Hashgraph

Achieving privacy & sovereignty via cross-ledger protocols



KASH BALHOTRA
Institutional Digital Assets Lead
kash.b@hashgraph.com

Privacy model on HashSphere

Transaction privacy is enforced cryptographically through zkSNARKs and encrypted balances so no party, including the network operator sees the transaction details.

Transactions **settle in under 3 seconds** and are mathematically guaranteed with hashgraph consensus for fair ordering

Independent network design–**not a subnet**. Your choice of governance.



HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Hashgraph

Atomic settlement without exposure

Atomic settlement across DvP, PvP, and private transfers without exposing any data to third parties

01

Three dimensions of transaction privacy

HashSphere enforces all three simultaneously — without sacrificing cryptographic trust or introducing centralized routing.

01

Confidentiality

AMOUNTS & BALANCES STAY HIDDEN

What it hides

Transaction amounts, asset types, and resulting balances

Example

Corporate treasury moves stablecoins between subsidiaries. Competitor on same network cannot infer liquidity or capital strategy

How

Private Balance encryption — arithmetic runs on ciphertext

02

Anonymity

COUNTERPARTY IDENTITIES STAY HIDDEN

What it hides

Who is trading with whom, when, and how frequently

Example

Tier-1 bank executing cross-border FX does not want competitors mapping its activity with Institution C on the same network

How

zkSNARKs — zero-knowledge proofs verify validity without revealing sender, recipient, or amount

03

Selective disclosure

AUTHORIZED PARTIES GET CRYPTOGRAPHIC ACCESS

What it hides

Full transaction detail — but only for designated parties (regulator, custodian, CSD)

Example

Regulator opens supervisory review 3 months later. No operator involvement needed — they hold a scoped read key from settlement

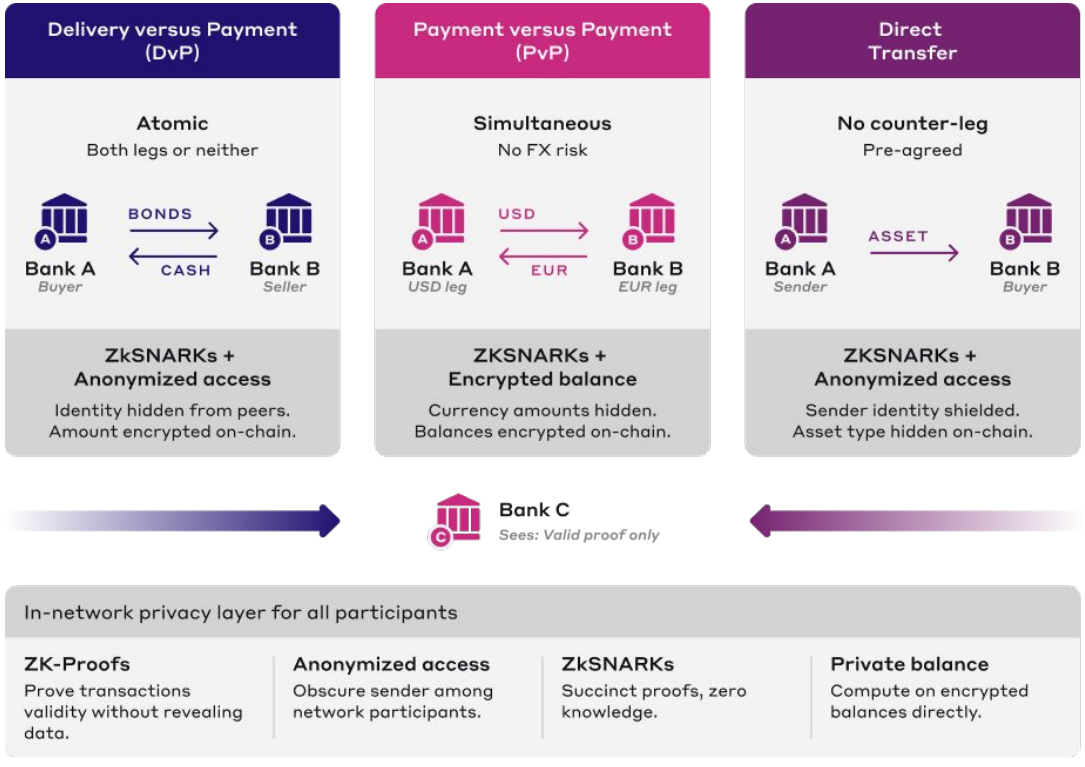
How

Read keys — scoped per transaction, cryptographically enforced, non-retroactive

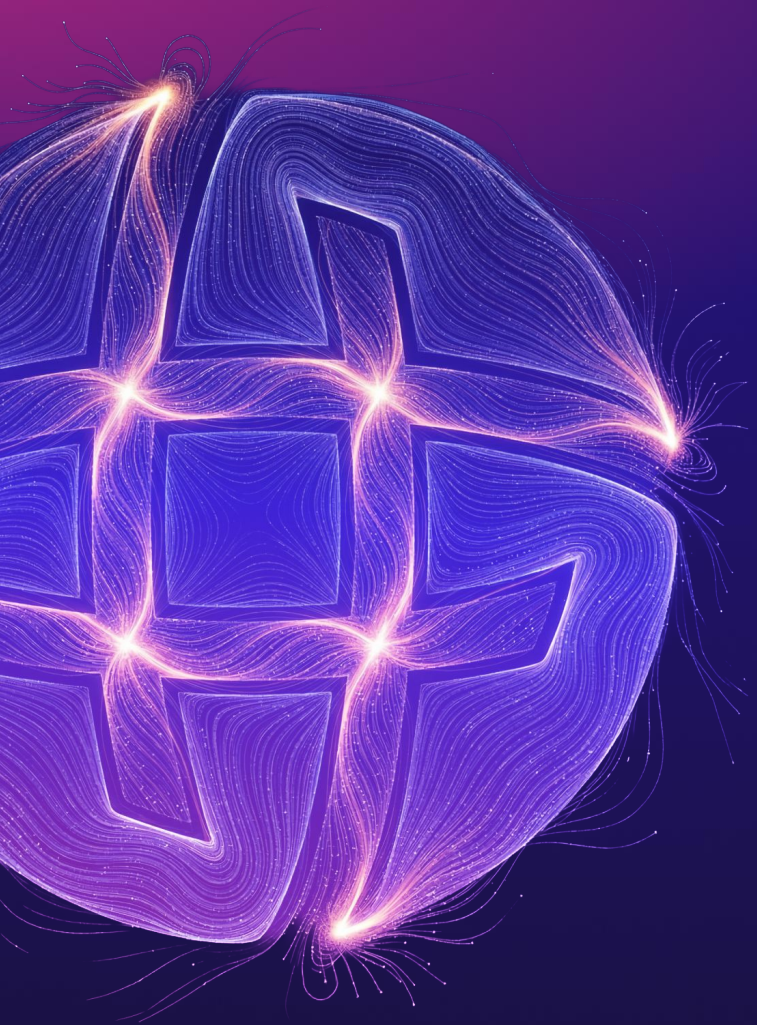
HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Transaction privacy approach across all settlement types

- All transactions (counterparty identities, amounts, asset types) can be encrypted on-chain (identities optional)
- Atomic finality–settlement completes or reverts fully. No partial exposure at any stage.
- All methods use ERC-20 on-and off-ramps (shield / unshield)



HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.



Demo of HashSphere Privacy in action

See Demo

 | © 2026 Hashgraph Foundry Inc. All rights reserved.

What HashSphere privacy delivers

Built around how institutions actually work — not around how a network wants them to work.

What HashSphere provides

- A private shielded account on your network — balances and trade activity visible only to you
- Simultaneous settlement for both legs of every trade — PvP or DvP — so neither side carries settlement risk
- A verified record of every settlement you control, available for regulatory audit on demand
- Integration with your existing signing and custody infrastructure — your HSM, your KMS, we never touch your keys

What your institution does

- Continue negotiating trades the way you do today — FIX, OTC desk, bilateral confirms
- Move funds into the private shielded account before trading (once per trading period, not per trade)
- Submit your settlement instruction with your standard trade reference
- Pull funds back to your main account when you are done

HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Setup & privacy configuration

Banks connect once. Privacy profile is chosen at pool deployment — not per transaction.

PREREQUISITES

- HashSphere node deployed in bank's infrastructure
- Signing key loaded via HSM/KMS (AWS, GCP, Fireblocks, DFNS)
- SDK connected: `connect(rpcUrl, contractAddress)`
- Key imported: `importKey(keyMaterial)`

Full privacy — selected

Counterparties AND amounts hidden. Deposit shuffling unlinks your institution from your settlement activity. Neither Bank B nor any other member can trace a settlement back to Bank A.

Also available

Confidentiality-only pool (amounts hidden, counterparties visible) for treasury flows that don't need full anonymity. Deploy multiple pools per use case — only pay the full-privacy cost when you need it.

SDK endpoint

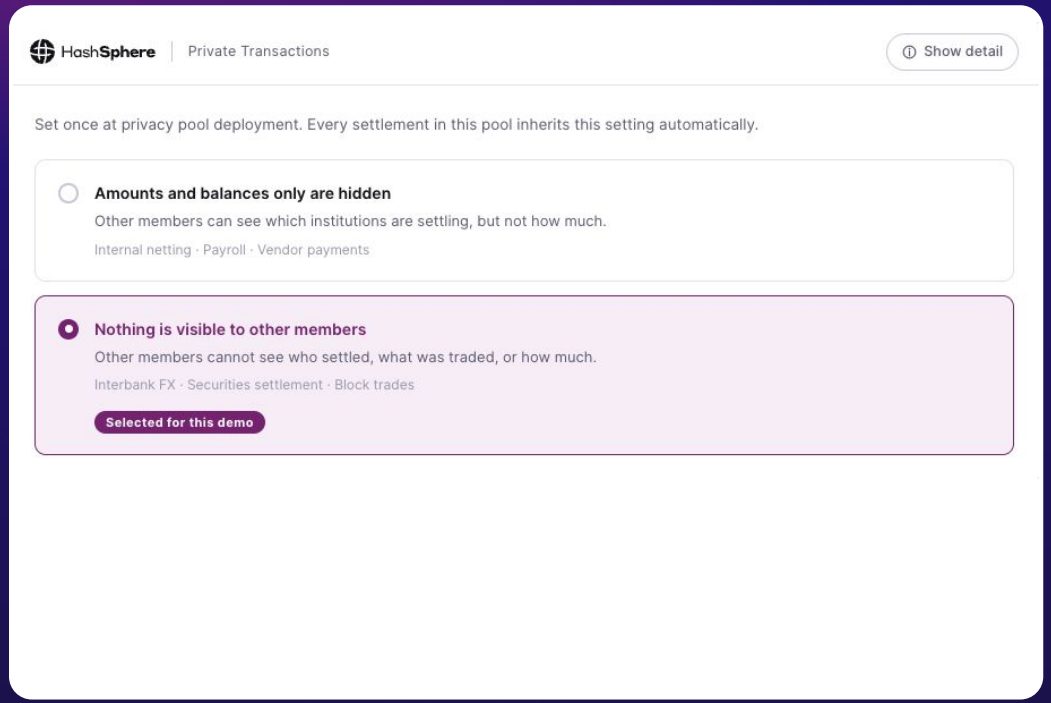
```
connect(rpcUrl, contractAddress) +
importKey(keyMaterial)
```


HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Choose your privacy profile

SETUP

Set once at pool deployment. Every settlement in this pool inherits this setting automatically.



What's happening

Bank A deploys this pool with 'Nothing is visible' selected. Every settlement in the pool inherits this setting. The choice affects latency and throughput — full anonymity takes ~2s for proof generation.

SDK calls

```
connect(rpcUrl, contractAddress)
```

```
importKey(keyMaterial)
```

What this means

- Pool profile is fixed at deployment
- 5–100 node consortium supported

Each settlement is recorded as an encrypted entry — like a locked safe-deposit box. The network confirms the box exists and rules were followed, without opening it. Only the two parties hold the key.

HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

The demo scenario

Bilateral PvP settlement between Bank A and Bank B on a six-node consortium. Terms agreed off-chain via FIX before any on-chain step.

BANK A

LEI: 549300EXAMPLEABC0001

PAYS
100,000,000
USDC

ATOMIC
PvP

BANK B

LEI: 549300VBWWV6BYQOWM67

PAYS
130,000,000
XSGD

TRADE REF	TYPE	FX RATE	VALUE DATE
HS-2026-0614-PVP-A001	PvP — Payment vs Payment	1.3000 USD/SGD	6 July (T+2)

DEMO STEPS

1. **Fund account**
   Move funds to shielded account
2. **Off-chain RFQ**
   FIX ticket agreed, UUID set
3. **Propose Settlement**
   Bank A proposed settlement based on agreed terms
4. **Complete Settlement**
   Bank B accepts settlement terms, balances updated
5. **Observer view**
   What Bank C sees
6. **Return funds**
   Back to main account
7. **Audit**
   Regulator verifies one trade

HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Shield funds

STEP 1/7

Bank A moves USDC into their private shielded account on HashSphere. Balance is encrypted — other participants cannot see the amount.

Bank A shields USDC into their private settlement account. Balance is encrypted — other participants cannot see the amount.

Fund Account | Settle Trade | Return Funds

INSTITUTION DETAILS

INSTITUTION

Bank A

SETTLEMENT POOL

Private Interbank FX

AMOUNT TO FUND

CURRENCY

USDC — USD Stablecoin

AMOUNT

42,482,967

What's happening

Bank A deposits 500M USDC. The SDK generates a random salt, computes a cryptographic commitment, and submits a zero-knowledge proof on-chain. The amount is never written to the ledger in plaintext. Deposits are anonymized post-deposit via balance shuffling.

SDK call

```
deposit(assetType, amount, proof)
```

Cryptographic primitives

- Random salt
- Hash (BankID | CurrencyID | Amount | Salt)
- Groth16 zero-knowledge proof
- Deposit anonymization via shuffling

Private / Visible

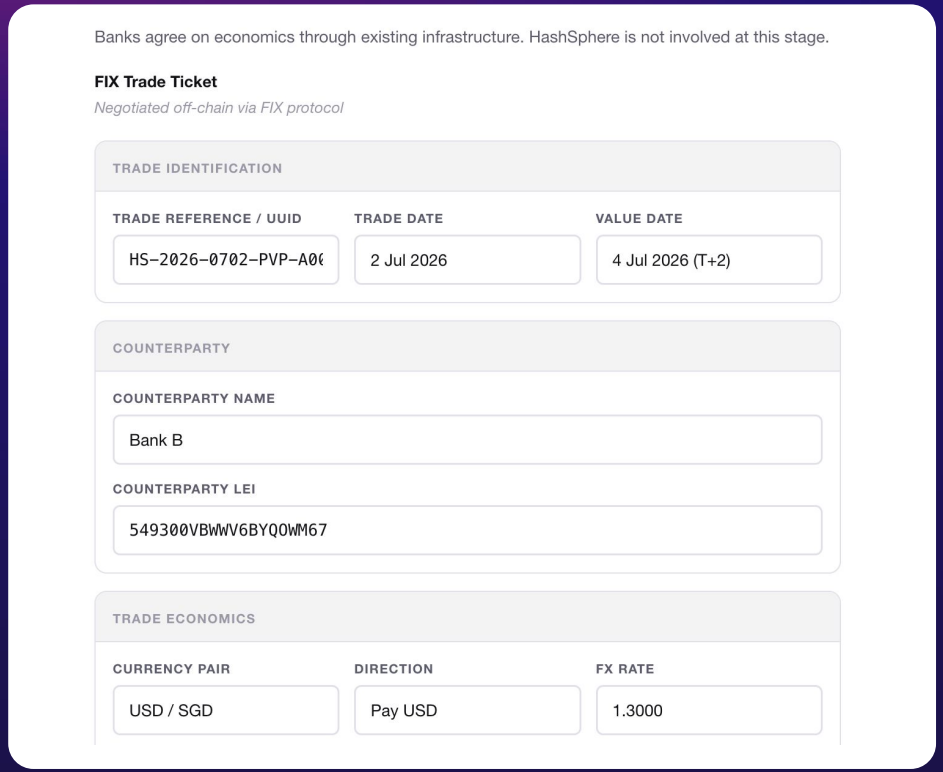
- Resulting account balance
- Amount deposited
- Bank A funded an account in this pool
- ZK proof verified on-chain

HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Off-chain term agreement

STEP 2/7

Banks agree on economics through existing infrastructure. HashSphere is not involved at this stage.



What's happening

Banks negotiate via FIX as they always have. The Trade Reference UUID agreed here links both settlement instructions on-chain. HashSphere never replaces FIX — it picks up after terms are confirmed.

SDK calls

None — this step uses your existing FIX infrastructure.

Key fields

- Trade Reference / UUID — links both legs on-chain
- Counterparty LEI — verified against pool membership
- Value Date — settlement window enforced on-chain

Private / Visible

- All trade economics
- Counterparty details
- UUID shared between counterparties
- No on-chain activity yet

HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Propose settlement - Bank A

STEP 3/7

At settlement, Bank A sends the proposed transaction to Bank B for review and acceptance

Bank A sends the proposed transaction to Bank B for review and acceptance.

Fund Account | **Settle Trade** | Return Funds

TRADE REFERENCE: HS-2026-0702-PVP-A001

VALUE DATE: 4 Jul 2026 (T+2)

LEG 1 — SEND

CURRENCY: USDC

NOTIONAL: 100,000,000

LEG 2 — RECEIVE

COUNTERPARTY: Bank B

CURRENCY: XSGD

NOTIONAL: 130,000,000

FX RATE: 1.3000 USD / XSGD

Submit Settlement Instruction

What's happening

Bank A calls the SDK to generate a settlement proposal which is sent to Bank B, including a zero knowledge proof of their balances. Bank B has the option to accept or reject the settlement proposal.

SDK call

```
initiateSwap(counterparty, terms, uuid)
```

Cryptographic primitives

- Transaction signed via bank HSM
- Groth16 proof per bank

Private / Visible

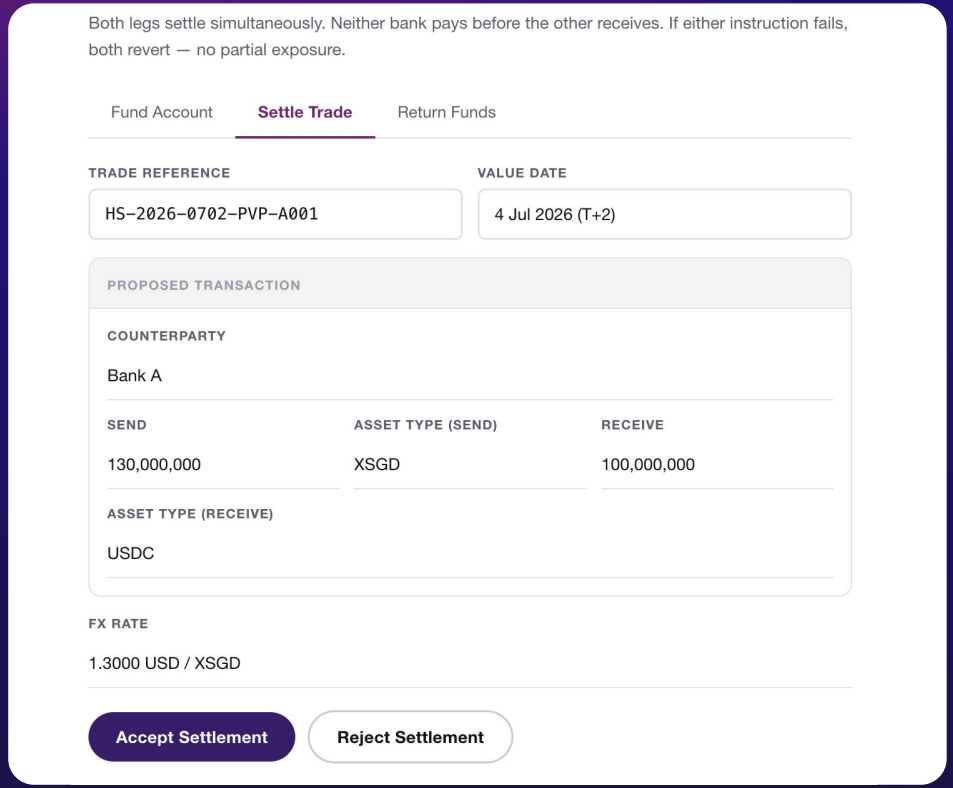
- Both banks' amounts
- Both account balances
- Institution identities
- Settlement occurred, randomized tx hash
- Both proofs verified on-chain

HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Complete settlement - Bank B

STEP 4/7

Both legs settle simultaneously. Neither bank pays before the other receives.
If either instruction fails, both revert — no partial exposure.



What's happening

Bank B verifies settlement terms and accepts the proposal, creating their own proof. Both halves of the transaction are submitted to the network. The contract verifies both proofs simultaneously and settles atomically. If either proof fails, no balances are updated. Balances are re-shuffled to anonymize participants

SDK call

```
matchAndExecute(uuid)
```

Cryptographic primitives

- Transaction signed via bank HSM
- Groth16 proof per bank
- Participant anonymity preserved via balance shuffling

Private / Visible

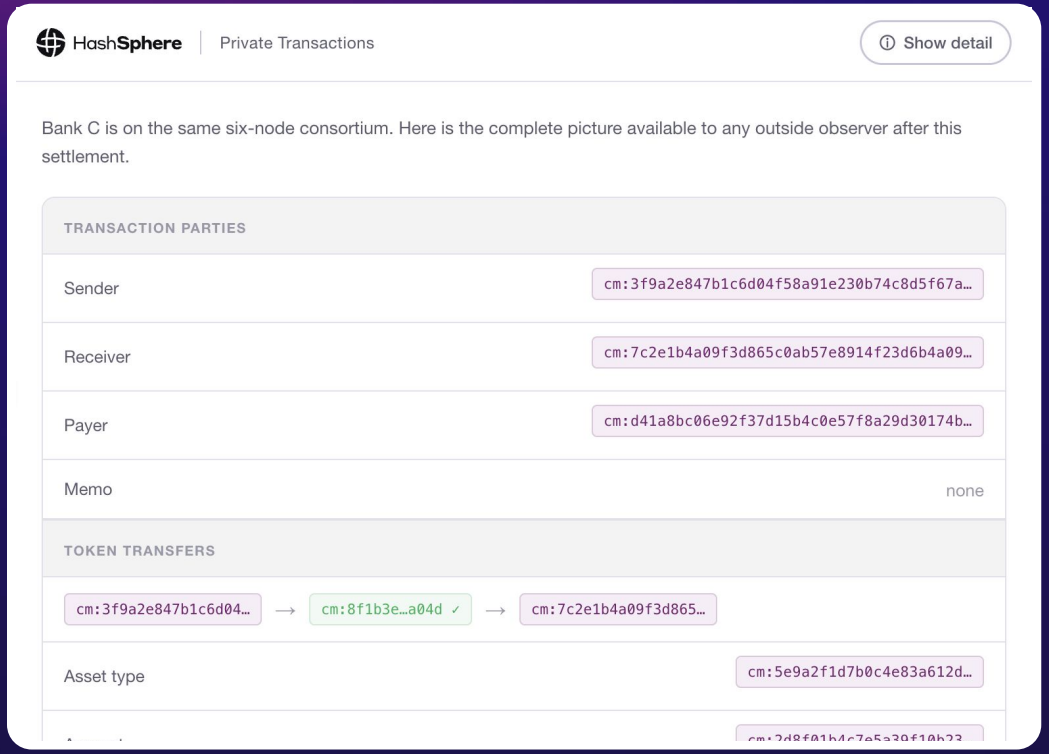
- Both banks' amounts
- Both account balances
- Institution identities
- Settlement occurred, randomized tx hash
- Both proofs verified on-chain

HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

What Bank C sees on the network

STEP 5/7

Bank C is on the same six-node consortium. Here is the complete picture available to any outside observer after this settlement.



BANK C CAN CONFIRM

- A valid settlement occurred
- Both proofs verified — rules were followed
- No double-spend — each leg consumed once
- Settlement is final (hashgraph consensus)

BANK C CANNOT SEE

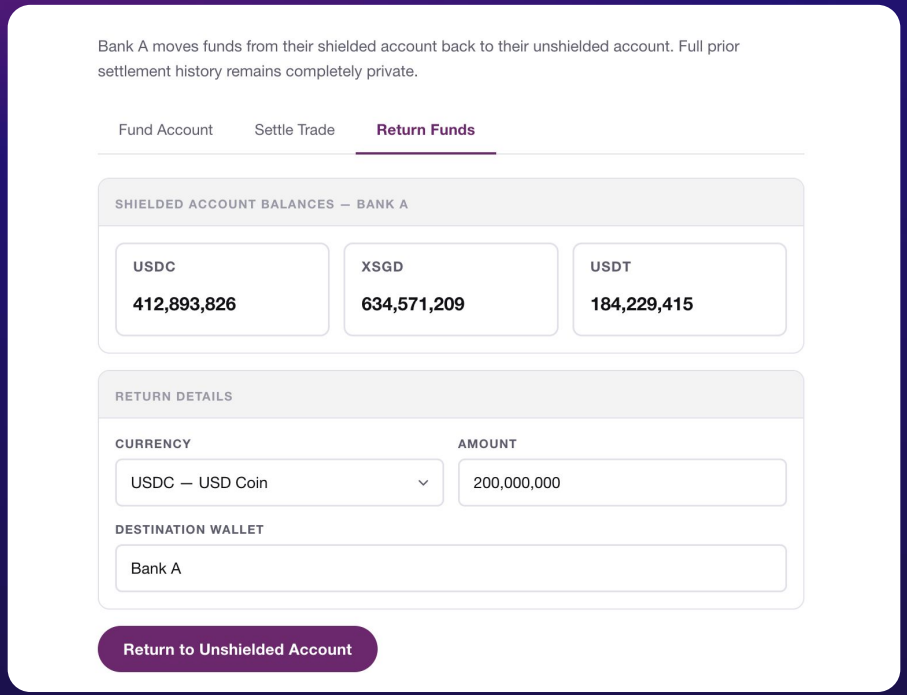
- Which institutions settled
- Amount or currency of either leg
- Either institution's remaining balance
- Any other trades in this pool

HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Unshield funds

STEP 6/7

Bank A moves USDC from their shielded account back to their main account.
Full prior settlement history remains completely private.



What's happening

Bank B generates a cryptographic proof of settlement account ownership and balance. The contract verifies the proof, removes the funds from the shielded account, and releases the equivalent USDC to Bank B's standard account.

SDK call

```
withdraw(slotID, proof)
```

Cryptographic primitives

- Groth16 proof of account ownership and balance
- Unlock public ERC-20 balance

Private / Visible

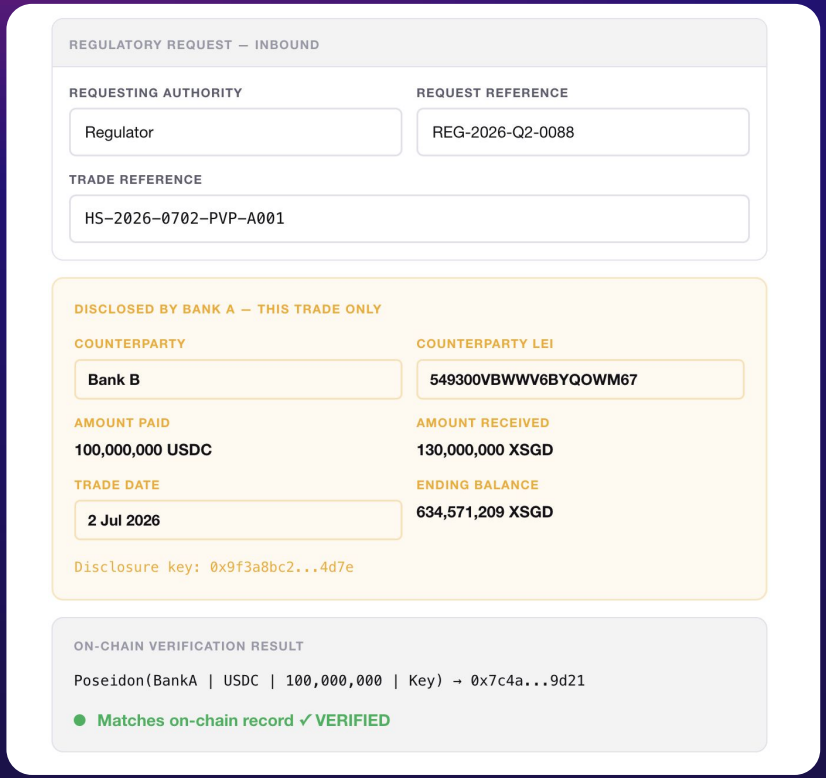
- Full prior settlement history
- Bank A's standard USDC balance increases

HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Regulatory disclosure — on demand

STEP 7/7

After settlement, Regulator requests verification of this specific trade.
Bank A discloses exactly what was asked. Nothing else is revealed.



What's happening

Bank A shares this trade's details plus the disclosure key. The auditor recomputes the hash and confirms it matches the on-chain record. No other Bank A trade is affected or revealed.

SDK call

```
verifyDisclosure(transactionDetails, auditorkey)
```

Cryptographic primitives

- Bank controls disclosure — not the network
- Auditor verifies — cannot browse
- Selective disclosure is granular — only reveals one trade

Private / Visible

- All other Bank A trades
- Bank B's full transaction history
- This one trade (disclosed by Bank A)

HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

The case in four points

HashSphere Privacy removes the last barrier to critical settlement on shared infrastructure.

Cryptographic, not trust-based

Privacy is enforced by mathematics. No network operator, validator, or other member can see what they shouldn't — not by policy, but by proof.

One chain, one audit trail

All encrypted data lives on-chain. No off-chain data stores, no Layer 2, no dual audit trail. One system, one source of truth.

Banks control disclosure

Regulators get access to specific trades when a bank chooses to share. One disclosure key reveals one trade. Everything else stays private.

Built for bilateral settlement

Atomic finality on both legs. In PvP and DvP, both sides settle or neither does — eliminating settlement risk by design.

Hashgraph.com | © 2026 Hashgraph Foundry Inc. All rights reserved.

Summary of atomic swap: Zero knowledge, zero exposure

Six stages from shield to settlement. No party — including the network — sees counterparty, amount, or resulting balance.

01
Shield
Lock public ERC-20
Mint shielded token

02
Agree terms
UUID assigned off-chain
OTC negotiation

03
Generate proofs
Each party proves: balance · recipient · UUID

04
Submit & match
Contract matches UUIDs
Validates both ZKPs

05
Atomic settlement
Encrypted balances updated simultaneously

06
Unshield (opt.)
Burns private token
Releases public ERC-20

↑ If proofs fail or UUIDs don't match → Transaction reverts. No balance change.

Cryptographic privacy
zkSNARKs verify validity without revealing sender, recipient, or amount

Atomic finality
Both legs settle together or neither does — enforced by the smart contract

Selective disclosure
Regulator / custodian / CSD hold scoped read keys — cryptographic, not policy

Fair ordering
Hashgraph consensus ensures no front-running, no manipulation of sequence

WHO SEES WHAT	IDENTITY	AMOUNTS	BALANCES	ASSET TYPE
Counterparties	● Visible	● Visible	● Visible	● Visible
Custodians / CSD	● Visible	● Visible	● Visible	● Visible
Regulator (read key)	● Visible	● Visible	● Visible	● Visible
Institution C	✕ Hidden	✕ Hidden	✕ Hidden	✕ Hidden

Hashgraph

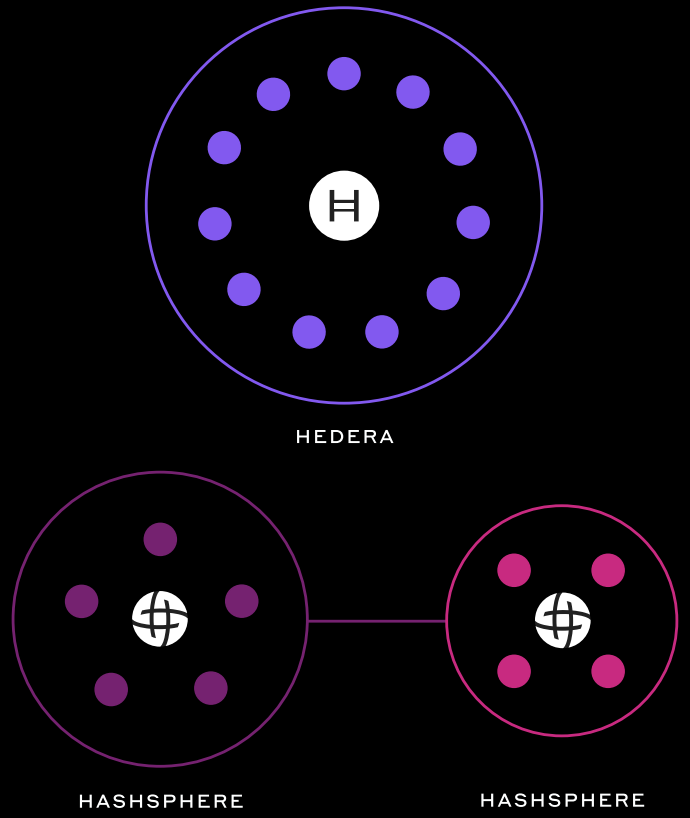
True network sovereignty

Strengthen privacy while eliminating single points of failure.

02

You control the governance model

- The validator set is chosen by the stand-alone private network, which is not a subnet on a public network.
- There is no central routing party, the network chooses its own governance model.
- The network is built on a standard EVM infrastructure, compatible with the broader ecosystem of tooling, custody and development talent



HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Hashgraph

Secure cross-network interoperability

Liquidity is fragmented. Bridges are broken. There's a better way.

03

Liquidity is trapped in walled gardens

Institutions cannot efficiently and compliantly move value across ledgers at scale

Assets remain trapped.

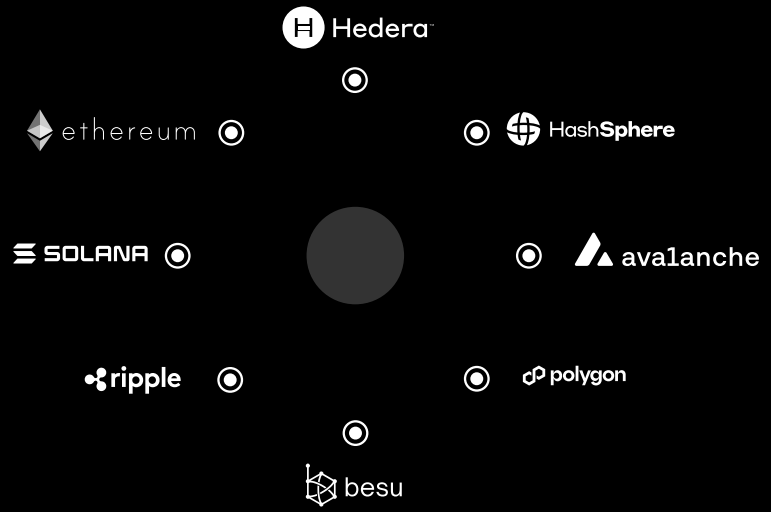
Capital stays locked within native networks—unable to move freely across ecosystems.

Bridges are workarounds.

Cross-chain bridges introduce centralization and have serious security vulnerabilities.

Walled gardens, not a system.

The digital asset economy is fragmented networks, not a unified interoperable financial system.



HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Current workarounds introduce unacceptable risks

$3B+ LOST TO BRIDGE EXPLOITS

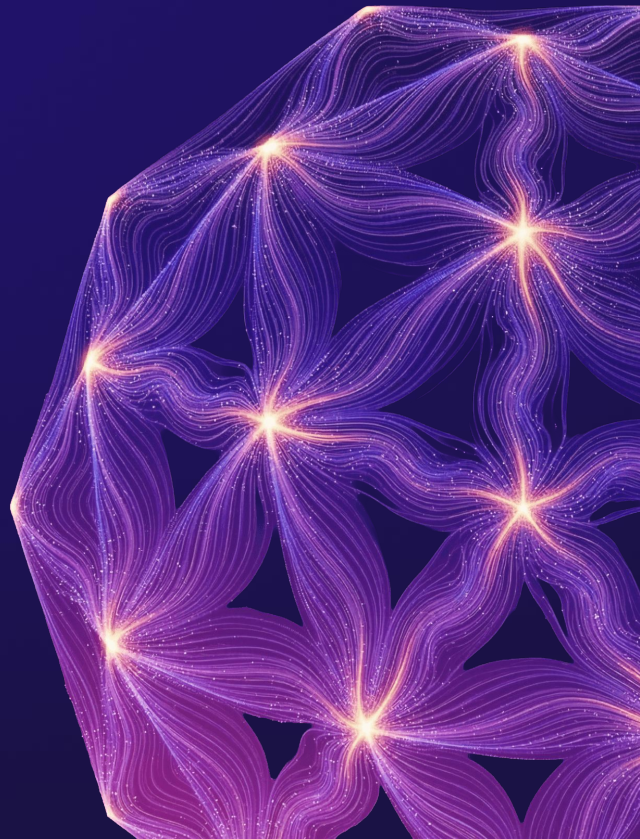
Cross-chain bridges and relayers

- Locks assets into bridge contracts–creating honeypots
- Exploits have created billions in losses (Ronin, Wormhole, Nomad, KelpDAO)
- Not truly cryptographically trusted bridges are a more vulnerable target

VENDOR LOCK-IN

Private, permissioned networks

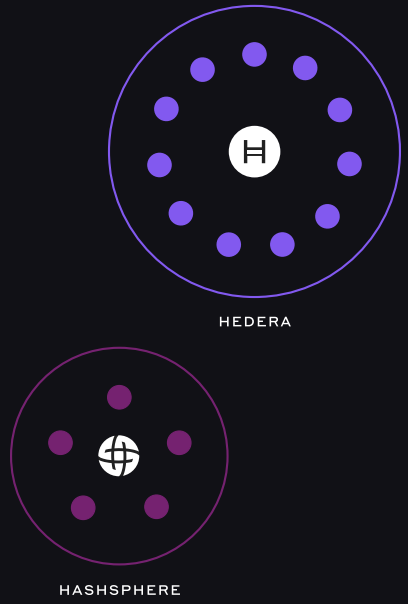
- Interoperability limited to networks within the same walled garden
- Central point of failure–single operator controls access
- Require costly maintenance of liquidity reserves on both sides



CLPR also delivers interoperability for HashSphere networks

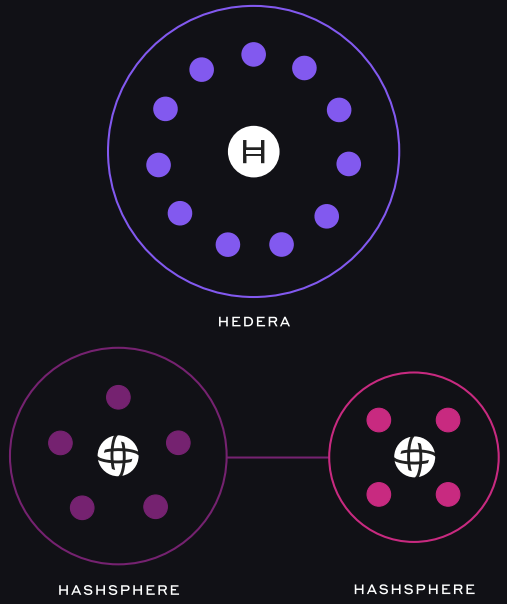
Private ‘Sphere’

A closed network between several individuals or organizations.



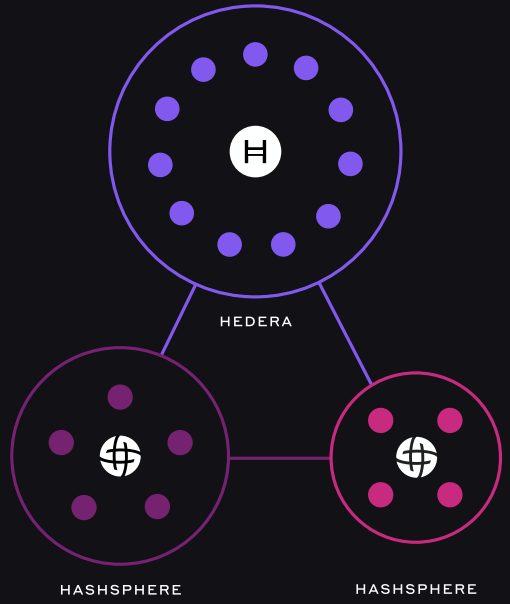
Private ‘Sphere’ Interoperability

A connection securely sharing data, workflows, and value with other private networks.



Hybrid Private & Public Networks

Connections where assets flow freely and securely between private networks and public networks such as Hedera



 | © 2026 Hashgraph Foundry Inc. All rights reserved.

Hashgraph

Applications reviewed on a rolling basis

25-30 seats

WHAT HAPPENS NEXT?

1. **Apply:** Submit your interest via the link or QR code
2. **Review:** Hashgraph reviews applications through end of June
3. **Join the cohort:** Confirmed Navigators onboarded in July

REGISTER YOUR INTEREST

HASHGRAPH.COM/
CLPR-EARLY-ADOPTER-PROGRAM/

HASHGRAPH.COM | © 2026 Hashgraph Foundry Inc. All rights reserved.

Hashgraph HashSphere

Thank you

DISCOVER MORE AT
HASHGRAPH.COM/HASHSPHERE

KASH BALHOTRA
kash.b@hashgraph.com