Espresso

Rethinking the Public / Private Distinction

How Layer 2 Architecture Across a Shared Consensus Layer Enables Configurable, Interoperable Privacy

Ellie Davidson
Chief Product Officer
Espresso Systems

The world runs on systems.

These systems interoperate everyday.

Blockchains are no different.

sovereign

The world runs on systems.

These systems interoperate everyday.

don’t have to be

Blockchains ~~are no~~ different.

Trust between these systems is ambiguous, everchanging, and **deeply complex**.

But we need to interoperate between these systems as **efficiently as possible**.

Trust-minimized base layers are the most effective way to interoperate between these systems.

Trust-minimized base layers are the most effective way to interoperate between these systems.

- Certificate authorities
- CSDs (DTCC, Euroclear, Clearstream, etc.)
- Google, Apple for identity
- Governments

Trust-minimized base layers collapse many, bilateral trust assumptions into **one multilateral trust assumption**

Blockchains aim to be this trusted base layer, but historically they have done so by forcing systems to give up desirable properties, such as sovereignty, customization, control, privacy, and performance.

don’t have to be

Blockchains ~~are no~~ different. → Decompose

Separate blockchains into these parts and optimize / customize each part according to our needs

Sequencing	Consensus	Execution
Ordering transactions (FCFS, fee auction, MEV-based	Canonicalizing data (determining what is irreversible)	Computing the new state of the system

Separate blockchains into these parts and optimize / customize each part according to our needs

Sequencing	Consensus	Execution
Fine–whatever the consensus protocol orders	Canonicalizing data (determining what is irreversible)	Computing the new state of the system

Separate blockchains into these parts and optimize / customize each part according to our needs

Sequencing	Consensus	Execution
Ordering transactions (FCFS, fee auction, MEV-based	Different executions are possible–double spends	Computing the new state of the system

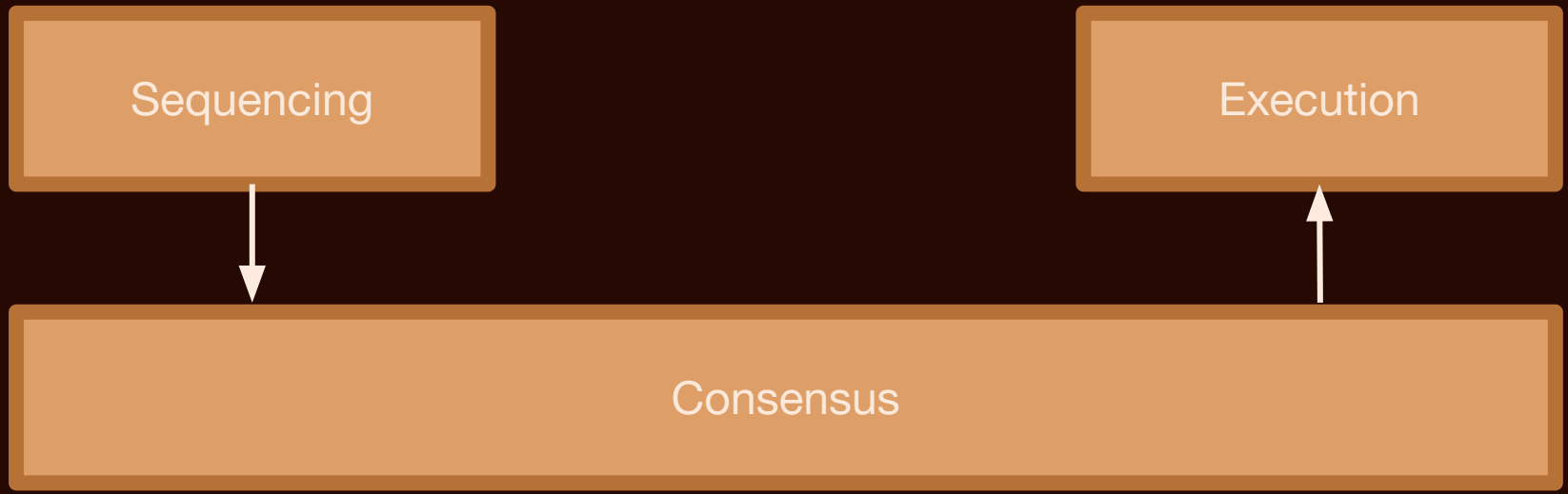
Separate blockchains into these parts and optimize / customize each part according to our needs

Sequencing	Consensus	Execution
Ordering transactions (FCFS, fee auction, MEV-based	Canonicalizing data (determining what is irreversible)	Don’t know the new state of the system, can’t act on it in the real world

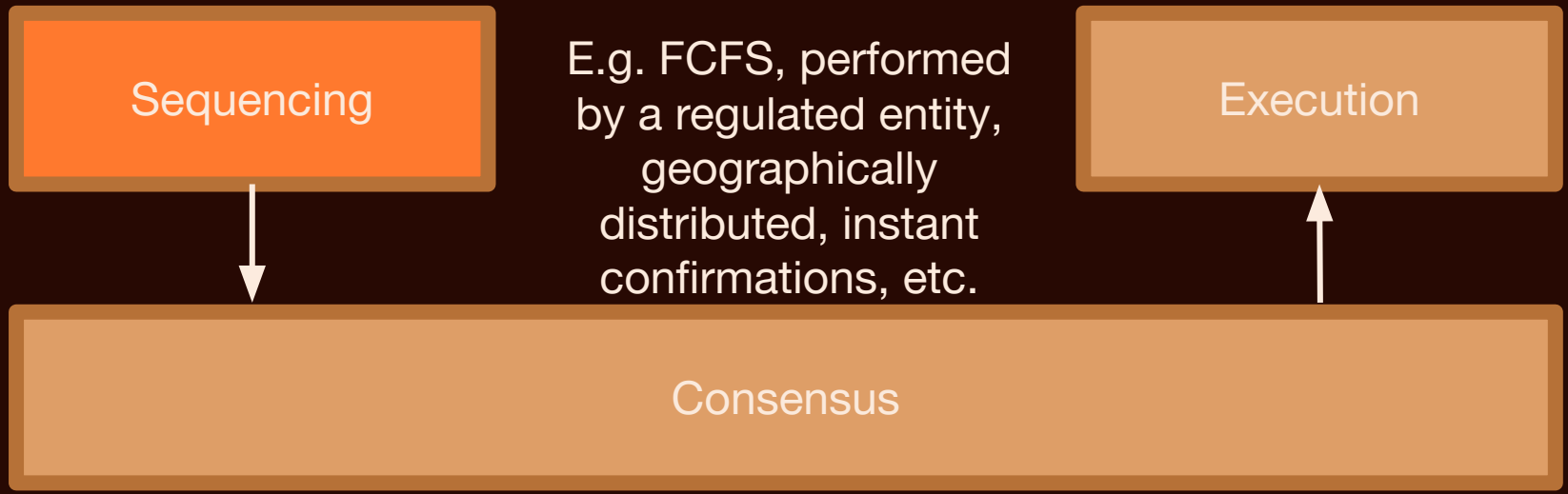
May want different privacy / permissioning guarantees for each component

Sequencing	Consensus	Execution
Ordering transactions (FCFS, fee auction, MEV-based	Canonicalizing data (determining what is irreversible)	Computing the new state of the system

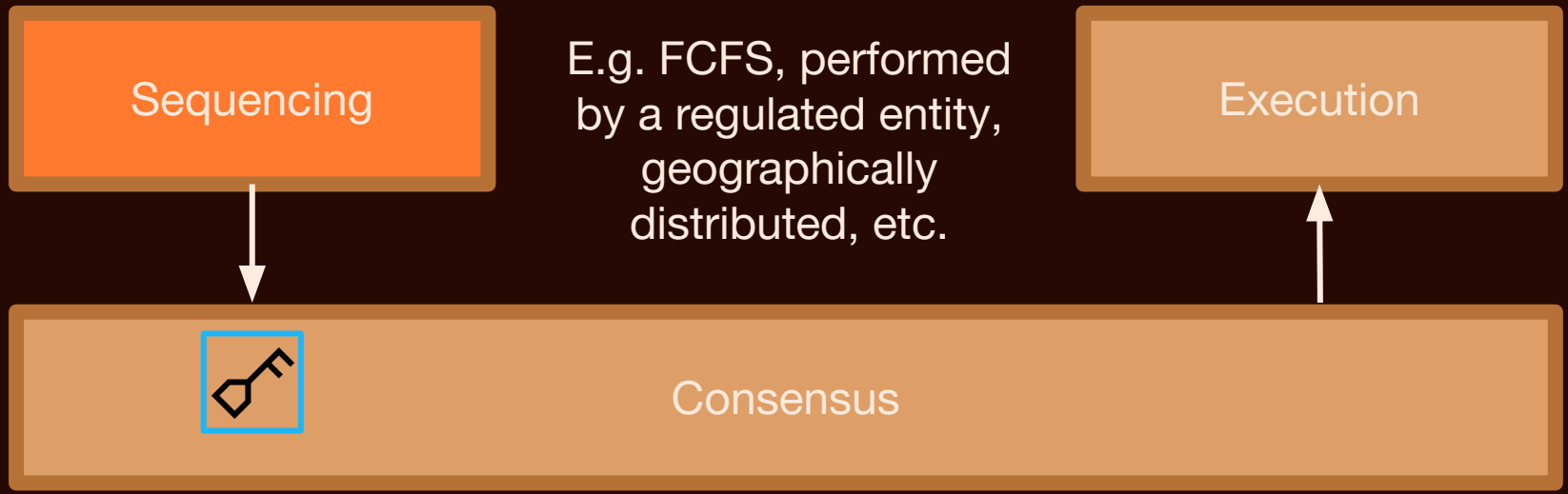
Separated / modular architecture via Layer 2s is more flexible.



Separated / modular architecture via Layer 2s is more flexible.



Separated / modular architecture via Layer 2s is more flexible.



Separated / modular architecture via Layer 2s is more flexible.



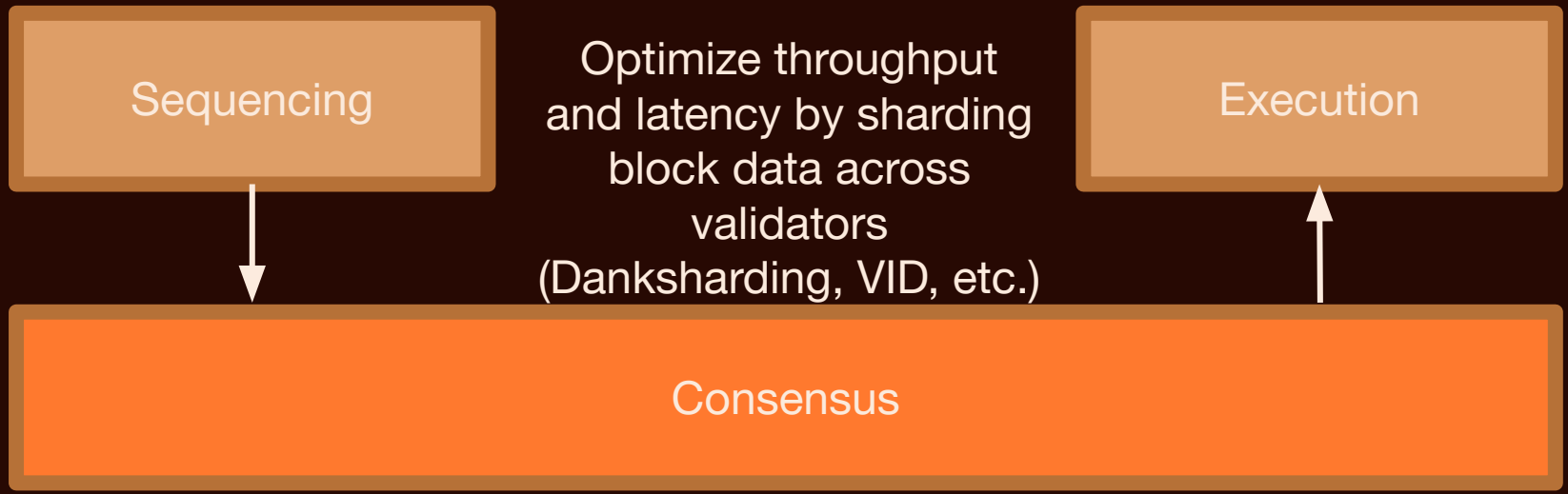
Separated / modular architecture via Layer 2s is more flexible.



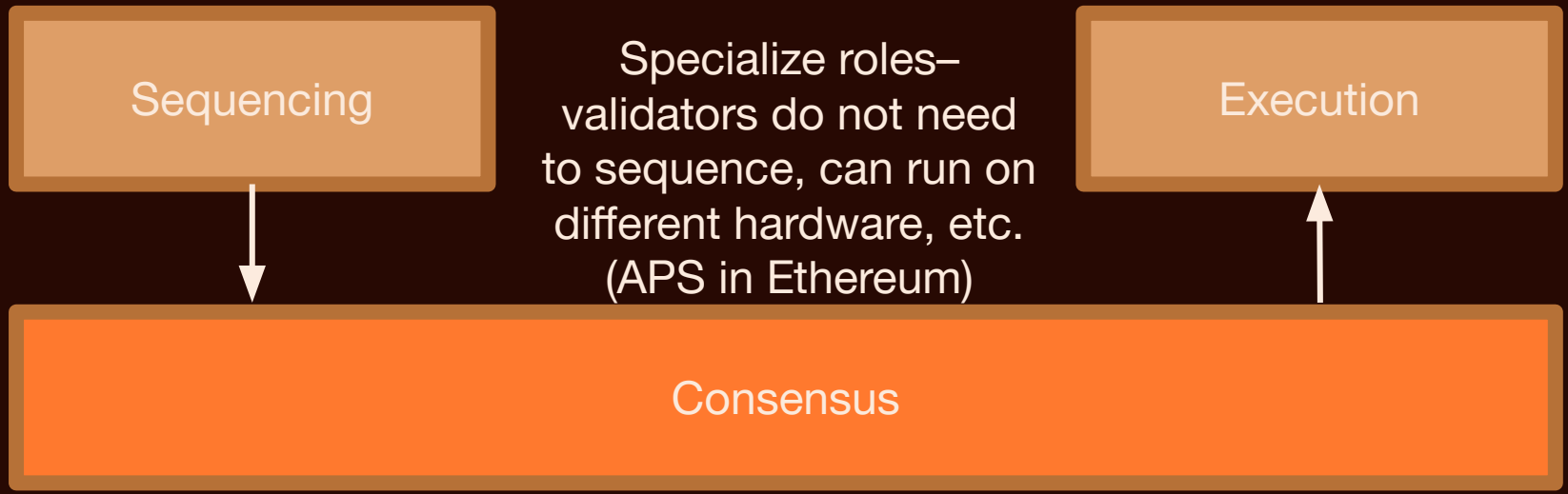
Separated / modular architecture via Layer 2s is more flexible.



Separated / modular architecture via Layer 2s is more flexible.



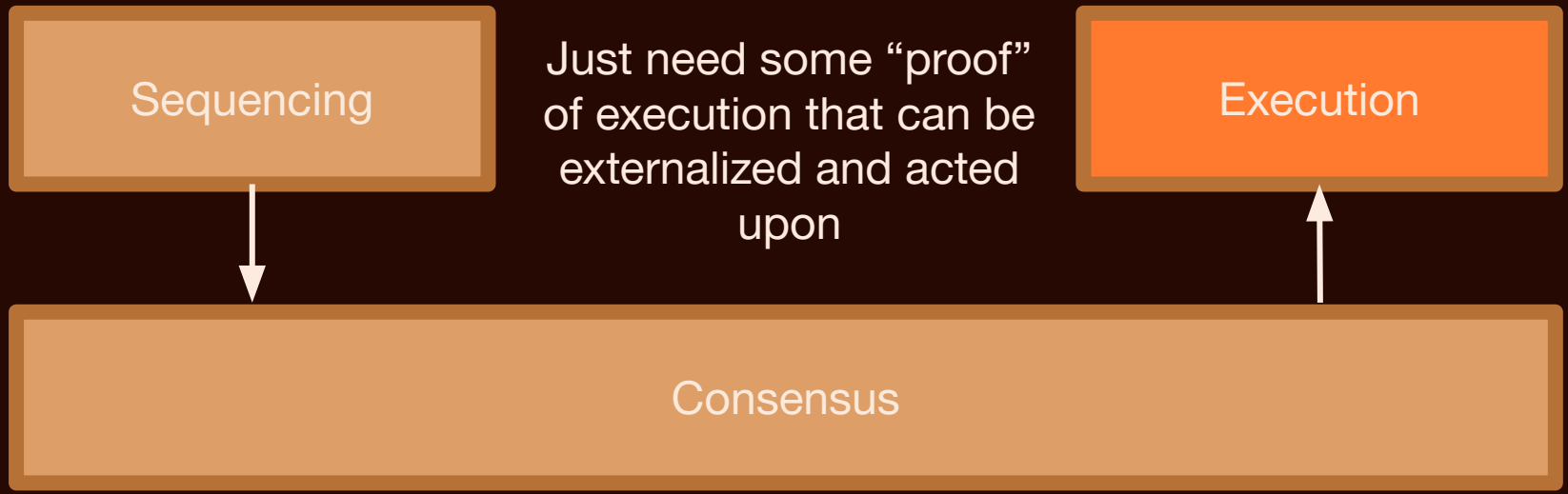
Separated / modular architecture via Layer 2s is more flexible.



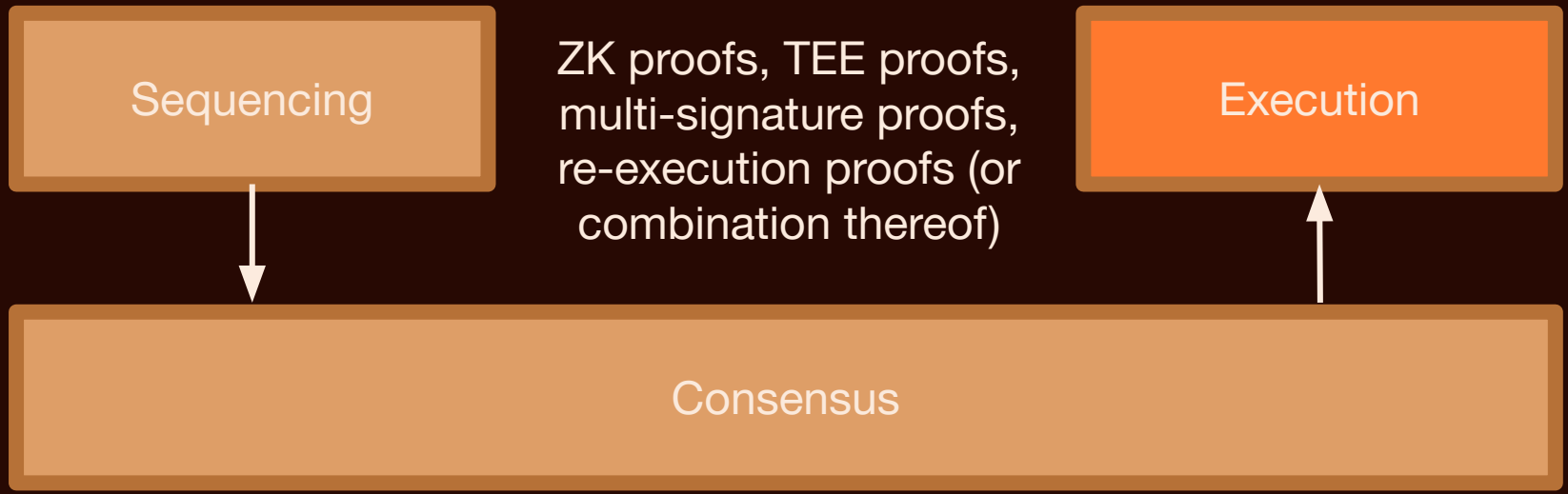
Separated / modular architecture via Layer 2s is more flexible.



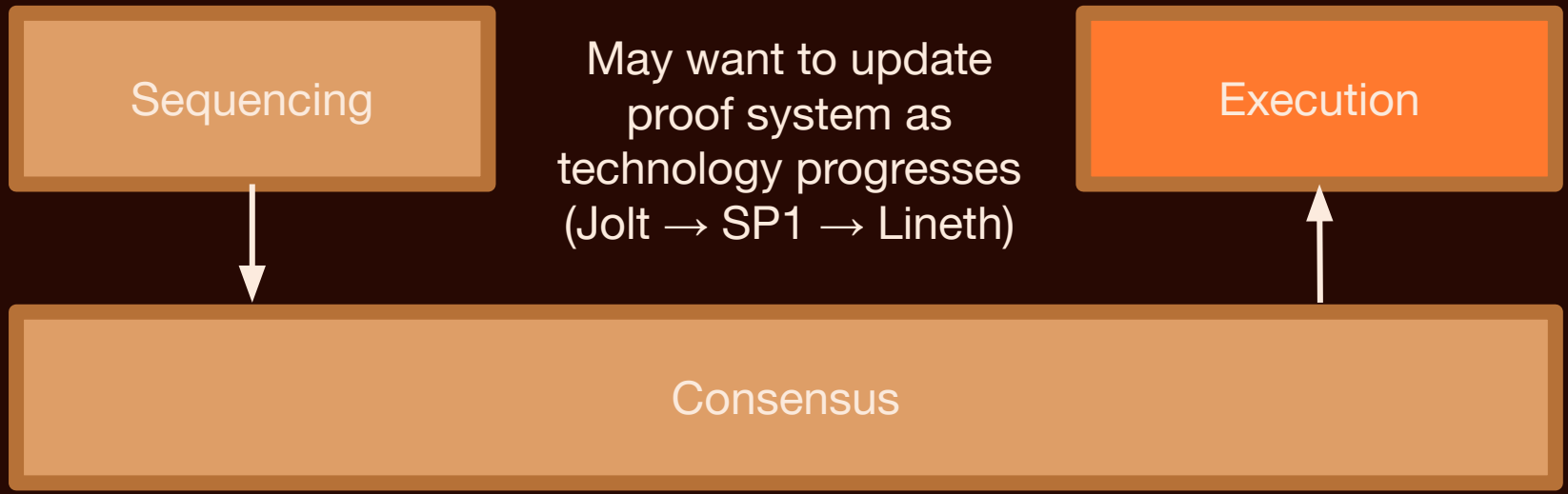
Separated / modular architecture via Layer 2s is more flexible.



Separated / modular architecture via Layer 2s is more flexible.



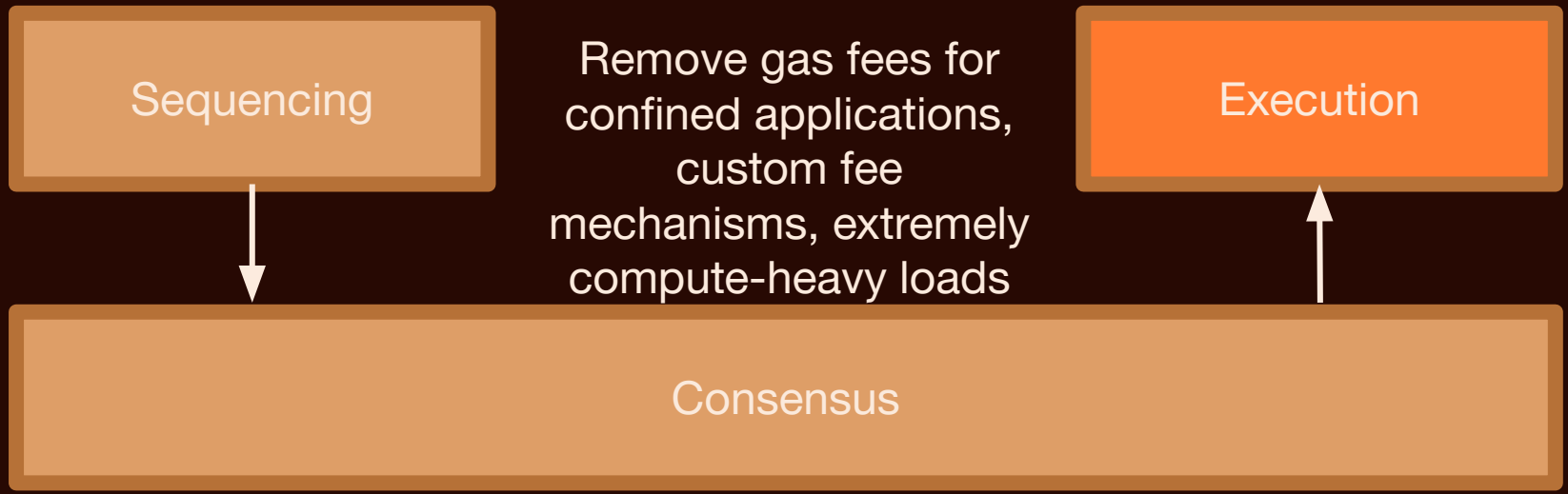
Separated / modular architecture via Layer 2s is more flexible.



Separated / modular architecture via Layer 2s is more flexible.



Separated / modular architecture via Layer 2s is more flexible.





Each component can sit in a different quadrant



Each component can sit in a different quadrant:

Sequencer

Consensus

Exeuction

Ethereum:



Each component can sit in a different quadrant:
Sequencer
Consensus
Exeuction

Collateral AppChain:



Each component can sit in a different quadrant:

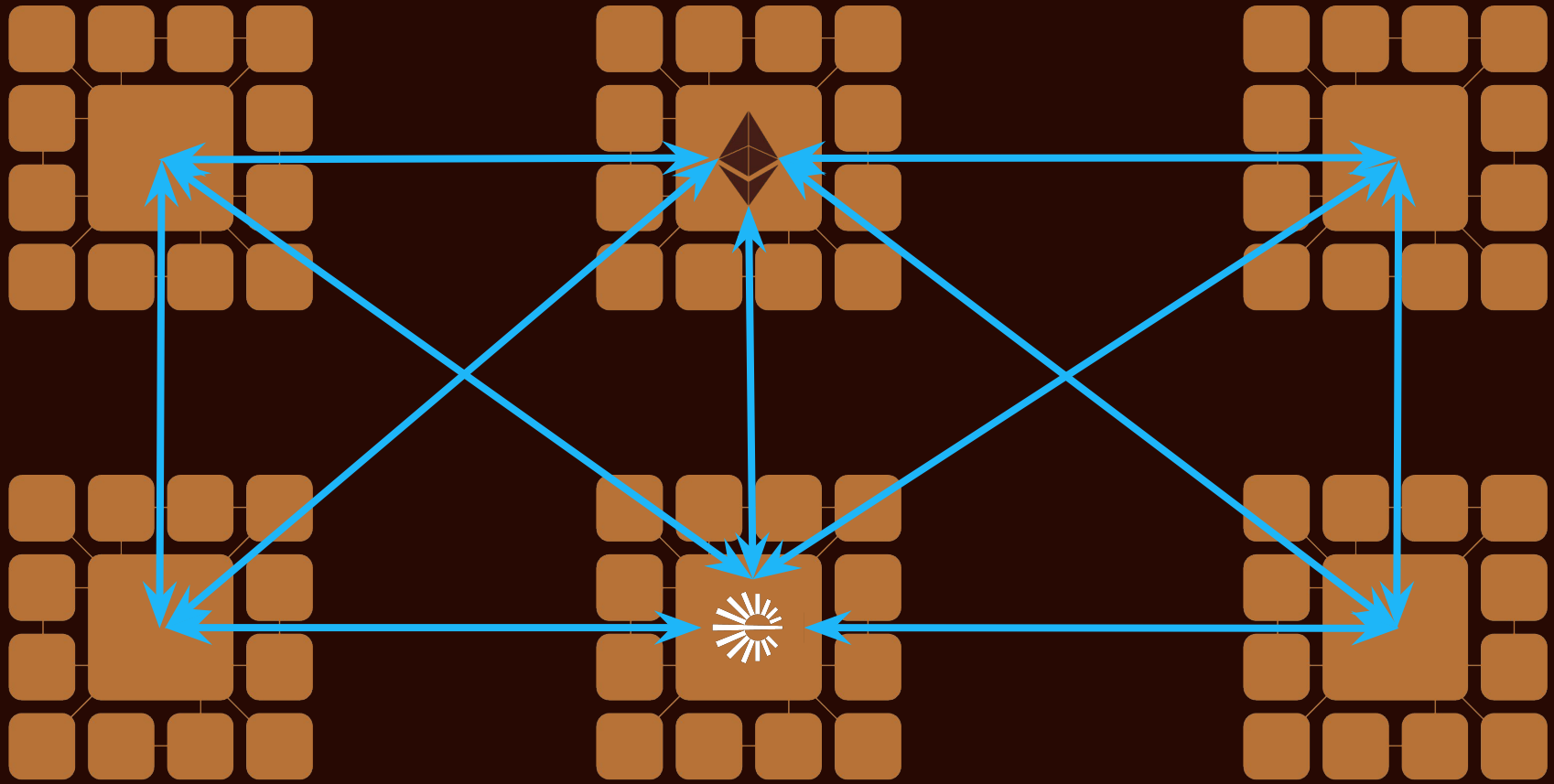
Sequencer

Consensus

Exeuction

Canton:

On Interoperability…



We have interoperability friction due to **Slow finality times + slow proving times + lack of standarization***

We have interoperability friction due to **Slow finality times** + slow proving times + lack of standarization*

Finality time is a comprised of consensus latency and throughput

We have interoperability friction due to Slow finality times + **slow proving times** + lack of standarization*

Interop is defined by assets / applications, not chains!

ZK proofs are not needed for all applications (e.g. USDC)

We have interoperability friction due to Slow finality times + **slow proving times** + lack of standarization*

Interop is defined by assets / applications, not chains!

TEE + ZK proofs can bridge the latency gap (though more difficult for private applications)

We have interoperability friction due to Slow finality times + slow proving times + **lack of standarization***

We have good standards! (ERC-7786, etc.)

Issue is political alignment, incentives, and neutrality

