

Talise Litepaper v1.0

Talise Invisible Stablecoin Settlement on Sui *Litepaper* · v1.0 · May 2026

"If the user has to think about the chain, the chain hasn't done its job yet." Adeniyi Abiodun, Co-founder, Mysten Labs. Sui Basecamp keynote, 2025.

1. Abstract

Talise is a consumer payments protocol on Sui. A user signs in with Google and claims a handle of the form `alice@talise.sui`. Anything sent to that handle, whether SUI, USDC, or USDT, is automatically converted to USDsui (Sui's native dollar-pegged stablecoin) and delivered to the recipient's wallet within a minute. The user holds no gas token, signs no seed phrase, and never sees a swap screen. Idle USDsui earns yield through Navi in one tap.

The protocol runs on Sui mainnet. Transactions are sponsored, so the recipient pays nothing to receive. Fees on the Talise stack itself are zero, and the only costs the user incurs are off-ramp fees at the destination, which Talise does not collect.

This paper describes what Talise does, the security model that keeps user funds safe, the products built on top of it, and the roadmap for the next twelve months.

2. The Problem

Sending money across borders is expensive and slow, and the Sub-Saharan corridor is the worst of any region. The World Bank's Remittance Prices Worldwide report (Q4 2024) puts the global average cost of sending USD 200 at 6.65%. Sub-Saharan Africa sits at 7.89%. A worker in London sending GBP 200 to Lagos through Western Union loses roughly GBP 14 to fees and another 2 to 4 percent to the FX spread. Settlement takes days. Cash pickup requires the recipient to physically present themselves with ID at an agent location.

Crypto has solved the technical problem. Sui finalizes transactions in under a second at sub-cent cost. Dollar-pegged stablecoins are abundant. The experience of using crypto, though, is still worse than fiat for the kind of recipient who needs it most. A wallet is an application the user must install, secure with a seed phrase, fund with a gas token, and consciously open

every time value moves. The recipient, who is typically less technical than the sender, on a worse device and a worse network, is asked to learn the vocabulary of mnemonics, gas, networks, and approvals just to receive the equivalent of a week of groceries.

"[Seed phrases] are not good enough... hardware wallets alone are not good enough... social recovery is better." Vitalik Buterin, Why we need wide adoption of social recovery wallets, 2021.

What is missing is a layer above the chain that gives the user three things at once: a human-readable handle that resolves to an address, automatic conversion of inbound coins to a single dollar-denominated unit, and gas paid by someone else so the user never holds the fee token. Each of these has been done in isolation. None of them have been put together into a payment product a non-crypto user can actually use. Talise is that layer.

3. Why Now, Why Sui

Three things had to arrive together for an invisible payments protocol to be possible. Sui has all three.

zkLogin removes the seed phrase. A user signs in with Google and gets a Sui address derived from the OAuth identity. There is no mnemonic to write down. There is no passkey to enroll on a second device. Recovery is Google account recovery: the same trade Apple Cash makes with iCloud. No equivalent exists at the protocol layer on any competing chain in production today.

Sub-second finality means payments feel like payments. Sui's Mystici consensus settles owned-object transactions in under a second and shared-object transactions in roughly one second. Parallel execution means ten thousand users converting their inbound coins in the same minute do not contend for a single sequencer. A consumer payments product cannot ship on a network whose tail latency goes to thirty seconds during congestion.

USDsui is a native dollar. USDsui is a dollar-pegged stablecoin issued natively on Sui. There is no bridge, no canonical-versus-wrapped distinction, no third-party operator to insure. The auto-swap loop has a single fixed destination because the user has expressed a preference for dollars and nothing else.

Sponsored gas (the ability for one address to pay the gas of another's transaction) is now first-class on Sui. Combined with zkLogin, it lets us put a user on the network without ever asking them to acquire SUI. None of this existed together a year ago. Building now means filling the gap before bridge-dependent wallets and region-specific custodial apps fill it instead.

4. How Talise Works

From the user's perspective, the entire system is four actions: sign in, claim a handle, send, receive. Everything else is the protocol's job.

Sign in. The user opens Talise on iOS or web, taps Continue with Google, and is signed in. A Sui address has been derived from their Google identity through zkLogin. They do not see this address. They do not need to.

Claim a handle. The user picks a name. If `alice` is available, they get `alice@talise.sui`. The handle is a SuiNS subname under the protocol-owned `talise.sui` domain, and it resolves to a Sui account that only the user can withdraw from. Once claimed, the handle is portable and persists across devices.

Send. The user types the recipient's handle (`mama@talise.sui`), the amount in their local currency, and taps Send. Talise pays the gas. The transaction settles on Sui in under a second.

Receive. When someone sends to your handle, the inbound coin is converted to USDsui and lands in your wallet within a minute. You do nothing. You see the new balance in your local currency, with USDsui balances and FX rates rendered behind the scenes.

Two pieces of protocol machinery deserve a sentence each. Every transfer to your handle is converted to USDsui before it touches your wallet, subject to a daily cap you control.

Privileged actions in Talise (changing roles, expanding allowlists) require multiple keys and a 48 hour delay before any change takes effect, so a single compromised key cannot move user funds.

5. Security and Recovery

Talise's job is to make sure the user's funds end up in the user's wallet and nowhere else. Three layers of protection make this true.

Funds always land in the user's wallet. The conversion path is hardwired on chain. There is no recipient parameter the worker can change. A compromised conversion worker cannot redirect the output of a swap because the destination is set in Move code to the user's own address.

Daily caps bound any compromise. Each user sets a per-transfer cap and a per-day cap on automatic conversion. The default is conservative. A compromised worker can convert up to

that daily budget before the protocol pauses, but it cannot exceed it, and it still cannot redirect the output.

Role separation with a delay. Privileged actions are split across four roles, with a 48 hour cancel window before any change activates. The full threat model is published in `SECURITY-V7.md`.

Slippage ceiling. Every conversion asserts on chain that the realized output is no worse than 2 percent below the quoted output. A misconfigured off-chain swap cannot quietly hand the user a bad price.

Recovery is Google recovery. A user who loses their phone signs in to a new device with Google and resumes against the same account. The account is a shared object on Sui; its address does not depend on the device. If the user loses access to their Google account itself, recovery is whatever Google provides: trusted contacts, backup codes, account-recovery support. This is the same trade Apple Cash makes with iCloud.

The Move package has 66 of 66 tests passing. An external audit is scoped before any deployment exceeds USD 10,000 in user funds. Candidate firms include OtterSec, Movebit, and Zellic.

6. Auto-Swap

Every transfer to your handle is converted to USDsui before it touches your wallet, subject to a daily cap you control.

The conversion happens automatically within a minute, and you do nothing. The user signs no extra transaction. The protocol watches for inbound coins on every Talise account, claims them on your behalf, routes them through a DEX aggregator to USDsui, and deposits the result directly into your wallet. The user sees a single activity entry: a deposit, denominated in their local currency.

Three properties make this safe. The destination is hardcoded to the user's own address on chain. The conversion is bounded by the user's daily cap. The price is bounded by a 2 percent slippage ceiling asserted in Move at the moment of deposit. The implementation details, including version history and the on-chain claim path, are documented in `AUTOSWAP.md`.

The user can disable auto-swap at any time, and the user can adjust the per-day budget at any time. Both actions are user-signed and sponsored. The protocol cannot raise the user's cap.

7. Surfaces

Talise is consumed through multiple clients. The protocol does not assume any one of them.

iOS app. A SwiftUI app that exercises the full protocol: sign-in, account creation, handle claim, send, receive, auto-swap settings, yield, and history. This is the primary surface for the launch corridor.

Web app. `talise.app` is the production landing surface. `app.talise.app` hosts the in-app web surface. The web codebase shares the same protocol bindings as iOS, and the auto-swap conversion runs on a Vercel cron sweep.

Onara sponsor. Onara is the gas-sponsorship layer. It is a Cloudflare Worker that signs as gas payer for every Talise transaction the user submits. The HTTP API is documented and self-hostable. An integrator who wants to sponsor its own users' Talise transactions deploys its own Onara against its own sponsor key.

8. Why Talise Is Different

	Talise	Wise / Remitly	Phantom / Suiet	Privy / Magic	Xend
Chain	Sui	(correspondent banking)	Sui	EVM (mostly)	Solana
Auth	zkLogin (Google)	Email + KYC	Mnemonic / passkey	OAuth + embedded key	WebAuthn passkey
Gas	Sponsored (zero SUI held)	N/A	User holds SUI	Per-app sponsorship	Sponsored
Recipient handle	<code>alice@talise.sui</code>	Bank details	Raw 0x... address	App-internal user id	Username
Inbound asset normalization	Automatic to USDsui	Local currency at the bank	None	None	Stablecoin
Stablecoin	USDsui (native)	N/A	Any	Any	USDC (bridged)
Fee on the stack	0% (off-ramp not included)	6 to 8% in Sub-Saharan corridor	DEX + gas (user pays)	Varies per integrator	Low

Finality	Sub-second	Days	Sub-second	Chain-dependent	Sub-second
----------	------------	------	------------	-----------------	------------

Versus Wise and Remitly. These are correspondent-banking products. Settlement requires a partner institution at the destination and is dominated by FX spread and inter-bank fees. Talise does not operate a balance sheet. Funds land on a public chain in the recipient's own account, denominated in USDsui from the moment the swap closes. The Talise stack itself charges zero. The only costs are the off-ramp at the destination, which Talise does not collect.

Versus Phantom and Suiet. Phantom and Suiet are wallets, which is to say applications a user opens, secures with a mnemonic, and funds with gas. They are excellent tools for users who want to be sovereign over a portfolio. They are the wrong abstraction for a recipient who just wants the equivalent of fifteen dollars in dollars. Talise inverts the assumption: the user sees a handle, a local-currency amount, and a balance.

Versus Privy and Magic. These are embedded-wallet SDKs. Each integration produces its own siloed account, so two apps using the same SDK produce two separate user-facing accounts. Talise's account is bound to the user's Google identity and resolves to a single handle across every Talise surface. Talise is the destination, not the SDK.

Versus Xend. Xend is the closest architectural neighbor. Xend is on Solana with WebAuthn passkeys; Talise is on Sui with zkLogin. Xend's bet is OS-level intent routing. Talise's bet is invisible auto-swap to a native dollar. Both are valid theses about where consumer crypto needs to close a gap. They are not the same product.

9. Use Cases

The diaspora sender. Amaka in London sends fifty pounds to her mother by typing `mama@talise.sui`. The recipient sees the equivalent in naira hit her wallet inside a minute, with no wire, no Swift fee, and no agent queue.

The freelancer. Tunde in Lagos invoices a US client in USDC, sharing only `tunde@talise.sui`. The payment lands in his wallet as USDsui, already dollar-denominated, ready to spend or to earn yield on.

The saver. Idle USDsui in any Talise wallet earns yield through Navi automatically. Withdraw any time; there are no lockups.

9.5 How Talise Earns

Talise transfers are free for the user. That is structural, not a promotion. The product earns on the boundary actions where the user moves between currencies and rails, the same way Wise, Revolut, and Chime do.

FX spread on auto-swap. When an inbound coin lands at a Talise handle and auto-swaps to USDsui through Cetus, Talise routes through a price that carries a small spread above the mid-market rate. The default is 30 basis points. On a 200 dollar inbound, that is 60 cents. The same transaction at Western Union costs 10 to 20 dollars.

Off-ramp margin. Sending USDsui between Talise handles is free. Converting USDsui to naira, cedis, shillings, or rand in a bank account is not. The conversion carries a spread, defaulting to 50 basis points, which sits inside the partner payout rail (Flutterwave, Paystack, M-Pesa). Compared against Wise's published corridor rates, Talise targets being equal or cheaper on the headline.

Yield rebate on idle balances. Talise's Earn surface routes idle USDsui to Navi. The displayed annual rate is the user's rate, not the underlying protocol rate. The delta between the gross Navi yield and the displayed yield is Talise's float income. On a 10 million dollar idle float, a 100 basis point delta is 100 thousand dollars per year.

On-ramp passthrough. Stripe Crypto Onramp handles fiat into USDsui. Stripe takes its cut, and Talise adds a small spread on top of the USD to USDsui leg, subject to Stripe terms.

None of these are hidden in fine print. Receipts will eventually surface the realised price against the contemporaneous mid-market reference, the same way Wise's app does today. The business model is transparent on purpose, because the African remittance audience compares Talise against Western Union and the spread Talise charges is an order of magnitude smaller.

10. Roadmap

Live (Q2 2026). Move package v7 on Sui mainnet with role separation, per-user daily caps, allowlists, global pause, and a 2 percent slippage ceiling. iOS in private beta. Onara worker running on Cloudflare with the conversion sweep. Web waitlist live at talise.app.

Q3 2026. External audit of the Talise Move package. Fiat off-ramp in Lagos through a Yellow Card or Onramper partner, closing the loop from USDsui in the wallet to naira in the bank account. Single-PTB onboarding so new users sign once.

Q4 2026. Multi-corridor expansion to Kenya (M-Pesa off-ramp), Ghana (mobile money), and South Africa (instant-EFT). TypeScript SDK so other Sui apps can offer auto-swap. Push

notifications on conversion so the recipient sees the inbound amount land without opening the app.

2027. Post-audit fixes shipped. Regulated entity in target markets, beginning with a payment-services license in Nigeria and a remittance license in Kenya. OS-level intent routing so handle taps in iMessage and select iOS apps open the Talise payment sheet directly.

11. Conclusion

Talise is consumer payments hidden inside a Sui address. The infrastructure is shipped: a Move package on mainnet with role separation, per-user caps, and a hardwired conversion path; an auto-swap loop that closes against mainnet's actual deposit semantics; a sponsor that lets the user transact without holding gas; and a zkLogin auth path that lets the user sign in with Google and never see a seed phrase.

The next twelve months are about distribution, not invention. External audit, fiat off-ramp, multi-corridor expansion, single-PTB onboarding, push notifications. None of these require a new primitive. All of them require careful product, partnership, and regulatory work in the markets that need the product most.

The applications are where the protocol becomes visible. The protocol is where the value lives.

12. References

1. Mysten Labs. *Sui: The Programmable Settlement Layer*. <https://sui.io/>
2. Mysten Labs. *zkLogin: A Privacy-Preserving Identity Solution for Sui*. <https://docs.sui.io/concepts/cryptography/zklogin>
3. Mysten Labs. *Sponsored Transactions on Sui*. <https://docs.sui.io/concepts/transactions/sponsored-transactions>
4. SuiNS Foundation. *Sui Name Service Documentation*. <https://docs.suins.io/>
5. World Bank. *Remittance Prices Worldwide Quarterly*, Issue 51, Q4 2024. <https://remittanceprices.worldbank.org/>
6. Cetus Protocol. *Cetus Aggregator API Documentation*. <https://cetus-1.gitbook.io/cetus-developer-docs/>
7. Navi Protocol. *NAVI Protocol Lending Documentation*. <https://naviprotocol.gitbook.io/>

8. OpenZeppelin. *OpenZeppelin Contracts for Sui v1.1.0*.
<https://github.com/OpenZeppelin/contracts-sui>
 9. Shinami. *Managed zkLogin Prover and Gas Station*. <https://docs.shinami.com/>
 10. Buterin, V. *Why we need wide adoption of social recovery wallets*. vitalik.ca, January 2021.
 11. Talise. *AUTOSWAP.md: Auto-Swap Architecture and Version History*.
[/move/talise/AUTOSWAP.md](#) .
 12. Talise. *SECURITY-V7.md: v7 Threat Model and RBAC*. [/move/talise/SECURITY-V7.md](#) .
-