

Remitly Circle: Money Movement (MXA)

Shipped Experience

Remitly Circle is now live as a standalone app on [iOS](#) and [Android](#).

[[Figma Prototype Link](#) - Placeholder]

You may explore the key screens in the [Solution section](#) below.

The Challenge

Remitly customers needed a way to add and withdraw money from their Circle accounts with zero friction. The traditional remittance model required full transaction flows with fees for each send, but Circle's stored-value model needed a different approach to money movement.

Key Problems:

- **Complex banking integration:** US banking partner integration for stored-value accounts was untested territory for Remitly
- **Load limits management:** Customers needed clear visibility into daily load/send limits without blocking the flow
- **Fraud prevention:** Balance storing required robust trust evaluation without adding friction
- **Payment processing:** Integration with PAX/MoMo platform for zero-fee deposits was technically complex
- **Withdrawal flexibility:** Money out needed to support multiple corridors (bank transfers, digital wallets, cash pickup) with varying capabilities

Research Insight: Customers wanted to "just load their wallet" and perform 2.5 downstream activities per load, indicating need for frictionless money-in experience.

Goals

Design the end-to-end money movement experience for Remitly Circle, enabling customers to add and withdraw funds seamlessly.

Success Metrics:

- [Placeholder: Load completion rate target]
- [Placeholder: Average load amount and frequency]
- Zero-fee deposits and withdrawals
- [Placeholder: Fraud prevention effectiveness without user friction]

Target Impact: Enable customers to load accounts quickly and withdraw to 170+ countries through multiple payout methods, all with zero fees.

The Solution

I designed the **Money Experience America (MXA)** flow – the core money movement experience for US customers. The design focused on:

Money In Flow

- **Frictionless card linking:** Debit card input via PAX/MoMo integration
- **Smart limits:** Proactive warning when approaching daily load limits (e.g., "\$100k entry triggers warning, customer adjusts to \$1k")
- **Trust evaluation:** Background Sift score check through Trust Platform for fraud prevention
- **Instant availability:** Funds credited to USD pool account immediately upon approval

Money Out Flow

- **Flexible disbursement:** Support for bank transfers, digital wallets (GCash, Nequi, etc.), and cash pickup
- **Corridor optimization:** Different money-out capabilities by corridor (GDN through Remitly Rail had withdrawal limitations)
- **Zero fees:** No withdrawal fees regardless of payout method
- **Real exchange rates:** Better rates than main Remitly app, with upfront transparency

Design Patterns

- **Stacked modals:** Created reusable pattern for limits and permissions UI
- **Loading states:** Clear "In progress... reviewing account" screens for trust evaluation
- **Error handling:** Graceful handling when transactions blocked due to low Sift scores
- **Sequence diagrams:** Full user flow documentation linked in Jira for engineering alignment

! [Placeholder: Money in home screen with balance]

Money in home screen showing current balance and add money CTA

![Placeholder: Card linking screen]

Debit card input screen with PAX integration

![Placeholder: Amount entry with limits]

Amount entry showing daily limit progress indicator

![Placeholder: Processing screen]

Trust evaluation loading state

![Placeholder: Success confirmation]

Successful deposit confirmation with updated balance

![Placeholder: Withdrawal method selection]

Money out flow with multiple payout options

![Placeholder: Withdrawal confirmation]

Withdrawal details with exchange rate and zero fee callout

Results

Launch Status: MXA shipped as core feature of Remitly Circle, enabling all US customer deposits and withdrawals.

Product Performance:

- [Placeholder: Load completion rate]
- [Placeholder: Fraud prevention success rate]
- Zero fees achieved while maintaining platform profitability through stored fund management

Customer Feedback:

- "I love it because it makes sending money to my sister or siblings back home so simple"
 - "Put in my deposits and Boom! It's in there already. My wife transferred it to her accounts so quickly"
-

Learnings

Technical complexity requires clear documentation:

- Sequence diagrams and detailed flow documentation in Jira prevented misalignment between design and engineering
- Stacked modal pattern became reusable component across the product

Limits need to guide, not block:

- Proactive warnings before limit hits performed better than hard blocks
- Showing progress toward limits increased customer confidence

Trust evaluation should be invisible:

- Background Sift scoring prevented fraud without adding customer friction
- Only showed loading states when necessary, maintaining flow momentum

Money scenarios vary by corridor:

- Different money-in/out capabilities required flexible design system
- Upfront research into corridor constraints prevented redesign later

Separated app was right choice:

- "Can we stand on our own?" philosophy led to cleaner, focused experience
- Integration with main Remitly app would have added complexity without clear customer benefit

Explorations

We explored various approaches to the money movement experience, focusing on how to present limits, handle trust evaluation, and structure the withdrawal flow.

![Placeholder: Limits presentation options - stacked modal vs. inline]

![Placeholder: Card input variations - single screen vs. multi-step]

![Placeholder: Withdrawal method layouts - grid vs. list]

![Placeholder: Loading state variations - progress bars vs. static messages]

![Placeholder: Error state explorations - retry flows and messaging]

Appendix

Project Context

Team & Role:

- **My Role:** Senior Product Designer, MXA Squad
- **Product Managers:** David, Molly
- **Engineering:** Anukul (tech lead), Dylan, team in Seattle and Tel Aviv
- **Collaborators:** Ian (Design Lead), Reut (IL Design Lead)

Timeline:

- Q4 2022 - Q1 2023: Discovery & design strategy
- Q2 2023: MVP design lock
- Q3 2023: Target launch
- 2024: Launched and live in limited markets

Squad Structure

MXA (Money Experience America) owned:

- End-to-end load wallet flow
- Money in portion of the send flow
- US banking partner integration
- Payment processing through PAX/MoMo

Clear ownership boundaries with other squads:

- NUX: Onboarding and first-time experience
- AMX: Account management and recipient experience (later consolidated)
- IWX (Israel): International wallet experience and localized flows

Cross-Team Collaboration

Design review process:

- Slack for questions → Jira for tickets → design reviews with engineering
- Regular design office hours for SoV product alignment
- Documentation through Loom videos and Figma for visibility

US-Israel coordination:

- Weekly syncs to align on component reuse and design patterns
- Shared understanding of technical constraints and regulatory requirements
- Clear communication protocols to prevent redundant work

Technical Architecture

Banking Integration:

- Credit transactions against USD pool account at external US banking partner
- PAX/MoMo platform for payment processing
- Trust Platform with Sift scores for fraud prevention
- Real-time balance updates across shared accounts

Regulatory Considerations:

- Euro holder citizenship requirements (EU/PHL compliance)
- Different money-in/out capabilities by corridor
- GDN through Remitly Rail limitations
- Watchlist evaluations for KYC compliance

Design Strategy

Influence without authority:

- Monthly jam sessions with PM + Tech Lead to identify customer challenges early
- Created design roadmaps showing end-to-end flows vs. individual features
- Regular artifact documentation increased visibility with leadership

MVP philosophy:

- "Can we stand on our own?" approach led to standalone app decision
- Shipped with feasible solutions rather than waiting for perfect implementation
- Focused on US market first, then expanded internationally

Customer insights:

- 84% of customers send to close friends/family
- Load intent: customers wanted to "just load their wallet" for future use
- Withdrawal flexibility: recipients valued multiple payout options in local currency

Component Library

Reusable patterns developed:

- Stacked modal patterns for limits and permissions
- Loading indicators for async operations (trust evaluation, KYC review)
- Card input patterns with PAX integration
- Amount entry with limit progress indicators
- Error states with retry mechanisms