



## WHITE PAPER

The purpose, the timeliness and the uniqueness of the project; what the network does better than existing solutions, and where its limits are. No formulas – for investors, partners, and anyone who needs the whole idea at once. The full technical specification lives in the Yellow Paper.

Xync Network L1 · July 2026

# Contents

---

1. The problem
2. The solution
3. Why it's cheap
4. What each unique feature unlocks
5. Offline payments: money where there is no signal
6. 255 currencies: a network where money is already exchanged
7. Decentralization without stopping the product
8. The XYNC token
9. Comparison
10. Growth vectors
11. Risks – honestly
12. Status and invitation

Xync Network is a Layer-1 blockchain that deals in money only: *send, request, exchange*. No virtual machine, no general-purpose contracts – and that self-restraint pays off: an irreversible, final transfer in ~0.3 seconds; a transaction of 80 bytes (3× smaller than an SMS); fees of a fraction of a cent – or zero; and a full wallet-node that runs right in your browser or inside Telegram. Currency exchange – swaps, liquidity pools, an interest-bearing deposit denominated in dollars – are not apps on top of the network but its own native operations (§6).

And unlike 99% of projects: **Xync already works**. The XyncPay payment service serves live customers in a Telegram MiniApp (t.me/XyncPayBot). This paper describes how a working product turns into an open, decentralized network – without stopping, and invisibly to its users.

---

## 1. The problem

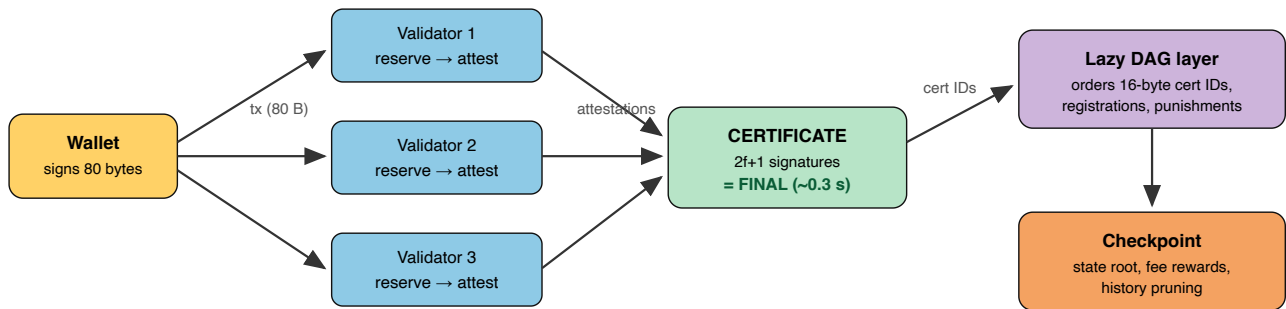
---

An international remittance: 3-7% in fees and days of waiting. A card payment: the merchant pays 1.5-3%, funds arrive days later and can be clawed back by a chargeback. Crypto promised a replacement – but Bitcoin finalizes in an hour, Ethereum in a quarter of an hour, fees swing from cents to tens of dollars, and the "general-purpose" chains are busy with everything (JPEGs, memecoins, derivatives) except the boring job of moving money from one human to another, cheaply and instantly.

Payments don't need a virtual machine. Payments need **speed, price, irreversibility, simplicity**.

## 2. The solution

Payment lifecycle: finality in one round-trip (fastpath), ordering later (lazy DAG)



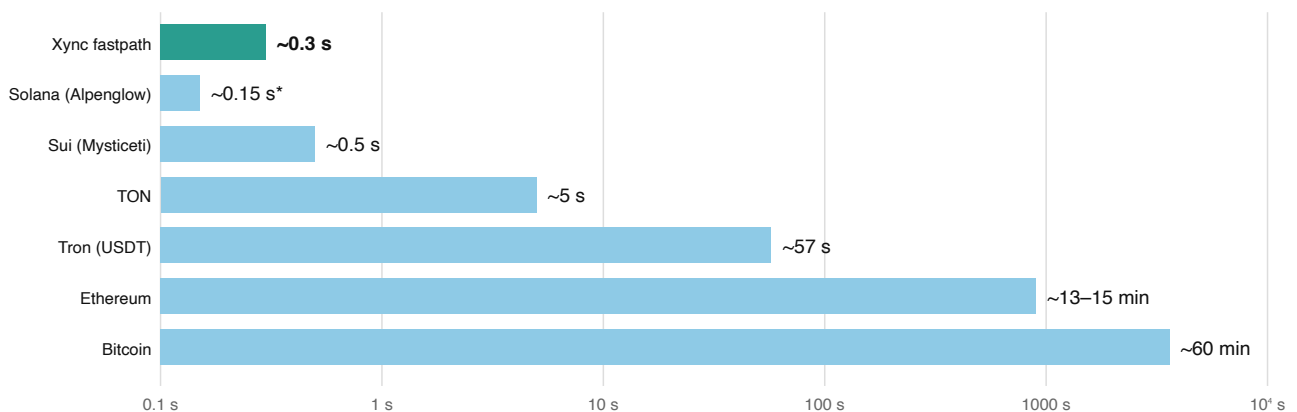
...n validators, any 2f+1 suffice

Money is final at the certificate step. The DAG never delays a payment — it only bookkeeps.

Xync has no mining and no queue to "get into a block." You sign a transaction – 16 bytes, it literally fits in a UUID – and hand it to the validators. Each validator instantly checks: *sender exists, funds are there, the operation number is right* – and signs. The moment the signatures reach a quorum ( $\frac{2}{3}$  of the network + 1), **that's it: the transfer is final**. One network round trip: roughly 0.15-0.4 seconds worldwide.

The math (proven in the Yellow Paper): two contradictory transfers can never both collect a quorum – just as one seat cannot be sold to two passengers if  $\frac{2}{3}$  of the ticket offices cross-check each other. You cannot cheat the network; you can only get caught: a double-spend attempt leaves a cryptographic proof, the fraudster's account is automatically banned and fined, and **half the fine goes to the node that caught them**.

Time to irreversible finality (log scale, typical values, 2026)

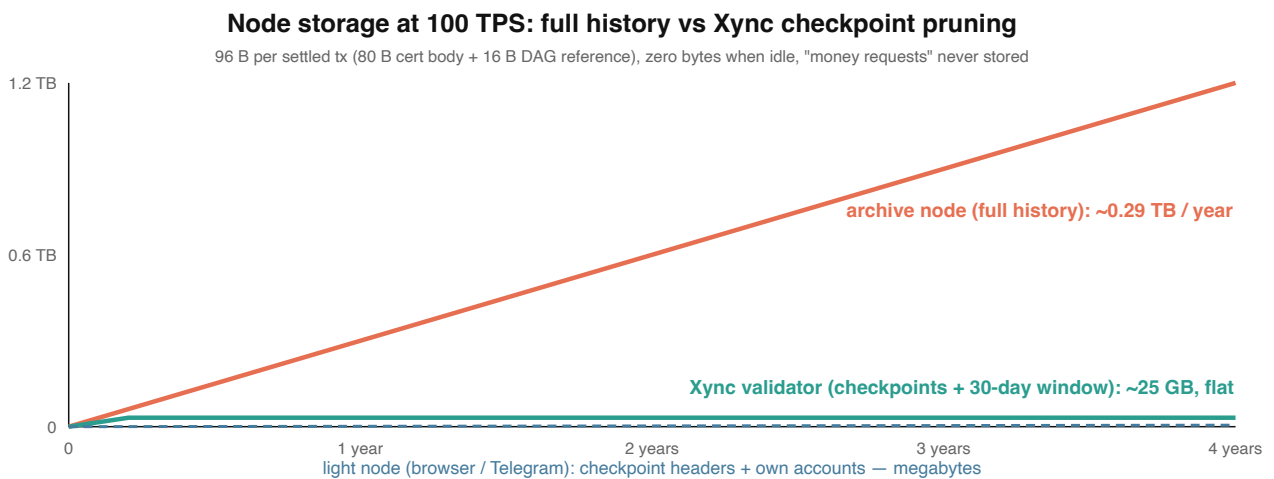


\* Alpenglow mainnet expected Q4 2026. Card networks authorize in ~2 s but settle in days and allow chargebacks.

### 3. Why it's cheap

A Xync transaction is 80 bytes with its signature – 3-4× smaller than the leanest competitors and thousands of times smaller than bank formats. Fewer bytes → less traffic, less disk, less electricity → lower cost → a base fee of 0.001 XYNC (a fraction of a cent). And when the network is idle it writes nothing at all – zero empty blocks (Ethereum and Solana stamp them around the clock).

There is also a **free lane**: every account gets a daily quota of zero-fee transactions. The good old `api.xync.net` stays forever as a "charity node" – customers of the classic XyncPay app keep sending money with no fees at all.



## 4. What each unique feature unlocks

---

**A transaction = a UUID (16 bytes).** A payment fits into a QR code with room to spare, an NFC tag, a receipt barcode, an SMS, a bank transfer memo field, an email signature. Scenarios: pay via a sticker on a market stall with no terminal; an invoice embedded in a paper bill; IoT micropayments (a sensor pays for bandwidth in 80-byte packets); offline hand-off of a signed tx over Bluetooth, submitted when connectivity returns.

**0.3-second finality, no reversals.** A checkout with no "waiting for confirmations": vending, transit, coffee. Selling digital goods with zero chargeback risk. Tipping a street musician by QR during the applause. Streaming payments: seconds of content or Wi-Fi paid as consumed. Settlements between bots and AI agents, for whom "days to clear" is nonsense.

**"Request money" – built-in invoicing that never touches the chain.** A request is a signed message with a TTL: splitting a restaurant bill (each friend gets a request for their share), collecting for a gift, scheduled rent, donations, a freelancer's invoice inside a Telegram chat, subscriptions. Payment settles the request automatically. The network's history stores none of it – zero bytes.

**255 currencies in one network.** USD, EUR, RUB, wrapped BTC, local-currency stablecoins, a coffee chain's loyalty points, an in-game currency, "corporate money" for inter-branch settlements – all identical 80-byte transactions at identical speed. This is the foundation of the network's uniqueness – section 6 is devoted to it: in-chain exchange, liquidity pools, and deposits directly in fiat.

**Payments without the internet – a class of its own.** Detailed in the next section: both the transaction and the proof of its execution fit in a QR code – money moves where there is no signal.

**Bank-receipt oracles (zkEmail class).** A DKIM-signed bank email is a cryptographic fact. On top of it: non-custodial fiat↔crypto exchange (escrow unlocks on proof of a bank transfer), invoice auto-settlement on a SEPA/FPS payment, stablecoin issuer proof-of-reserves.

**Short account numbers.** Account #518 is as memorable as an apartment number. "Send it to five-one-eight" works in a voice message. Early numbers carry status and collectible value (remember ICQ).

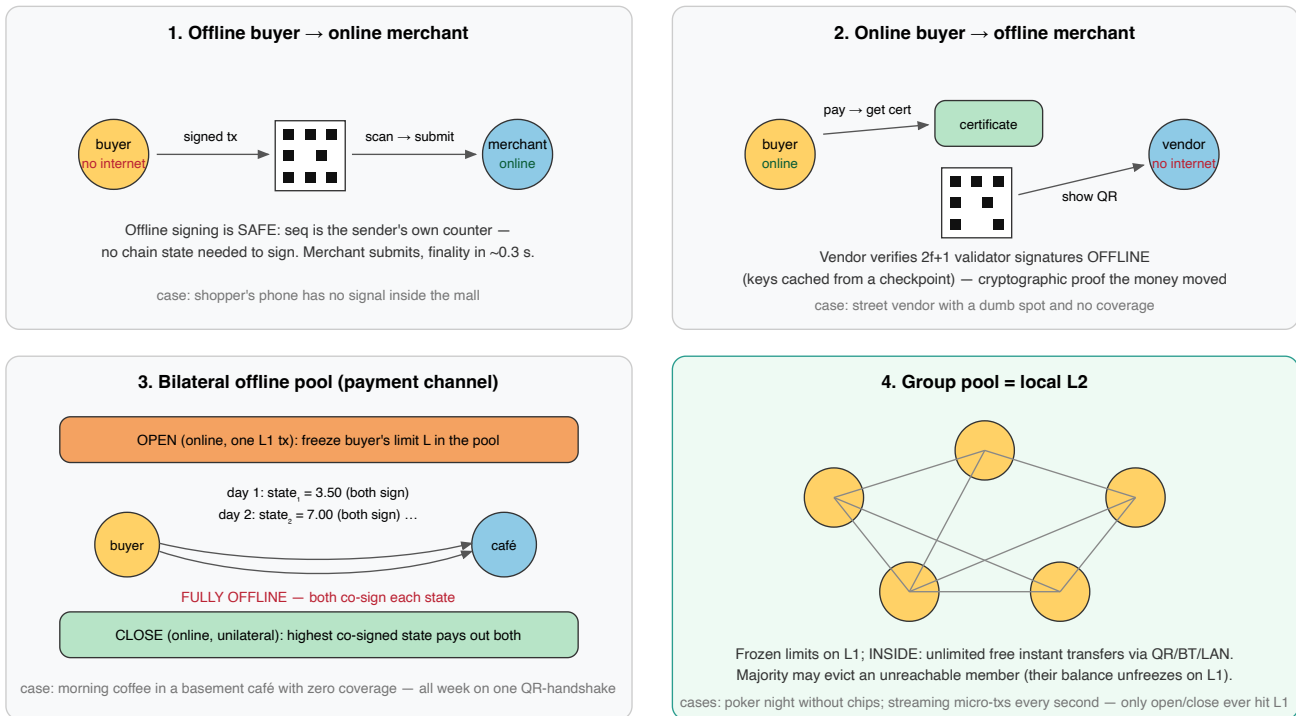
**A node inside the wallet.** A light node starts in seconds (checkpoint headers + your own accounts) and runs in a browser extension or Telegram MiniApp. You don't trust someone's server – you verify the quorum's signatures yourself. A full node fits on a Raspberry Pi (~25 GB at 100 TPS, thanks to history pruning).

# 5. Offline payments: money where there is no signal

A Xync transaction is 80 bytes; the proof of its finality (the certificate with quorum signatures) is about 280 bytes. Both fit comfortably in an ordinary QR code. Out of this single fact grows a class of capabilities no other payment network offers:

## Offline payments: everything fits in a QR code

tx = 80 bytes → tiny QR; certificate (tx + quorum signatures ≈ 280 B) → still a small QR, verifiable OFFLINE against cached validator keys



**Mode 1: an offline buyer pays an online merchant.** Signing works fully offline — nothing is needed beyond the sender's own operation counter (our seq property). The buyer shows a QR, the merchant scans it and submits the transaction — finality in 0.3 s before both their eyes. *Case: your phone loses signal in a basement store — the purchase still happens.*

**Mode 2: an offline merchant accepts from an online buyer.** The buyer pays from their own phone and shows a QR with the certificate. The merchant's wallet verifies the validator quorum's signatures OFFLINE (keys cached from the latest checkpoint) — a cryptographic proof the money has already arrived, not a promise. *Case: a street vendor in a dead zone takes payments all day and goes online only in the evening.*

**Mode 3: a buyer ↔ merchant offline pool (a payment channel).** While at least one of the two is online, they open a pool: a spending limit is frozen on the buyer's balance with a single transaction. From then on — full offline for both: every purchase is an exchange of QR codes co-signing a running tab. Either party can close the pool unilaterally (online); both balances settle by the latest co-signed state. *Case: a person with no mobile data gets coffee every morning in a basement café with no coverage. One handshake a week — seven days of offline coffee.*

**Mode 4: a group pool – a pocket L2 blockchain.** A group, by prior agreement (one member online is enough to collect the signatures), freezes per-member limits. Inside – unlimited instant transfers to each other in full offline, as long as members stay in contact (QR/Bluetooth/LAN). The majority may evict an unreachable member – their current balance unfreezes on the network; returning, or joining as a newcomer, requires going online. *Case 1: poker night without chips – every pot settles with a QR scan. Case 2 (the online variant): the same mechanics as a local L2 – opening and closing the pool are the only two on-chain transactions, while inside you can STREAM payments every second, free and unlimited: per-second content billing, in-game economies, settlement for a swarm of IoT devices.*

**Mode 5: a free escrow agent – one who never holds the money.** A group pool closes on the signatures of a majority of its members. Form a pool of three – a buyer, a seller, and anyone both of them trust – and any two of you are that majority. The buyer freezes the price, the seller freezes a bond, the guarantor freezes nothing.

If the deal goes well, buyer and seller sign the outcome between themselves: that alone releases the money, and the guarantor takes no part and earns nothing. If a dispute arises, the guarantor signs the outcome together with whichever side he judges right. And stealing is physically impossible for him: his lone signature is not enough, and the second will only come from the person he did not rob. The money sits frozen on the blockchain the entire time, never in an intermediary's account – unlike every classical escrow, where the intermediary owns it and charges 1-5% for the privilege.

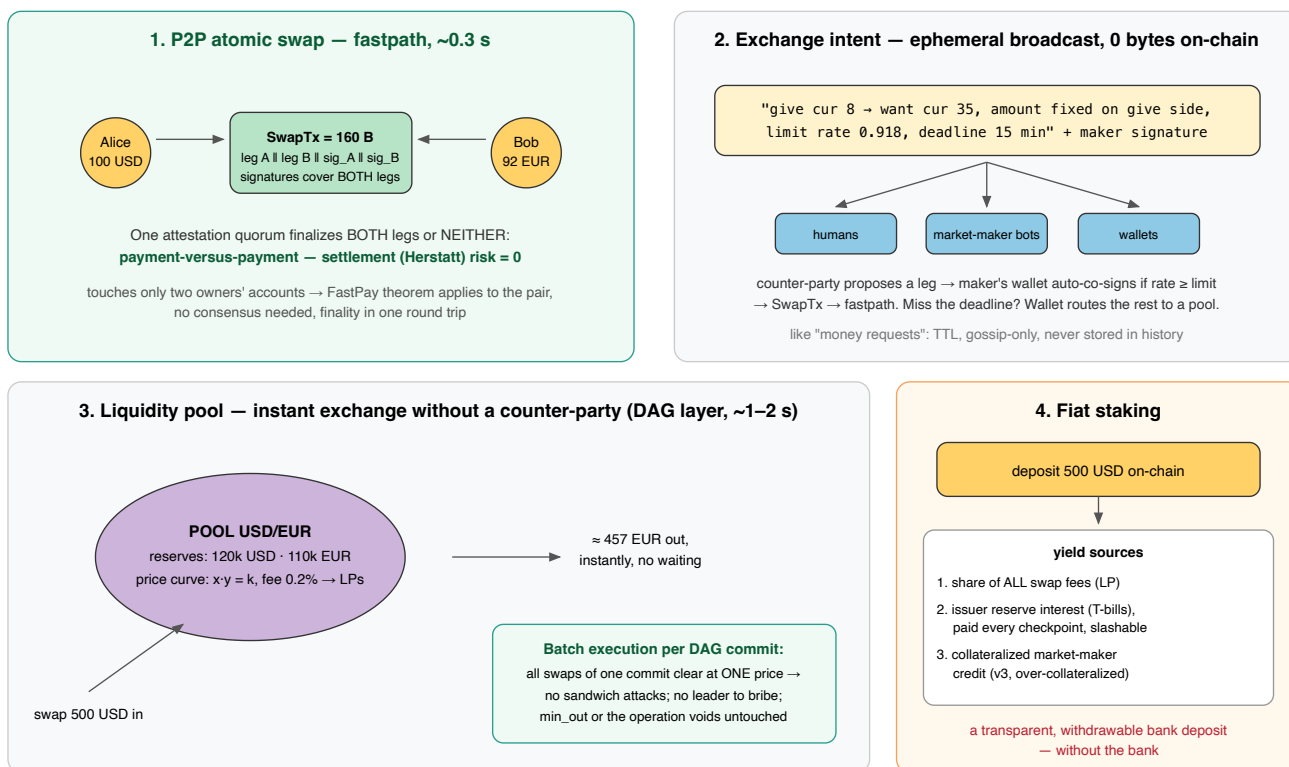
*Case: buying a car from a stranger, a marketplace deal, a freelance job, swapping cash for crypto in person. The internet is needed twice – to open the pool and to close it; the deal itself happens offline, by exchanging QR codes. Honest caveat: a guarantor who colludes with one side robs the other. That is the ceiling of every "two of three" scheme – which is why the guarantor is chosen by both.*

Of all this activity, only the opening, evictions and closing ever reach the blockchain. Thousands of inner payments cost the network – and its users – zero.

# 6. 255 currencies: a network where money is already exchanged

An ordinary blockchain knows one currency – its own. Everything else there is either a contract token with its own bugs, or someone else's coin "wrapped" and shipped over a bridge (the biggest hack source in crypto history: Ronin – \$624M, Wormhole – \$326M...). In Xync **255 currencies are wired into the protocol itself**: the dollar, the euro, a regulated issuer's bitcoin wrapper, a coffee chain's points – they are not contracts but native citizens, with the same 80 bytes, the same 0.3 s finality and the same offline capabilities. There are no bridges – so there is nothing to hack.

## The 255-currency exchange layer: three mechanisms, one network



## Currency exchange inside one blockchain – three ways.

*Instantly, through a liquidity pool.* No waiting for a "matching wisher": pools hold a reserve of both currencies, the swap executes immediately by a transparent formula, and the 0.2% fee goes to those who provided the liquidity. The anti-manipulation trick: all swaps of one network "beat" execute at one shared price – inserting a trade "just before yours" (the sandwich attack, the plague of DEXes) is mathematically impossible.

*P2P, by broadcast.* You announce to the network: "swapping dollars for dirhams, amount fixed on the dirham side, rate no worse than X, valid 15 minutes." The announcement travels the gossip network (never written to the chain – zero bytes), a counter-party – human or market-maker bot – responds, and the wallets settle the deal. Deadline passed? Your wallet auto-fills the remainder through a pool.

*Atomically, with zero settlement risk.* Any P2P exchange is ONE operation with two legs: either you received the dirhams AND gave the dollars, or nothing happened at all. Neither side can "receive without giving."

### **An on-chain deposit directly in fiat – the bank deposit replaced.**

Put 500 USD "at interest" with one tap – and, unlike a bank, see exactly where the yield comes from:

1. **Exchange fees.** Your deposit works as liquidity: everyone who swaps currency through the pool pays 0.2% – split among the pool's depositors. People always exchange money – the deposit always earns.
2. **Issuer interest.** On-chain dollars are backed by the issuer's reserves (T-bills); the issuer shares the reserve yield with deposit holders – like a bank deposit, except the rate is declared on-chain, payouts arrive automatically every network beat, and the reserves are checked by cryptographic proofs (proof-of-reserves), not by a fine-print annual audit.
3. **Market-maker credit (next stage).** Deposits lend to professional participants against over-collateral, with automatic liquidation – one more source of interest.

Withdrawal – at term, or instantly for a small exit fee.

### **What this unlocks (scenarios closed to single-token chains):**

- salary in USD, rent in EUR, coffee in local currency – one account, no exchange offices along the way;
- transfer-with-conversion as one operation: you send dollars, grandma receives tenge (the wallet routes through a pool by itself);
- a merchant prices in their own currency and accepts ANY – the network converts on the fly at an honest rate;
- corporate treasury: balances in ten currencies, rebalancing in seconds, PvP safety on every conversion;
- loyalty points swappable to dollars in one tap (a café's currency is a currency like any other);
- citizens of weak-currency countries hedge inflation with one swipe "local → USD" without leaving their payment network;
- an FX market without an exchange: market makers earn spreads right in the protocol, deepening liquidity for everyone.

**This already works – not a roadmap.** The exchange layer is implemented and verified in the prototype: 46 protocol invariants and end-to-end tests on a real network of four validators. The implementation yields advantages that ordinary DeFi protocols lack by construction:

- **Constructive anti-MEV, not a promise.** On ordinary DEXes a bot sees your trade and inserts its own around it (the sandwich attack), skimming part of your amount; fighting this is an endless arms race. Here all swaps of one beat execute at ONE clearing price, and only the amounts – not the order – enter the computation, so there is nowhere to insert a trade "before yours." It is a property of the algorithm; it cannot be worked around.

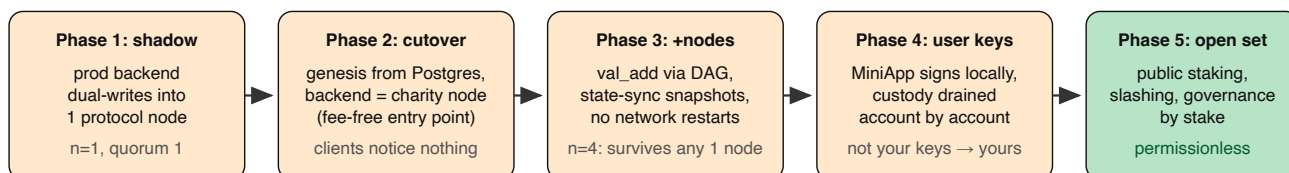
- **One result for everyone – money that does not "depend on the node."** Any pool or deposit operation produces a byte-for-byte identical outcome on every network node: the arithmetic is integer, the order is set by consensus. No "one node shows this balance, another shows a different one."
- **Exchange with no settlement risk.** A P2P swap is one indivisible operation: either both sides received theirs, or nothing happened. The situation "I sent, the counter-party vanished" – the one for which bank FX builds expensive clearing houses – is impossible here.
- **A deposit with an honest yield source.** The interest is not conjured from thin air: it is a share of real exchange fees and the issuer's declared on-chain rate, visible on the blockchain; accruals arrive automatically every beat. If the issuer does not pay, the network records it on-chain rather than hiding it.

Honestly about the prototype's boundaries: some mechanisms are deliberately simplified (details and the development plan – Yellow Paper §13.7 and the decision log [docs/decisions.md](#)); this is a working reference implementation, not final production code.

The technical mechanics (160-byte atomic swaps, intents, the pool formula, batch pricing) – Yellow Paper §13.

## 7. Decentralization without stopping the product

### Seamless decentralization of a LIVE payment product (XyncPay)



**Trust milestone: at  $n = 4$  validators (quorum 3) the operator loses unilateral power over payments** – every transfer needs signatures from independent parties; every phase is reversible until announced final.

Legacy history is anchored in genesis (legacy\_root hash) – the new chain cryptographically references its centralized past.

Most networks launch empty and then spend years hunting for users. Xync walks the opposite path: a working centralized service gradually gives its power away. Customers notice nothing – except a new line, "finalized in 312 ms." The entire legacy history is cryptographically anchored in the new network's genesis (the legacy\_root hash): the past is verifiable, balances are auditable. At four validators the Xync operator physically loses the ability to unilaterally confirm (or censor) payments; beyond that, the set opens up by stake.

## 8. The XYNC token

---

XYNC is the network's fuel: all fees are paid in it. Demand comes from transactions and validator staking; supply compresses: 25% of every fee is burned, and emission starts at 4%/yr decaying to a 0.75% floor (formulas in the Yellow Paper). Validator rewards depend on uptime and contribution – the network pays for quality of service, not farm size. Registration fees (\$1 per account) fill the development treasury.

Genesis allocation (proposal): 40% treasury/ecosystem (4-year vesting), 20% team (1-year cliff), 15% launch staking reserve, ≤15% investors, 10% community for testnet activity.

## 9. Comparison

---

	<b>Xync</b>	<b>Visa/MC</b>	<b>Bitcoin</b>	<b>Ethereum</b>	<b>Solana</b>	<b>TON</b>	<b>Tron</b>
Irreversible finality	<b>~0.3 s</b>	days (chargebacks)	~60 min	~13 min	~0.15 s*	~5 s	~57 s
P2P fee	<b>~\$0.001 / 0</b>	1.5-3% (merchant)	\$0.5-50	\$0.1-50	~\$0.0003	~\$0.01	~\$0.3 (USDT)
Tx size	<b>80 B</b>	–	~250 B	~110 B	~230 B	~300 B	~200 B
Wallet node in browser	<b>yes</b>	–	no	partially	no	no	no
Free tier	<b>yes</b>	no	no	no	no	no	no
Live product with users	<b>yes (XyncPay)</b>	yes	yes	yes	yes	yes	yes

\* Alpenglow, mainnet ~Q4 2026. Honest about the boundary: Xync is not a smart-contract platform; anything that isn't "send/request money" deliberately doesn't fit. That is the price of speed and simplicity.

## 10. Growth vectors

---

1. **Remittance corridors** (CIS, Gulf↔South Asia) via the oracle on-ramp – a \$700B+/yr market suffering 3-7% fees.
2. **B2B invoicing**: "request money" as an API/SDK for POS, CRM and Telegram bots of small businesses; fees an order of magnitude below card acquiring.
3. **Micropayments & streaming money**: content, gaming, AI agents, IoT – segments physically impossible above \$0.01 per tx.
4. **Programmable payments without a VM**: typed escrows (delivery-versus-payment via a courier-API oracle), recurring payments, splits – as protocol types, not contracts.
5. **Local stablecoins and loyalty currencies** – 255 registry slots.
6. **The Rust/WASM node** in every wallet: verifiability as a mass feature.

## 11. Risks – honestly

---

Technical: the fastpath punishes buggy wallets (mitigated by the iron signing rule and graduated fines); the DAG layer is sensitive to bad networks (never touches payments; plan B is a Raptr-style path). Market: a narrow domain means less speculative hype – the bet is on real payment volume. Regulatory: payments are regulated; the strategy is non-custodial design, licensed gateway partners, geographic validator diversity. Competitive: giants (Visa, Solana Pay, TON) may move into the niche – our answer is the iteration speed of a specialized protocol and XyncPay's already-live user base.

## 12. Status and invitation

---

Working today: the XyncPay production service (TG MiniApp), the network prototype (4 validators × 5 microservices, one `docker compose up`), a 46-invariant protocol simulation, scripts for seamless validator onboarding and live-database migration. We are opening early-validator slots for phase 3 and hiring Rust core engineers.

*Mechanics in depth – the Yellow Paper. The product – [t.me/XyncPayBot](https://t.me/XyncPayBot).*