Bringing Decentralized Search to Decentralized Services

Mingyu Li[1], Jinhao Zhu[1], Tianxu Zhang[1], Cheng Tan[2],
Yubin Xia[1], Sebastian Angel[3], Haibo Chen[1]

Northeastern University

Penn
University of Pennsylvania

Outline

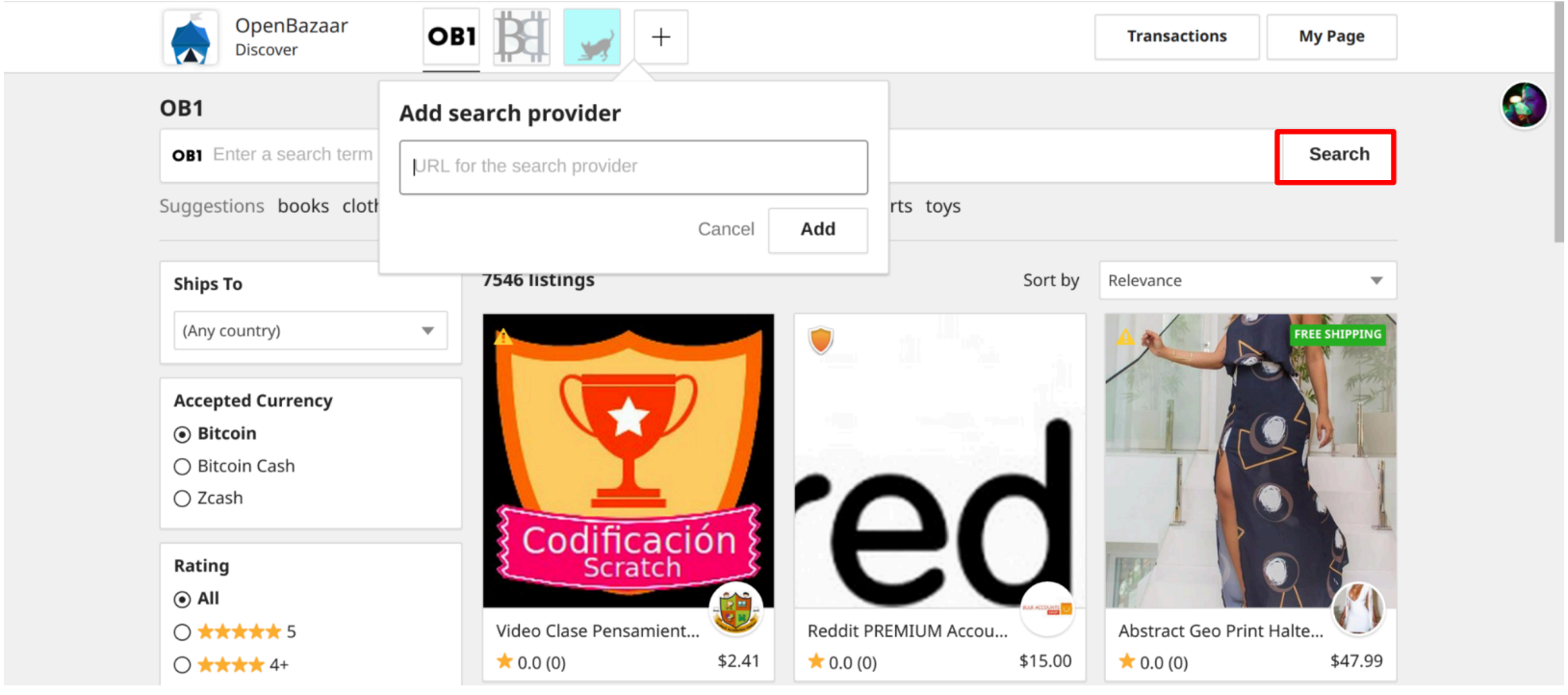
- **Problem Statement**
- Design: decentralized search
- Design: verifiable search
- Design: private search
- Evaluation

Status Quo of DApps

- **Decentralized applications are booming**

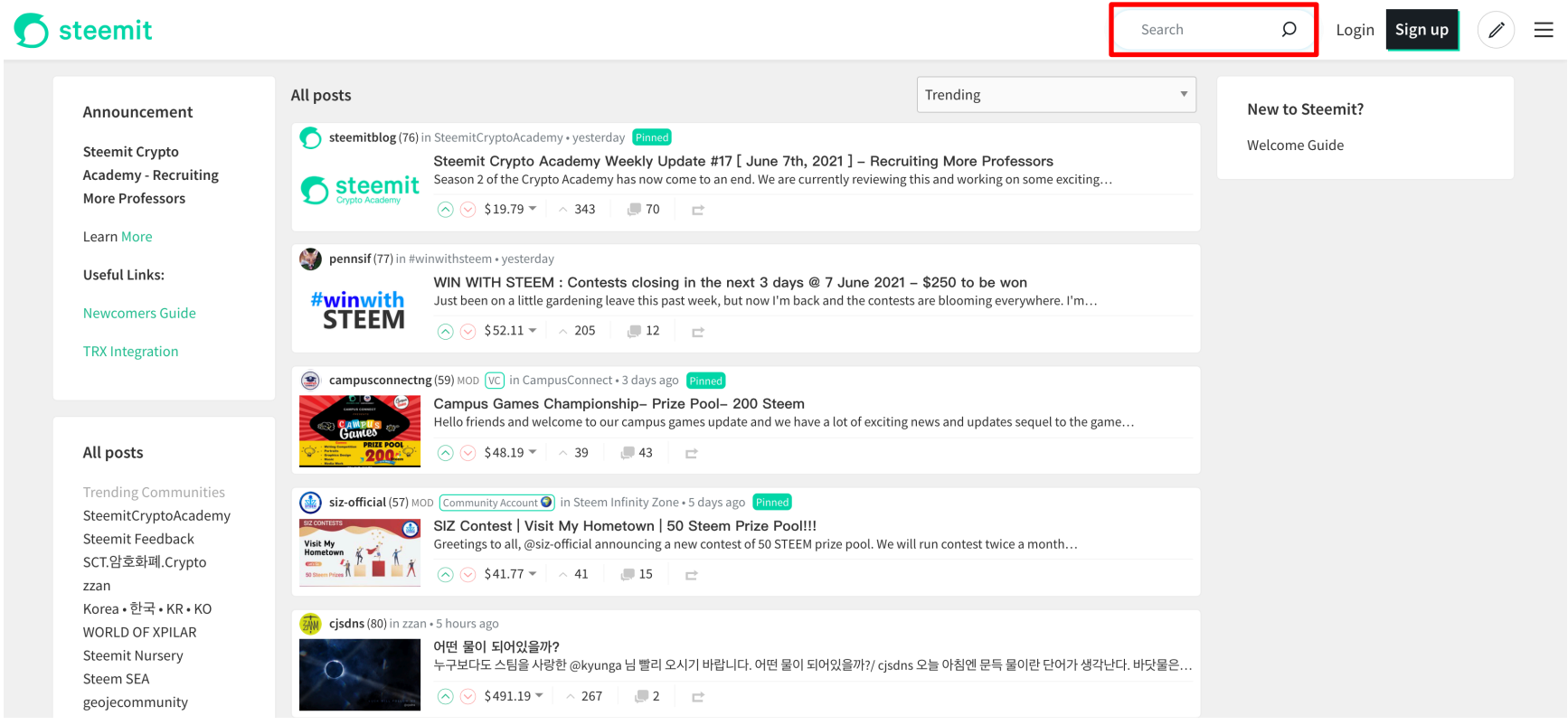
Service	Centralized	Decentralized
E-commerce	eBay	OpenBazaar
Social Media	Twitter	Steemit
Video Sharing	Youtube	DTube
Public Storage	DropBox	IPFS
Messaging	Slack	Matrix
Video Conference	Zoom	Zipcall
…	…	…

Search as DApps' Entrance



OpenBazaar users need to search shopping items

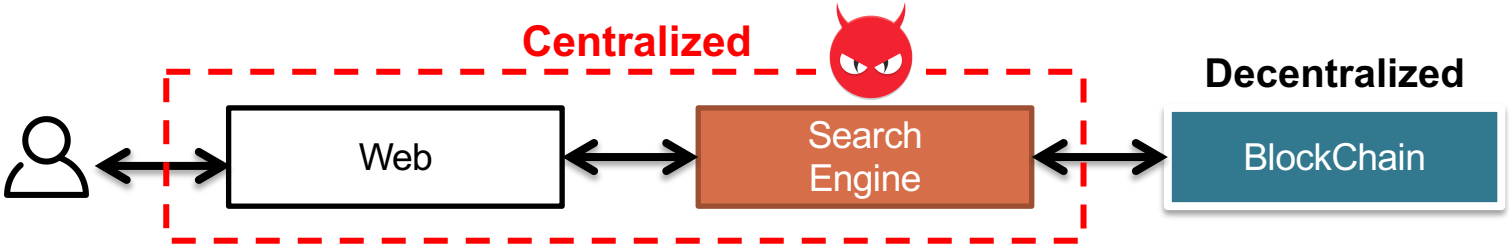
Search as DApps' Entrance



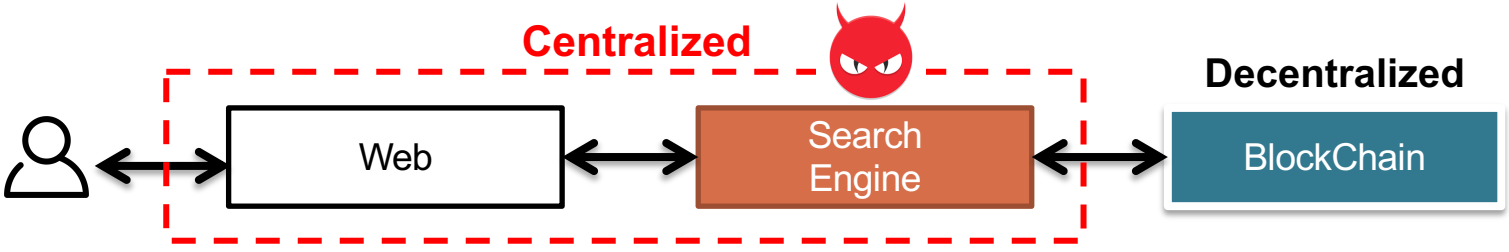
Steemit users need to search blogs of interest

Why centralized search not fit?

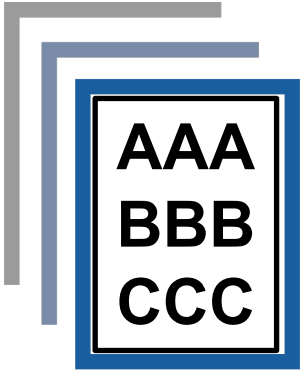
Why centralized search not fit?



Why centralized search not fit?



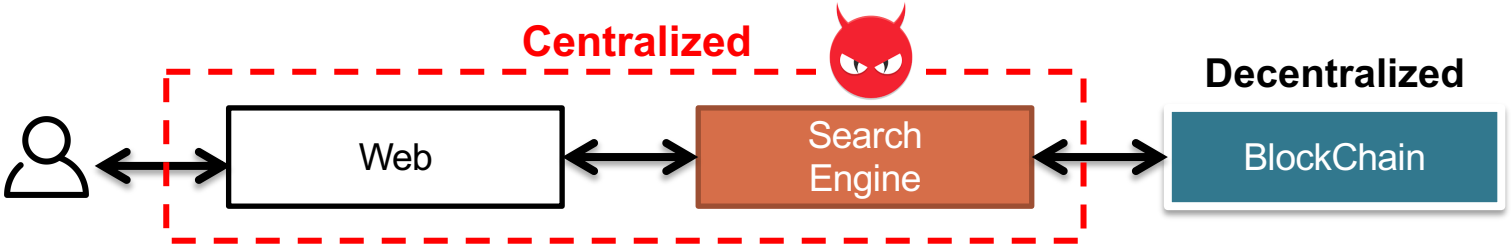
Expected



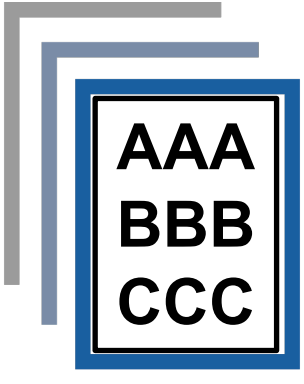
Opaque Censorship[1]

[1] **Steemit censoring users on immutable social media blockchain's front-end.**
https://cryptoslate.com/steemit-censoring-users-immutable-blockchainsocial-media/

Why centralized search not fit?

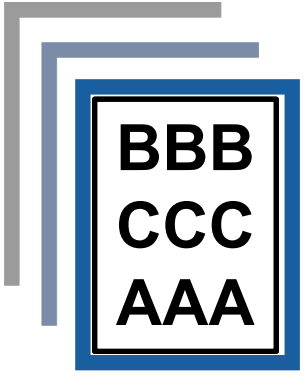


Expected



Opaque Censorship[1]

Paid/Bias Listings



[1] **Steemit censoring users on immutable social media blockchain's front-end.**
https://cryptoslate.com/steemit-censoring-users-immutable-blockchainsocial-media/

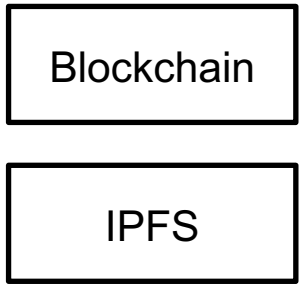
Why not build a decentralized search engine for decentralized apps?

Why not build a decentralized search engine for decentralized apps?

DeSearch

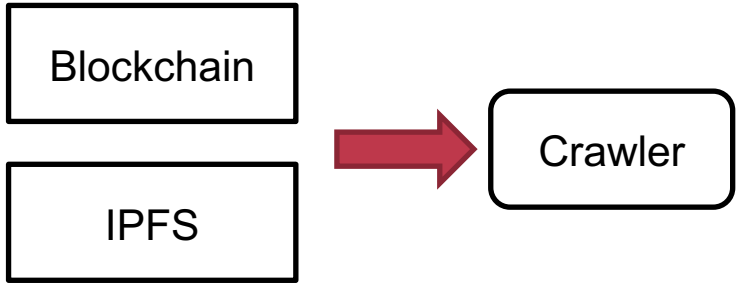
Search Pipeline

Public data source



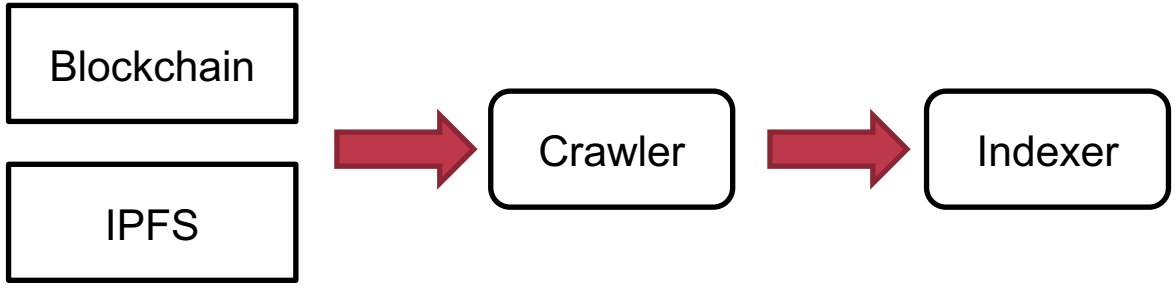
Search Pipeline

Public data source



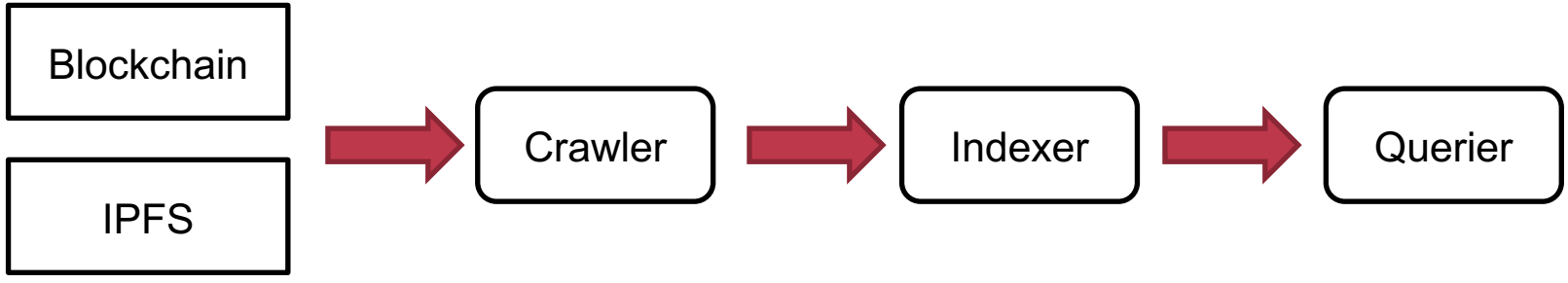
Search Pipeline

Public data source

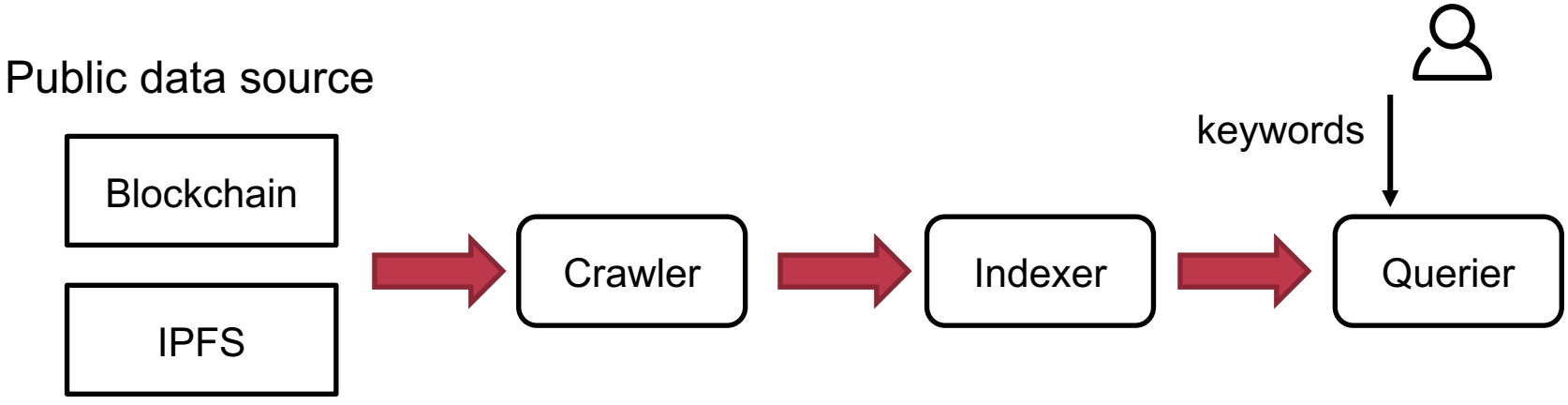


Search Pipeline

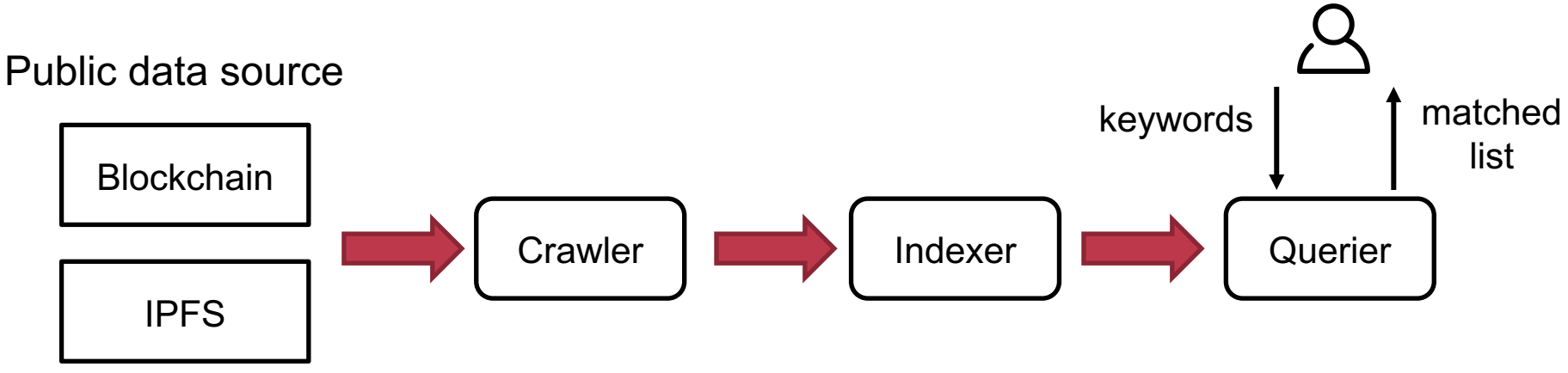
Public data source



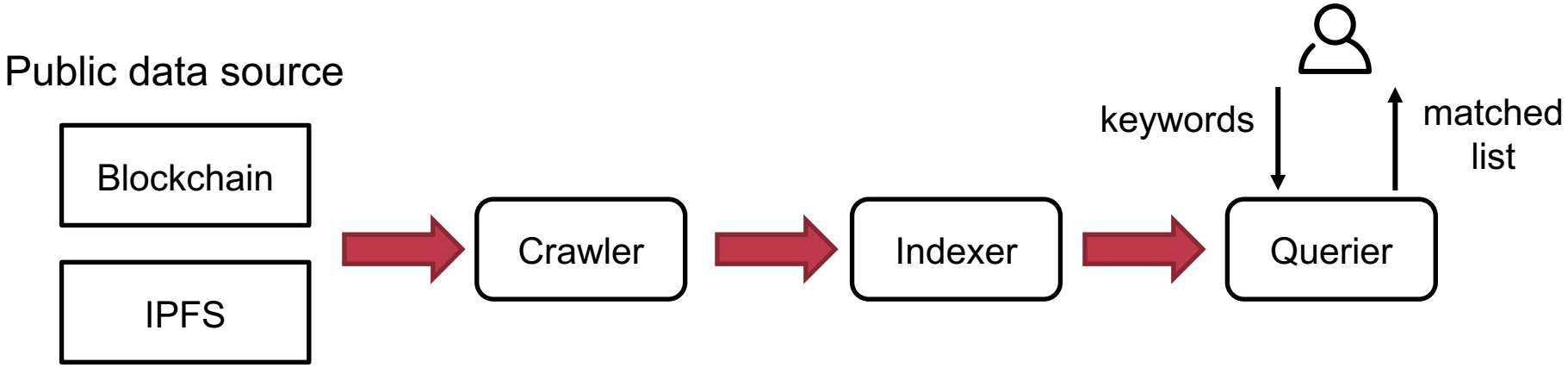
Search Pipeline



Search Pipeline

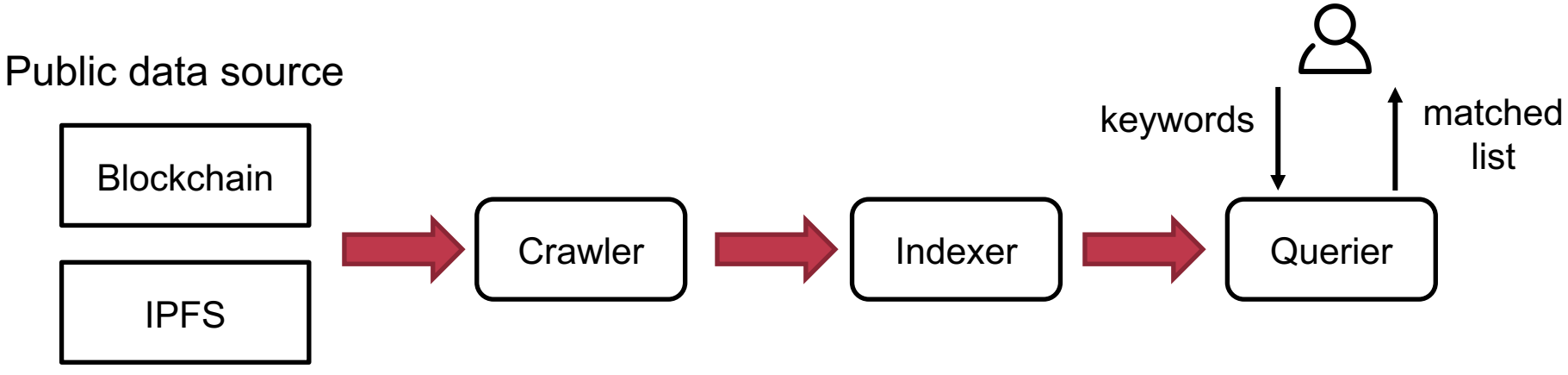


Search Pipeline



Can we use smart contracts?

Search Pipeline

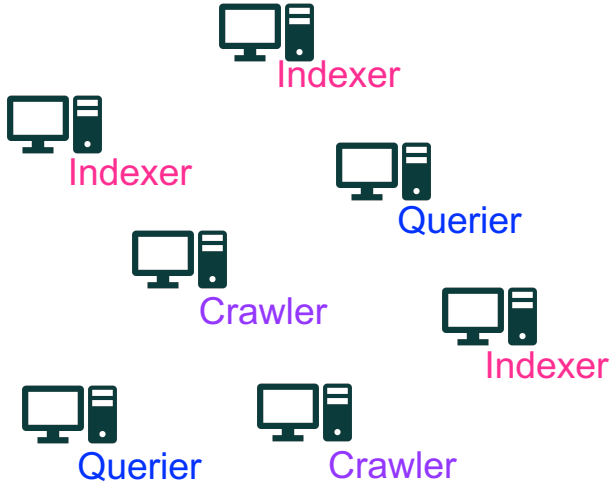


Can we use smart contracts?

Not practical <= long search latency

DeSearch Challenges

Decentralized P2P Network



How to ensure state availability?

DeSearch Challenges

Decentralized P2P Network

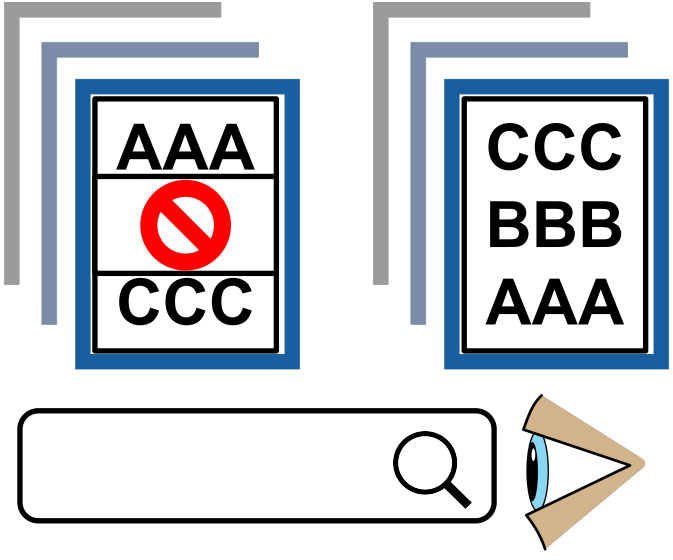
How to ensure state availability?

DeSearch Challenges

Decentralized P2P Network

How to ensure state availability?

Untrusted Environments

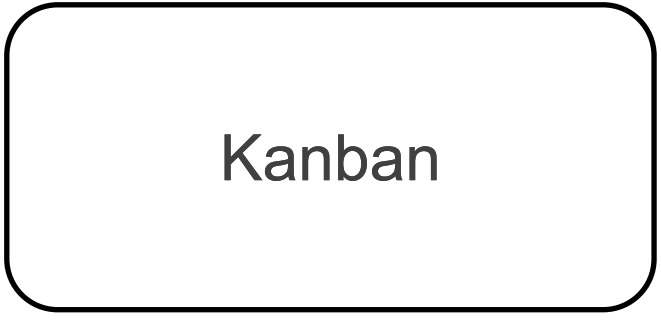


How to detect dishonest behaviors?

Outline

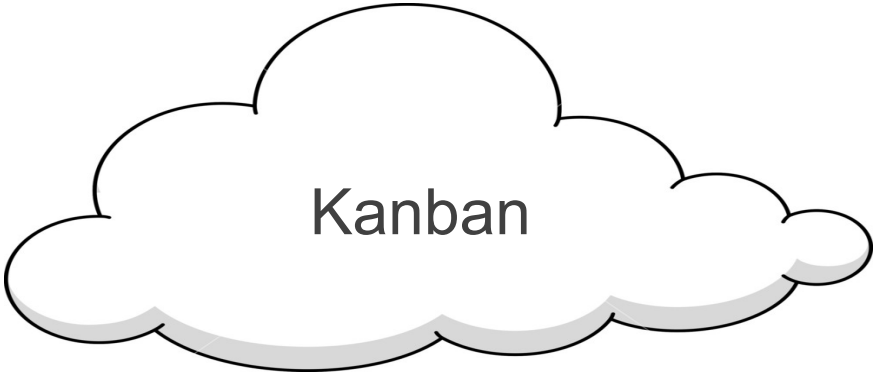
- **Problem Statement**
- **DeSearch Design: decentralized search**
  - how to ensure state availability?
- **Design: verifiable search**
- **Design: private search**
- **Evaluation**

Decouple states from computation



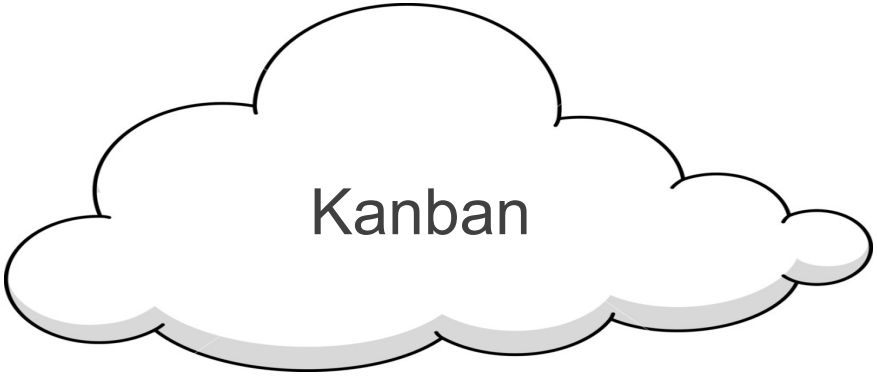
States with high availability

Decouple states from computation

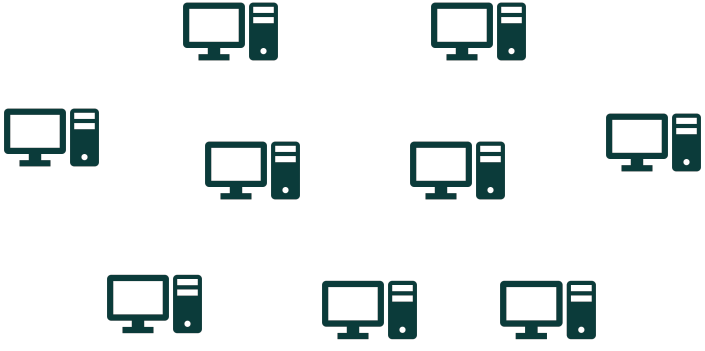


States with high availability

Decouple states from computation



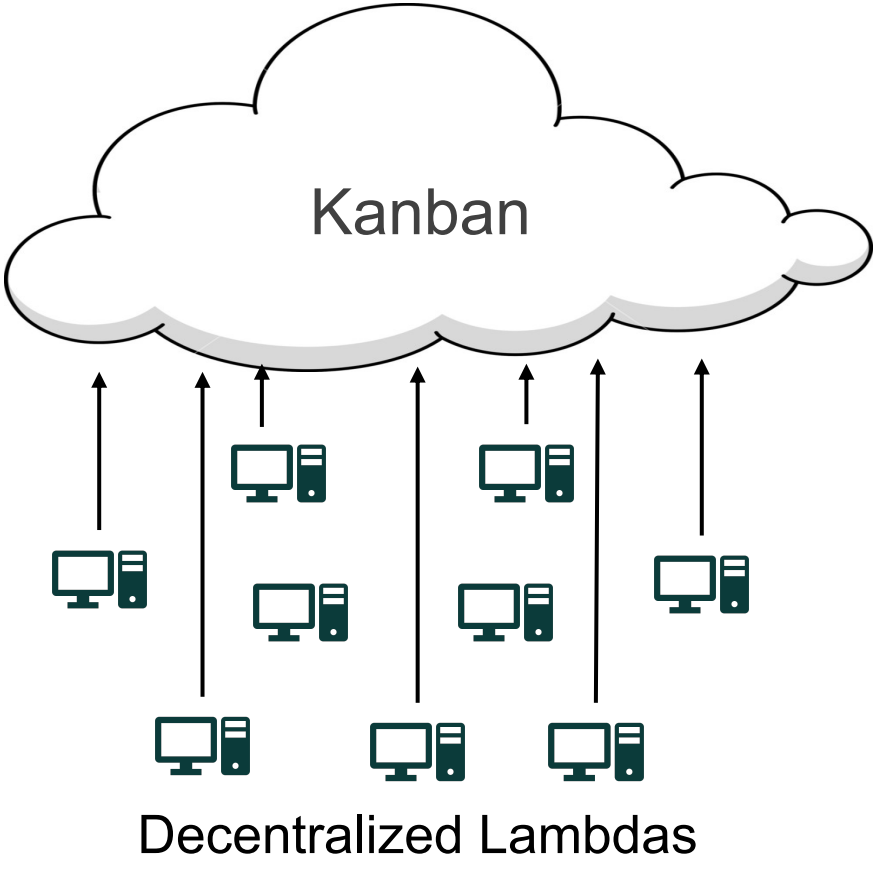
States with high availability



Stateless short-lived tasks
(crawlers, indexers, queriers)

Decentralized Lambdas

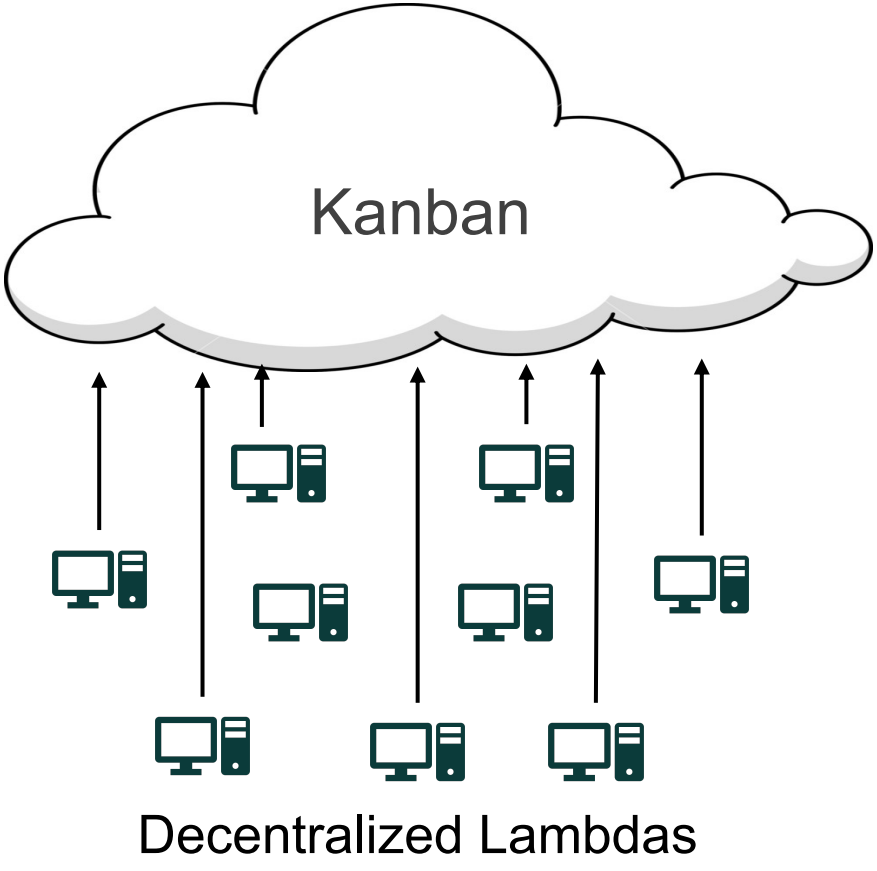
Decouple states from computation



States with high availability

Stateless short-lived tasks
(crawlers, indexers, queriers)

Decouple states from computation



States with high availability

Stateless short-lived tasks
(crawlers, indexers, queriers)

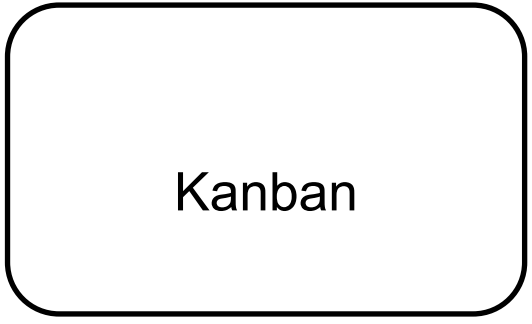
good scalability

good fault tolerance

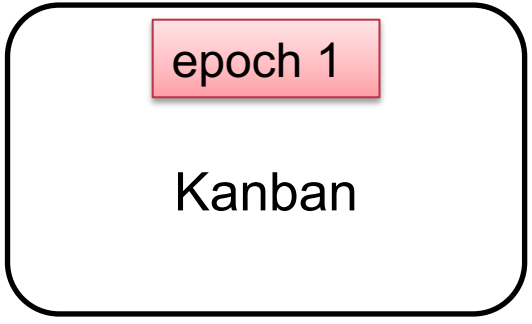
Outline

- **Problem Statement**
- **DeSearch Design: decentralized search**
- **DeSearch Design: verifiable search**
  - Verifiable Kanban
  - Verifiable lambda
- **Evaluation**

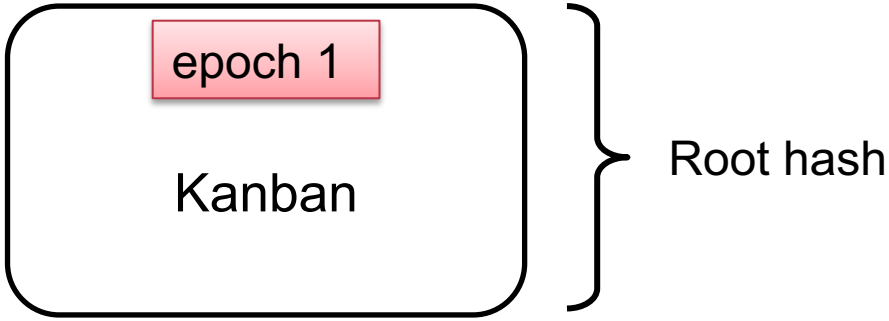
Verifiable Kanban



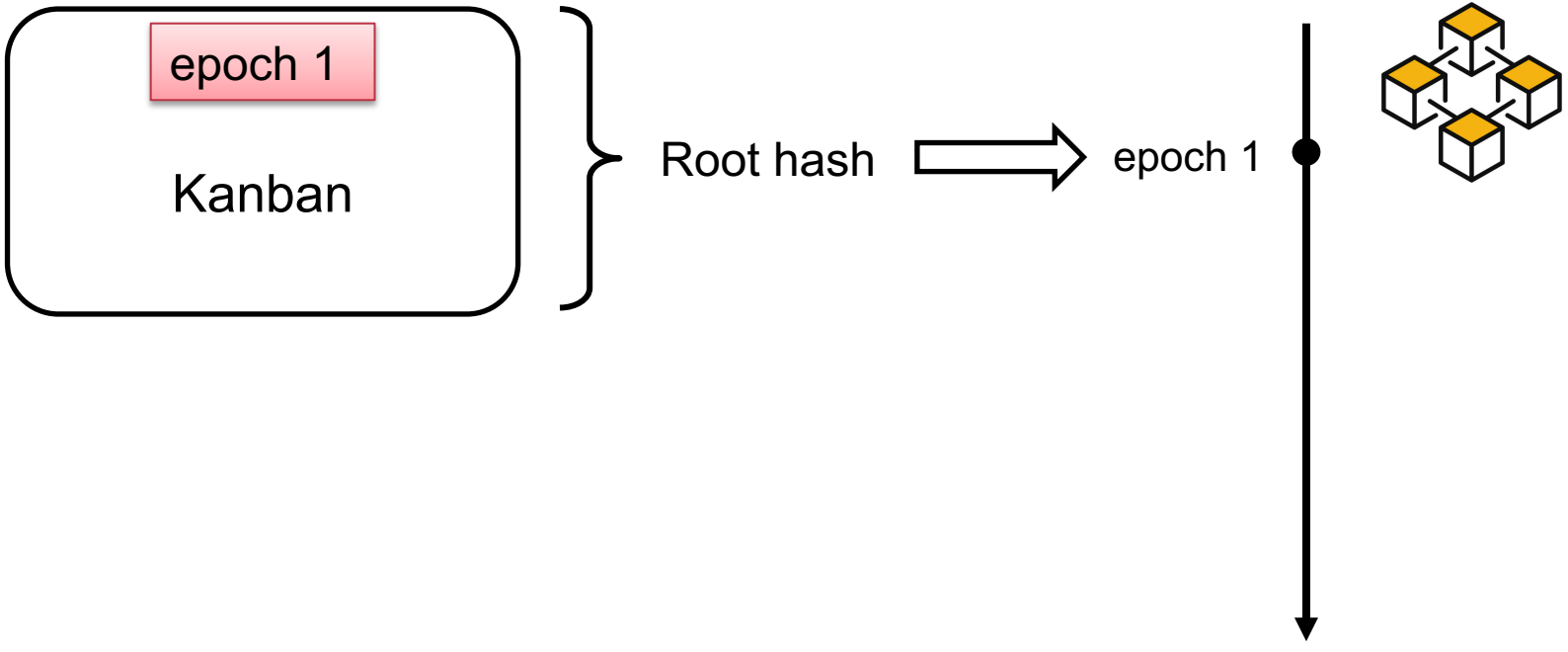
Verifiable Kanban



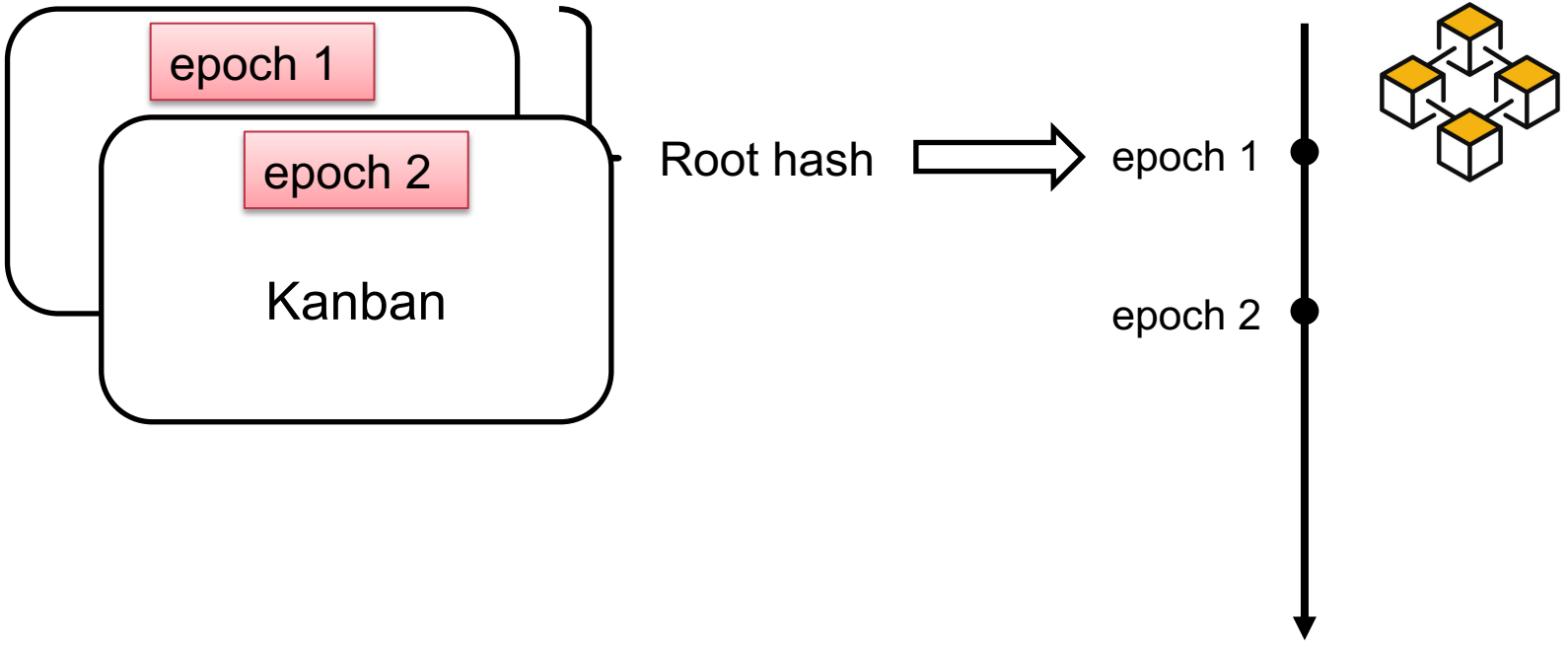
Verifiable Kanban



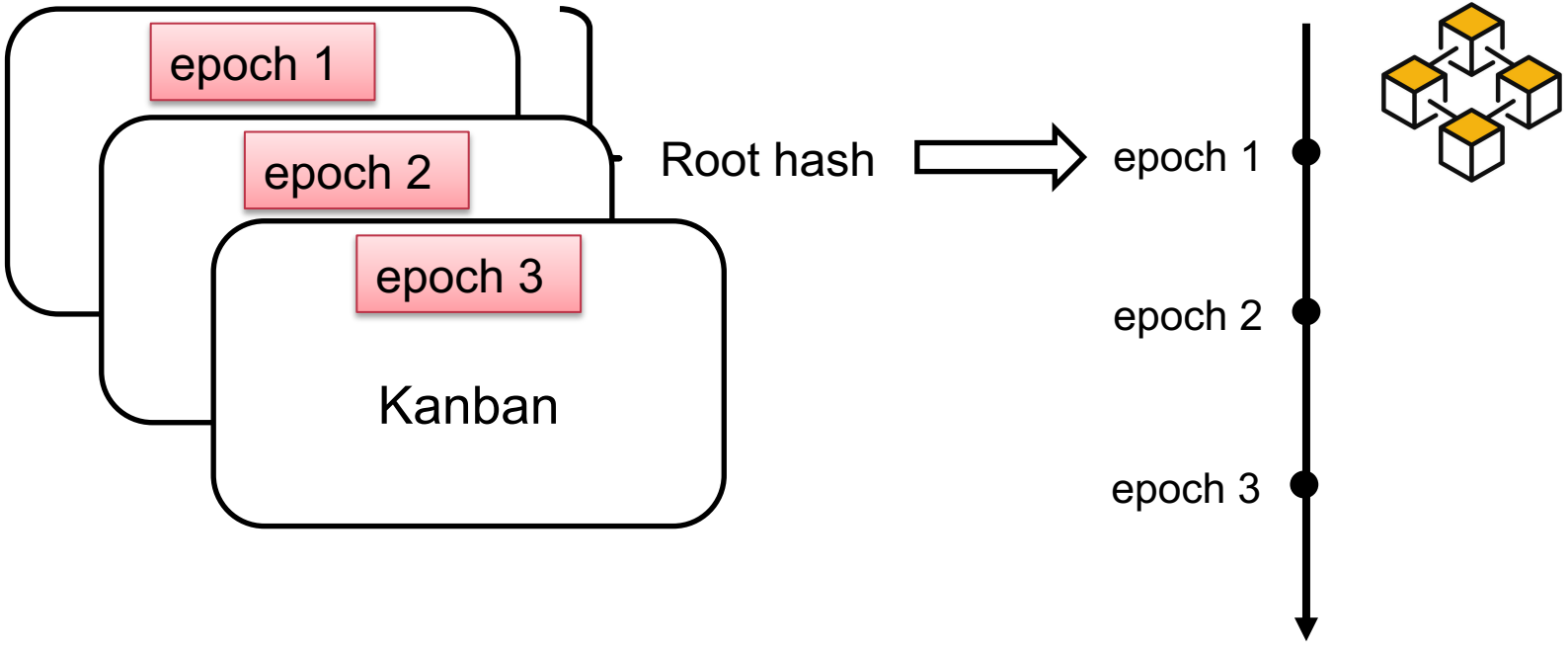
Verifiable Kanban



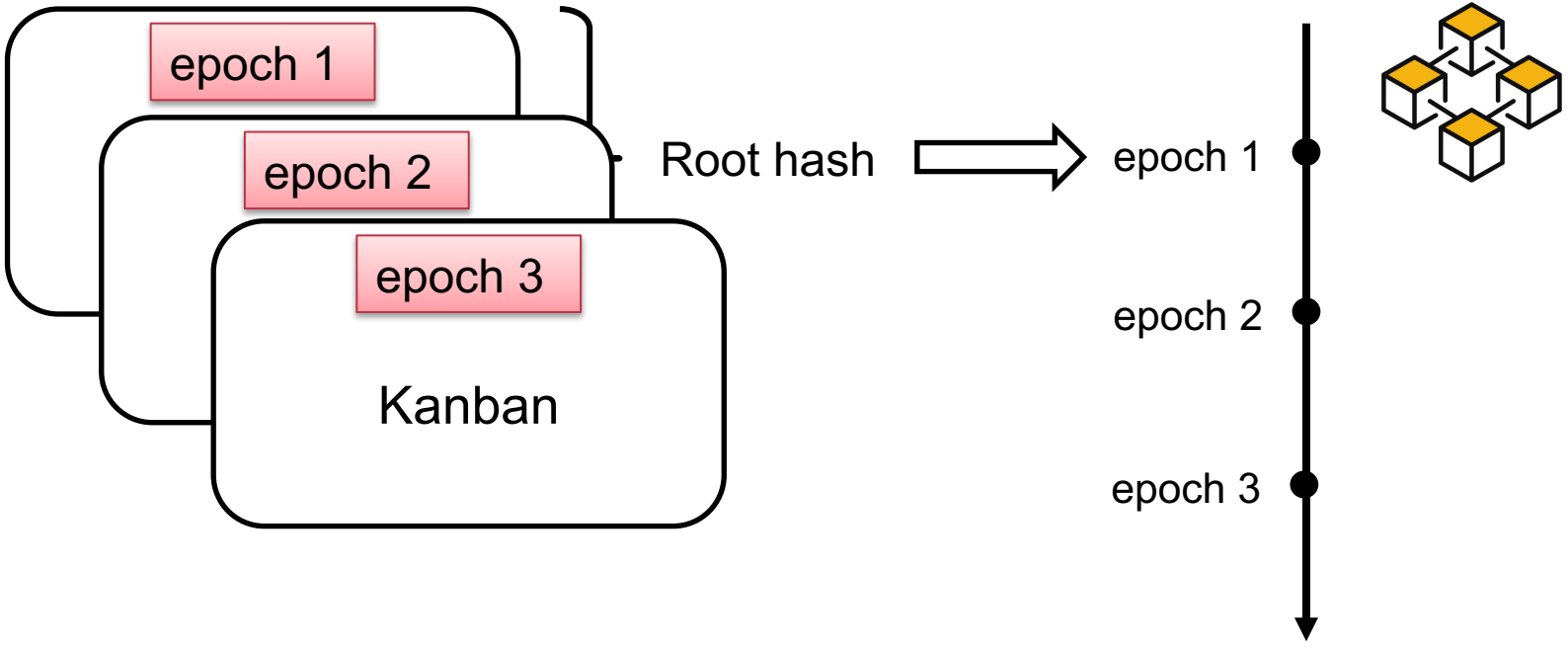
Verifiable Kanban



Verifiable Kanban



Verifiable Kanban



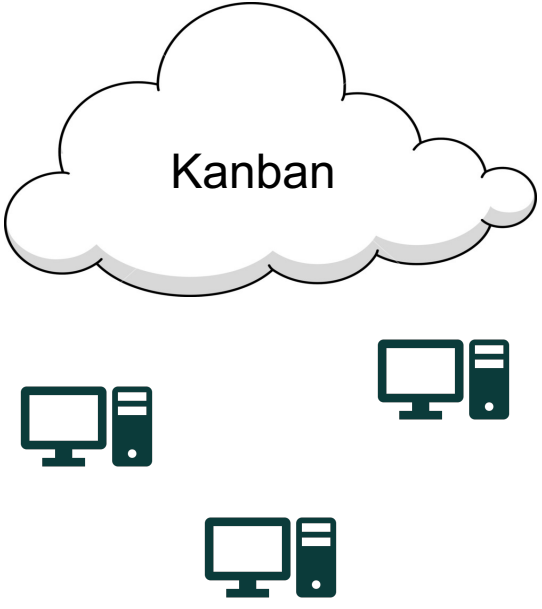
Cloud

High availability + big storage

Blockchain

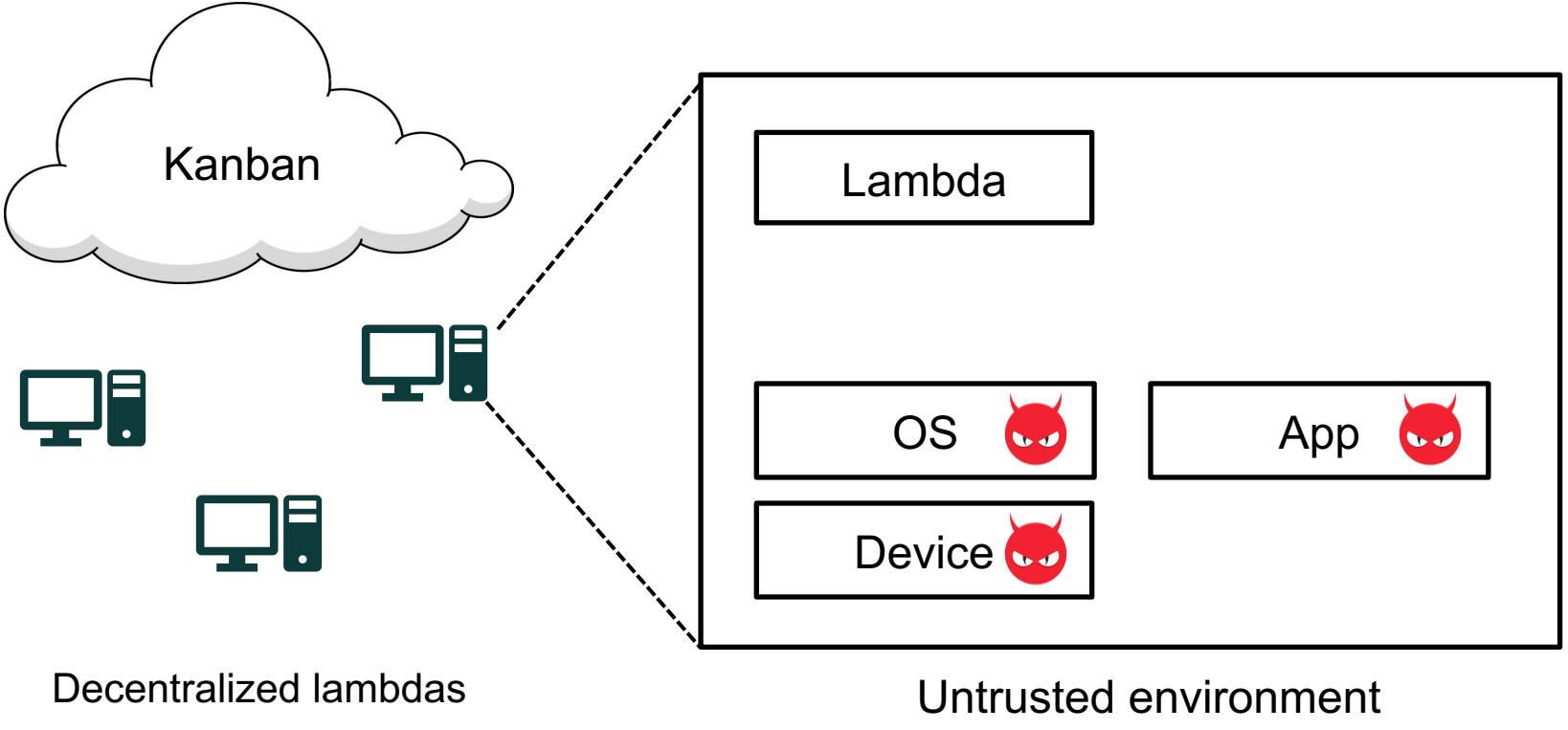
Undeniable + short hash

TEE-based Lambda

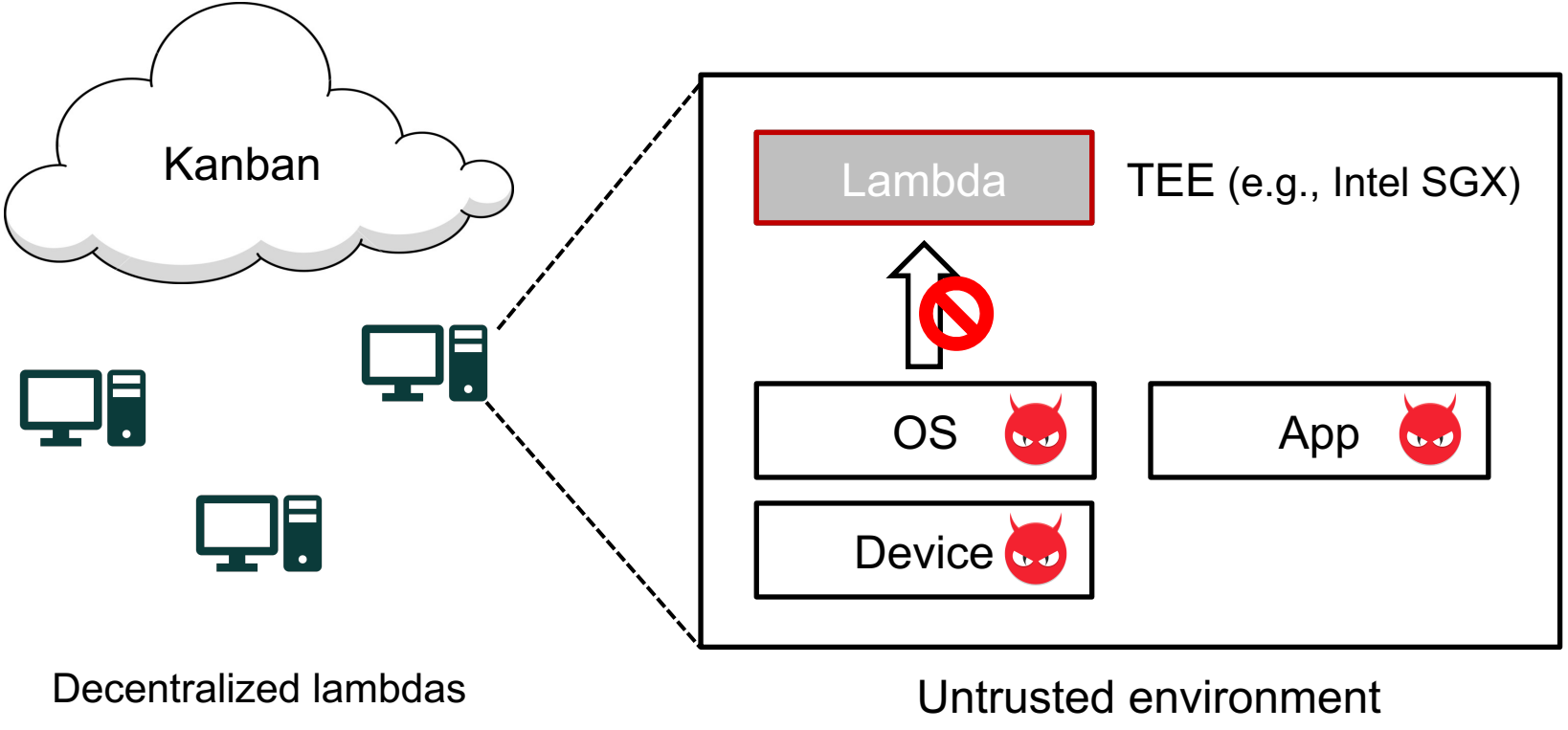


Decentralized lambdas

TEE-based Lambda

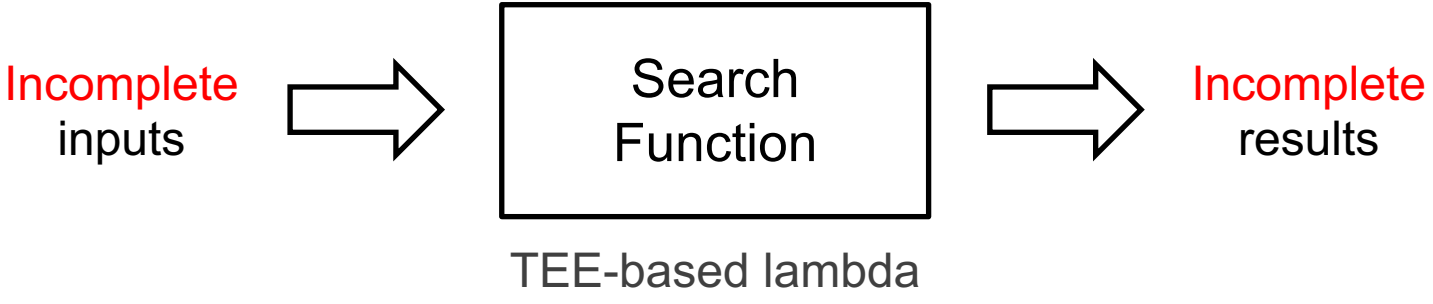


TEE-based Lambda



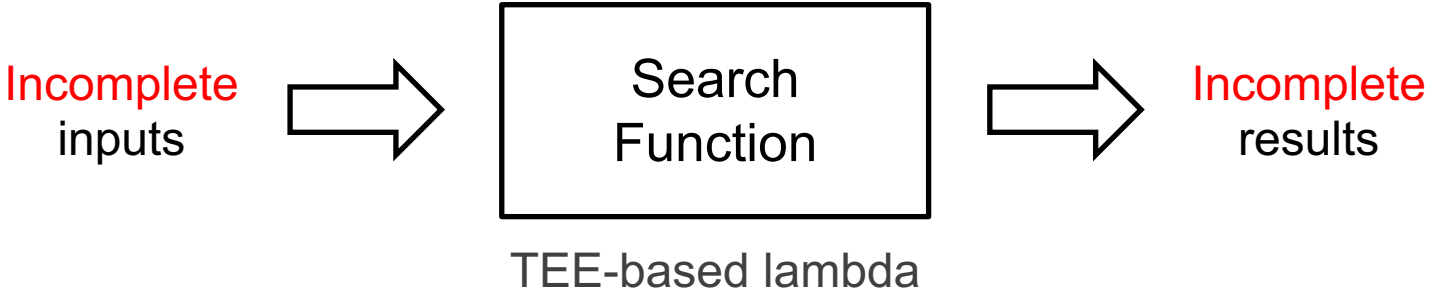
Why TEE is not enough?

- **Execution integrity ≠ Data integrity**



Why TEE is not enough?

- **Execution integrity ≠ Data integrity**



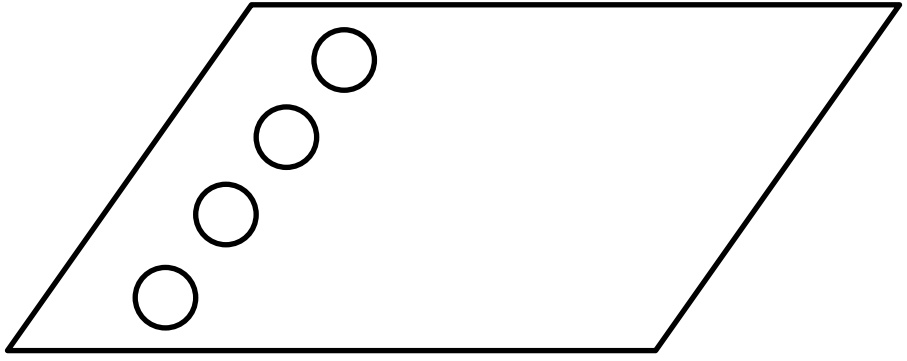
- **Put verification code into lambdas?**
  - Lambdas are ephemeral (verification time >> execution time)
  - Lambdas are scattered (lacking a global view)

Decouple verification from computation

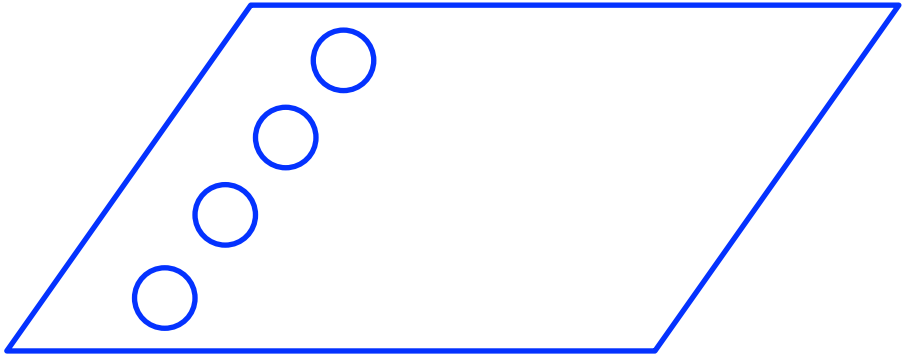
Computation Plane

Verification Plane

Decouple verification from computation

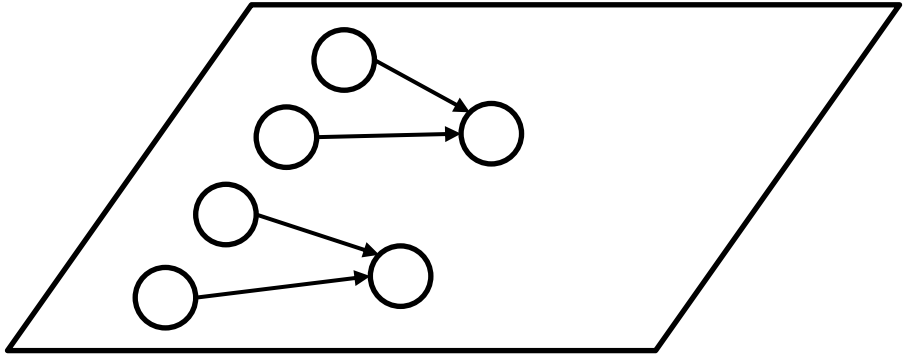


Computation Plane

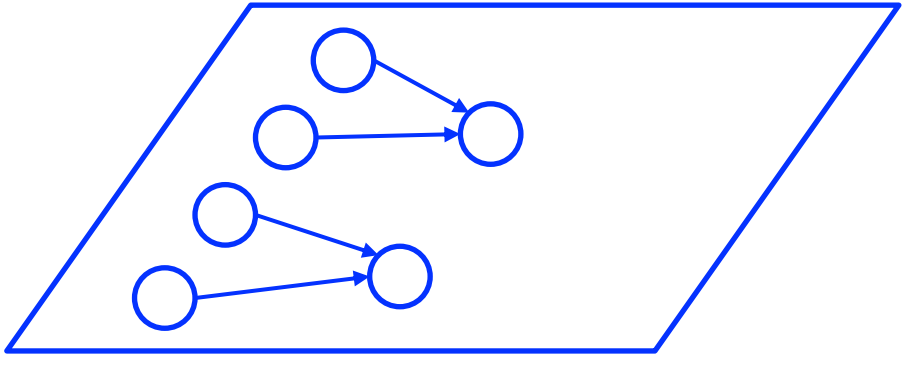


Verification Plane

Decouple verification from computation

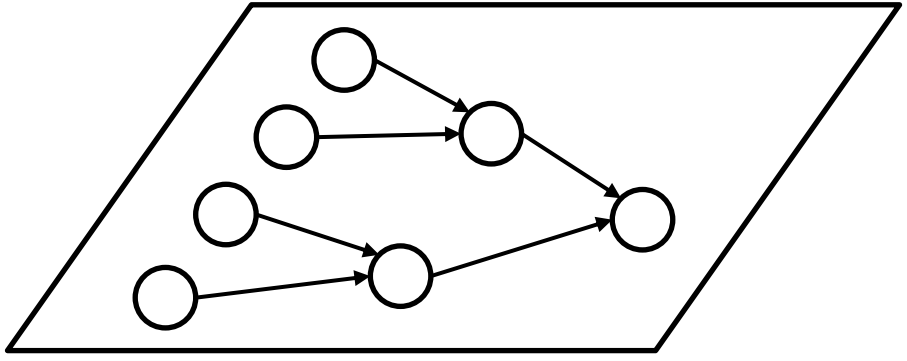


Computation Plane

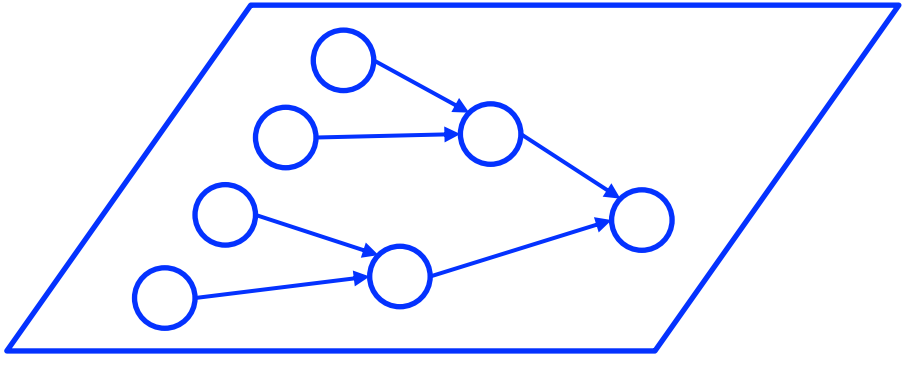


Verification Plane

Decouple verification from computation



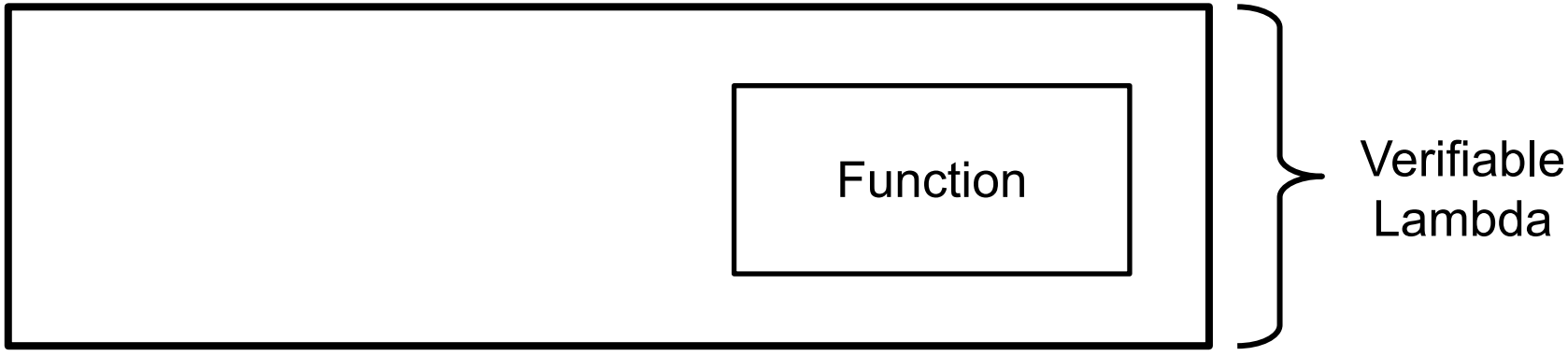
Computation Plane



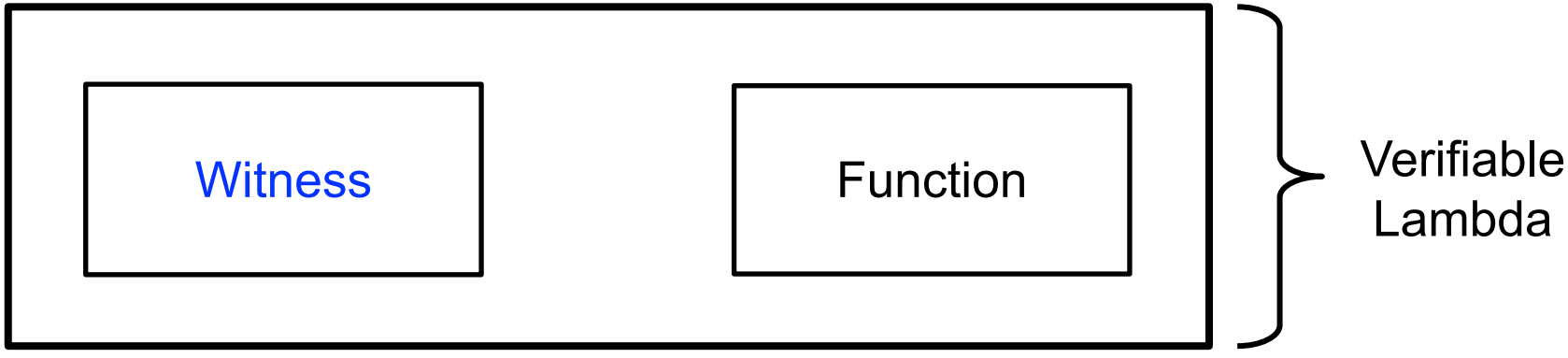
Verification Plane

Verifiable Lambda and Witness

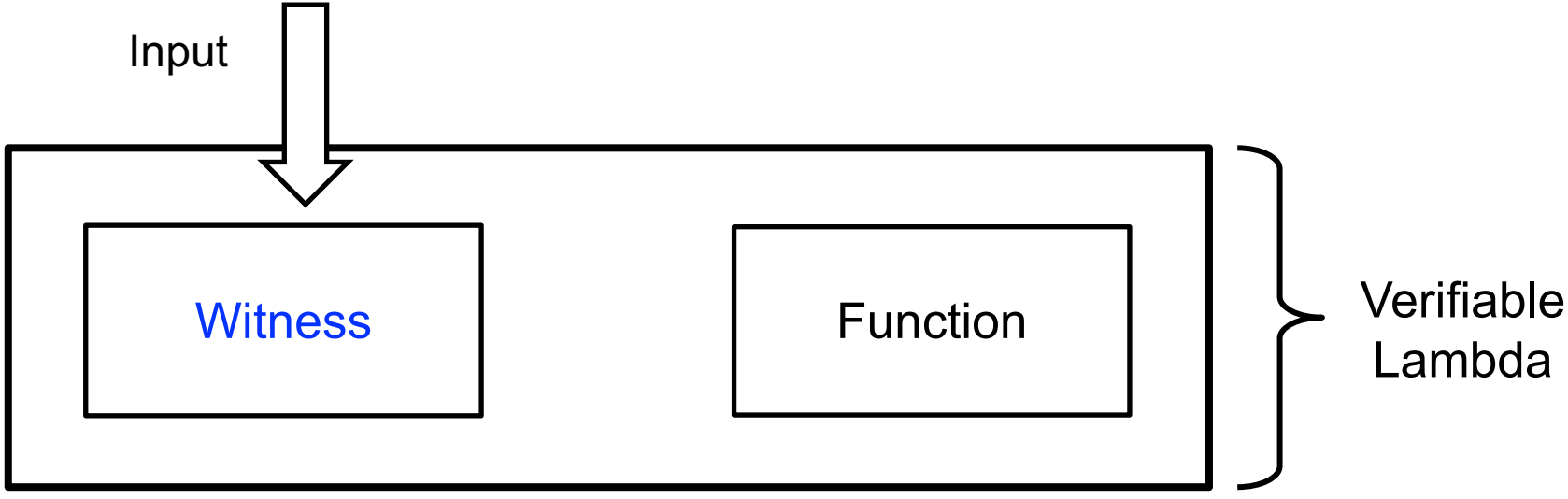
Verifiable Lambda and Witness



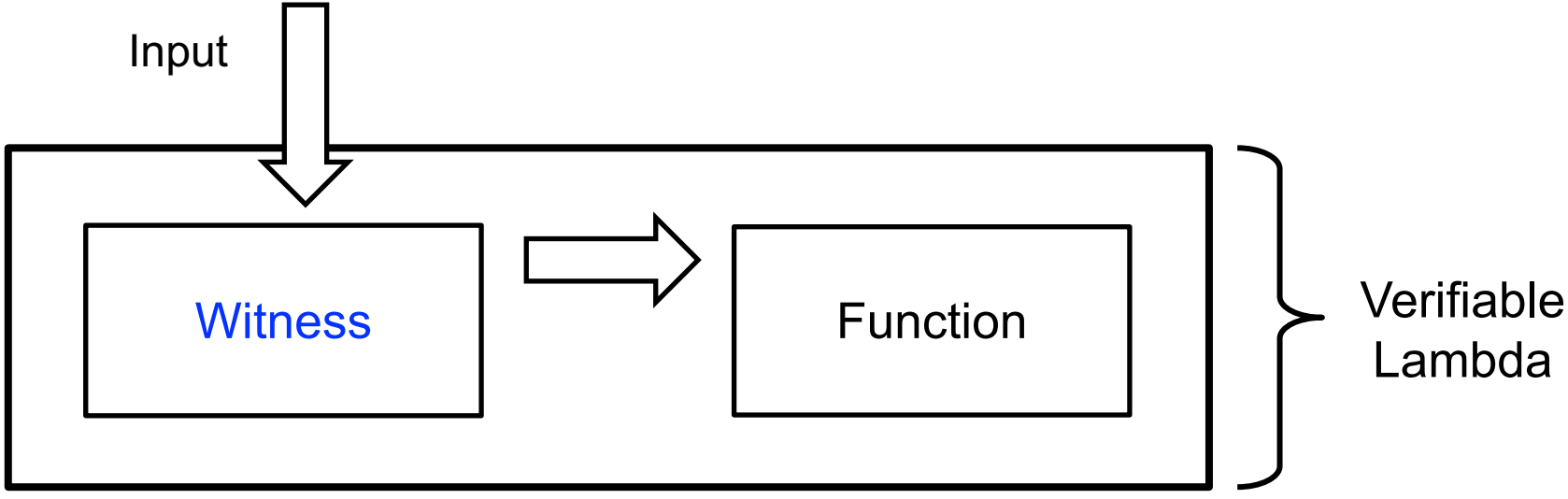
Verifiable Lambda and Witness



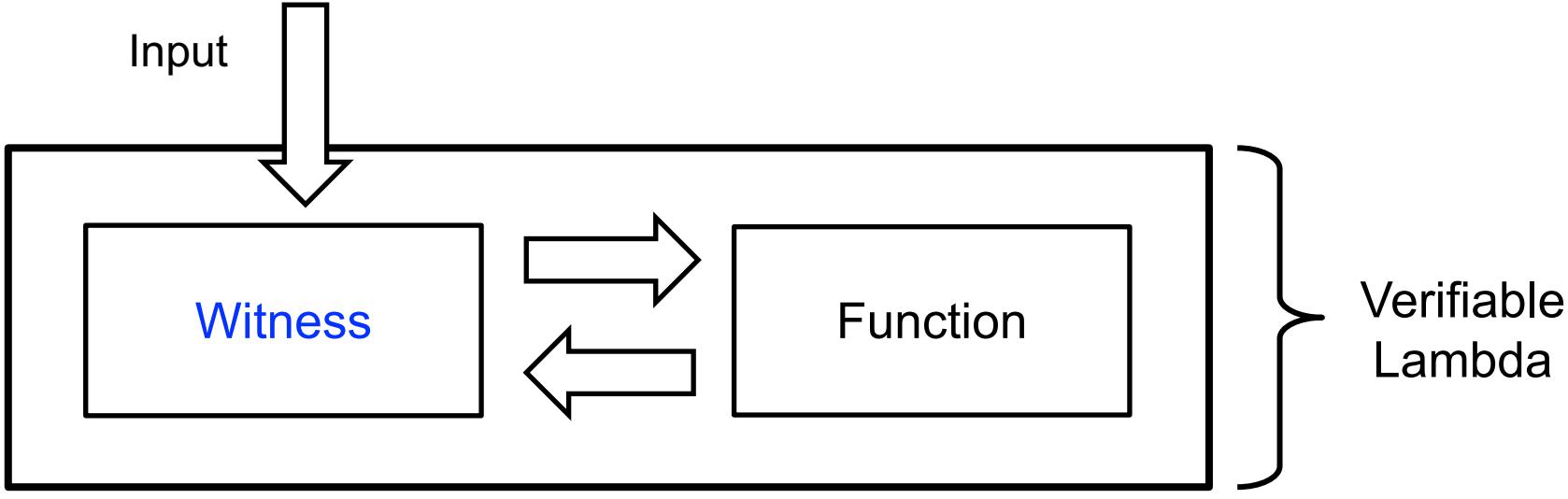
Verifiable Lambda and Witness



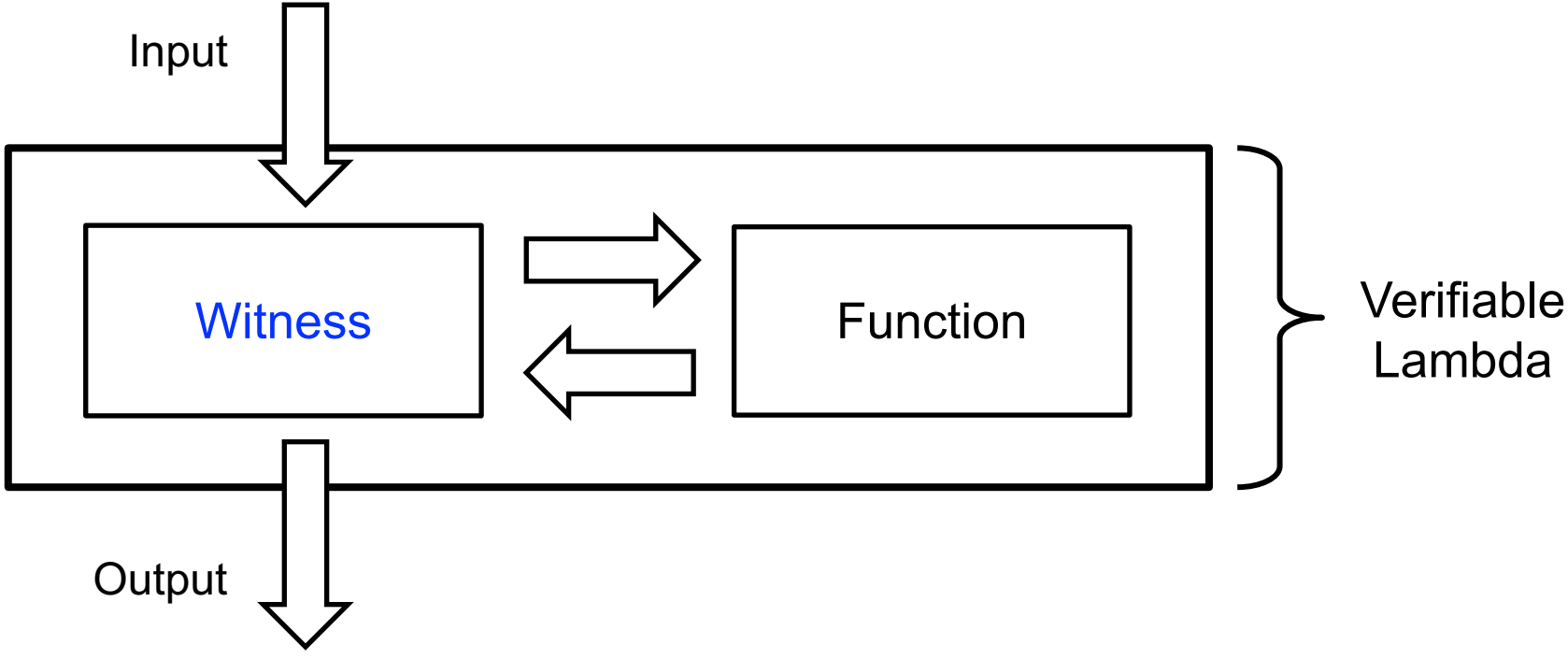
Verifiable Lambda and Witness



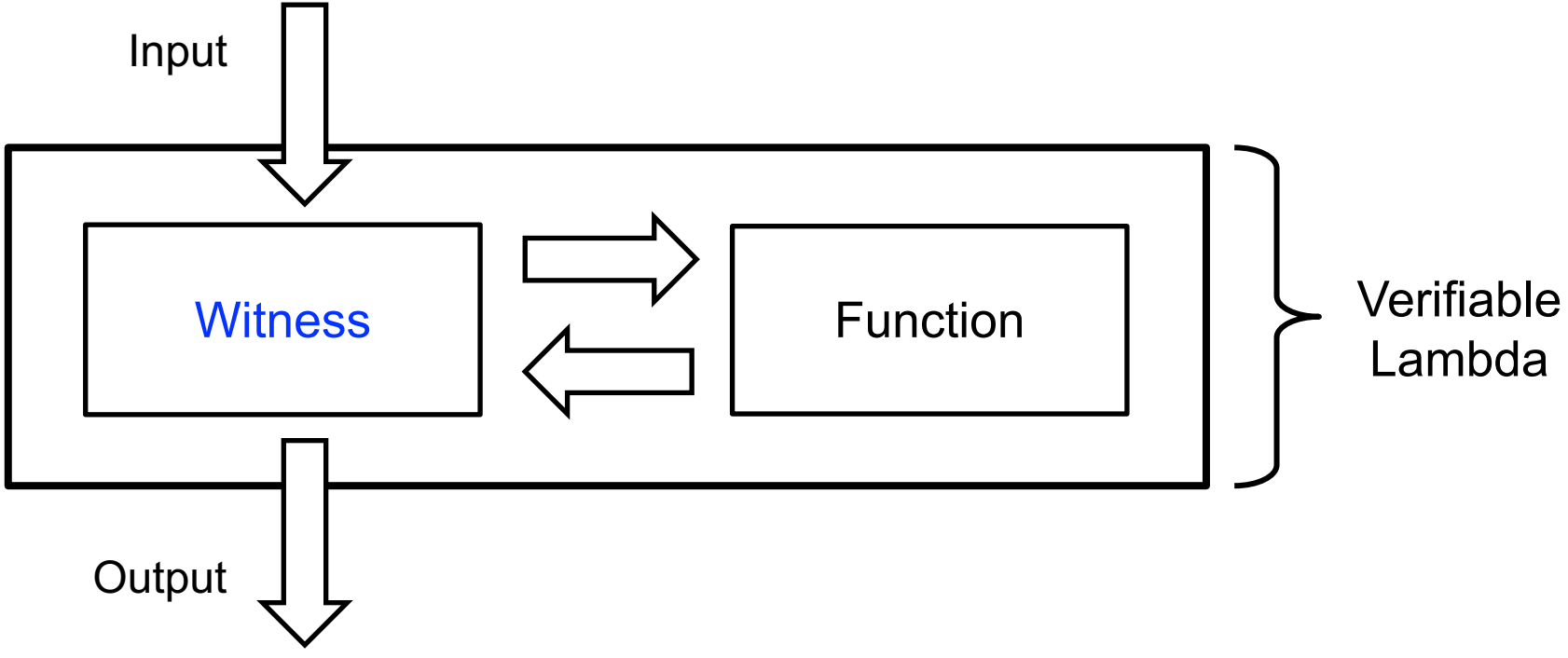
Verifiable Lambda and Witness



Verifiable Lambda and Witness

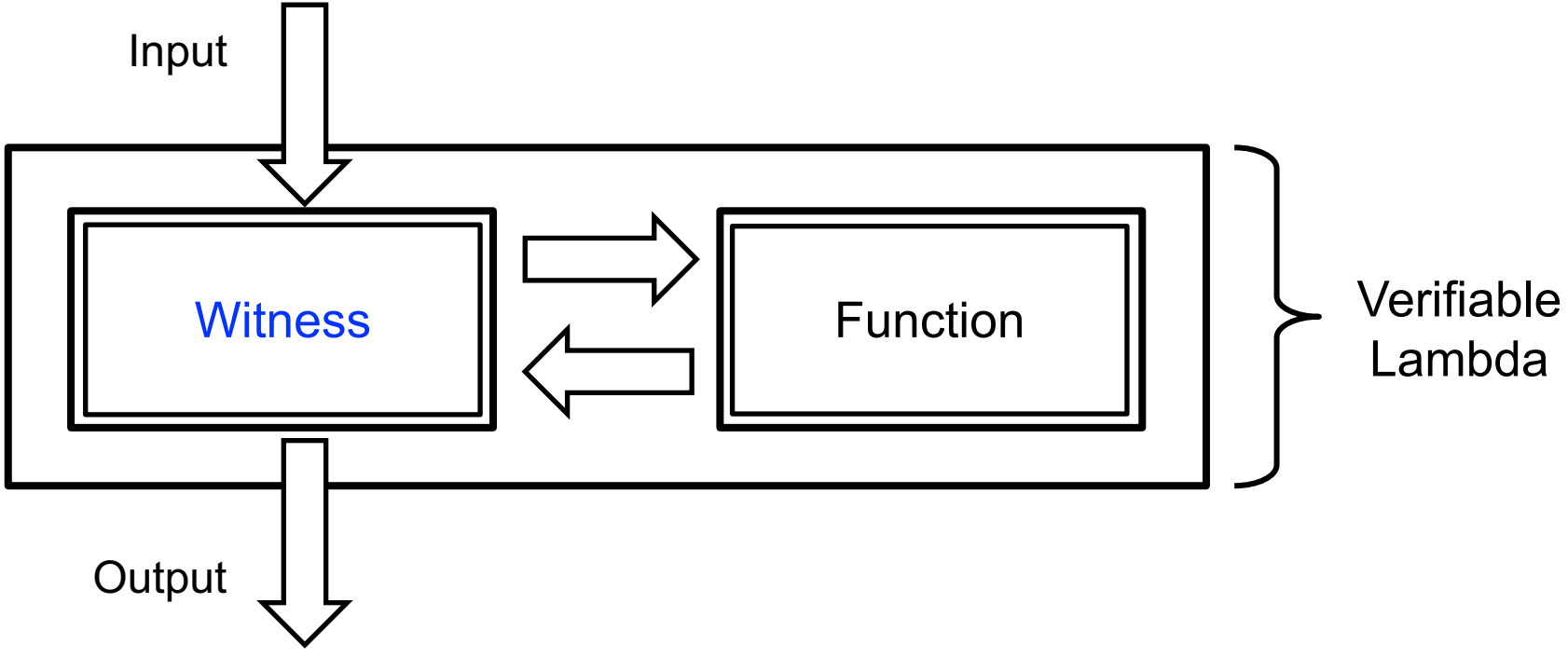


Verifiable Lambda and Witness



<[Hash(Input)], Hash(Function), Hash(Output)>$_{signed}$

Verifiable Lambda and Witness



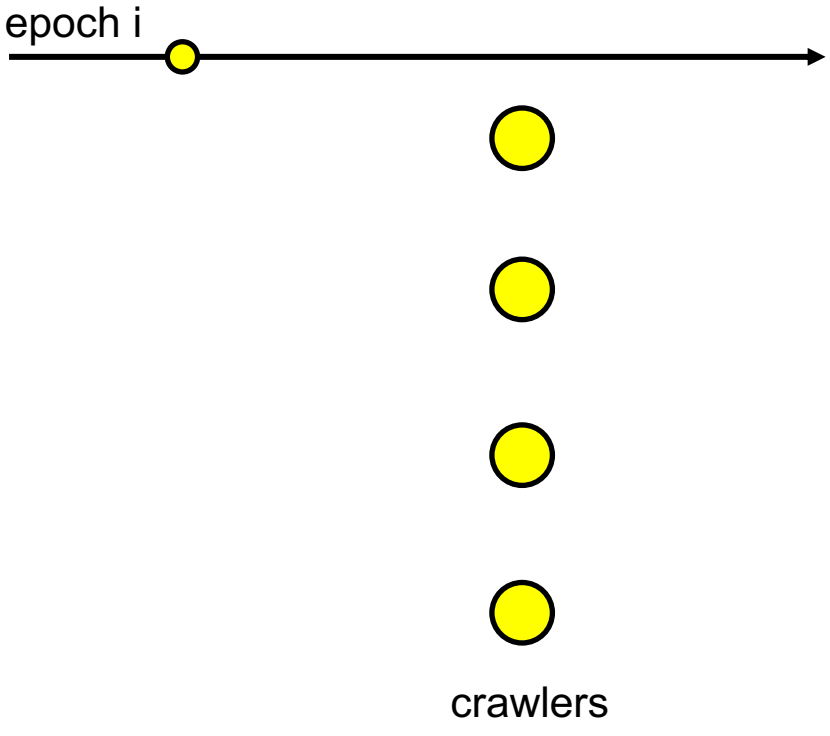
<[Hash(Input)], Hash(Function), Hash(Output)>$_{signed}$

Providing Verifiable Search

- **Combing Lambda witnesses and Kanban epochs**

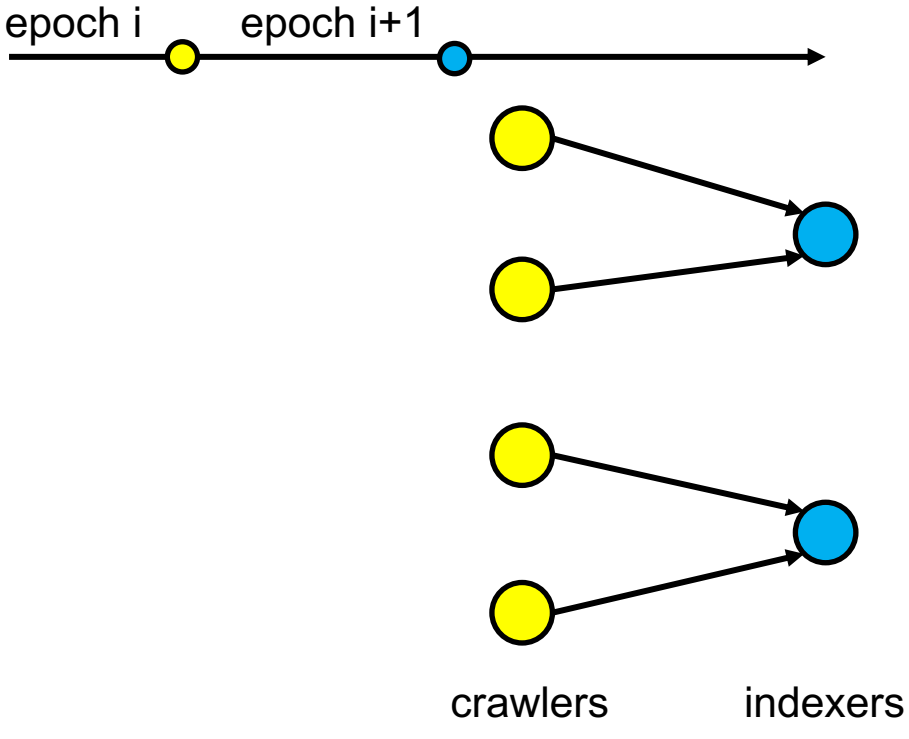
Providing Verifiable Search

- **Combing Lambda witnesses and Kanban epochs**



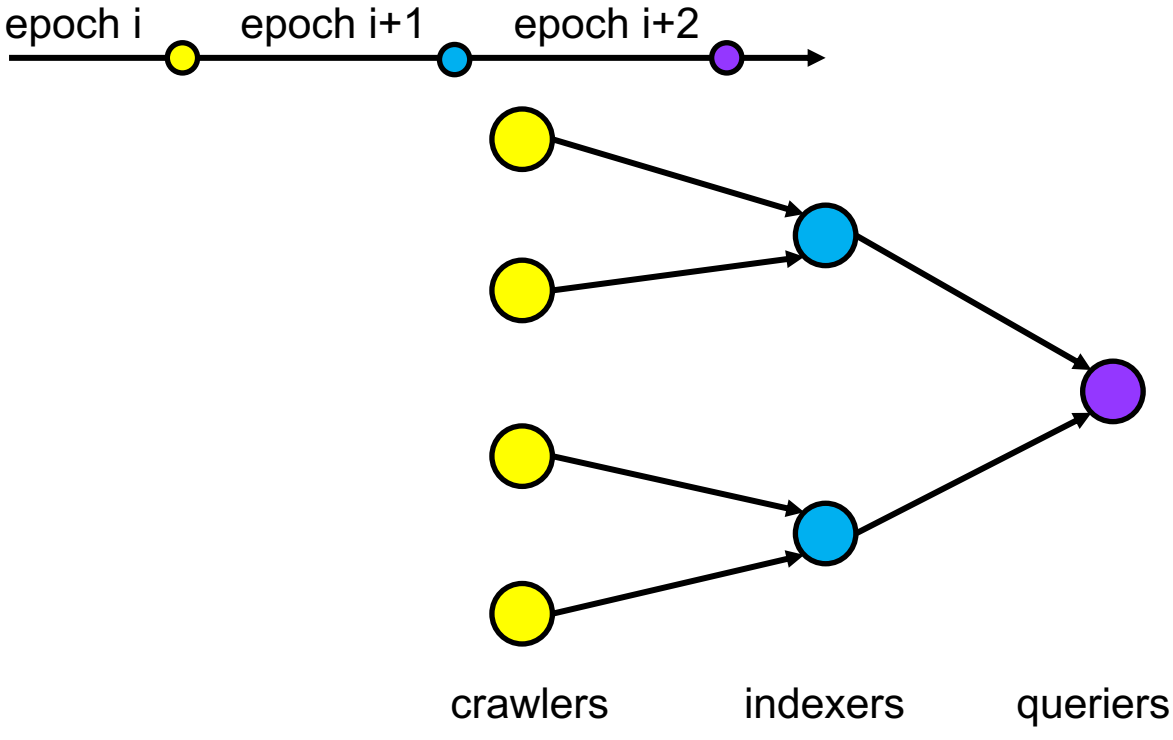
Providing Verifiable Search

- **Combing Lambda witnesses and Kanban epochs**



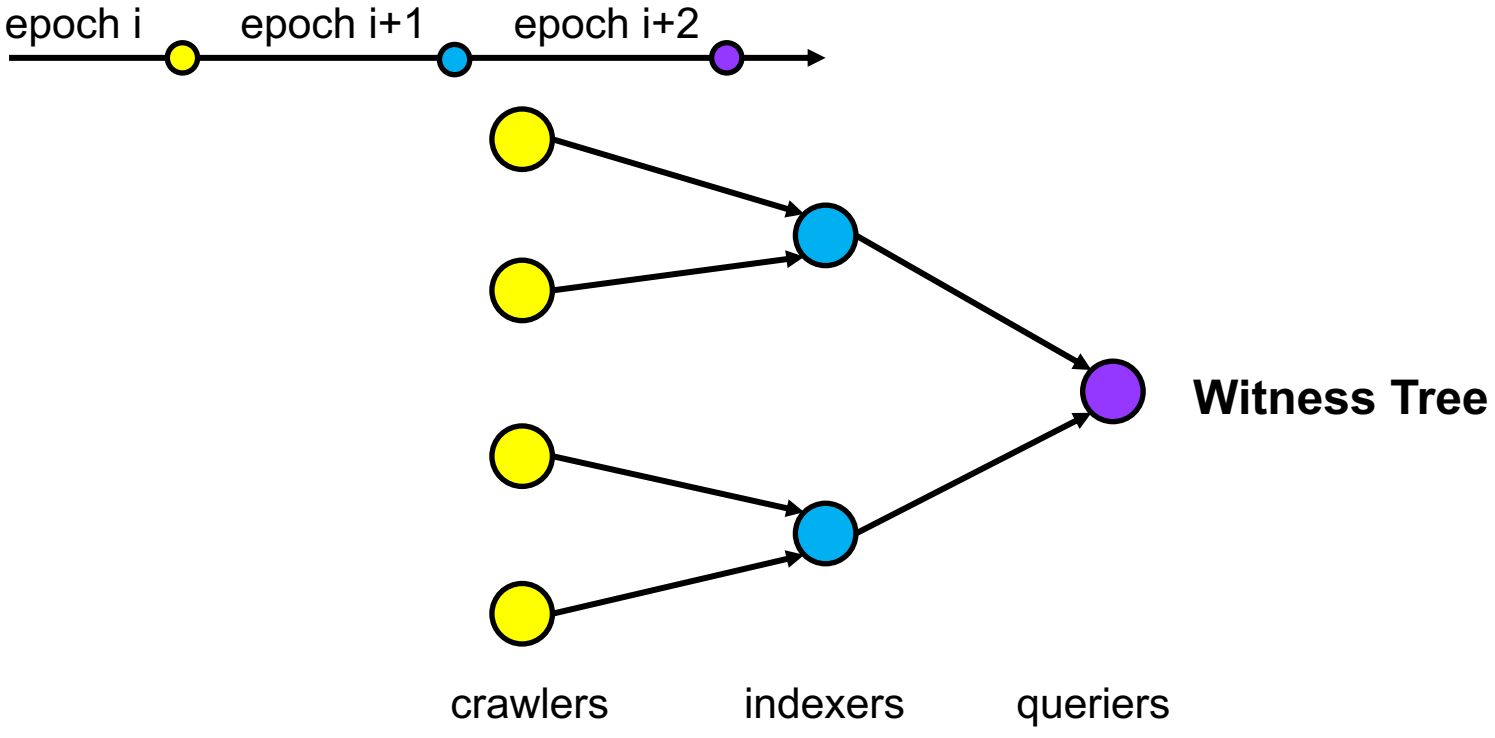
Providing Verifiable Search

- **Combing Lambda witnesses and Kanban epochs**



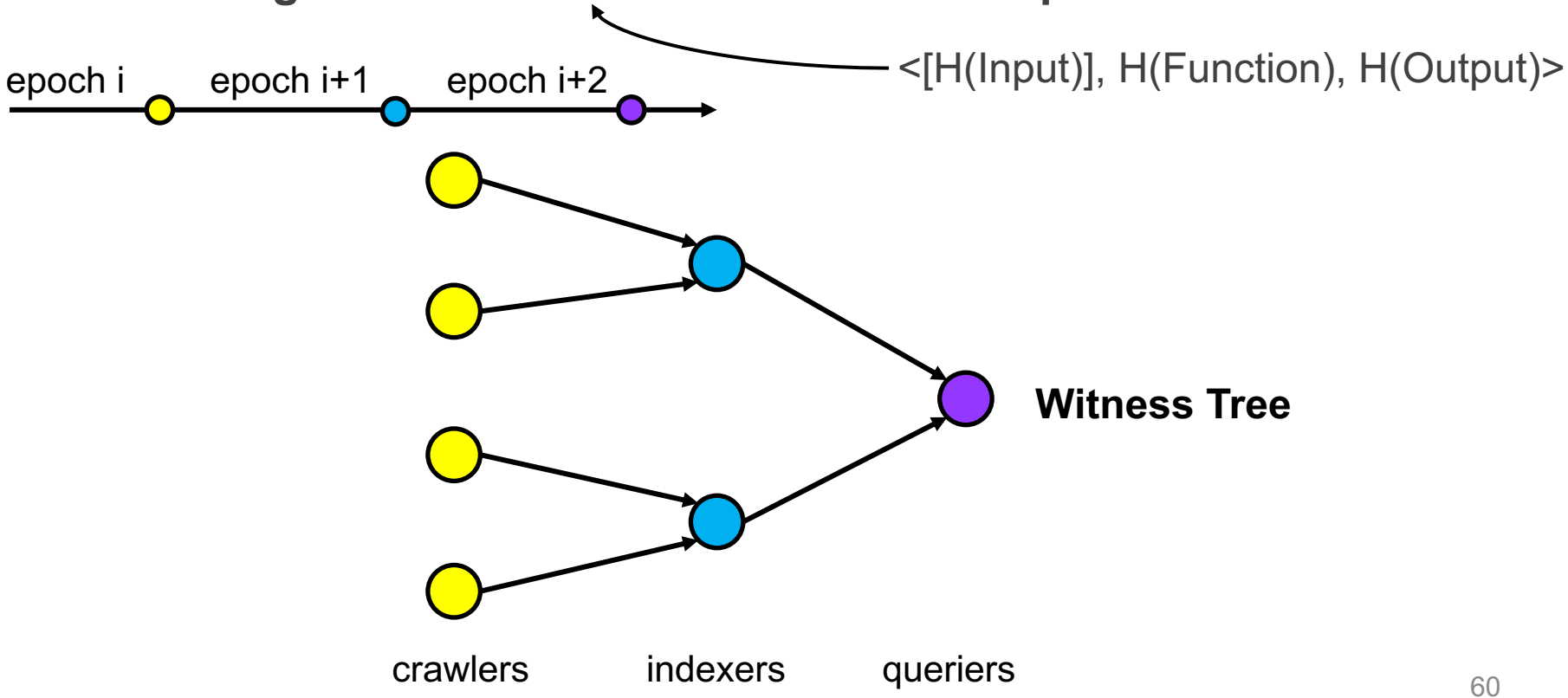
Providing Verifiable Search

- **Combing Lambda witnesses and Kanban epochs**



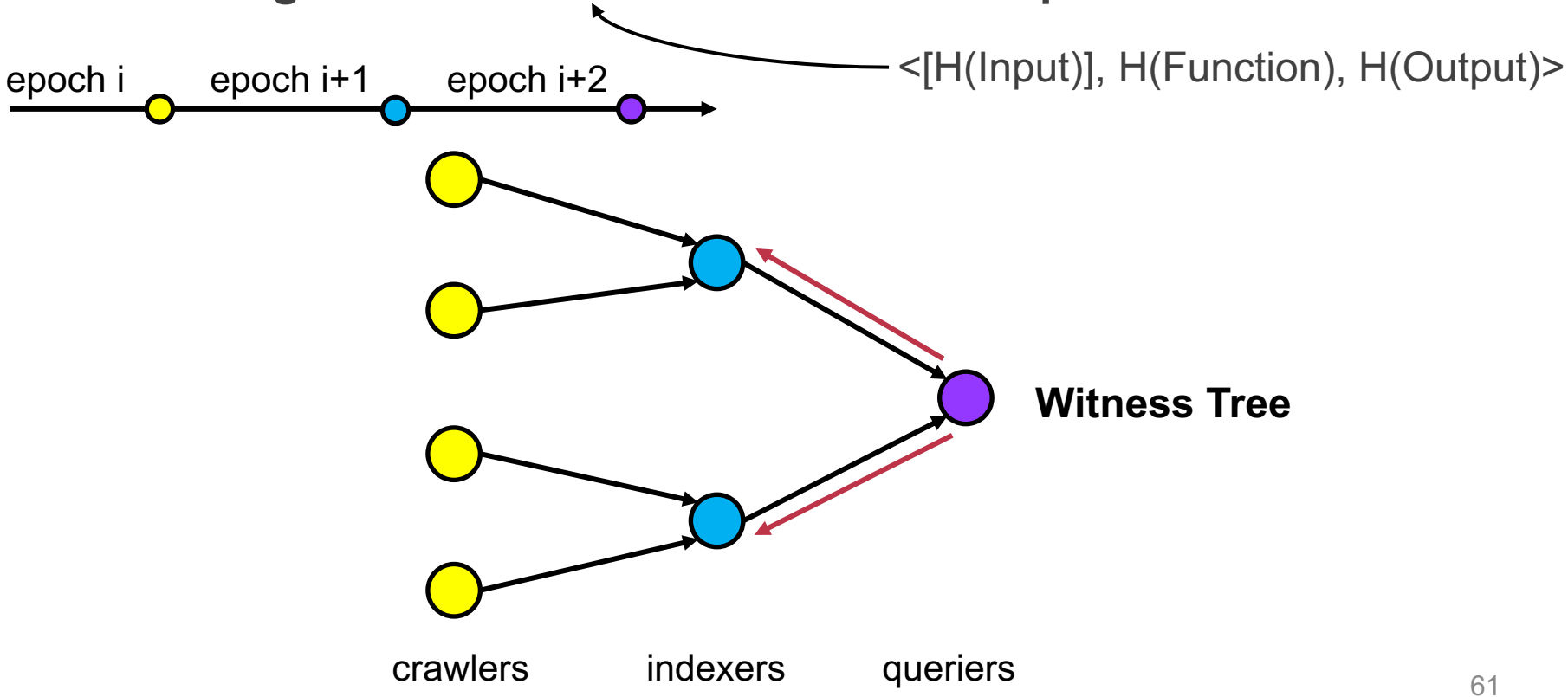
Providing Verifiable Search

- **Combing Lambda witnesses and Kanban epochs**



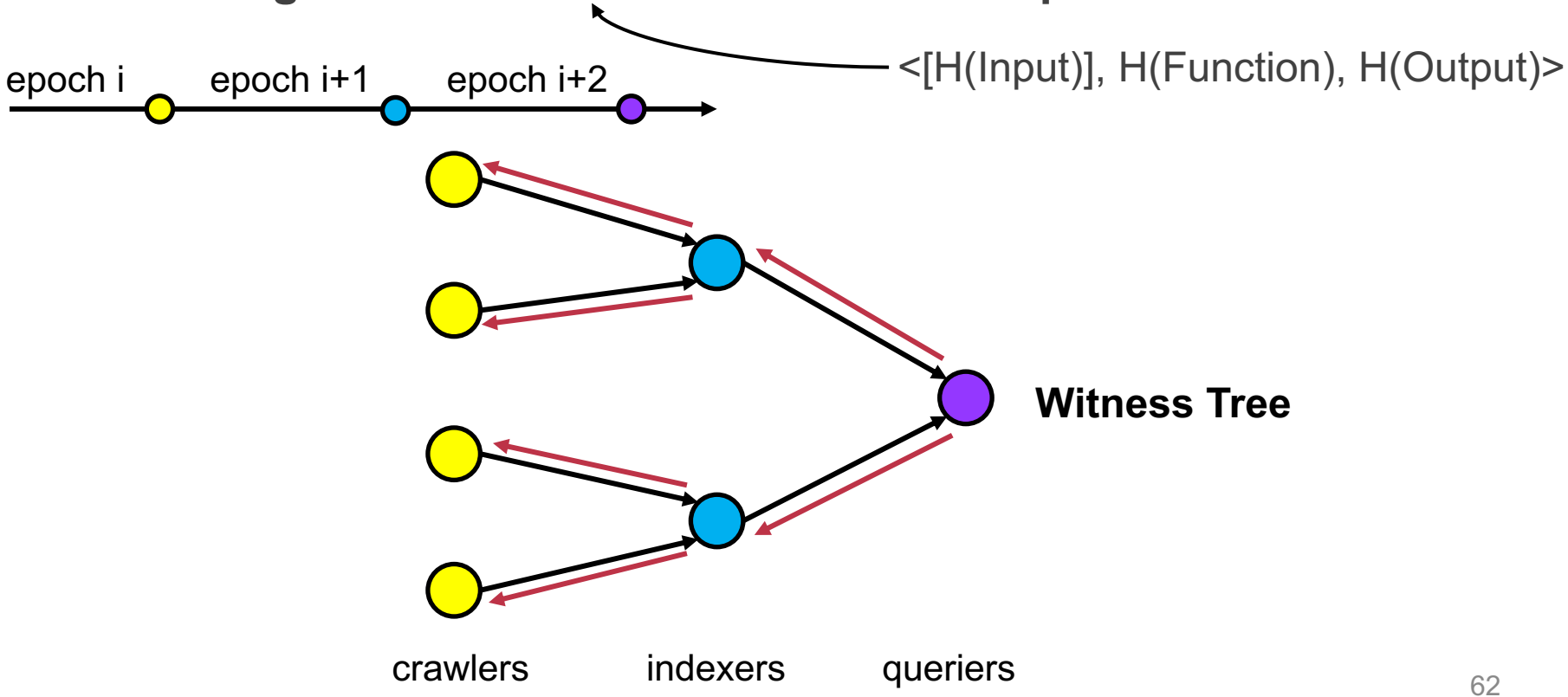
Providing Verifiable Search

- **Combing Lambda witnesses and Kanban epochs**



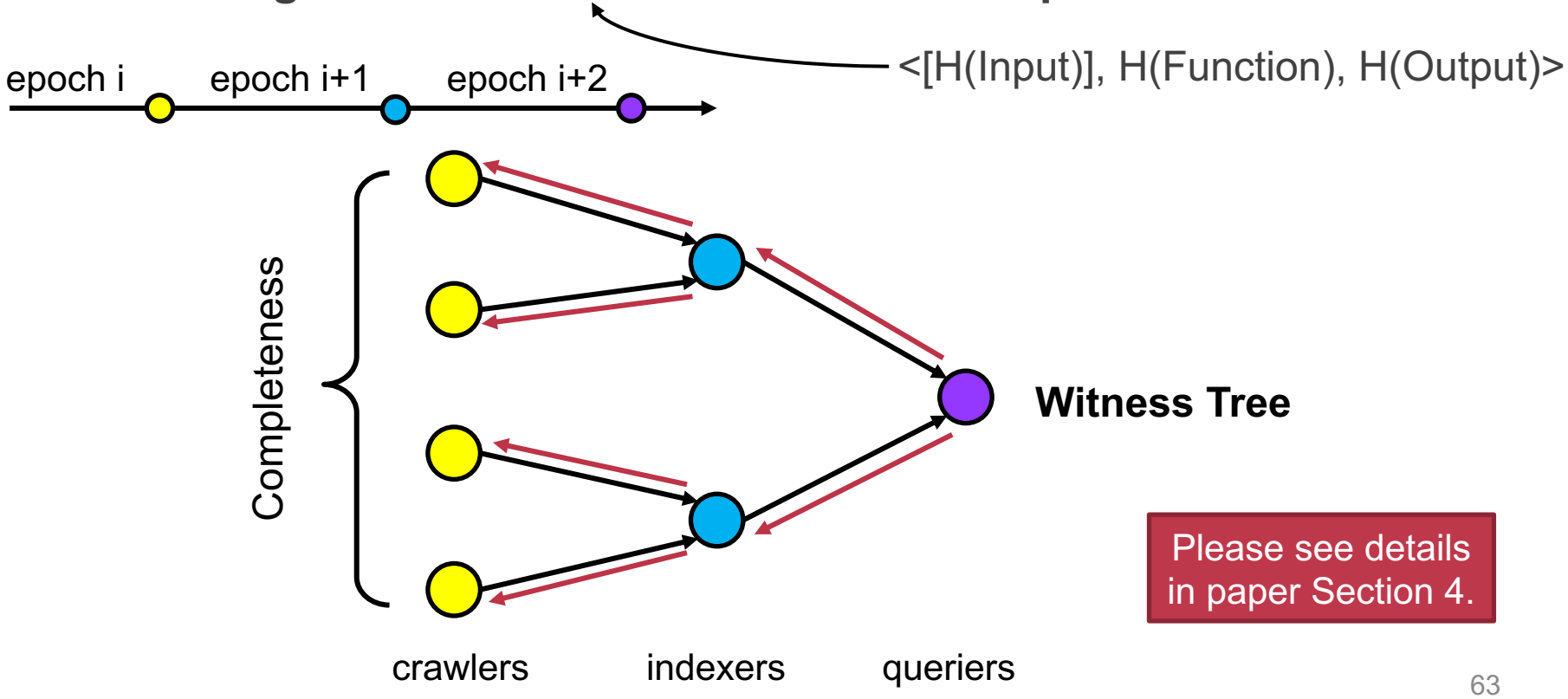
Providing Verifiable Search

- **Combing Lambda witnesses and Kanban epochs**



Providing Verifiable Search

- **Combing Lambda witnesses and Kanban epochs**



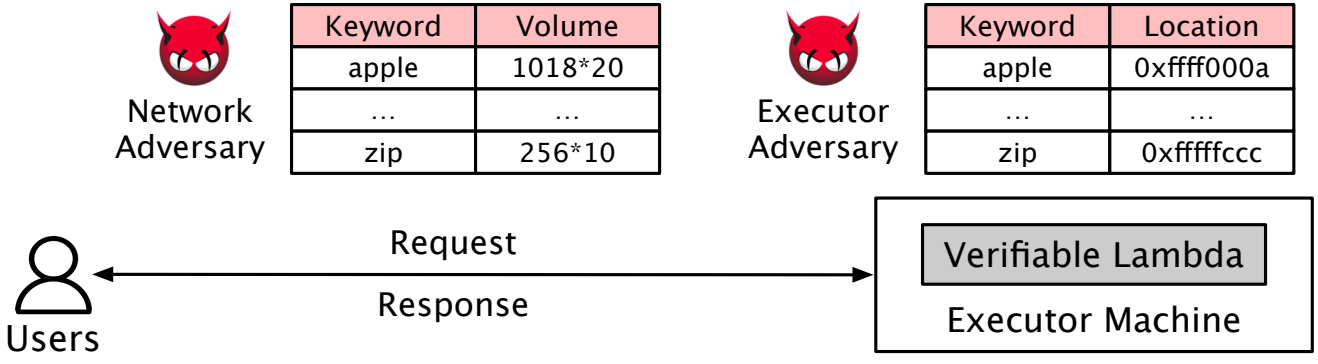
Please see details in paper Section 4.

Outline

- Problem Statement
- Design: decentralized search
- Design: verifiable search
- **Design: private search**
- Evaluation

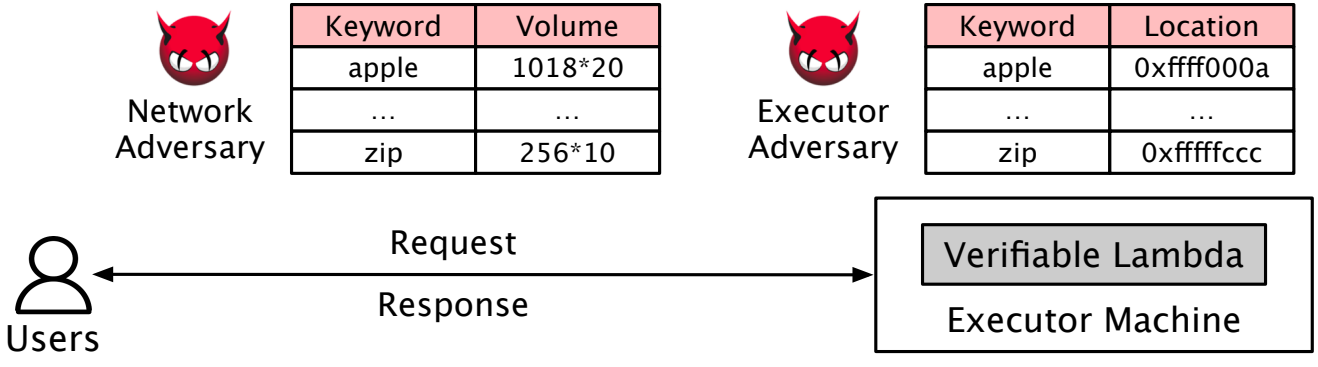
Private Search

- **Two threats**
  - Network adversary: infer search keywords via message lengths
  - Executor adversary: infer keywords via memory access pattern



Private Search

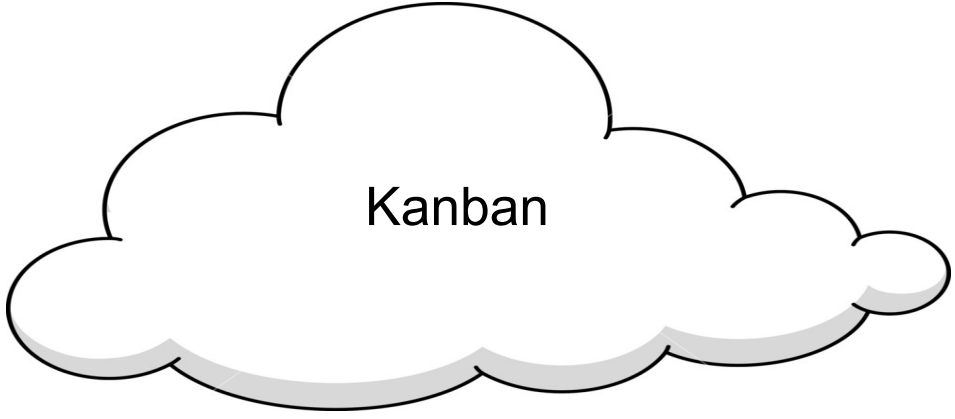
- **Two threats**
  - Network adversary: infer search keywords via message lengths
  - Executor adversary: infer keywords via memory access pattern



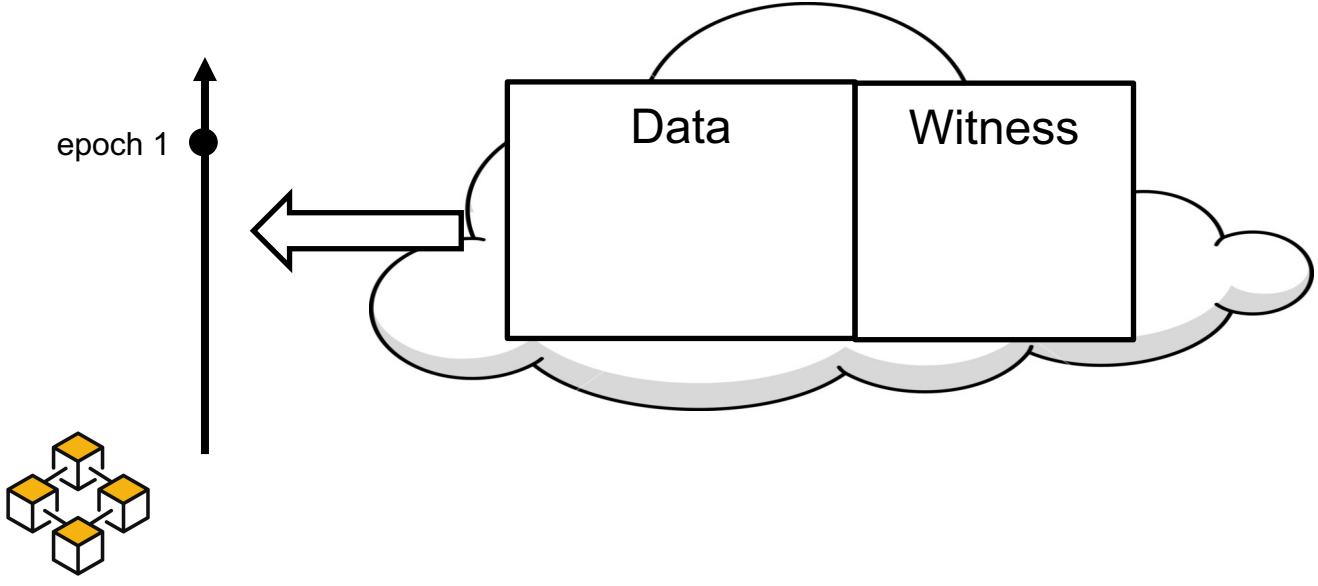
- **Countermeasures**
  - Network adversary: equalizing all message lengths
  - Executor adversary: SGX + Circuit-ORAM

Putting all together

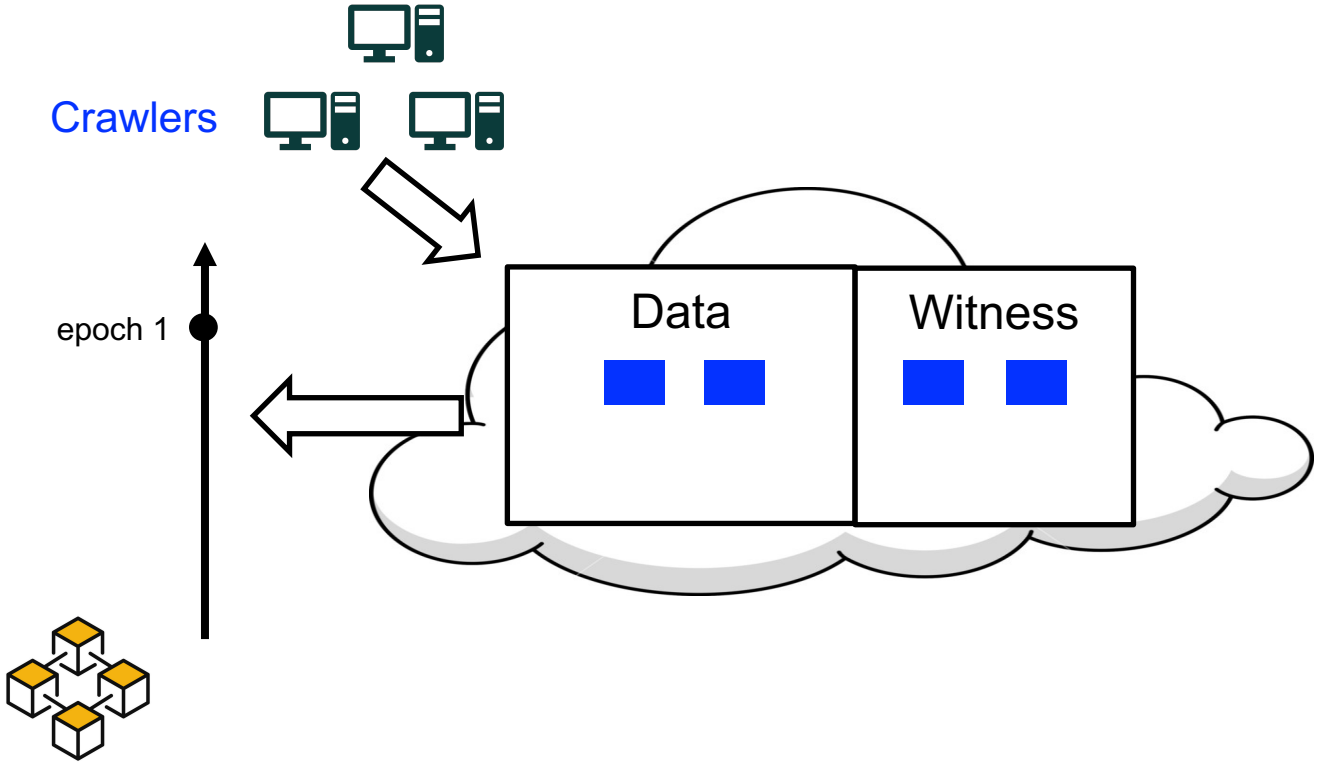
Putting all together



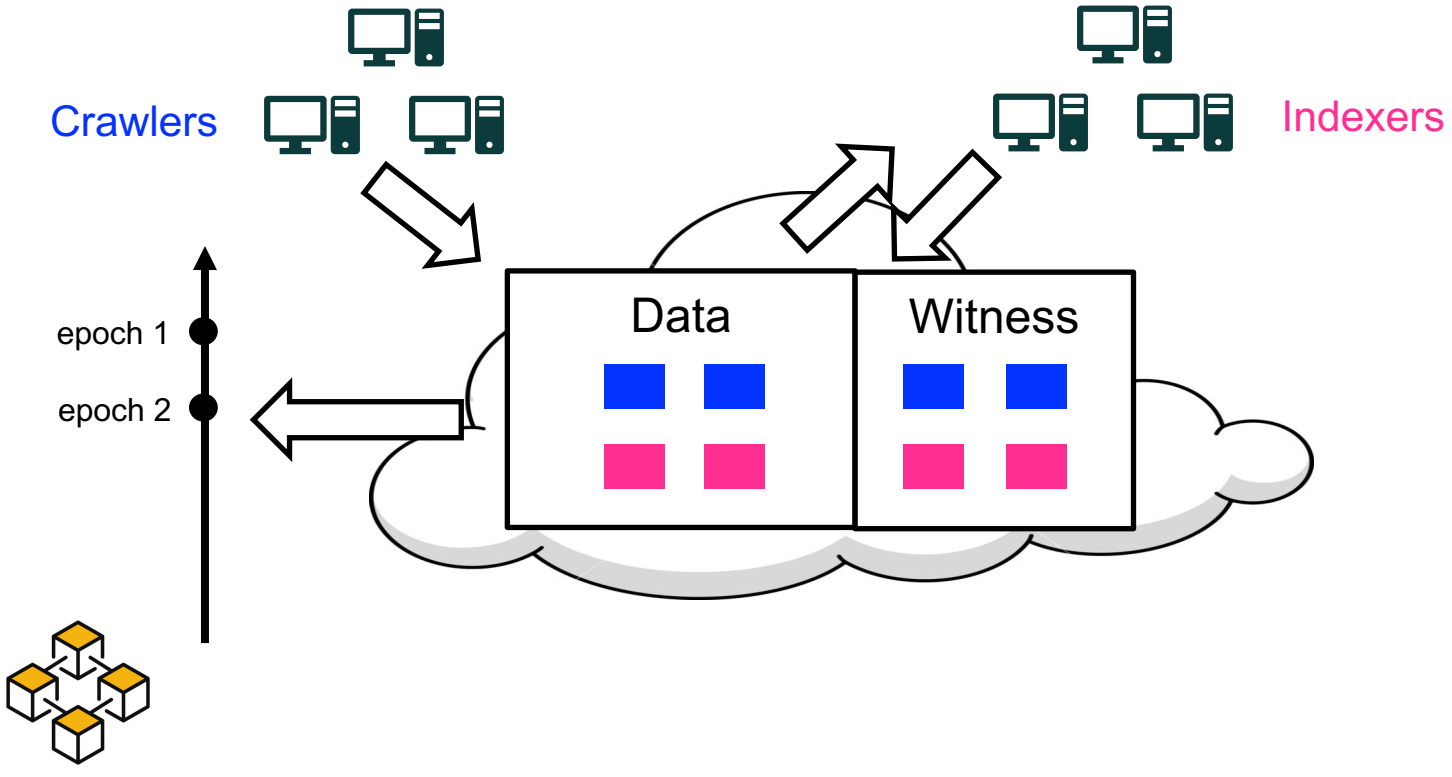
Putting all together



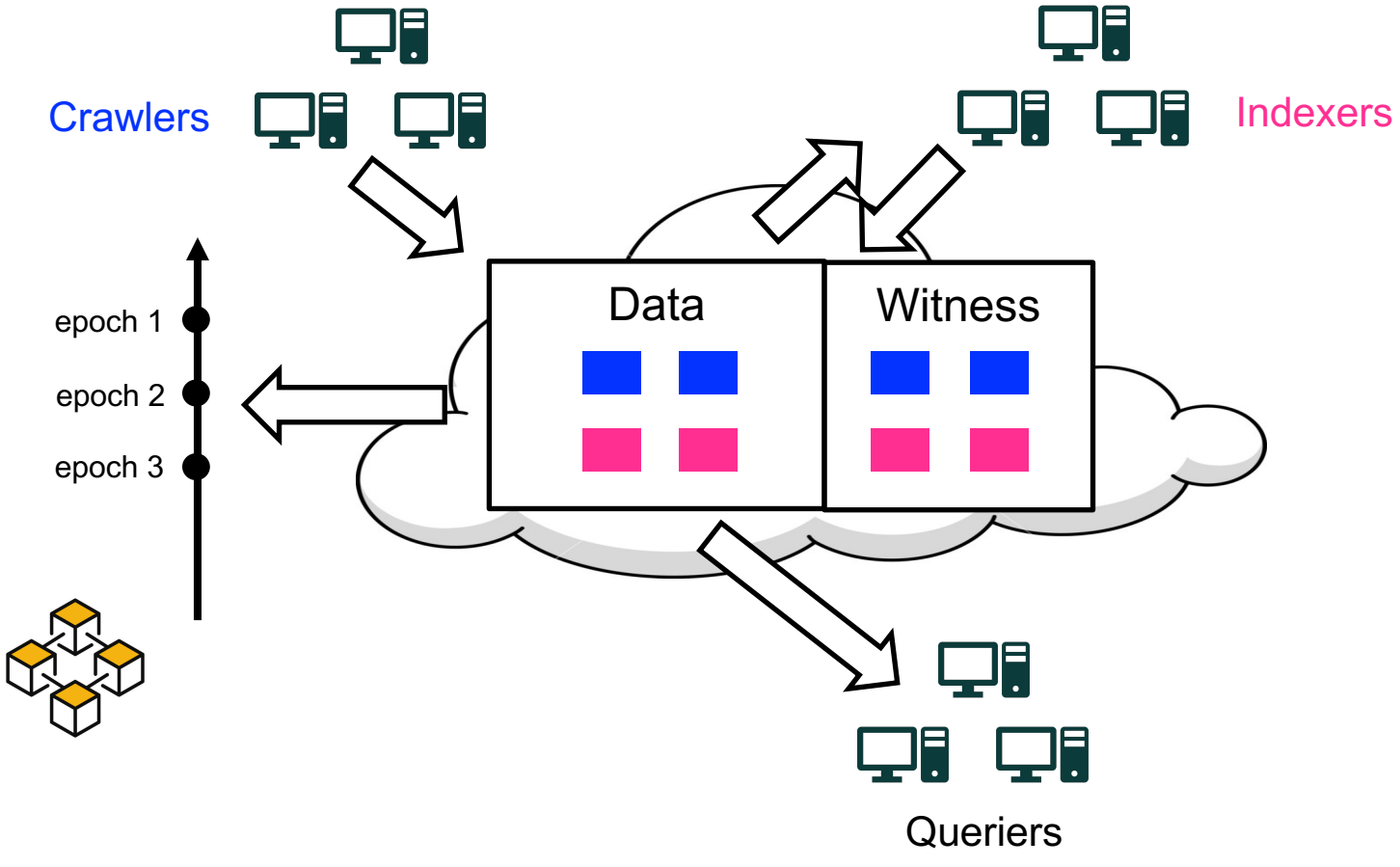
Putting all together



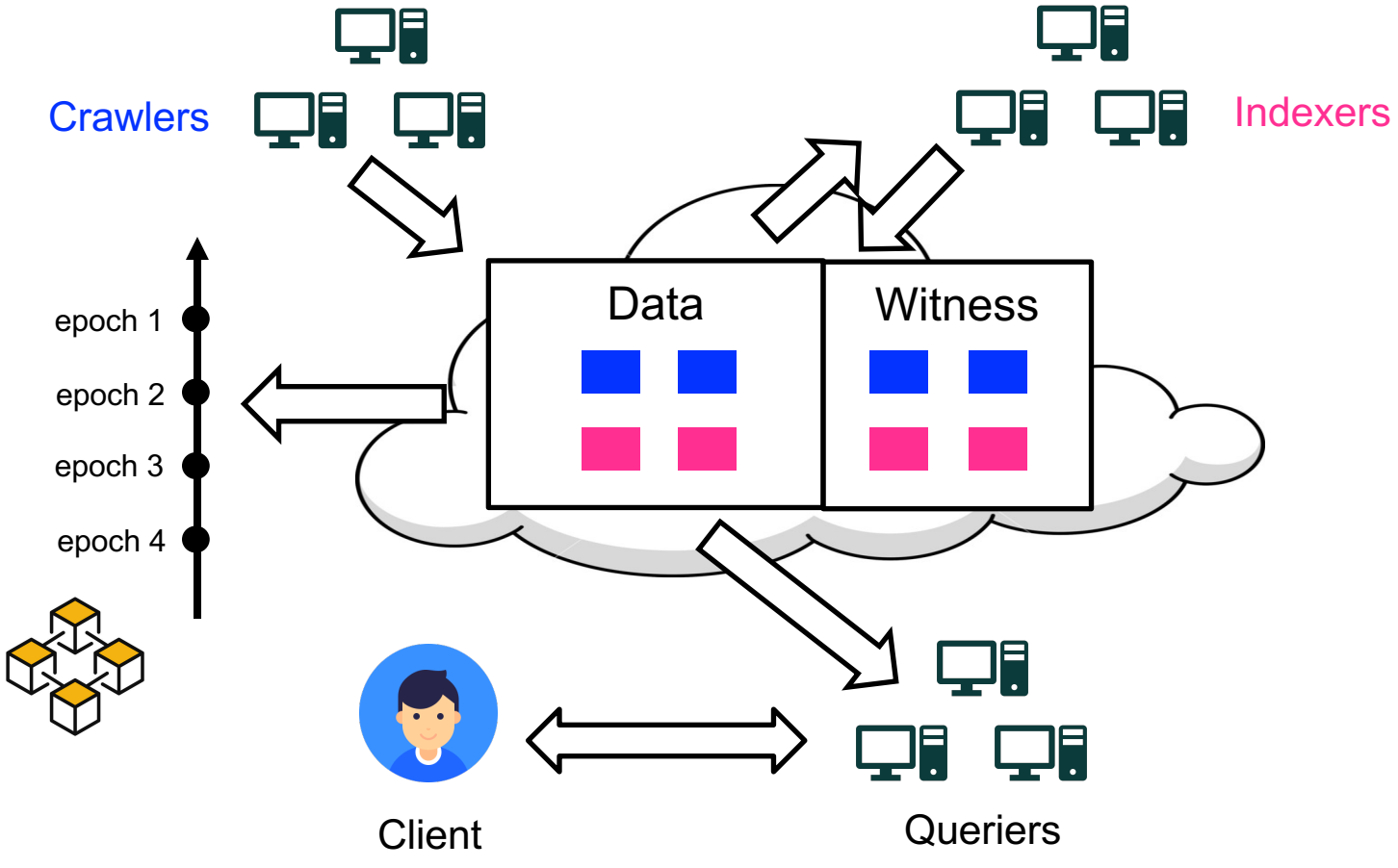
Putting all together



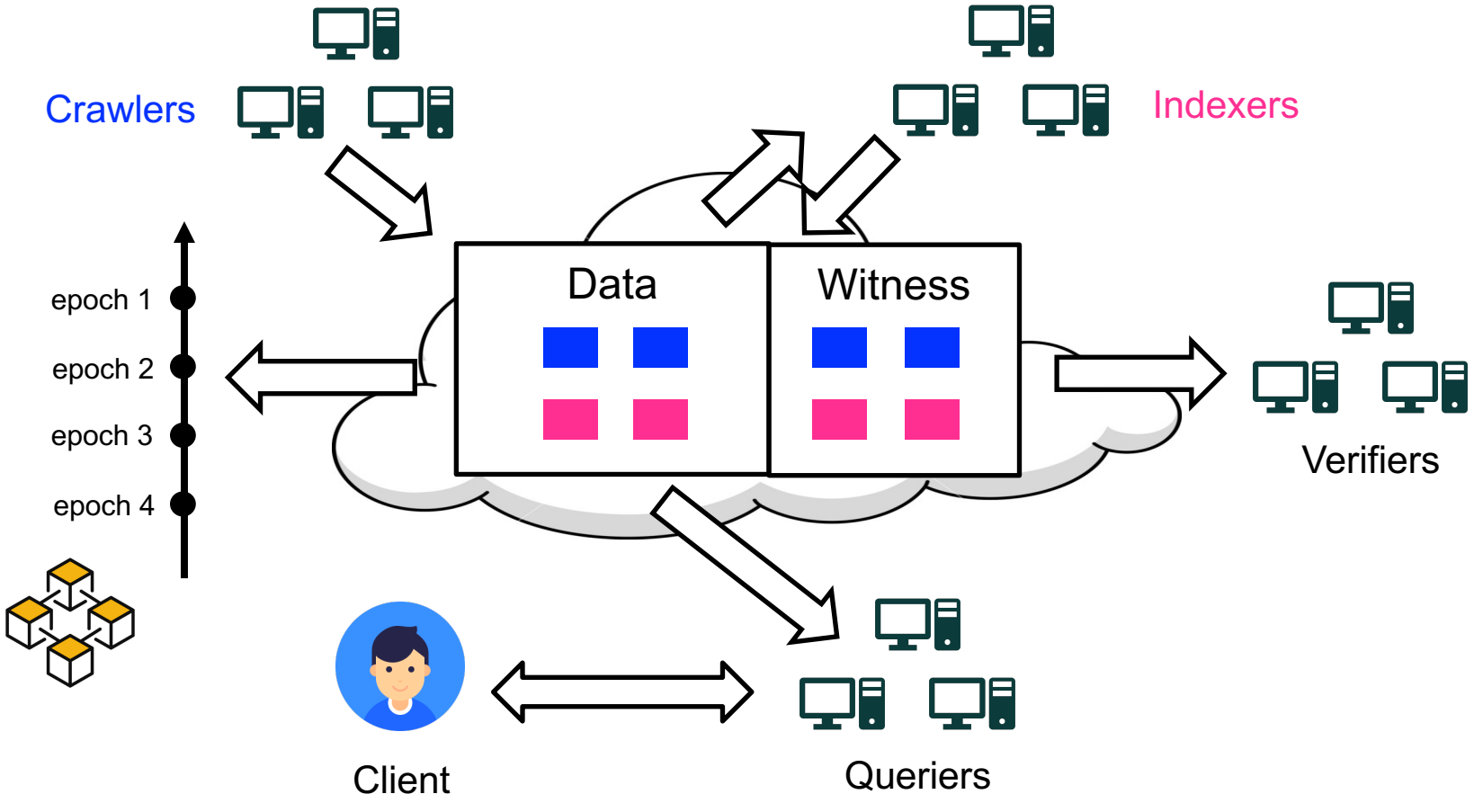
Putting all together



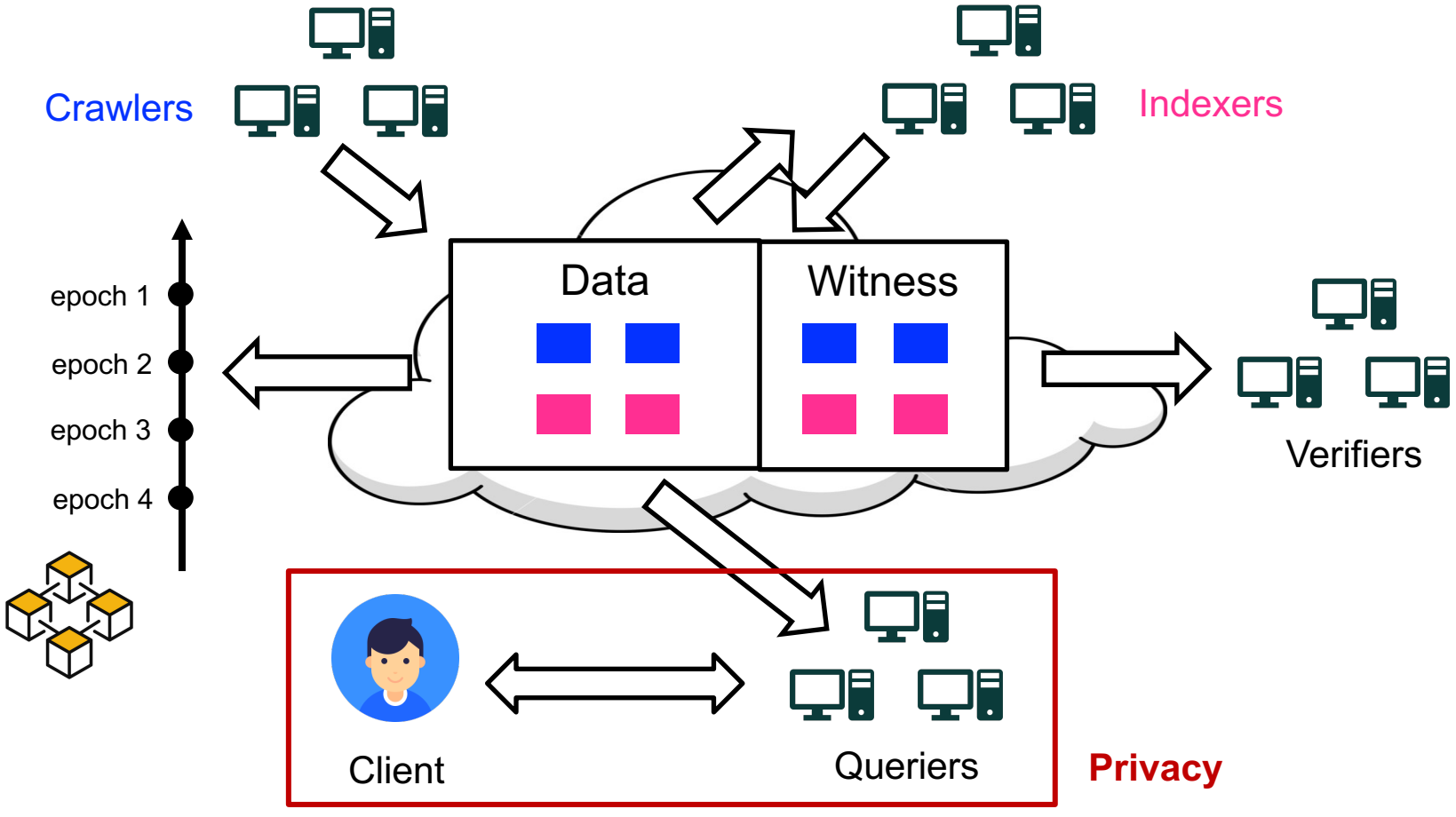
Putting all together



Putting all together



Putting all together



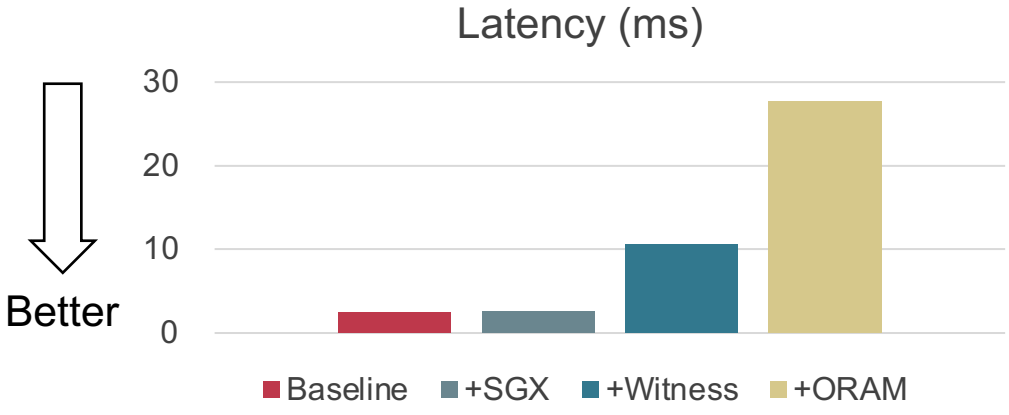
Outline

- Problem Statement
- Design: decentralized search
- Design: verifiable search
- Design: private search
- **Evaluation**

Evaluation Setup

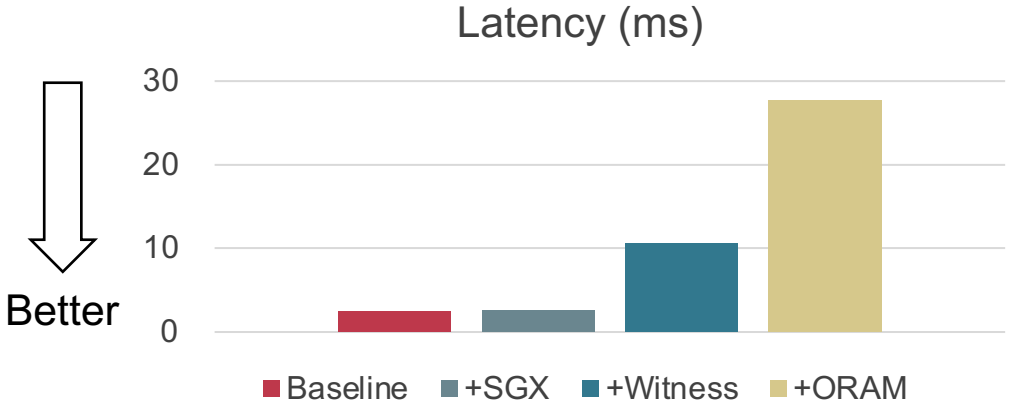
- **Datasets：**
  - Steemit: 234GB on-chain items
  - OpenBazaar: ~20K items per day
- **Lab testbed**
  - 9 SGX-enabled physical machines: at least 8-core, 8GB DRAM
- **AWS geo-distributed large-scale deployment**
  - 1300+ t2.medium EC2 instances: 2-core vCPU, 4GB DRAM

Microbenchmark

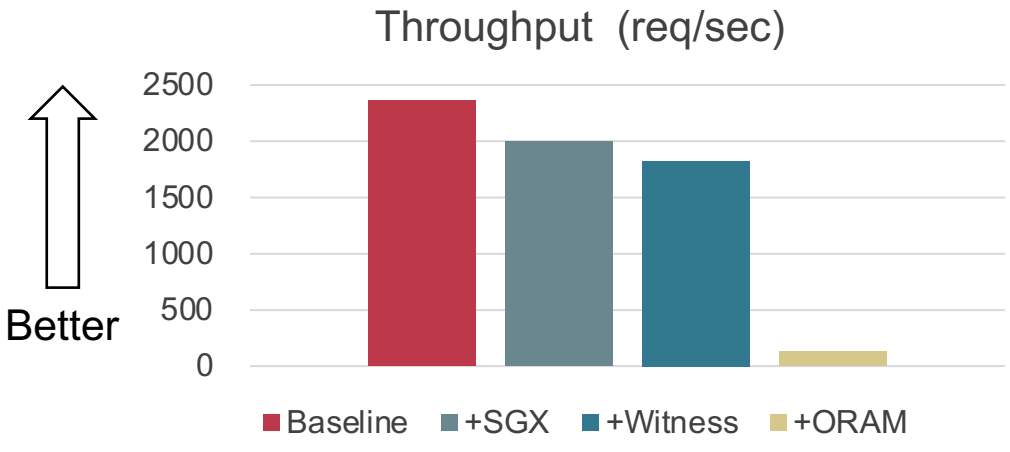


- **End-to-end latency**
  - < 50ms

Microbenchmark



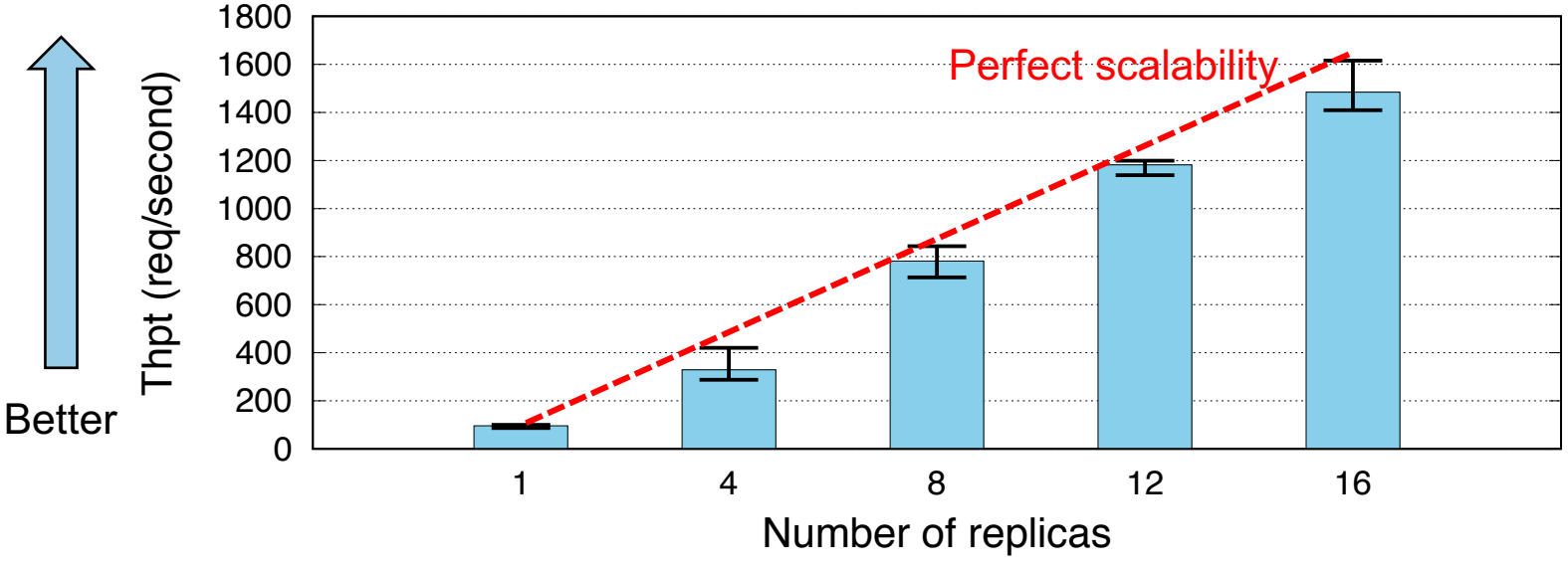
- **End-to-end latency**
  - < 50ms



- **Throughput :**
  - SGX: +15.2%
  - Witness: +7.6%
  - ORAM: +1260% (ORAM does not support concurrency)

Large-scale Deployment

- **Four geographic regions:** Singapore (Asia), London (Europe), West Virginia (East America), California (West America)



From 1 to 16 replicas (each with 82 workers): nearly horizontally scalable

Verification and Speedup

	Native	Delegated
Execution Integrity		
Witness Download	517s	-
Signature Verification	4h25min	-
Data Integrity		
Witness Tree Verification	202s	-
Final-phase verify		
Verifier Interaction	0s	1.0s
Signature Verification	0s	0.2s
Total Time	4h33min	1.2s

13681x

Witnesses can be reused to accelerate the verification

Summary

- **DeSearch is a decentralized search system for DApps**
- **Kanban: decouple states from computation**
  - cloud: avoid replicated storage and computation
  - blockchain: ensure Kanban contents verifiable
- **Witness: decouple verification from computation**
  - remove verification from lambda's fast paths
  - public dataflow proofs: enable data reuse and efficient verification
- **DeSearch project will be available at:**

https://github.com/SJTU-IPADS/DeSearch

Thanks!