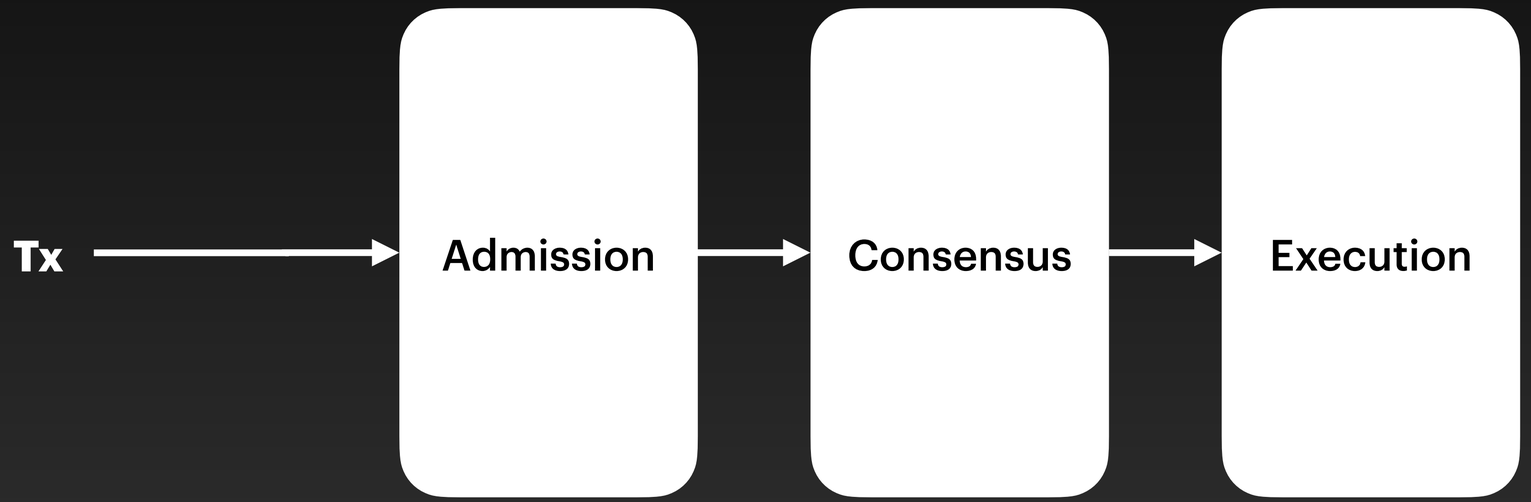
The Sui Validator

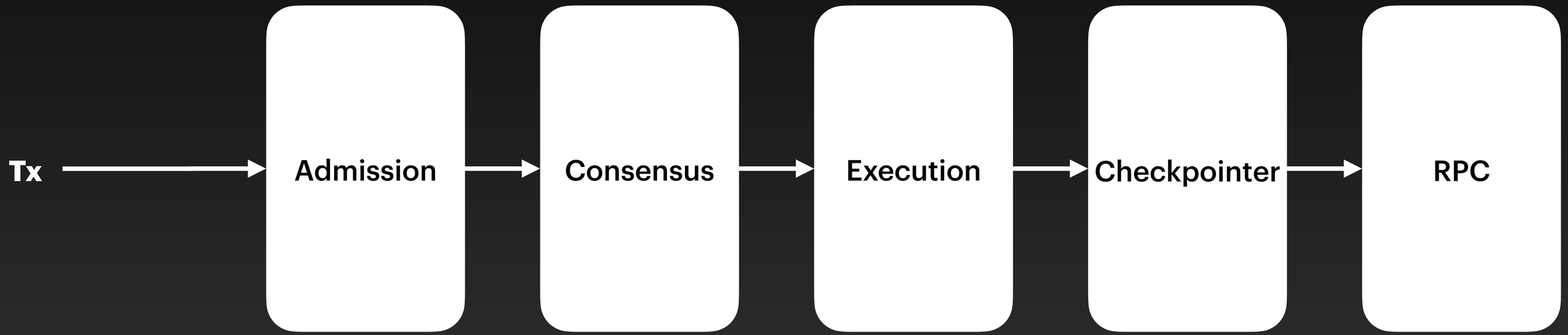
In 3 years from now

Lisbon Research Offsite

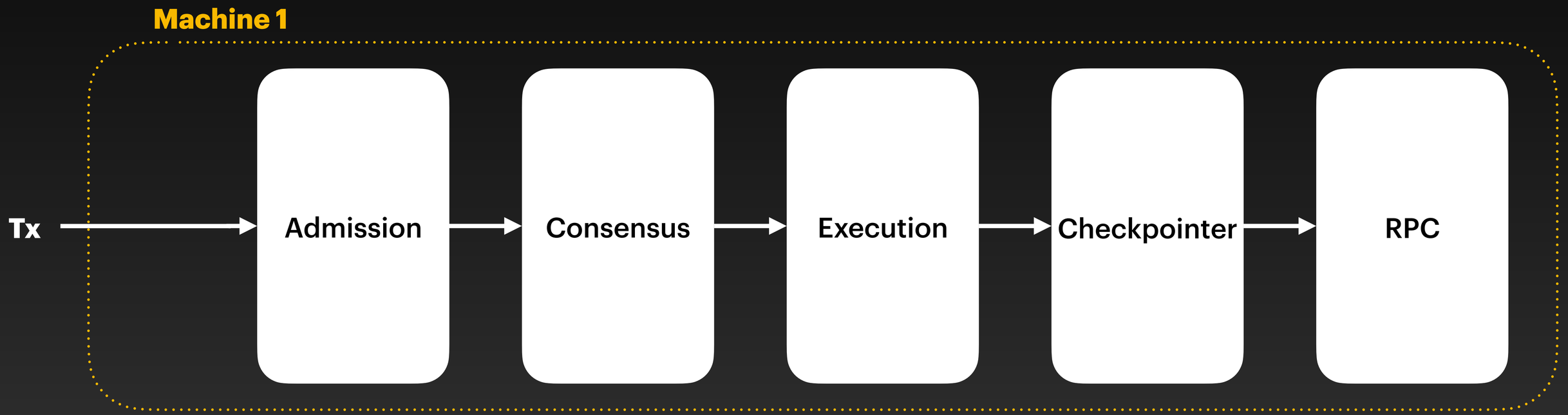
Current State



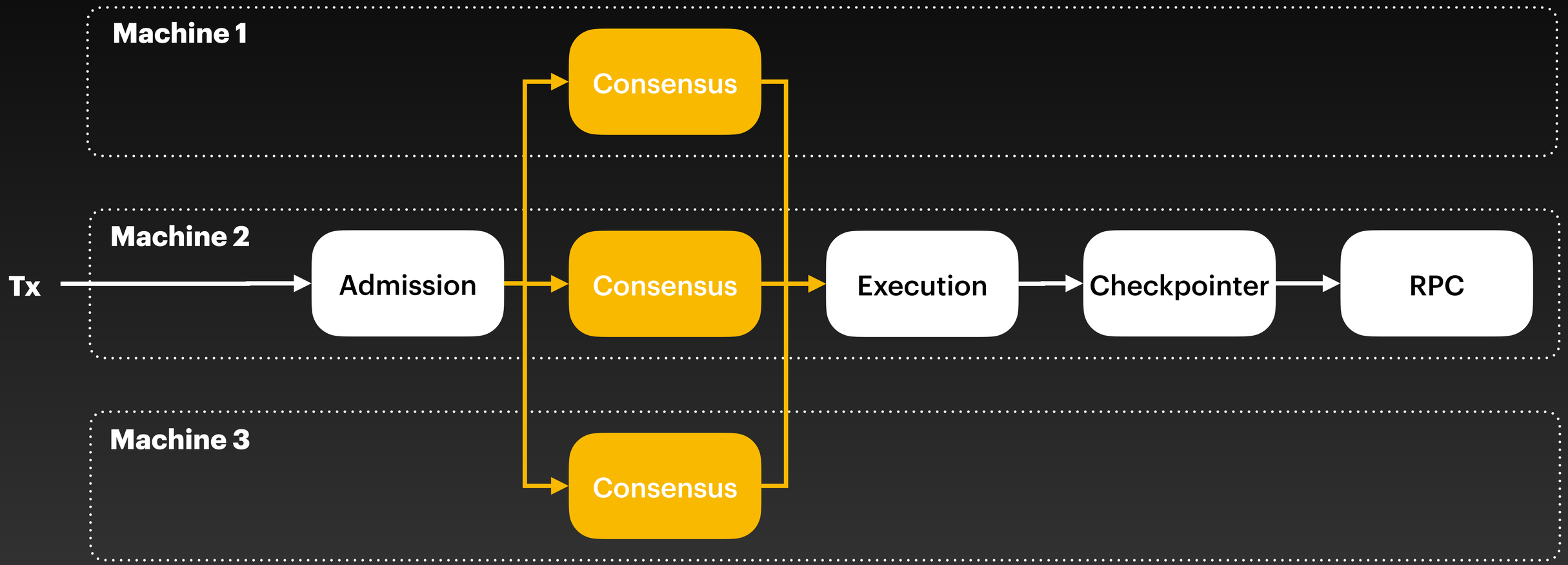
Current State



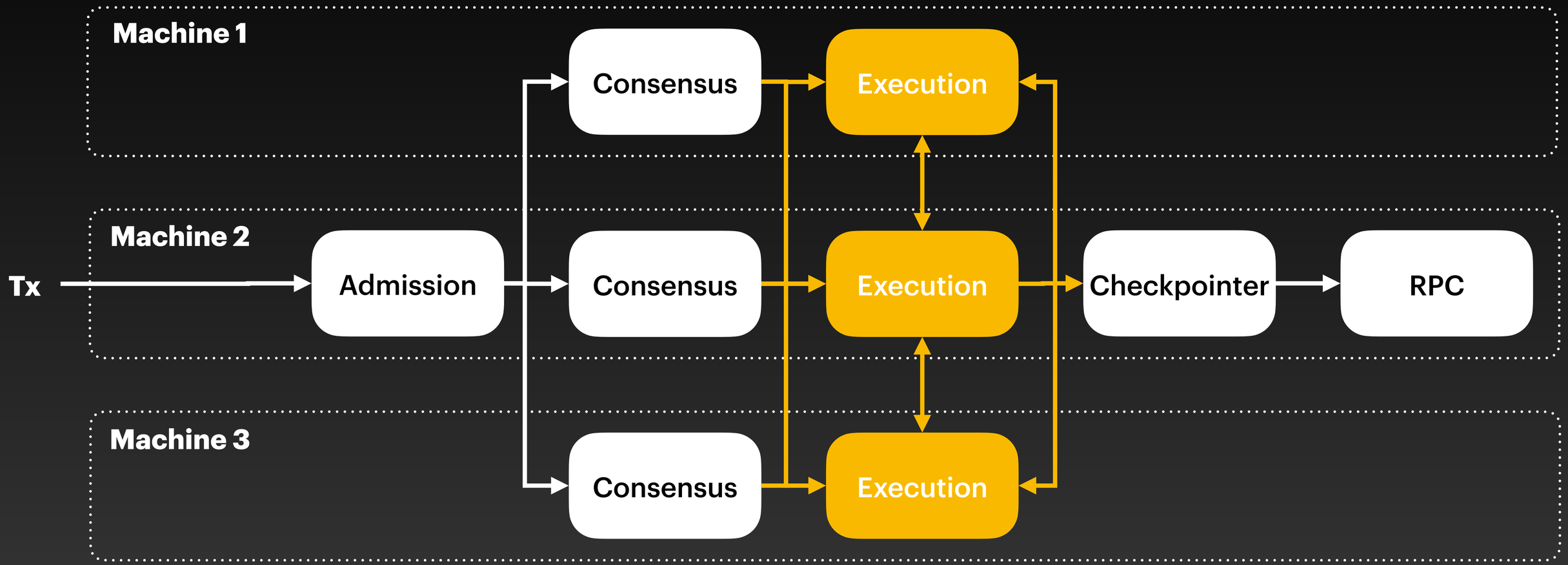
Current State



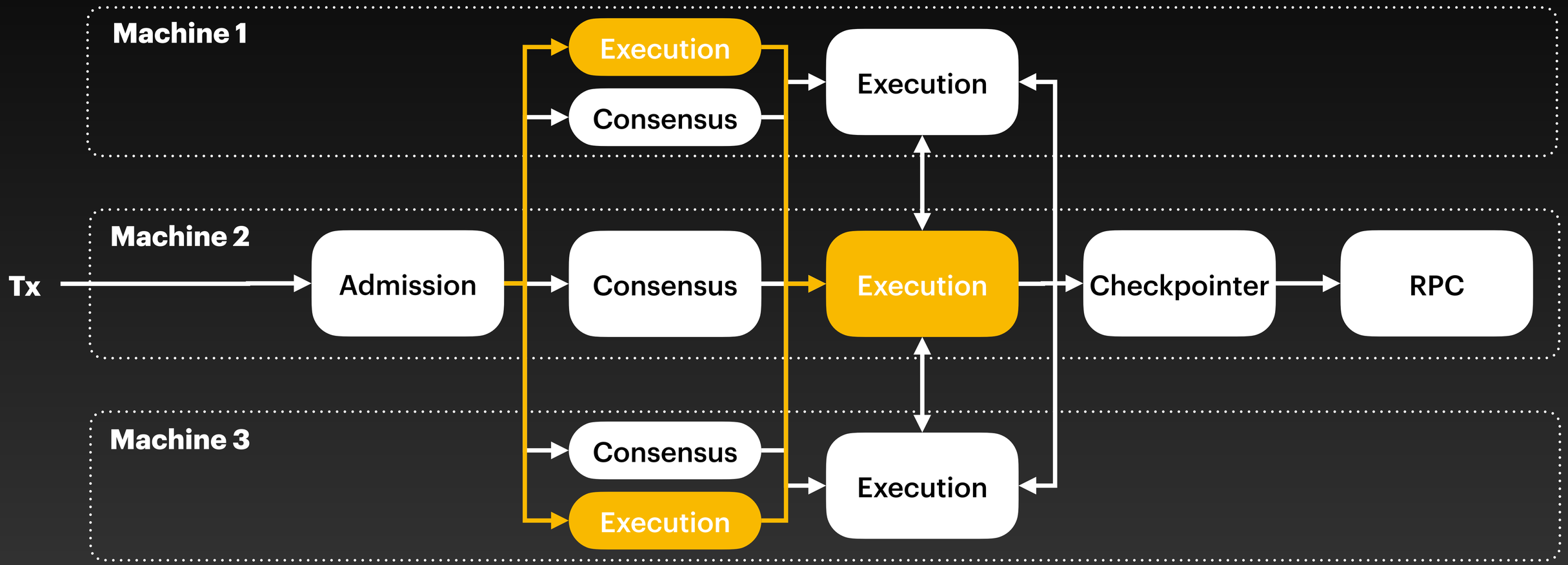
Narwhal: Scaling Consensus



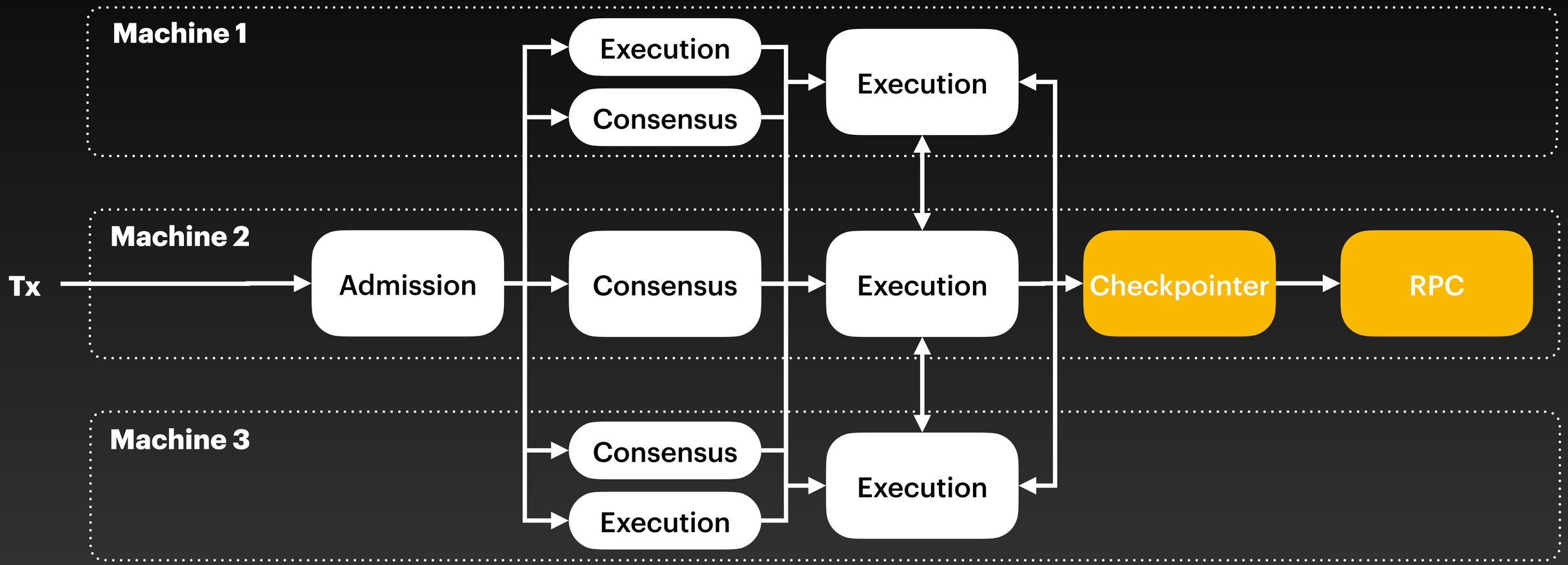
Pilotfish: Scaling Execution (1)



Remora: Scaling Execution (2)



Other Components?



Writes

Reads

How to fix that?

Controversial

Checkpoint Builder

Purposes

- Light clients can read state
- Quick comparison of validators state

Issues

- All clients wait for seconds for all reads
- Hard to optimise / mostly unspecified

Checkpoint Builder

Purposes

- Light clients can read state
- Quick comparison of validators state

Issues

- All clients wait for seconds for all reads
- Hard to optimise / mostly unspecified

What else?

RPC

Purposes

- Serve state to clients

Issues

- Resource intensive
- Limit to serve only a few full nodes

RPC

Purposes

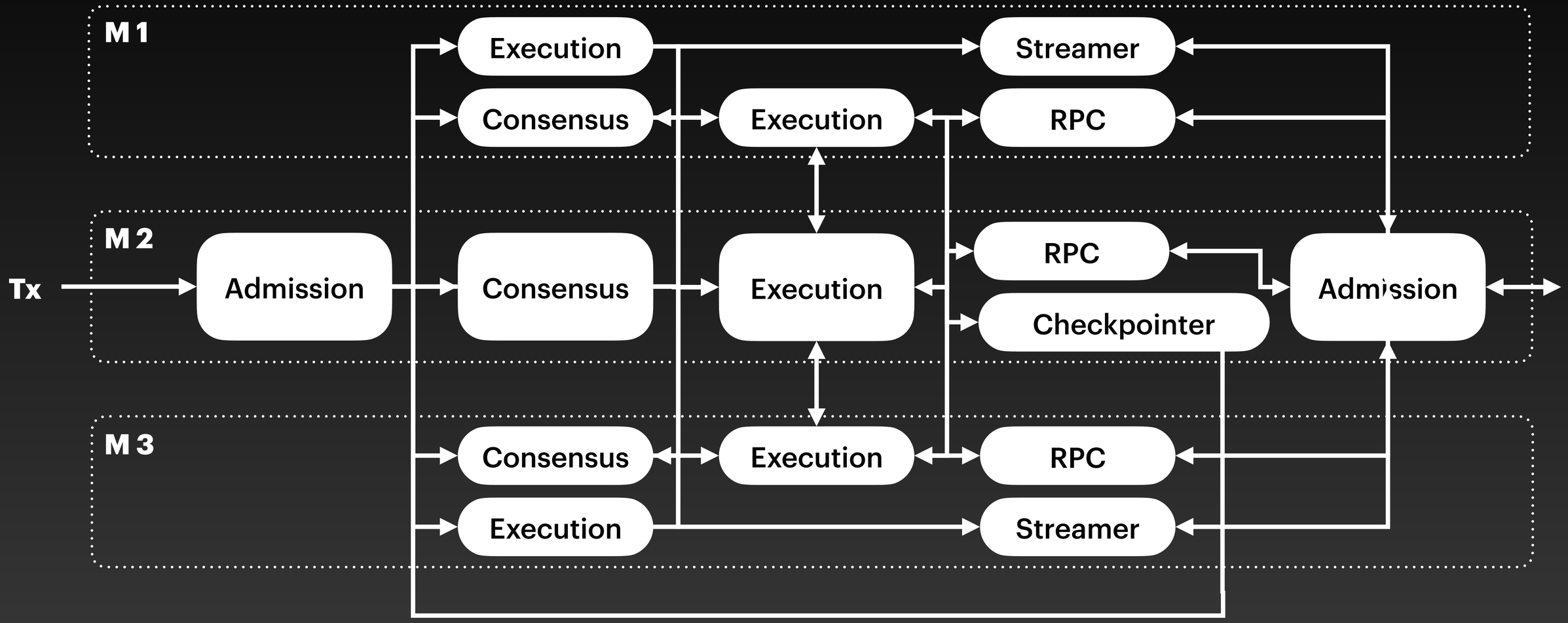
- Serve state to clients

Issues

- Resource intensive
- Limit to serve only a few full nodes

What else?

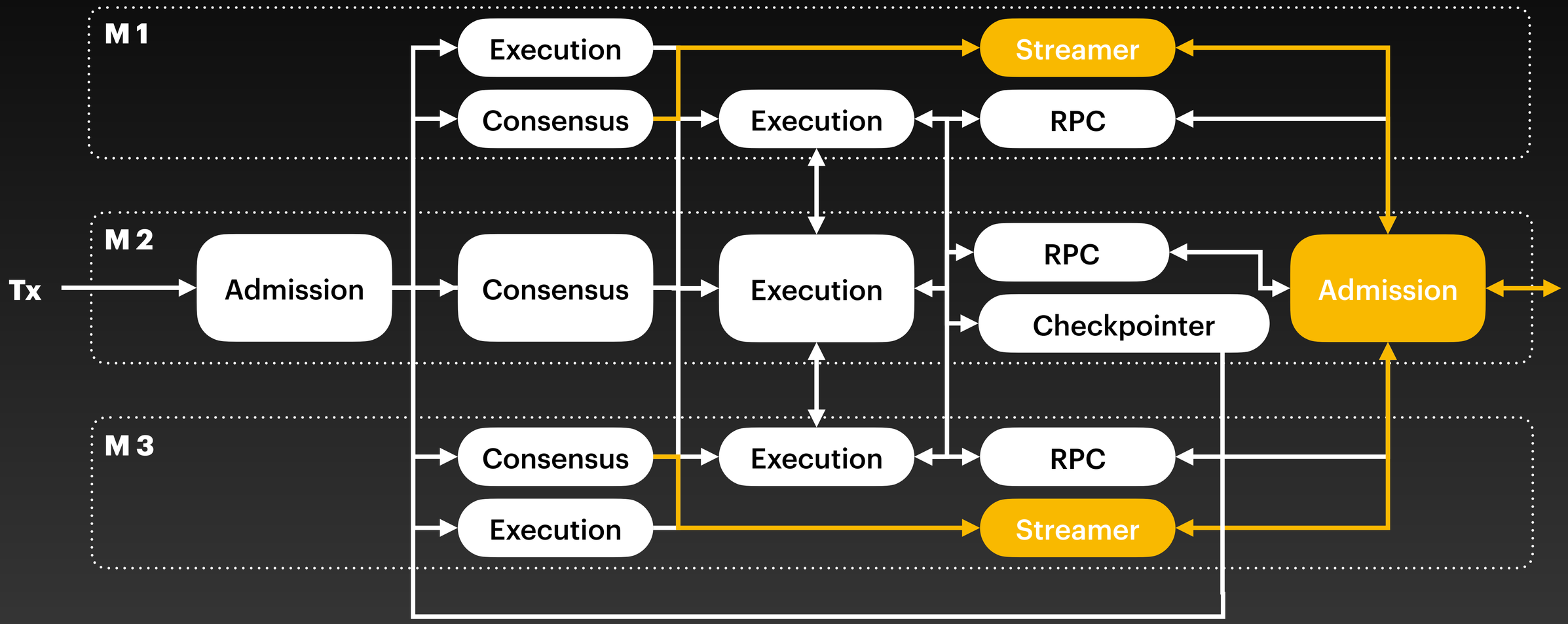
Suggestion


M 1
Execution
Streamer
Consensus
Execution
RPC
M 2
Tx
Admission
Consensus
Execution
RPC
Checkpointer
Admission
M 3
Consensus
Execution
RPC
Execution
Streamer

A Weekend Project

— the Walrus Team

Stream the DAG



Streamer

Advantages

- **Low latency for full clients**

Challenges

- **Full client requires high bandwidth**

Streamer

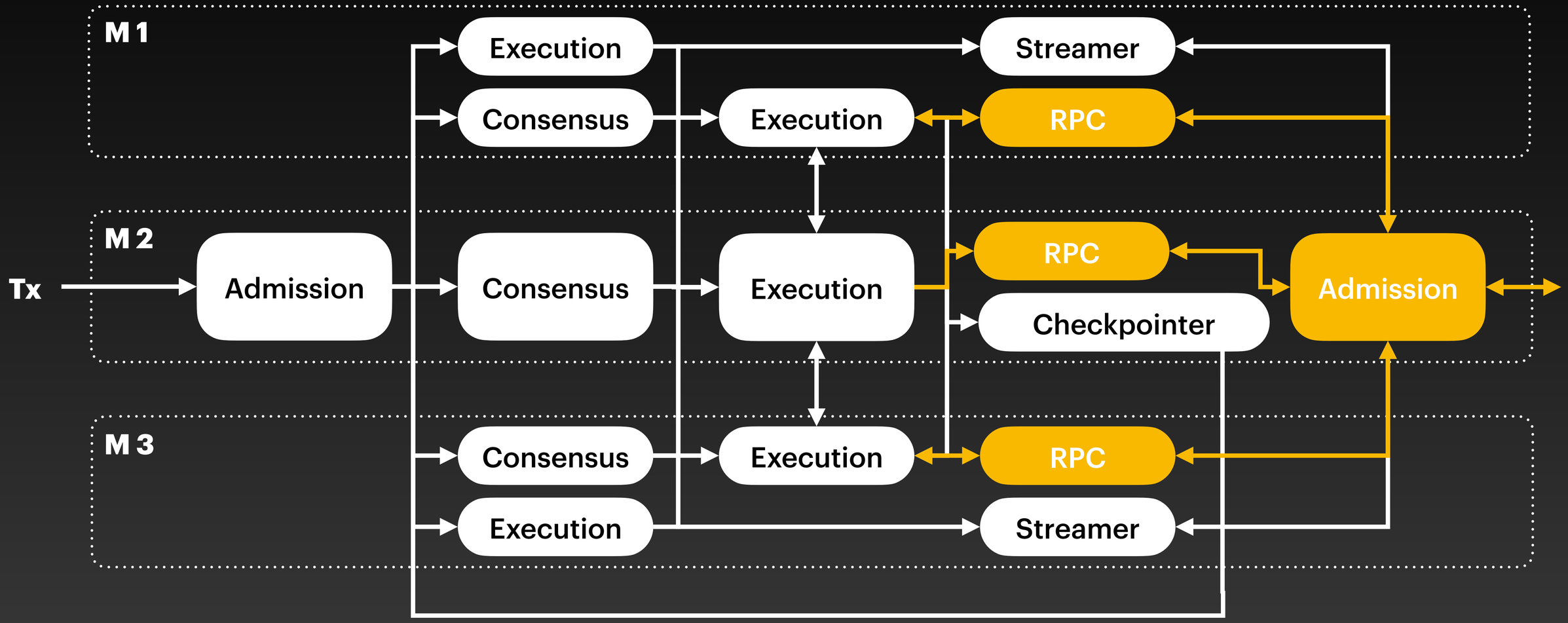
Advantages

- Low latency for full clients
- Easy sparse nodes
- Clients can run their own execution
  - Rollups, L2
- Clients can run their own consensus
  - Higher quorums (O-Flex)
  - Run async consensus!
- Client choose their finality rules
  - Weak/soft finality with rollbacks

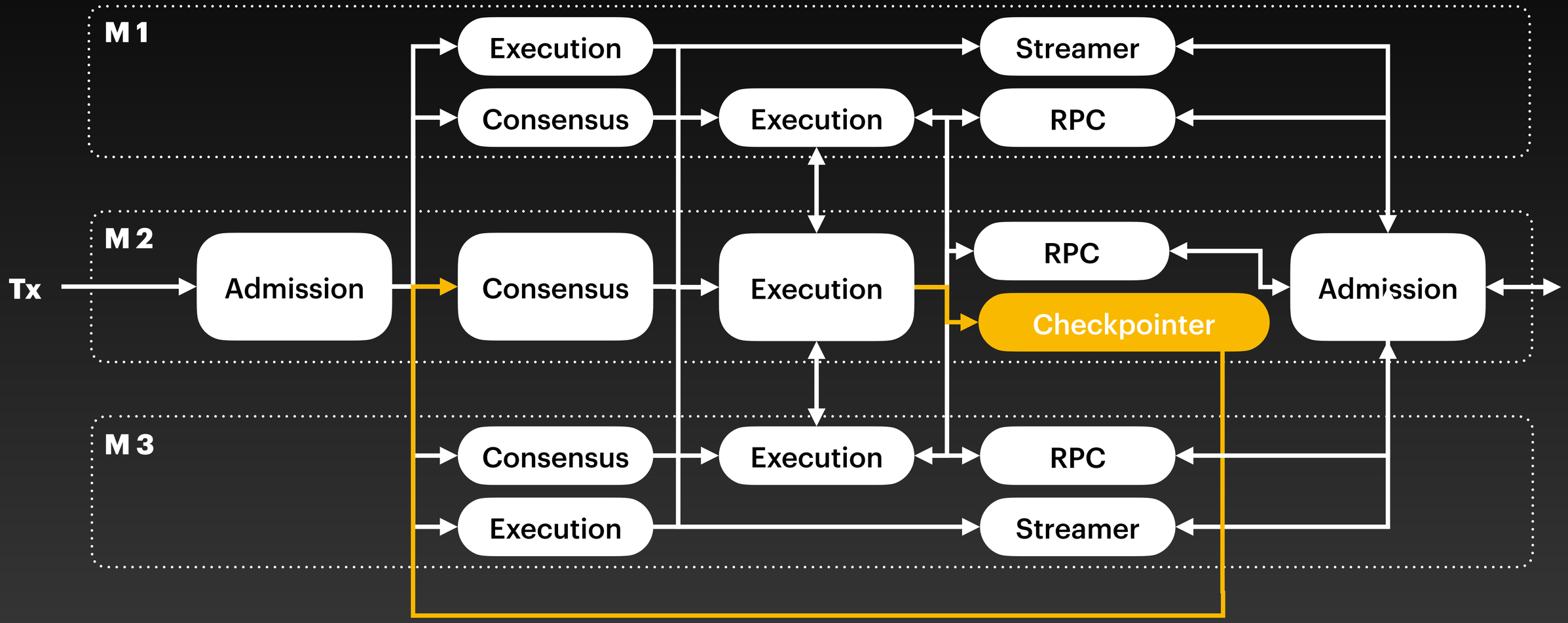
Challenges

- Full client requires high bandwidth

Serving State



Checkpointer



Checkpointer

- Should we send back entire state through consensus or a commitment of it?
- Should we have :
  - Checkpoints without consensus, through a separate component
  - One checkpoint per epoch that goes through consensus (for validator state agreement)