TACEO

Private Finance

Through the Lens of MPC and CoSNARKs

Aisling Connolly, CSO, TACEO

LFDT Privacy Workshop, London, 1 July 2026

Hiding transaction amounts is easy.

If we want to seriously discuss bringing finance into the real (decentralised) world, we need a lot more than private payments.

One person

What can you infer when amounts are private?

Current account, one month

Tue 04	City Pharmacy	
Wed 05	Fertility clinic	
Fri 07	Family law solicitors	
Sat 08	Online betting	
Tue 11	City Pharmacy	
Sat 15	Online betting	
Tue 18	City Pharmacy	
Thu 20	Family law solicitors	

Every amount is hidden.
The merchants are the edges. The pattern is the graph.

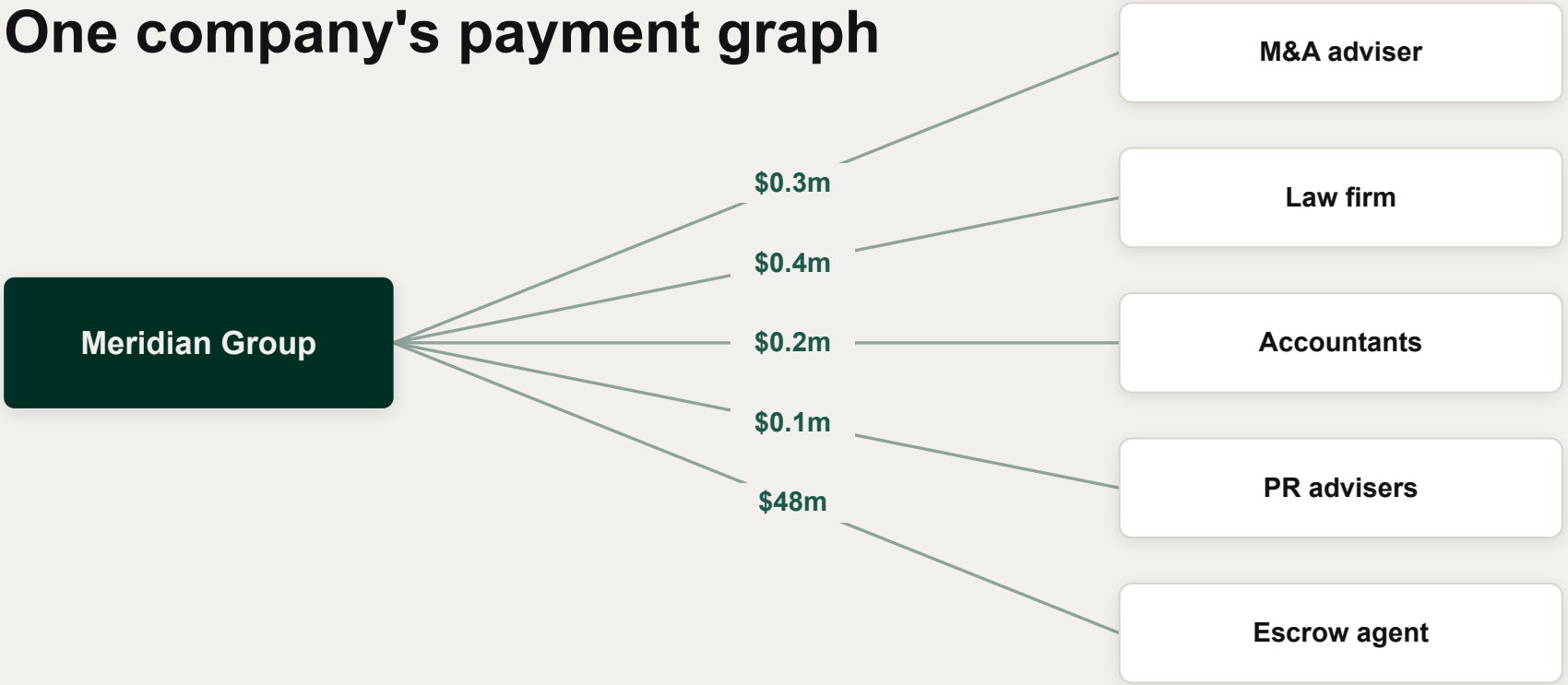
Chronically ill

Going through a divorce

Gambling

Let’s look at a company

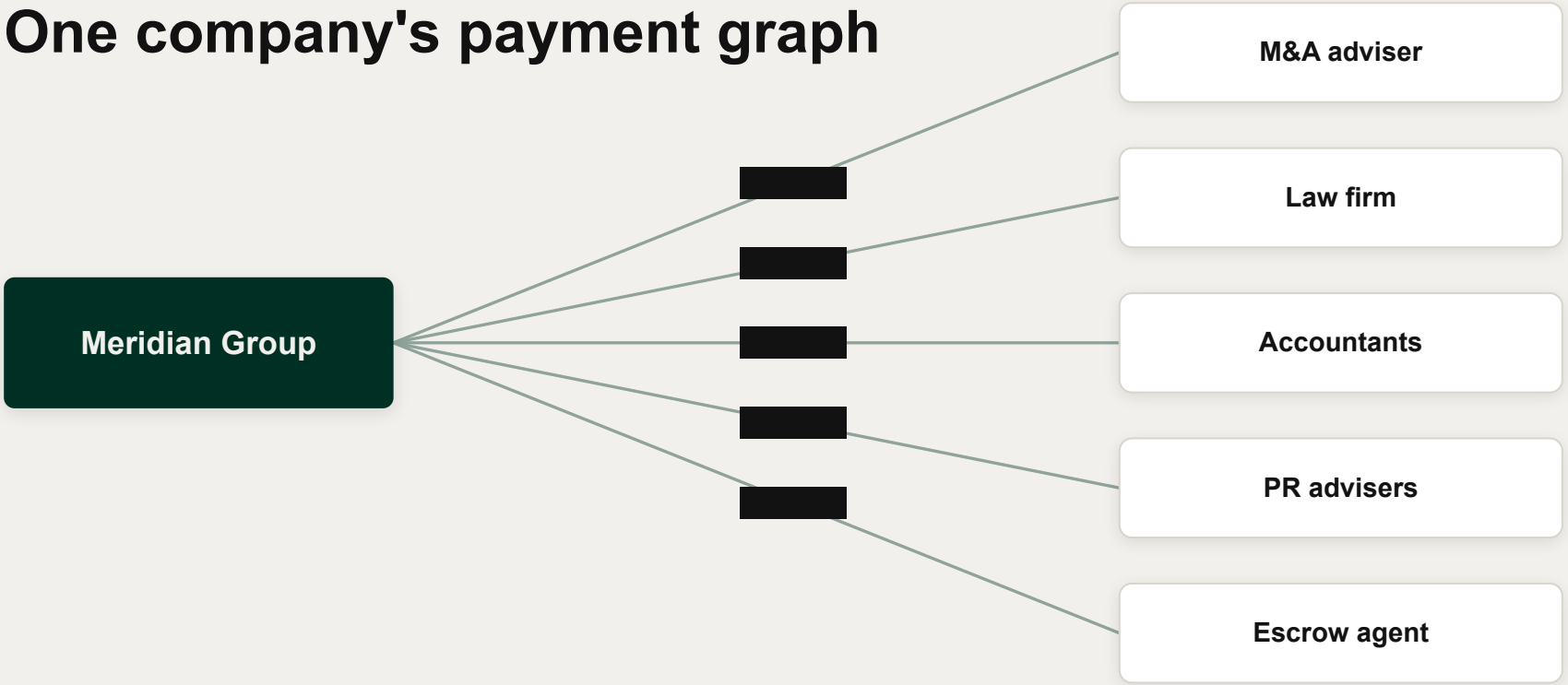
One company's payment graph



Outbound payments over a quarter with names and amounts visible.

Amounts gone

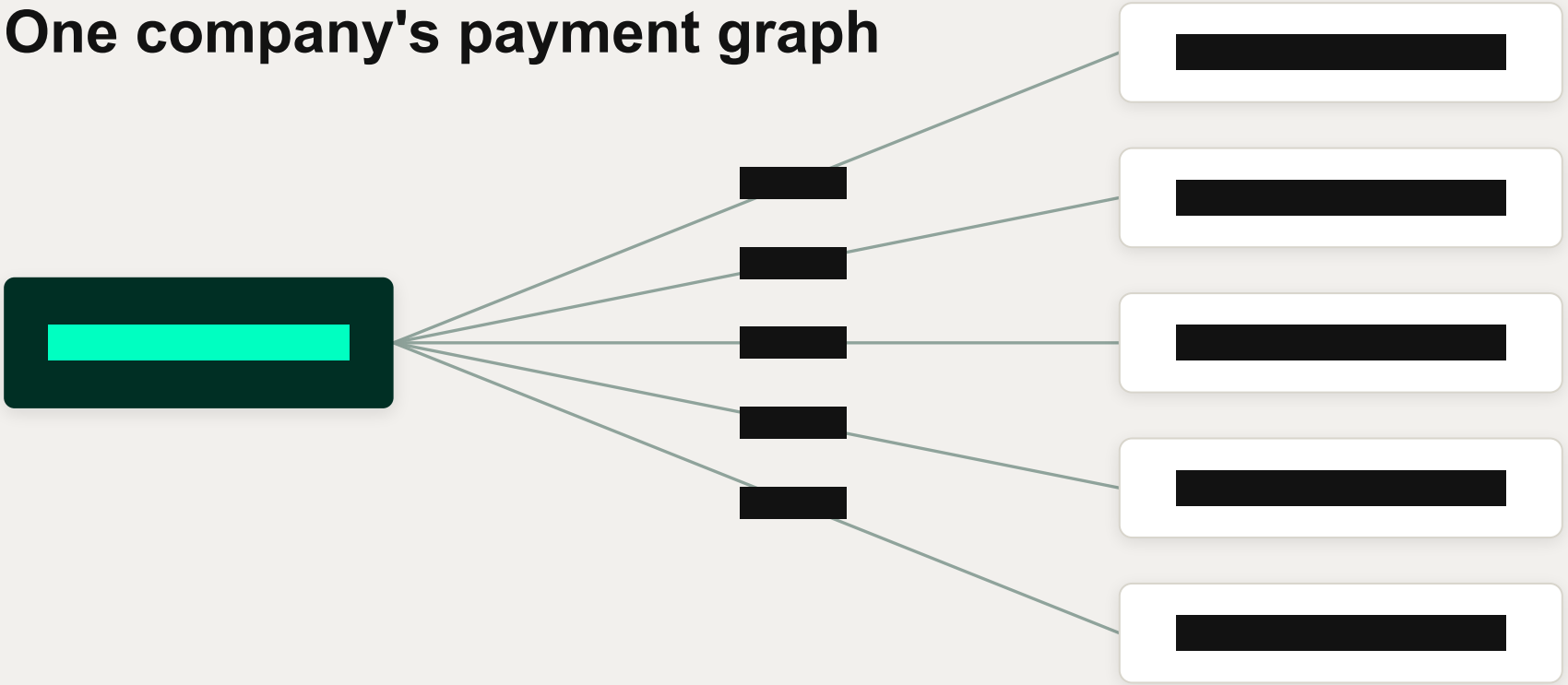
One company's payment graph



If we take away the amounts the context is still clear.

Names gone

One company's payment graph

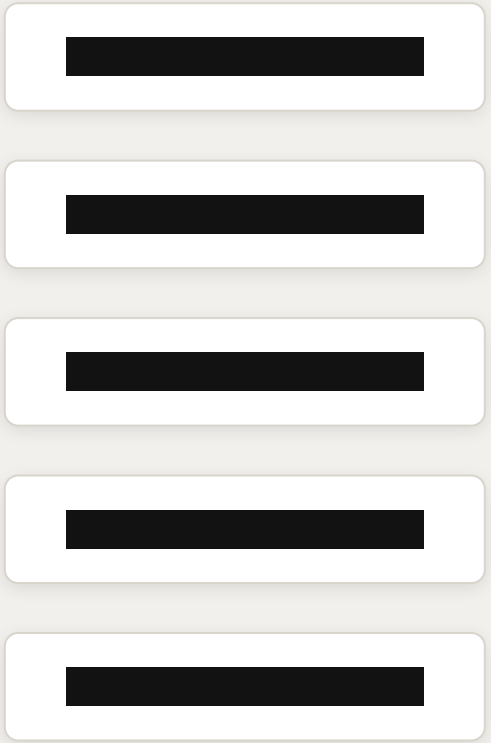


Even without names, the structure still tells a story.

Edges cut

One company's payment graph

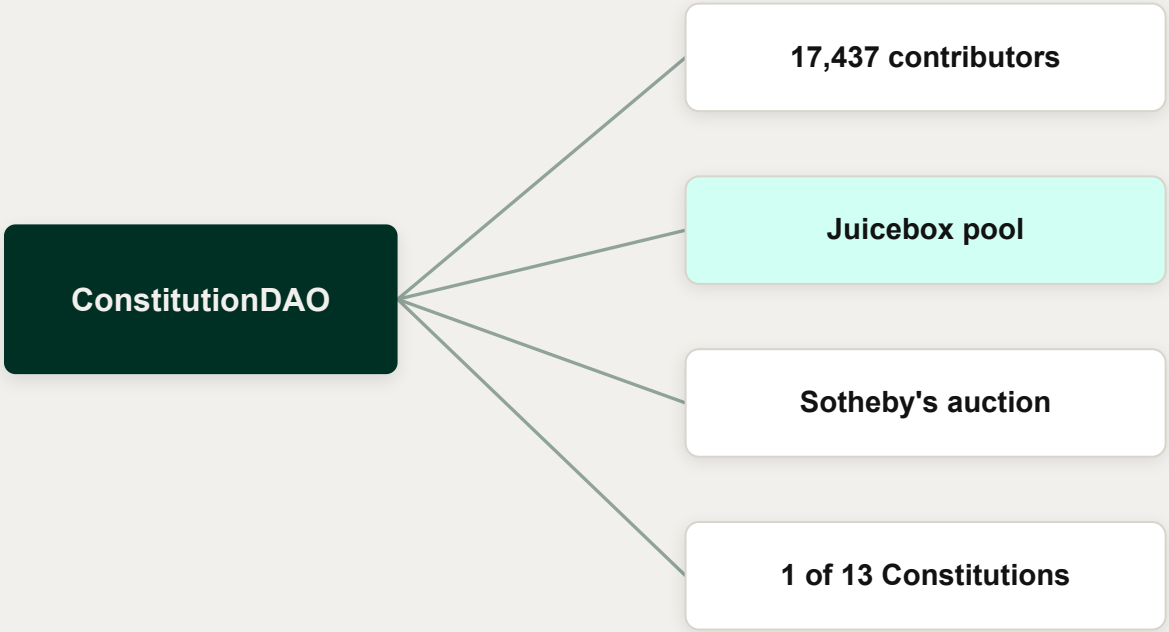




Hiding the edges makes us oblivious to the moves of a particular entity

Long live the $People

Remember the Constitution Dao?



Similar structure to the previous deal, and all elements were public.

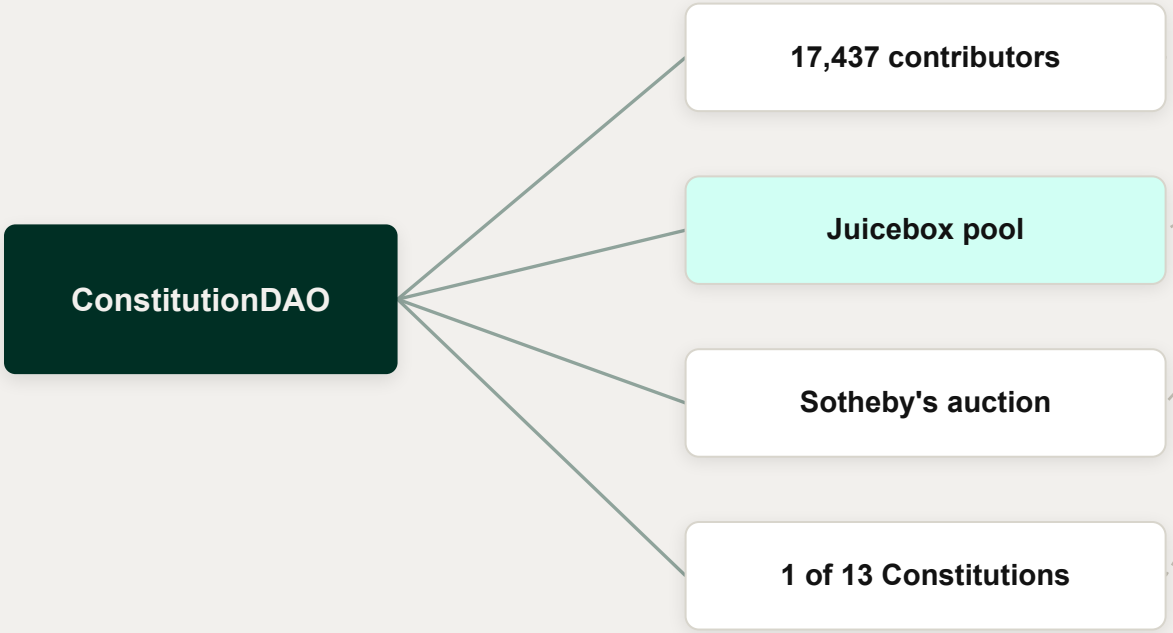
Long live the $People

Remember the Constitution Dao?

Ken Griffin (Citadel)

watched it all, then bid just over

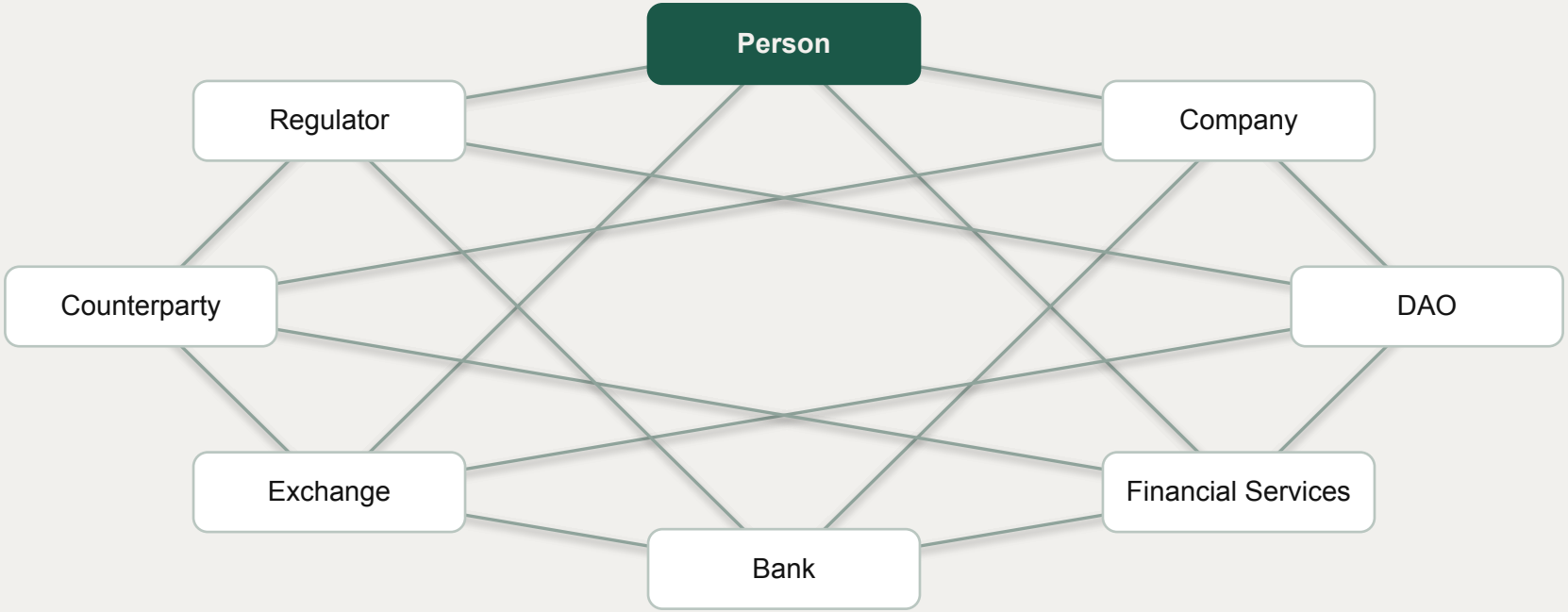




Even if the juicebox amount was hidden, a lot can be inferred from who's interacting

Zooming out

Not just a one way flow



Entities interact with each other and operate on each other's holdings.

Four steps to hiding the graph

1. **Compute on hidden data:** MPC
2. **Prove it ran correctly:** collaborative SNARKs
3. **Make it a shared ledger:** private shared state
4. **Hide which record moves:** oblivious maps

Each step hides more by composing primitives along the way.

The easy half

Hiding the amount is essentially solved

Hiding balances even has a name, confidential transfers, and there are several approaches.

FHE | **ZK proofs** | **MPC** | **TEEs**

Hides
the balance and the transfer amount.

Leaves exposed
who pays whom, and who is who. The graph.

Maturity and efficiency of the primitives vary, but the basics are there.

Who is wealthier, without saying how much?

Yao's millionaires' problem, 1982: two people want to learn who is wealthier, while neither reveals their holdings.

Solving it launched multi-party computation. In finance the same question becomes: is there enough balance, is this trade valid, who owes whom. Computed on numbers nobody reveals.

Part one: mpc

Compute on private data

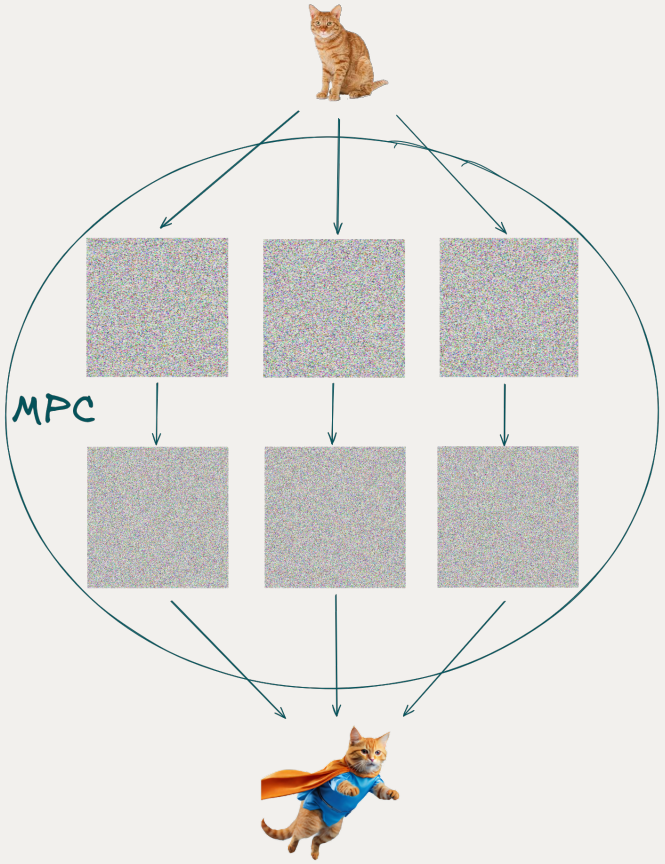
Several mutually distrusting parties compute on their combined data without any of them seeing it.

Split. Every input is split into secret shares, one per party. A single share is pure noise; on its own it reveals nothing.

Compute. The parties evaluate the function directly on the secret shares, exchanging messages as they go. Inputs and intermediate values never appear in the clear.

Reveal. Recombining the shares yields the result, and nothing else. Privacy remains assuming parties don't collude.

Arbitrary computation on hidden data.



Part two: integrity

MPC gets us privacy. It does not make it trustworthy.

The network computes off-chain, on data no one can see. So how does the chain, a regulator, or your counterparty know the right function ran on the right inputs, and that nobody deviated?

MPC on its own

Private, yes. But the answer is a black box. You have to trust the operators.

MPC plus a proof

Private and checkable. Anyone can verify it was computed correctly.

Collaborative snarks

Build the proof inside the MPC

A collaborative SNARK is a proof the network produces about its own work: this function ran correctly on these committed inputs, with nothing revealed.

Programmability

from MPC: run any function on shared data.

Provability

from the SNARK: a succinct, on-chain-checkable receipt.

Privacy

from both: inputs and steps never revealed.

Verifiability of private computation

Putting it together

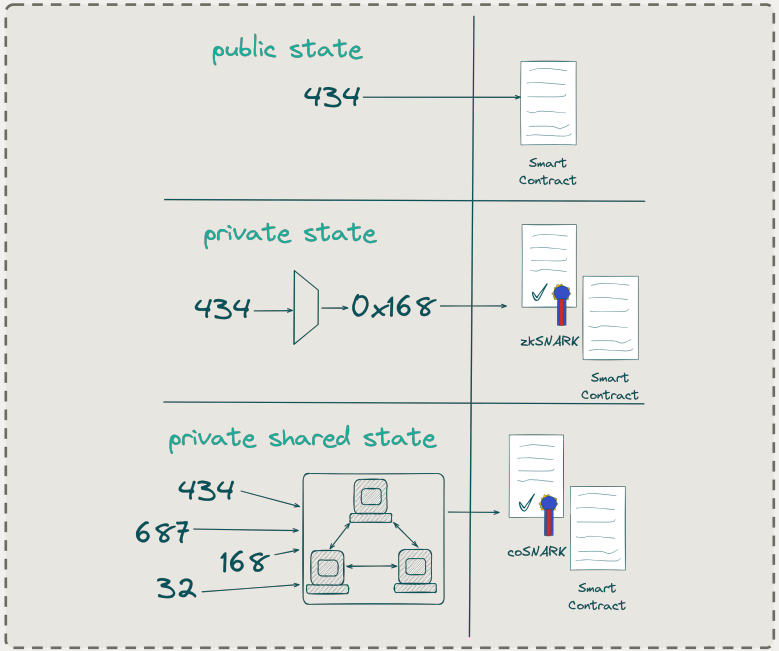
Private shared state

Public state: everyone sees it.

Private state: one owner, kept secret.

Finance is neither. It is many owners who must compute together without opening their books to each other.

Private shared state: secret-share the state, compute off-chain, and post a commitment to the result plus a proof onchain.



The round trip

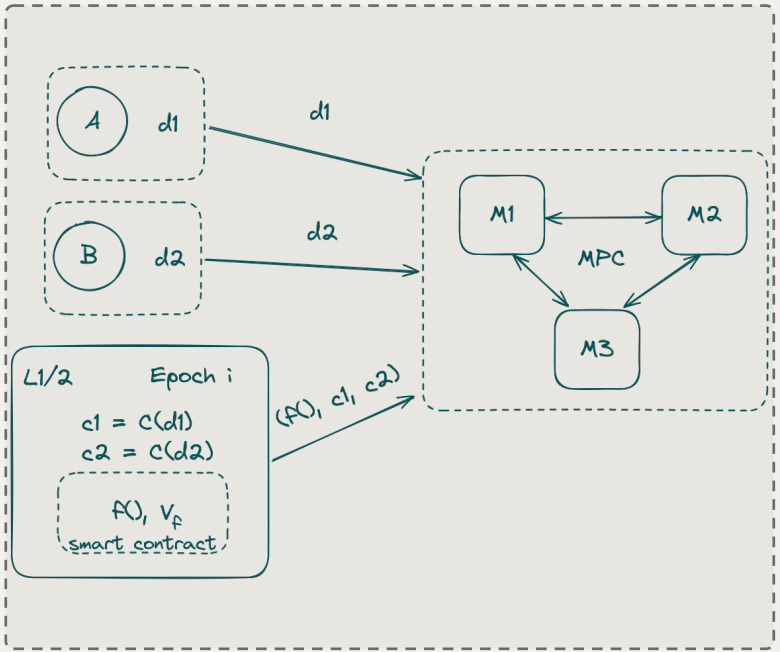
How it runs, end to end

Secret-share. Users split balances into random shares; the network sees only noise.

Compute. The MPC network runs the function on the shares, off-chain. No party sees inputs or intermediate values.

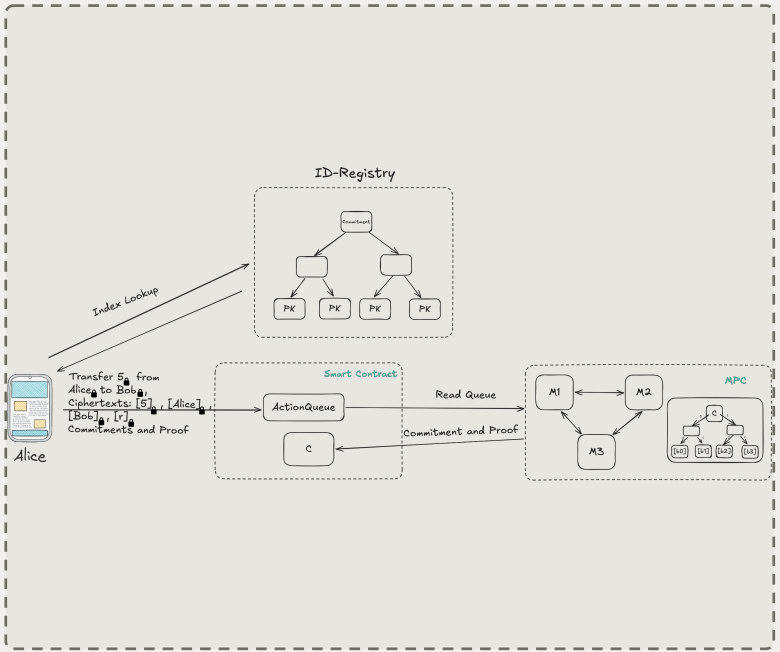
Prove. It builds a coSNARK: this exact function ran correctly on the committed inputs.

Verify. The smart contract checks the proof and advances state. No data ever touches the chain.



A private wrapper for any token

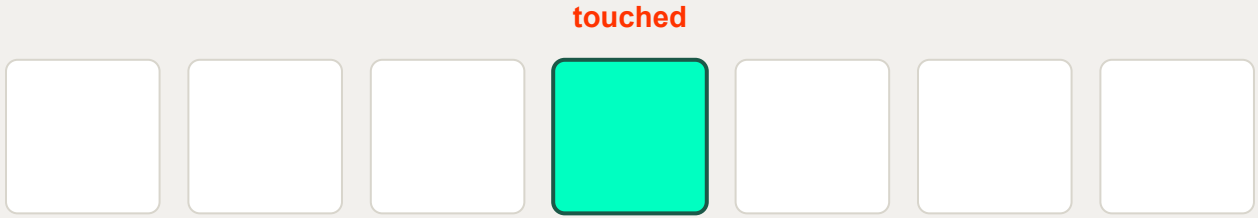
- Wrap any token, ETH or USDC, on any EVM chain. Hide the balance, the sender, the receiver and the amount.
- Balances live as private shared state: an ERC-20 mapping(address → balance), kept secret on the network.
- The contract holds only commitments, plus the total locked underneath.
- A transfer updates balances off-chain; a coSNARK proves it was valid: sender authorised, enough balance, amount positive.
- Ref: eprint.iacr.org/2026/850



Gotcha

That hides the values. It still leaks the shape.

Amounts are gone, but anyone who sees which entries a transfer touched can redraw who paid whom. **We're back at slide one: the access pattern is the graph.**

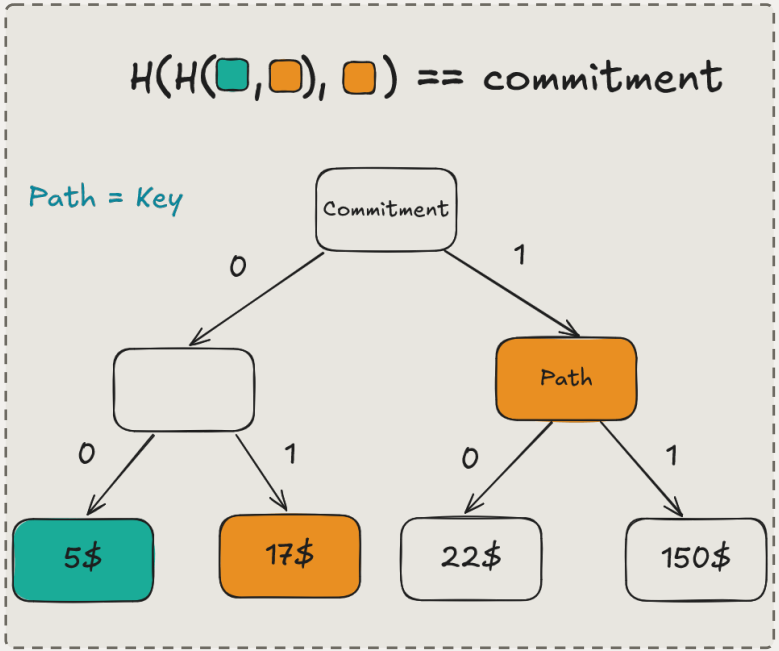


To hide the graph, you have to hide which record you read and write.

Our approach

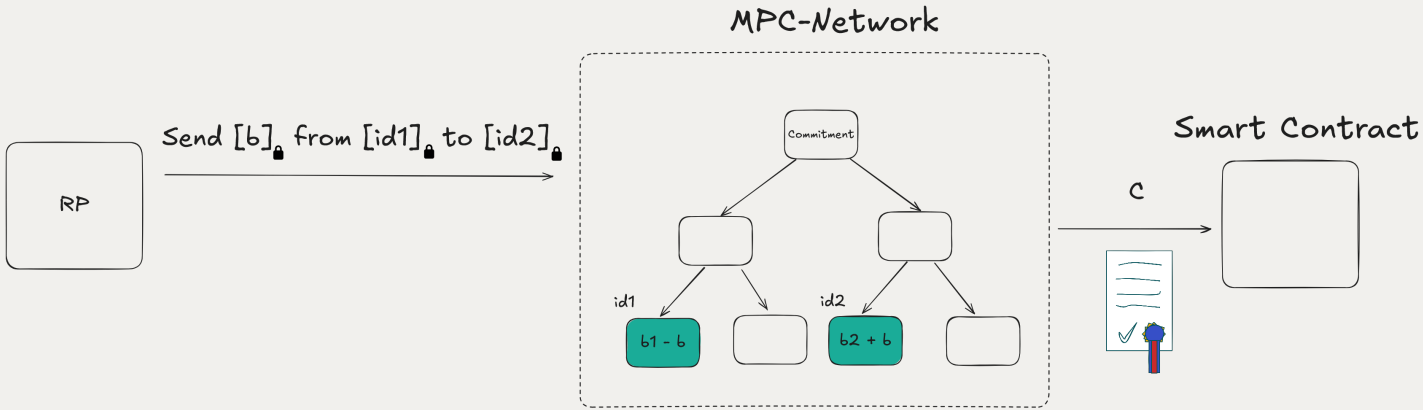
An oblivious map: every access looks the same

- Balances sit in an oblivious hashmap on MPC network: mapping(address → balance).
- Hashmap implemented as a Merkle tree. The root is the onchain commitment, the leaves are balances.
- Reads and writes walk the tree without revealing which leaf moved.
- Each layer of tree is an oblivious RAM.
- **When do you need it? On every transfer, because the access is the edge.**



Inside a private transfer

- A private transaction is an oblivious map update
- Sender ZK proof for authentication
- Deduct balance from sender and add to receiver
- MPC ZK proof: 4 merkle proofs, sender has enough balance, amount is positive
- Checking balance is a read from the map



TACEO Private Finance solution

Merces

A confidential token system on MPC and collaborative SNARKs.

Balances hidden by default

Amount privacy comes for almost free.

Graph privacy when you pay in efficiency

Sender, receiver and access pattern, possible to hide, but slower.

Running on public testnets

Millions of demo transactions, mainnet eta Q3.

MPC network in production

Securing the data of over 18 million people.

Building a general purpose network

MPC, coSNARKs, Omap underpin many features

Compliance

Gate and audit. Enforce before, reveal under proper authority after, with no standing backdoor.

Agent payments

Confidential x402 / agent-to-agent transfers. Running code today.

Identity

Private uniqueness and matching without needing plaintext data.

Yield

Confidential positions that earn and settle.

Where we stand

There are some tradeoffs. Here are the numbers.

Benchmarked across the four privacy models, May 2026.

Configuration	Hidden on-chain	Verifiable	Throughput
Confidential	Amounts	No	**~6000 TPS**
Confidential + proof	Amounts	Yes	**~300 TPS**
Private + proof	Users + amounts	Yes	**~50 TPS**
Graph-private + proof	Users + amounts + graph	Yes	**~4 TPS**

The private tier hides who and how much onchain; the network still sees the shape. The graph-private tier hides that too, and is where our active research sits.

Where the work is

Where the research is going

Throughput to 10k

The private tier runs about 50 TPS today; the target is 10k, from efficient primitive design proven at scale and from real hardware headroom. The confidential tier already scales horizontally past 10k.

Graph privacy is the frontier

The fully graph-private model sits near 4 TPS. That is the honest number, and it is where the active research is.

Post-quantum, in progress

The MPC core and the oblivious maps are post-quantum today. The proof system and the share-encryption key exchange are the pieces still to land.

We stay precise about what ships, what scales, and what is still open.

What are we best known for

TACEO

From cryptographic foundations, to large-scale deployments, to network orchestration.

Poseidon

Co-designed the hash function at the core of modern ZK systems.

Iris MPC

Optimized protocol securing the data of over 18 million people.

MPC network in Prod

A live MPC network, run by independent operators.

We work primarily across finance, identity and agents.

Thank you

Aisling Connolly, Strategy @ TACEO
taceo.io