SOLANA

Contra

Public liquidity. Private execution.

The private execution infrastructure for internet capital markets.

Research & Development

Why Solana?

Over the past five years, Solana has emerged as the undisputed platform for digital finance. With more trading volume, more developers, and more institutional adoption than any other network, Solana has proven it can handle real-world finance at scale.

$1T+	$0.001	25%	<1s
2025 DEX Volume	Median Tx Cost	Nasdaq daily trades	Finality

The fastest-growing financial ecosystem

Solana leads in developer growth, application revenue, and new asset creation. Institutions including Visa, PayPal, BlackRock, and Apollo have launched products on Solana. This ecosystem velocity means access to battle-tested tooling, proven infrastructure, and a talent pool that can ship production-grade applications in weeks, not years.

Deep, liquid capital markets

With $5B in daily DEX volume and $1T+ traded in 2025, Solana hosts the most liquid onchain markets globally. Solana has battle-tested infrastructure, liquidity with tight spreads, and institutional participation. Solana is the right choice for real-world trading, global state management, and treasury operations.

High throughput

Solana consistently performs over 1,000 transactions per second (TPS) in production. Last quarter, Solana processed 8.5 billion transactions, more than all other chains combined. Network upgrades like Alpenglow are expected to deliver sub-150ms finality and capacity for 100,000+ TPS.

Introducing Contra:
Public Liquidity. Private Execution.

Contra is a private payment channel with access to Solana's global capital markets. Contra delivers instant finality within a controlled environment.

Private Operations

All transaction data stays within your infrastructure. Execute thousands of transactions instantly with complete privacy. No public mempool, no front-running, no data leakage. You control who participates, what rules apply, and how data is accessed.

Regulatory Control

Implement custom KYC/AML rules, jurisdiction-specific compliance requirements, and operational policies. By owning the entire stack, you maintain complete control of your data, enabling full audit trails without public disclosure.

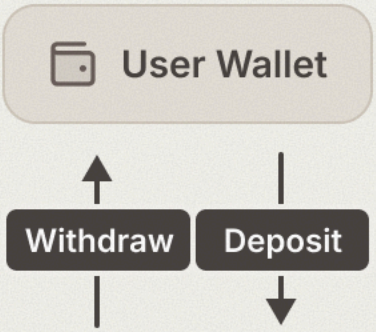
Global Liquidity Access

Contra payment channels maintain direct access to liquidity on Solana, including billions in stablecoins, institutional counterparties, deep liquidity pools, and a broad spectrum of digital and real-world assets.

How Contra Works

Mainnet

$100B+ Liquidity



Contra Payment Channel

Private

Transactions by default are configured to be completely private.

Instant

Up to thousands of transactions per second. 100ms settlement batches.

Your Rules

Set access controls, rate limits, compliance requirements.

Zero Fees

r

Contra

Technical Specifications

Contra Escrow

The Escrow and Withdraw programs move assets in and out of Contra channels. Deposits lock tokens in escrow; withdrawals use sparse Merkle tree (SMT) proofs to prevent double-spends

Role-Based Security Model

Businesses create private Contra instances with isolated rules and authorities.

The program implements three privilege levels:

- **Admin:** Creates instances, manages token mint whitelists, designates operators
- **Operator:** Executes withdrawals with cryptographic proofs
- **User:** Deposits tokens (subject to mint whitelist)

Deposits & Mint Whitelisting

Admins invoke AllowMint for each SPL token, creating an AllowedMint PDA and initializing the instance's Associated Token Account.

This whitelist prevents unauthorized deposits. Admins revoke via BlockMint, blocking new deposits while preserving withdrawals for existing balances.

Withdrawal Proofs

The protocol tracks withdrawals using a binary SMT (height 16, capacity 65,536 leaves). Each withdrawal receives a monotonically increasing nonce. Withdrawals require dual proofs:

- **Exclusion proof:** The nonce does NOT exist in the current SMT root
- **Inclusion proof:** The nonce DOES exist in the new SMT root

This ensures withdrawals can't be processed twice — a second attempt fails exclusion verification.

At capacity, ResetSmtRoot increments the tree index. Nonces encode their generation, preventing cross-generation reuse.

Contra DB

Contra uses PostgreSQL for settled state with a hybrid in-memory/ persistent architecture optimized for hot account access.

Schema Design

Three core tables store settled state:

1. **Accounts:** bincode-serialized AccountSharedData containing lamports, owner, program data.
2. **Transactions:** complete transaction history for audit trail.
3. **Blocks:** slot progression for settlement coordination.

To maximize write throughput, the schema uses only primary key indexes. Queries must look up records by pubkey, signature, or slot.

The Best-of-Both Cache

BOB (Best-Of-Both) is an in-memory HashMap that caches account state during execution. When the SV executor needs account data, BOB checks its cache first. On a hit, the account is returned from memory. On a miss, BOB synchronously fetches from PostgreSQL, blocking execution until the query completes.

Write/Read Separation

Contra separates write and read nodes. The write node connects to the PostgreSQL primary for transaction execution. PostgreSQL continuously streams write-ahead log (WAL) entries to a replica database in real-time to serve read node queries.

Indexer

The indexer monitors onchain events and writes deposit records to PostgreSQL. It operates independently of the payment channel.

The indexer supports two datasource strategies:

1. **RPC Polling:** Fetches blocks via getBlock RPC calls. Configurable batch size, poll interval, and starting slot. Handles skipped slots with exponential backoff. Suitable for low-to-moderate volume.
2. **Yellowstone gRPC:** Real-time block streaming via gRPC (Yellowstone protocol). Uses Vixen parsing framework to decode instructions. Lower latency, reduced RPC load. Requires Yellowstone gRPC endpoint.

Both parse Escrow/Logger program instructions into a common type before writing to PostgreSQL.

Operational Considerations:

- **Channel Forwarding:** Parsed instructions flow through an async channel, decoupling block fetching from database writes with automatic back-pressure.
- **Database Persistence:** The transaction processor writes to PostgreSQL's transactions table: signature (unique), slot, addresses, mint, amount, and type.
- **Backfills/Missed Blocks:** On restart, the indexer backfills any gap between the last checkpoint and Solana's current slot using parallel RPC fetching. Then the configured datasource resumes real-time streaming.

Deploying Contra

Contra ships as a Docker Compose stack with pre-configured components for production deployment. The architecture separates write/read paths and includes full observability infrastructure.

Core Services:

- **Irite Ilode:** Processes transactions, connects to PostgreSQL primary (8899)
- **Read Ilode:** Serves queries, connects to PostgreSQL replica (8900)
- **PostgreSQL Primary/Replica:** Physical streaming replication with replication slot
- **Indexer:** Polls Solana Mainnet for deposit events, writes to separate PostgreSQL instance

Observability Stack:

- **Prometheus:** Scrapes cAdvisor metrics every 15s
- **Grafana:** Pre-configured dashboard for node metrics, database lag, container health
- **cAdvisor:** Exports CPU, memory, disk, network metrics per container

Deployment Steps

1. Configure admin keys in **docker-compose.yml** (CONTRA_ADMIN_KEYS)
2. Run `docker compose up -d to start all services`
3. Monitor Grafana at `localhost:37429` (default credentials: **admin/admin**)
3. Write `Node exposes RPC` at `localhost:8900`

Get started with Contra

Reach out to your Solana Foundation team for more details.