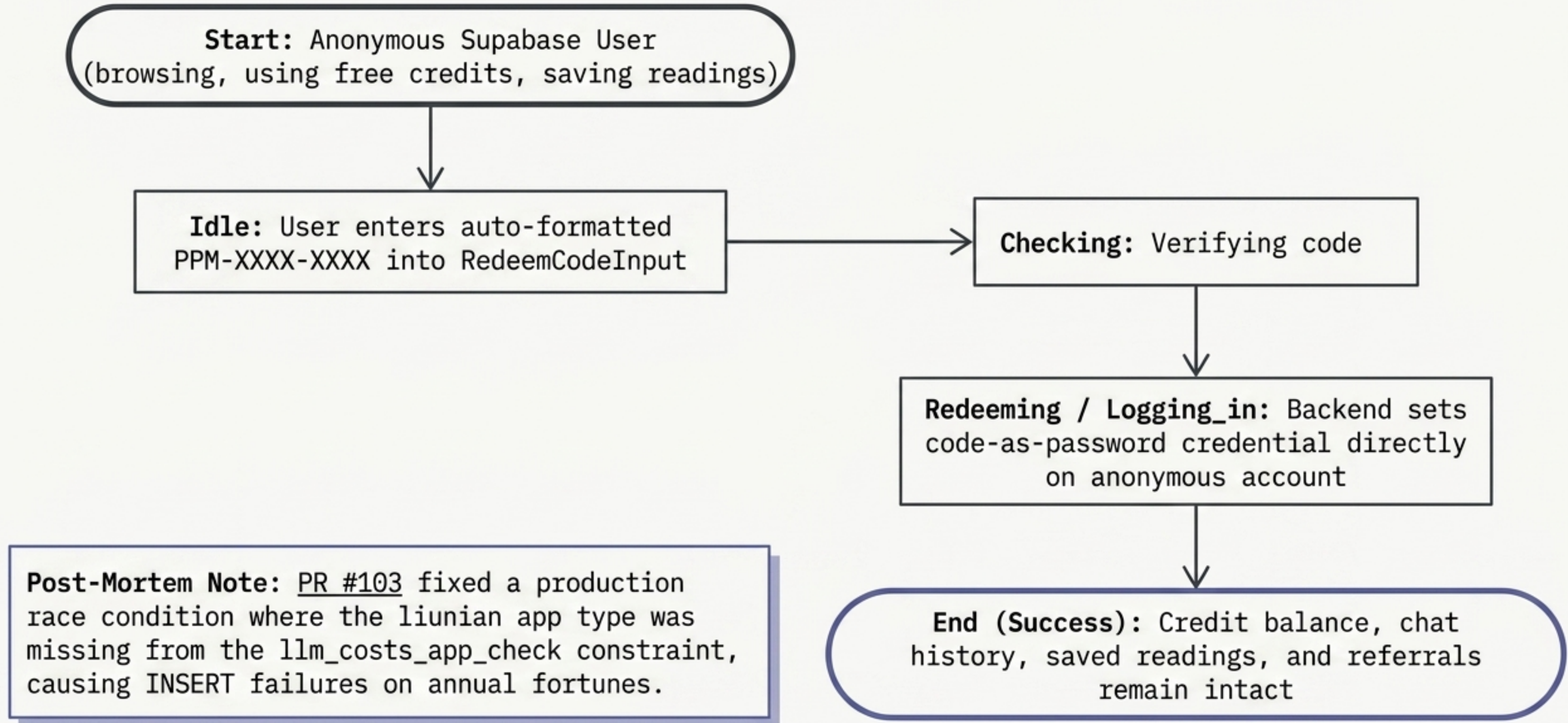
Milestone 1: Architecting the Credit Economy

PR #2 – 100 Files Changed

Transitioning a user base to a paid economy requires rebuilding the foundation. The primary engineering challenge was eliminating friction while preserving complex user states.

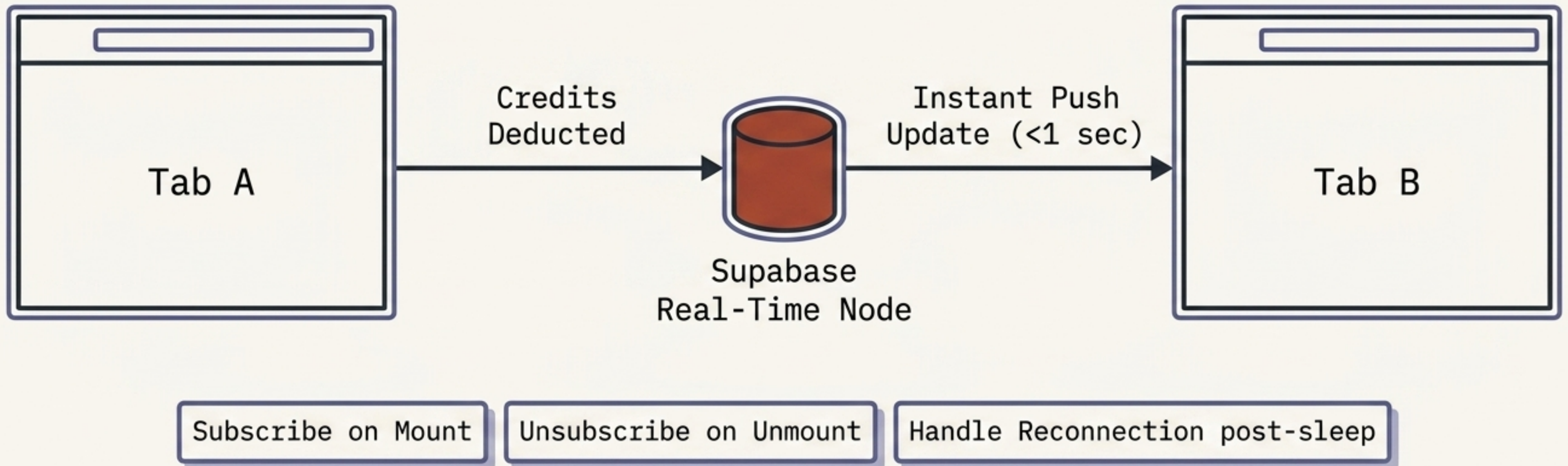


Frictionless Handoff: Anonymous to Authenticated



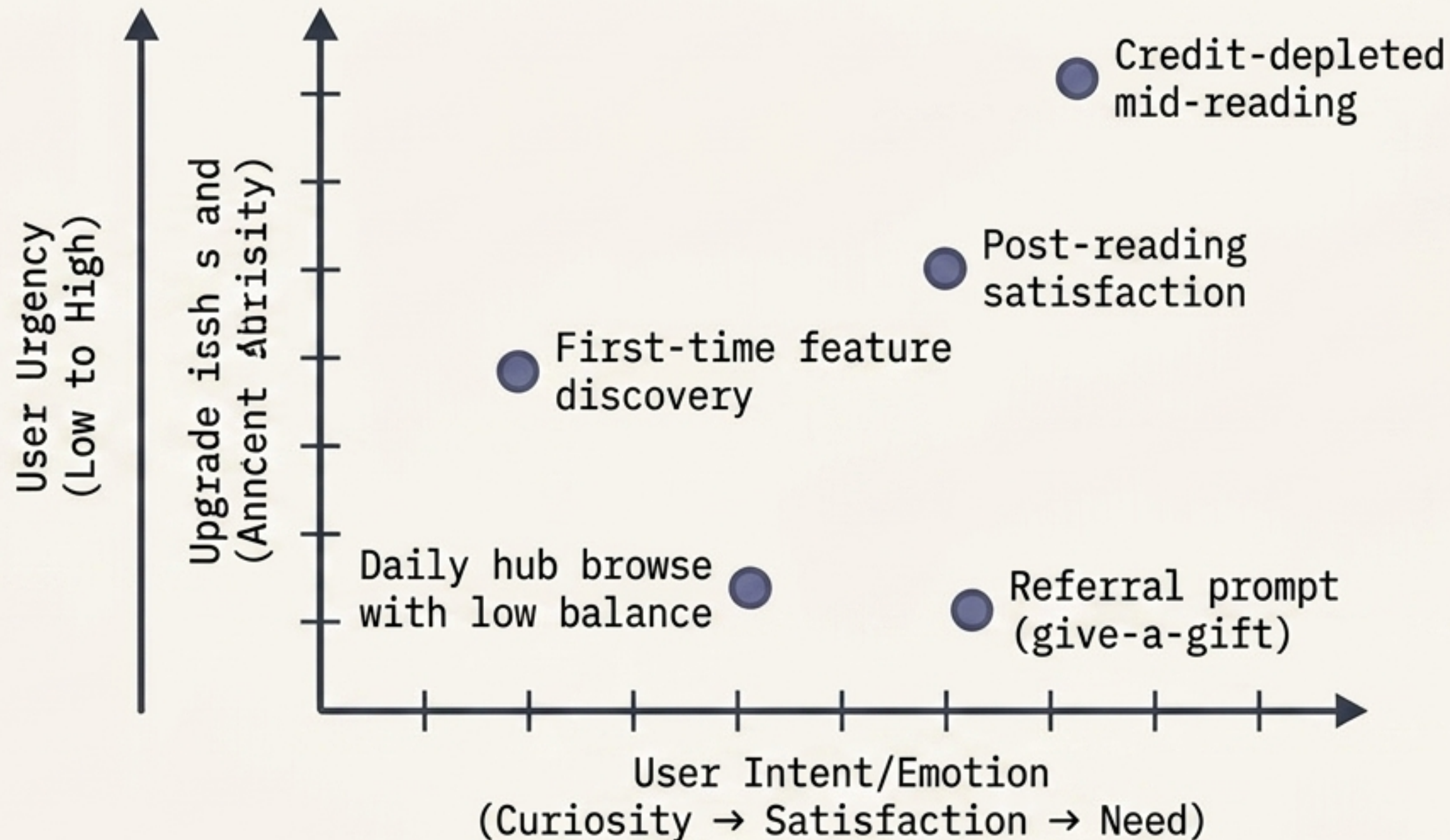
Eliminating the Double-Charge Bug

Without real-time sync, users buy credits in one tab, see old balances in another, assume failure, and purchase again. Fixing this requires precise management of the subscription lifecycle.



Mapping the 7 Contextual Upgrade Triggers

The UpgradePrompt component triggers dynamically based on the emotional context of the user journey, maximizing conversion without burning trust.



Available in Card & Banner variants + 24-hour dismiss cooldown to prevent harassment.

Milestone 2: The Engagement Architecture

PR #4 – 100 Files Changed

Fortune-telling is naturally event-driven: users arrive with a question and leave with an answer. The Daily Hub engineers a daily habit loop to drive persistent retention.



A Product Insight Disguised as Architecture

Pre-generating content via a secure, rate-limited cron job in the Asia/Shanghai timezone eliminates loading states. The **perceived quality** difference between **instant** delivery and a loading **spinner** is dramatic.

On-Demand Generation	Pre-Generation Architecture
Trigger: User opens app	Trigger: /api/cron/daily-content (Midnight Beijing Time)
Latency: 3-second loading spinner	Engine: Gemini AI → daily_content table (7-day auto-cleanup)
System Load: High variance (spikes at 8 AM)	Security: Constant-time XOR for secrets, 30 req/min IP limit
Perceived Quality: Lower (feels computed)	User Experience: Instant load, zero spinners. High perceived magic.

Zhanxing Overhaul: Math, Aesthetics, and Translation

A complete overhaul of the astrology module required simultaneously upgrading the mathematical precision, the visual language, and the global accessibility of the tool.



i18n Engine

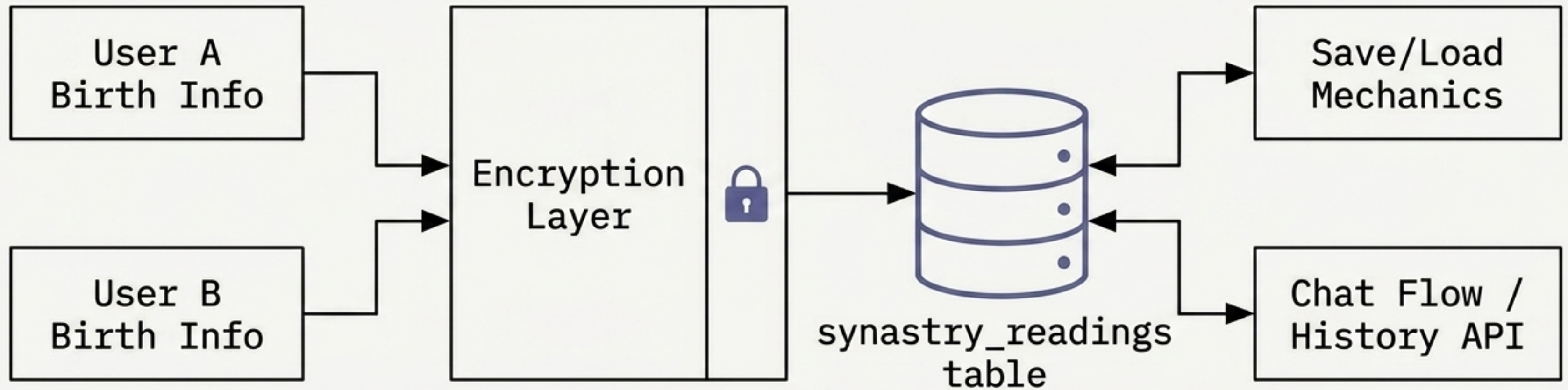
347-line trilingual dictionary (zh-CN / zh-TW / en) covering planets, signs, aspects, houses, and 70+ result strings.

QA

78 dedicated tests for chart math and placement functions.

Synastry Persistence and Data Privacy

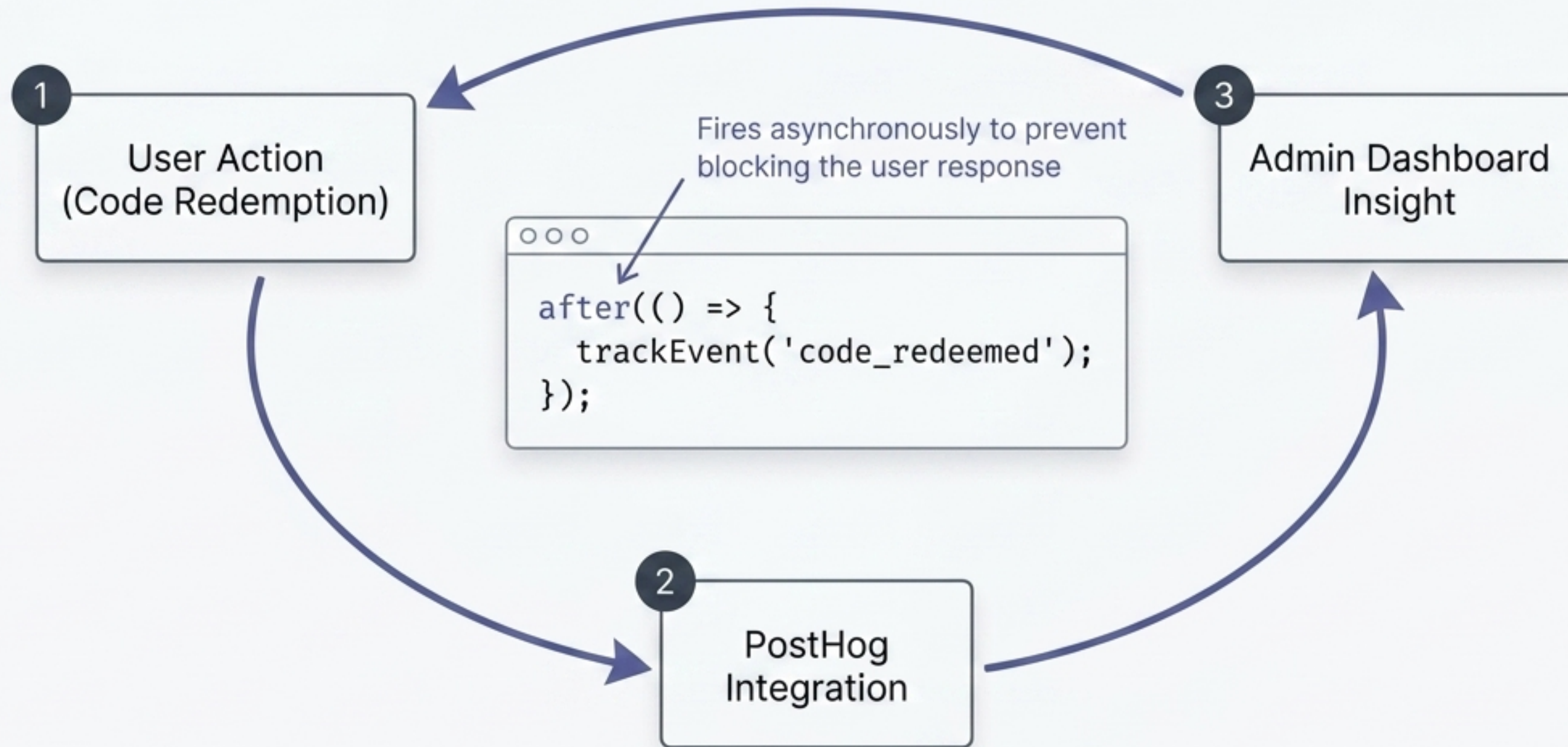
Compatibility readings inherently involve a second person's sensitive birth data. Integrating full save/load/chat history required architecting a strict encryption requirement before data ever hits the persistence layer.



Milestone 3: Closing the Reality Loop

PR #10 – 11 Files Changed

The smallest PR delivered the highest leverage. By tracking events without blocking the main execution thread, the system reveals what users actually do, rather than what the developer assumes they do.



The Paradigm Shift

Building a company requires realizing that retention and funnels are business problems that just happen to require engineering. This sprint marked the end of feature collection and the beginning of behavioral design.

Before M1-M3: Engineering Focus

Core Question:

What feature should I build next?

Deliverables:

- Repositories
- Endpoints
- Feature lists

After M1-M3: **Business Focus**

Core Question:

What **behavior** should I drive next?

Deliverables:

- Conversion funnels
- Retention hooks
- Daily engagement
- Referral incentives