



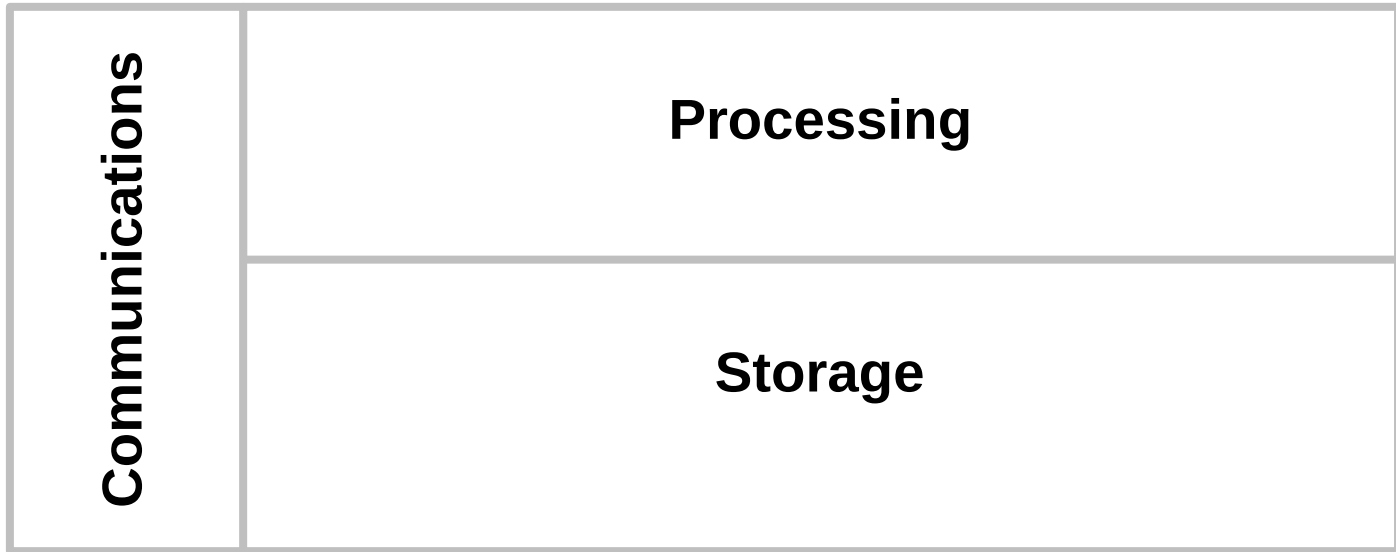
BigchainDB: A Scalable Blockchain Database

Trent McConaghy

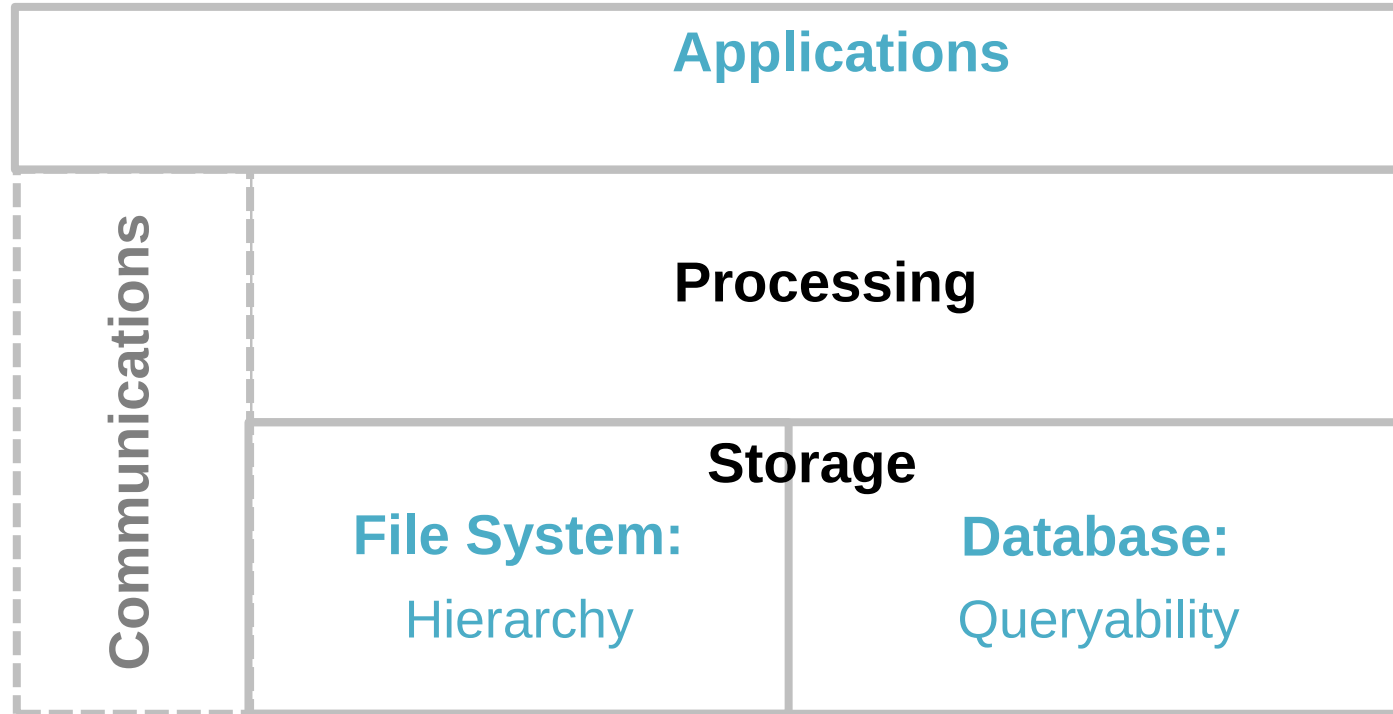
BIGCHAIN^{DB}

ascribe[®]

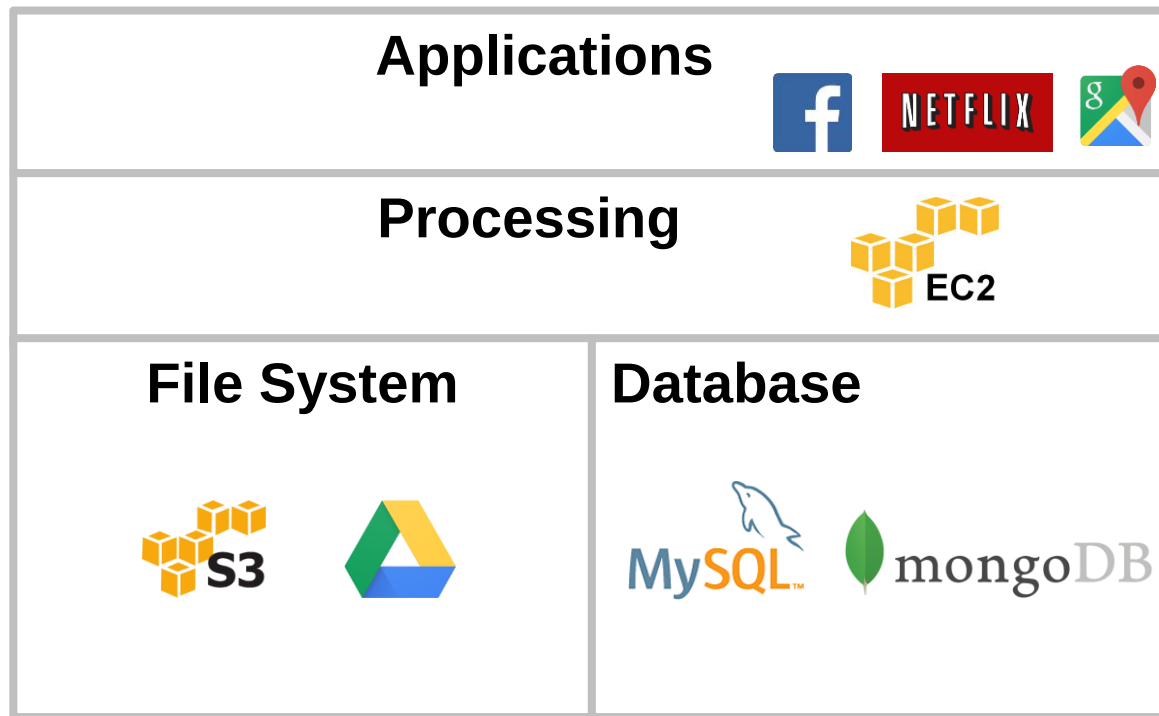
The Elements of Computing



Modern Application Stacks



The modern cloud application stack



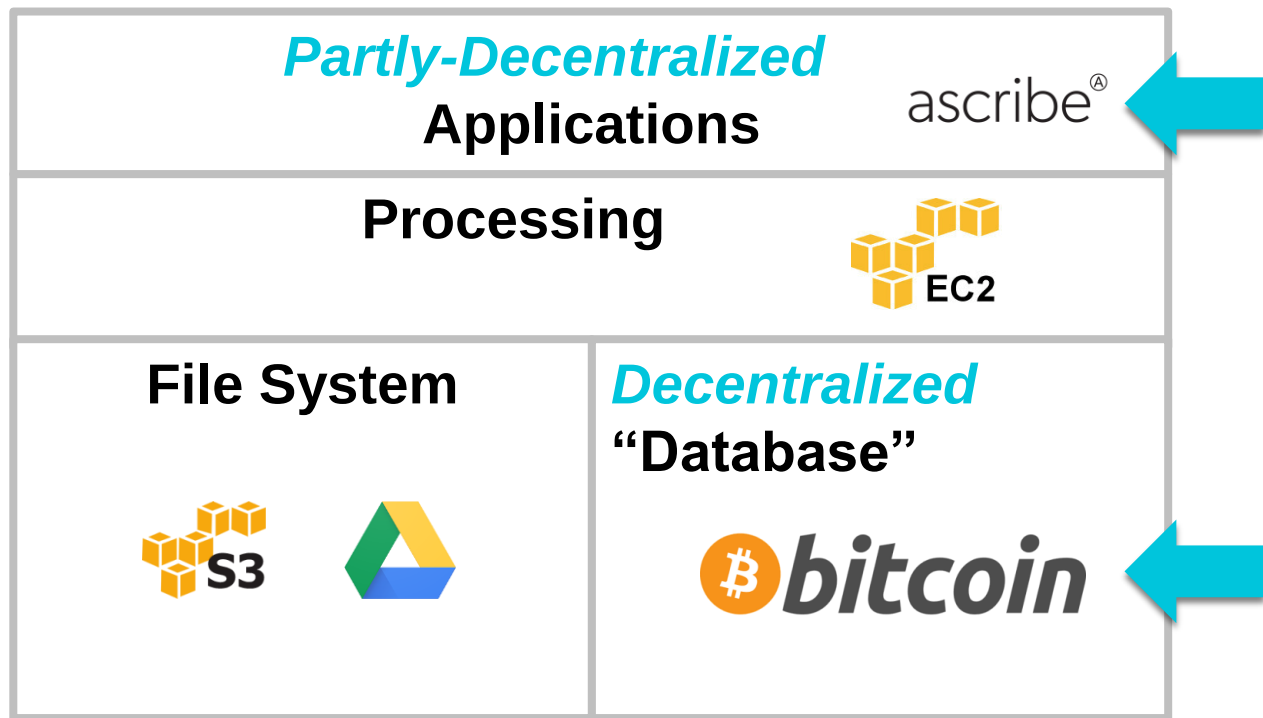
Along came Bitcoin...

“Magic Internet Money”



Bitcoin sparked a revolution

Truly own digital assets, supply chain visibility,



1.5 tx/s

50GB

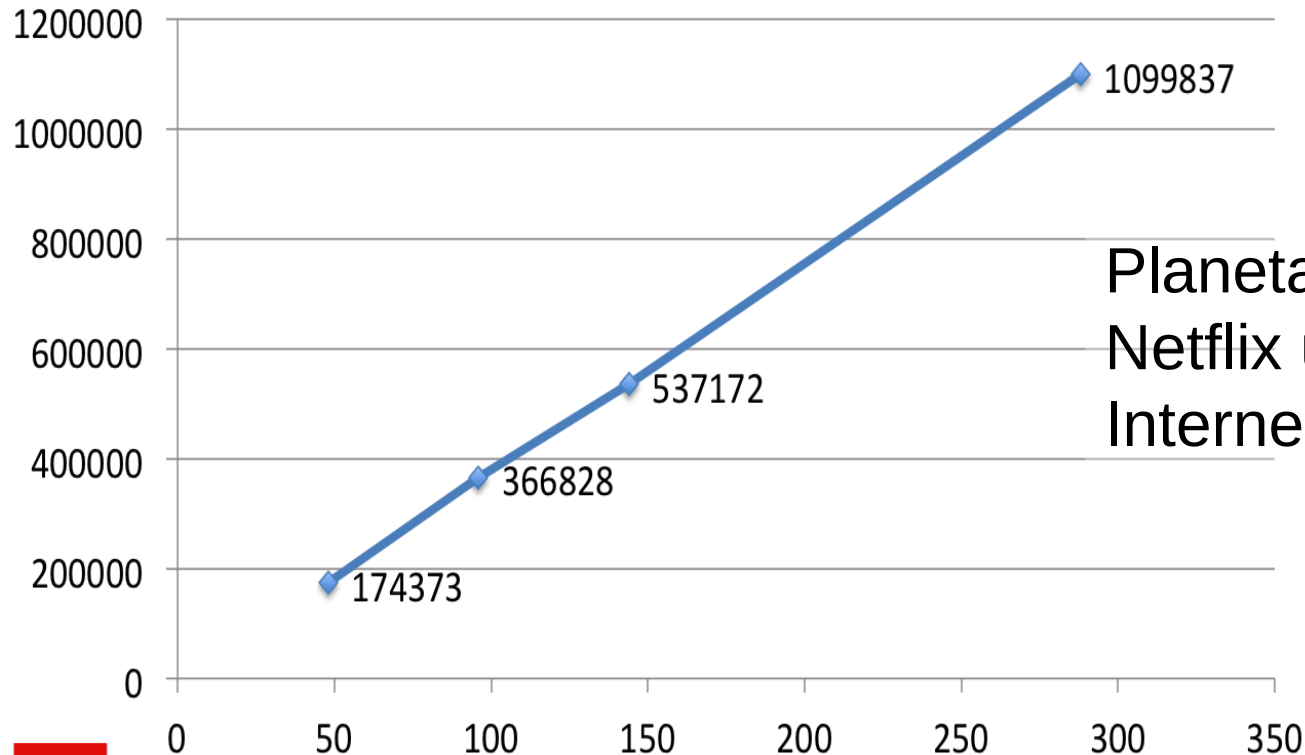
What about **planetary** scale?



Planetary scale:
Netflix uses 37% of
Internet bandwidth

“Big data” Distributed DBs

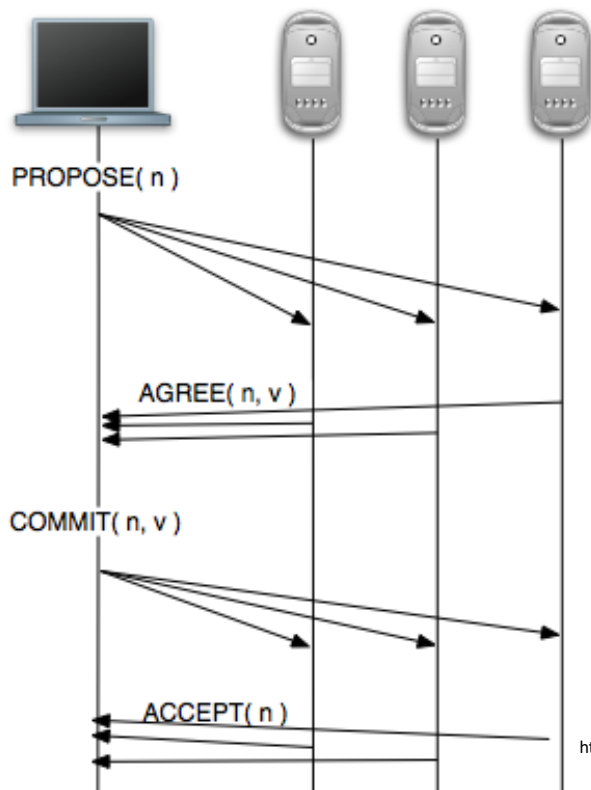
Writes / s vs. # nodes



Planetary scale:
Netflix uses 37% of
Internet bandwidth



To be Distributed, Big Data DBs Must Solve Consensus



Byzantine Consensus
(1982)

Paxos (1990/1998)

Two ways to scale up

Big data-fy the blockchain

- Builds on man-decades of work
- Significant scalability hurdles?

<or>

Blockchain-ify big data

- Builds on man-centuries (millennia?) of work
- Scalability challenges already resolved
- How to blockchain-ify? ...

“Blockchain-ify”

Decentralization: no single entity owns or controls

Immutability: tamper-resistant

Assets: Can issue & transfer assets

Blockchain (noun): hashed-together chain of blocks (1991!)

Blockchain (noun): storage that is decentralized + immutable + assets

Blockchain (*adj*): decentralized + immutable + assets

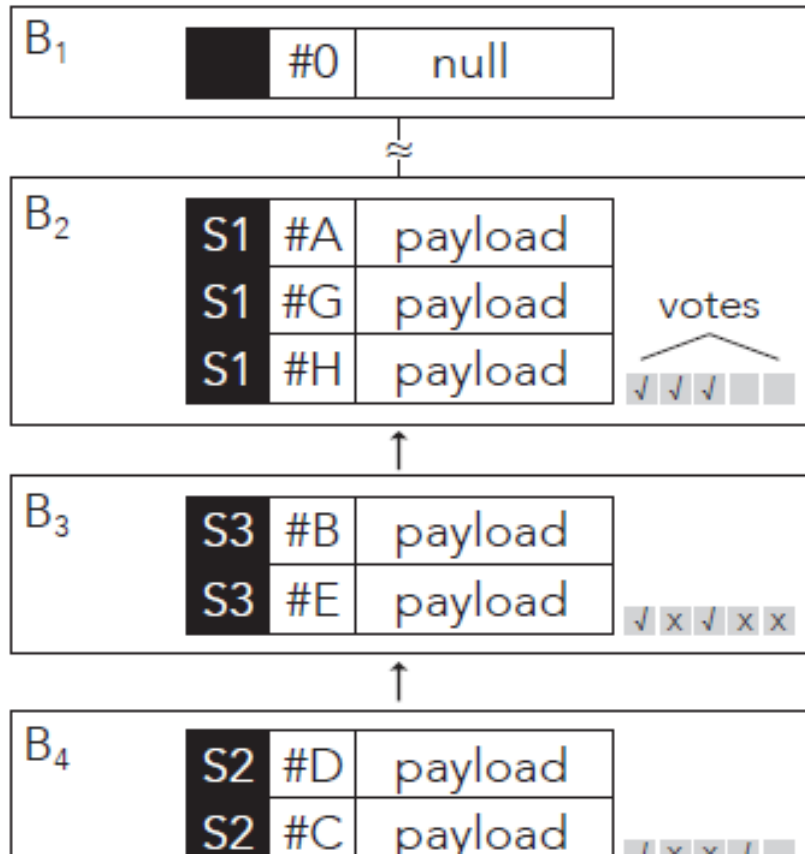
How to Blockchain-ify Big Data

Retain Big Data DB's Performance

- Let the Paxos derivative *solve order*. Get out of its way!
- It naturally builds a log of *all* txs

Add in blockchain characteristics

- **Decentralization:** federation voting on txs. Group into blocks for speed.
- **Immutability:** hash on prev. blocks
- **Assets:** Digital signatures etc.



Benchmarks 1/2

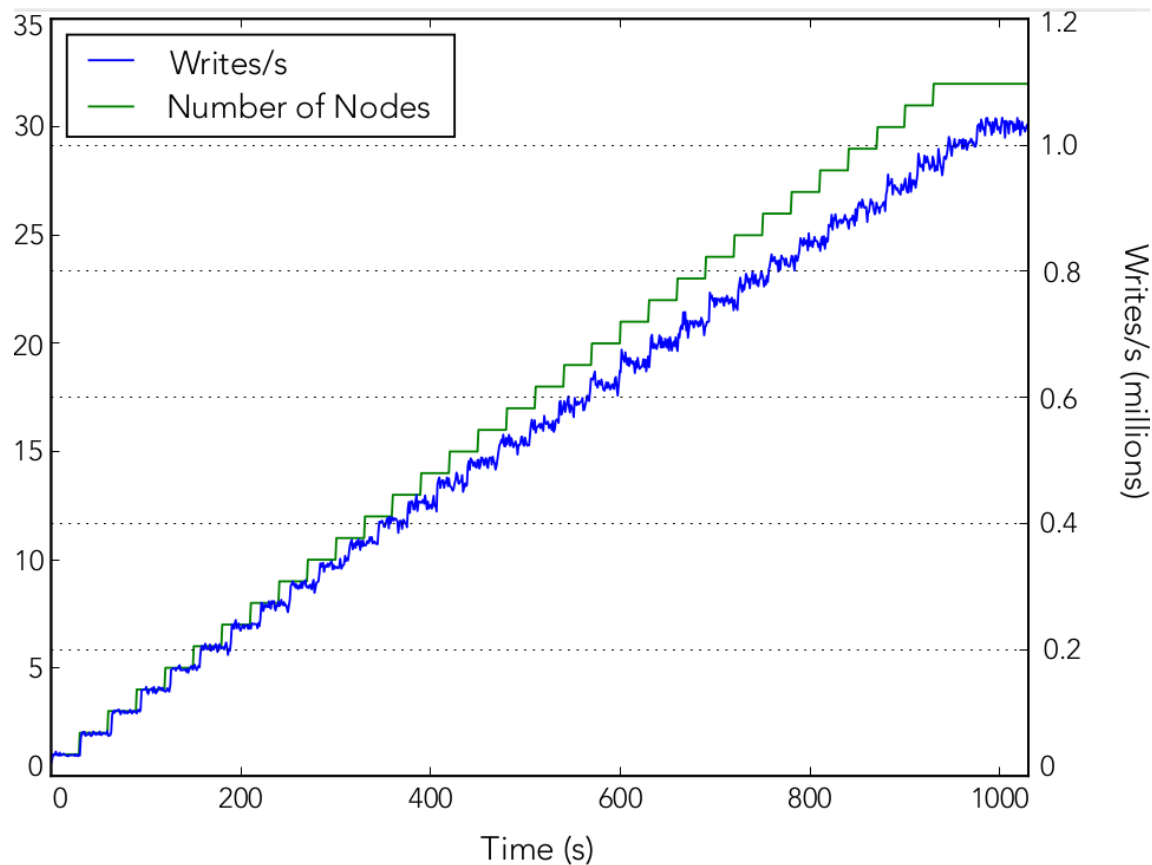
Storage: SSD

Nodes: 32

EC2 instance: c3.8xlarge

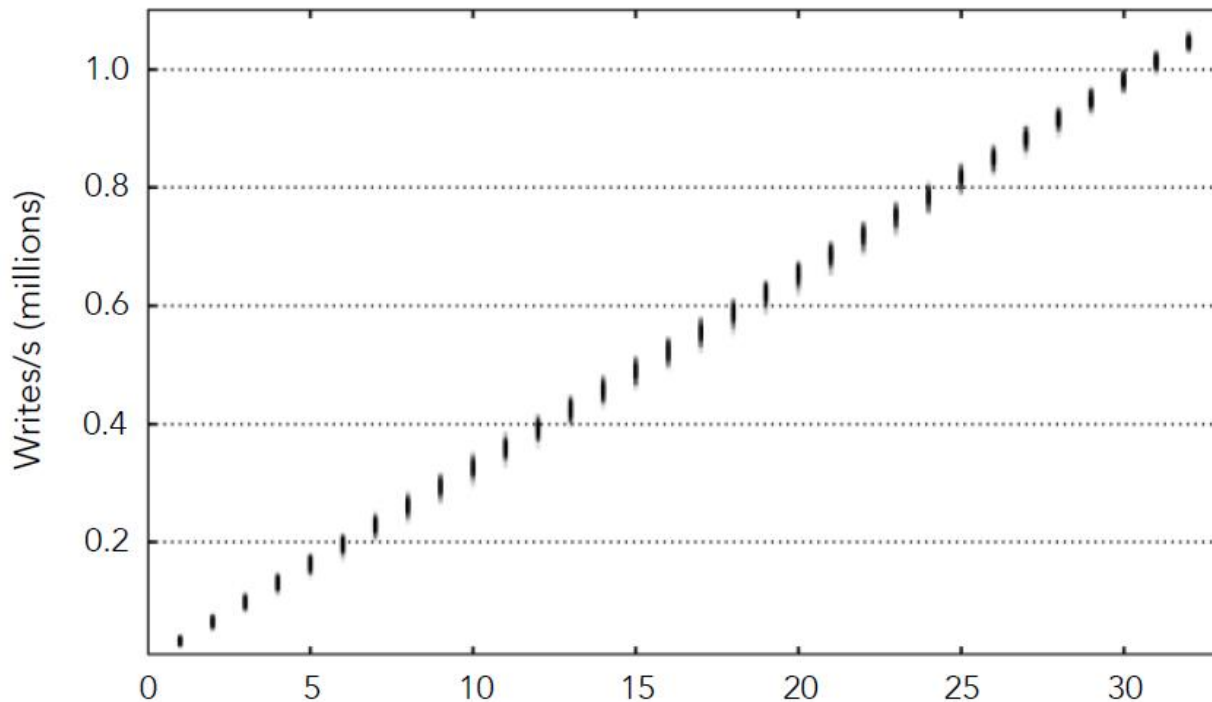
Cores: 32

Network: 10Gbps



Benchmarks 2/2

Writes / s vs. # nodes



This is **BIGCHAIN**^{DB}



Throughput
>1,000,000 writes/s



Latency
Less than 100 ms



Capacity
Petabytes with each
node adding 48TB



Scalability
Performance increases as
nodes are added



Query
Database is fully
queryable

Traditional
blockchains



Big Data

BIGCHAINDB

Immutability



Decentralized control



Assets



High Throughput



Low Latency



High Capacity

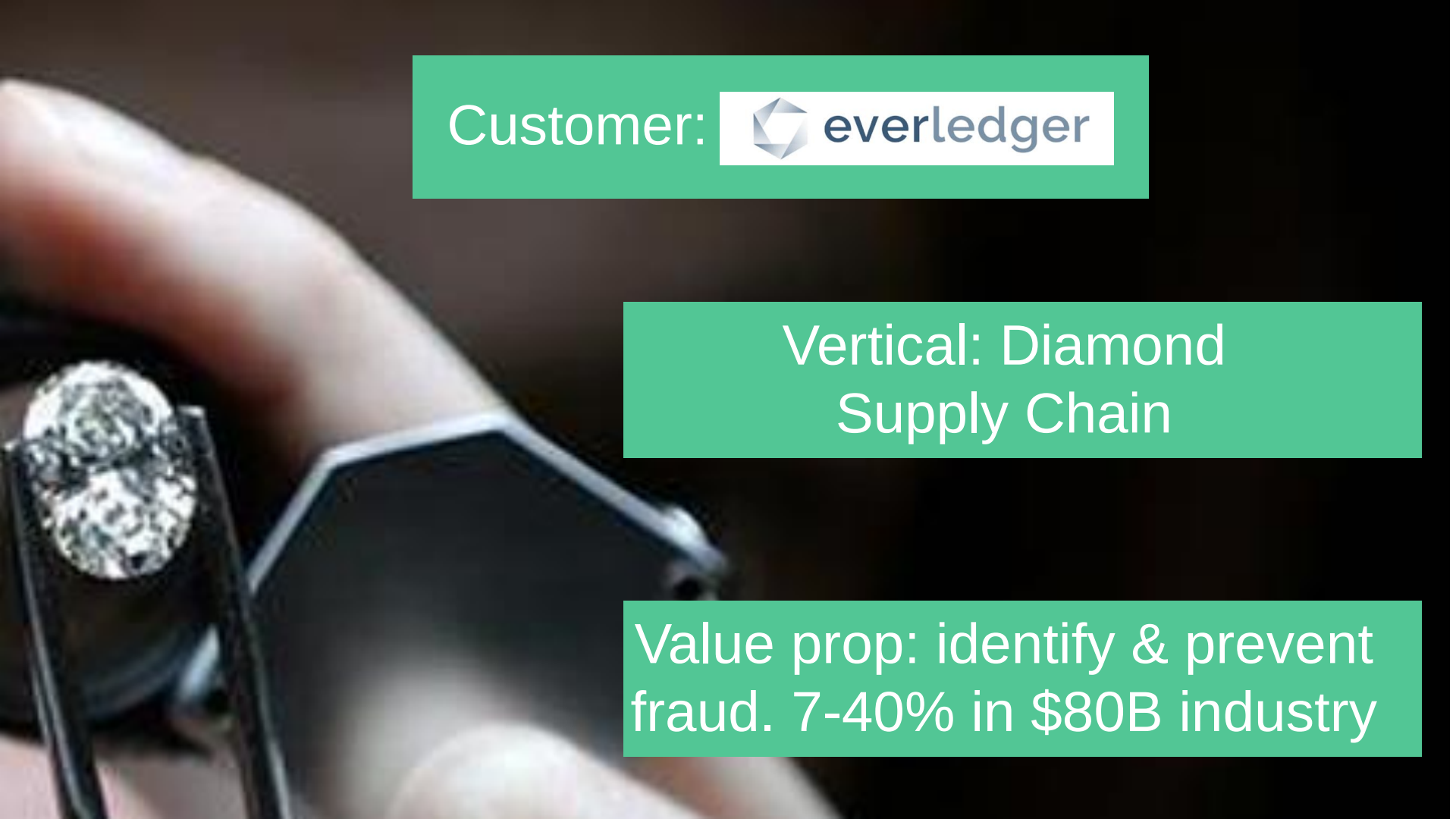


Rich Permissioning



Query Capabilities





Customer:



everledger

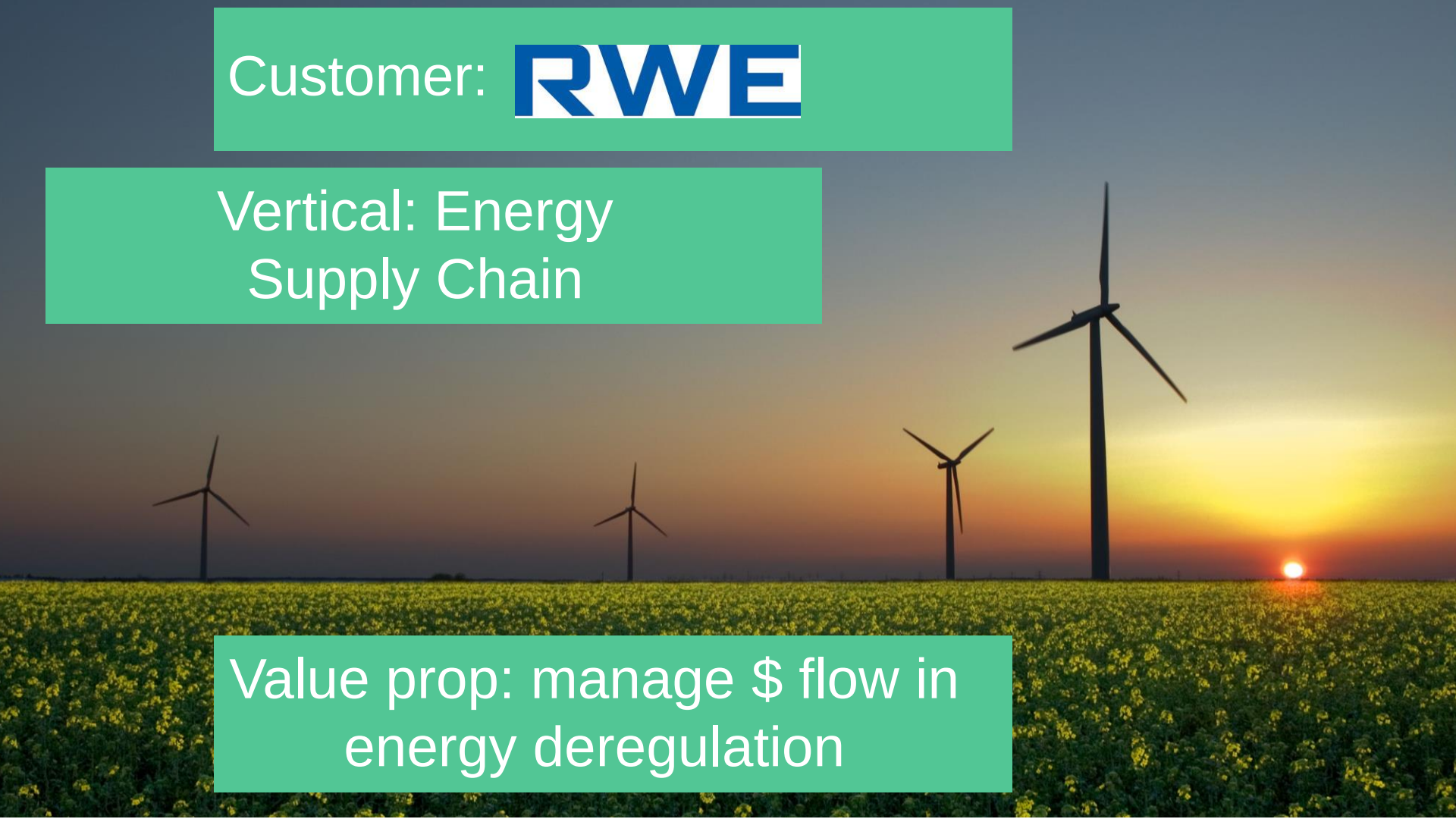
Vertical: Diamond
Supply Chain

Value prop: identify & prevent
fraud. 7-40% in \$80B industry

Customer: **RWE**

Vertical: Energy
Supply Chain

Value prop: manage \$ flow in
energy deregulation





Vertical: Medical Journals /
Supply Chain

Customer: **Tangent⁹⁰**

Value prop: government-
mandated transparent \$ flow

Customers: ascribe.io, 5000 artists,
25 marketplaces & non-profits

Verticals: Art Supply Chain,
Intellectual Property

Value Props: secure provenance
in \$64B art industry, IP mgmt.



Enterprise Use Cases Made Possible by **BIGCHAIN^{DB}**



Capital Markets

Determine exposure in real-time
and position on trades

Streamline Back Office functions

Faster Reconciliations



IoT & Supply Chain

Prevent fraud via transparency,
certificates

Detect fraud and leakage using
analytics

See bottlenecks and delays in
real-time



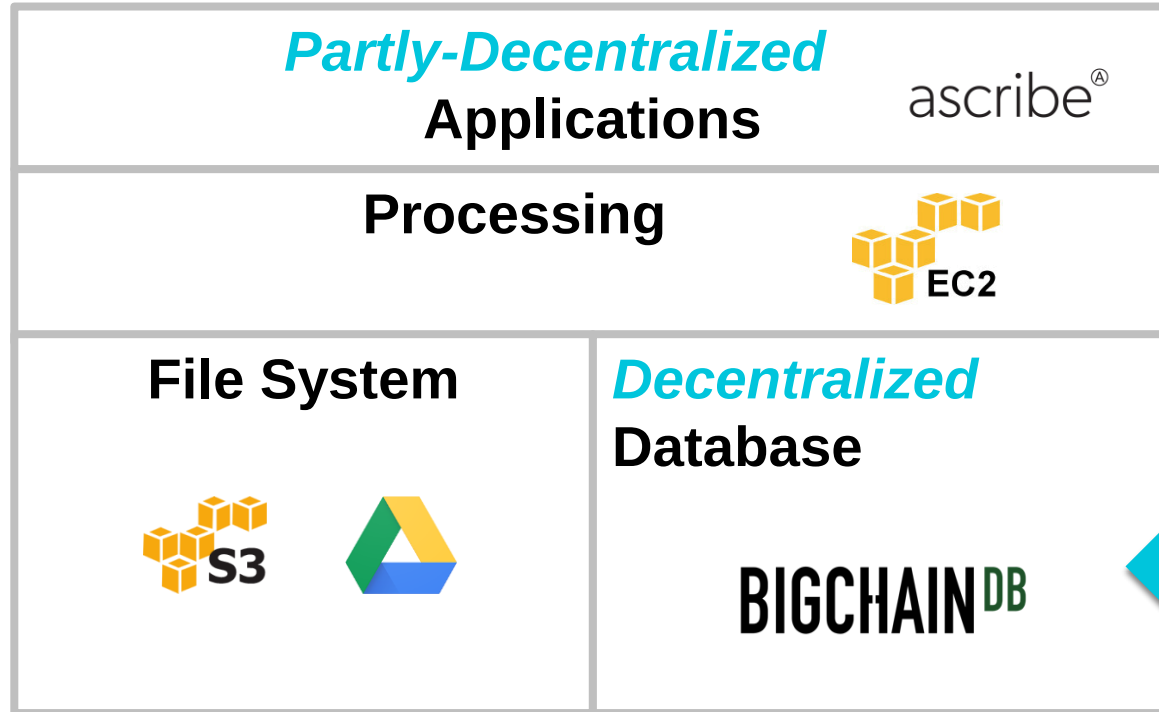
Payments

Reduce risk and cost of escrow

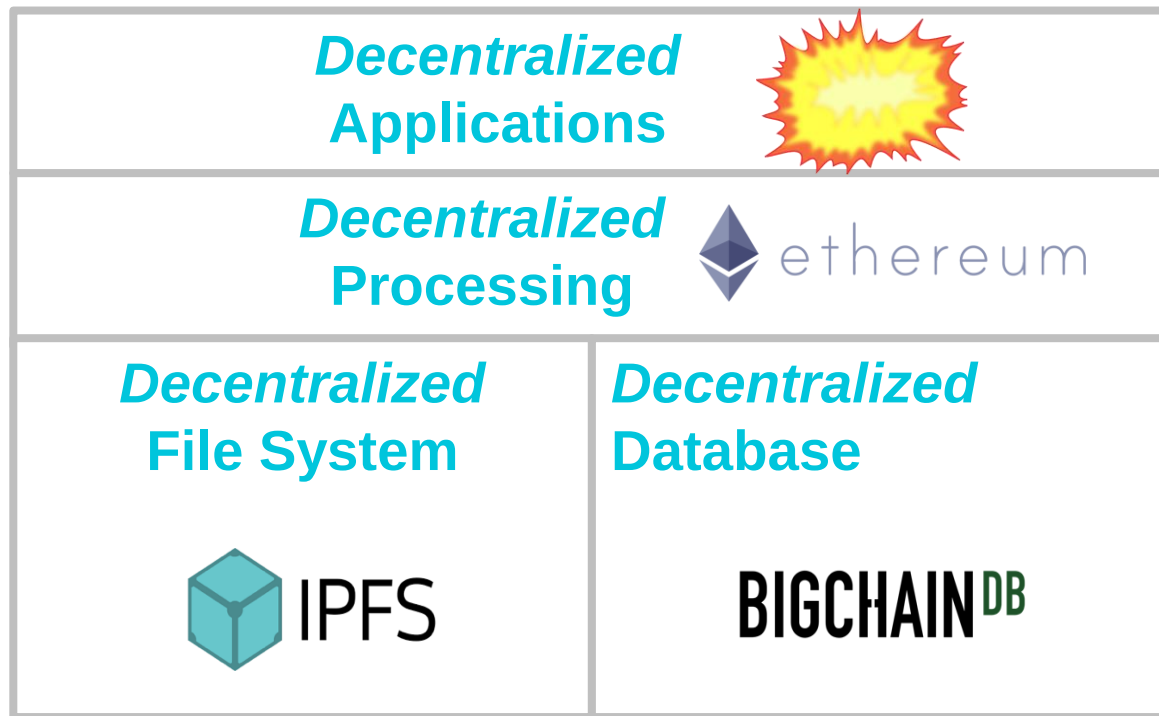
Reduce time to transmit funds

Provide audit trail and receipts

A Decentralized DB for the Planetary-Scale Cloud



Bonus: A DB for the Emerging Planetary-Scale *Decentralized* Cloud



RELATION TO OTHER DBS

What is BigchainDB?

It's an open source distributed blockchain database

- ★ Federated
- ★ Scales linearly with the number of nodes
- ★ High performance / low latency (kinda like RethinkDB)
- ★ **Decentralized** control (each cluster node is controlled by a different entity)
- ★ Federation nodes **update** the database (prevent Sybil Attacks without PoW)
- ★ **Publicly verifiable**
- ★ **Cryptographically secure** (only an owner can transfer its digital asset)
- ★ **Immutable**

What we tested

BigTable Clones

Lineage: Google's [BigTable paper](#).

Data model: Column family, i.e. a tabular model where each row can have an individual configuration of columns.

Tested: HBase, Cassandra

Good at: Handles size well. Stream massive write loads. High availability. Multiple-data centers. MapReduce.

Document Databases

Lineage: Inspired by Lotus Notes.

Data model: Collections of documents, which contain key-value collections.

Tested: RethinkDB, MongoDB, Elasticsearch

Good at: Natural data modeling. Programmer friendly. Rapid development. Web friendly, CRUD.

Key-Value Stores

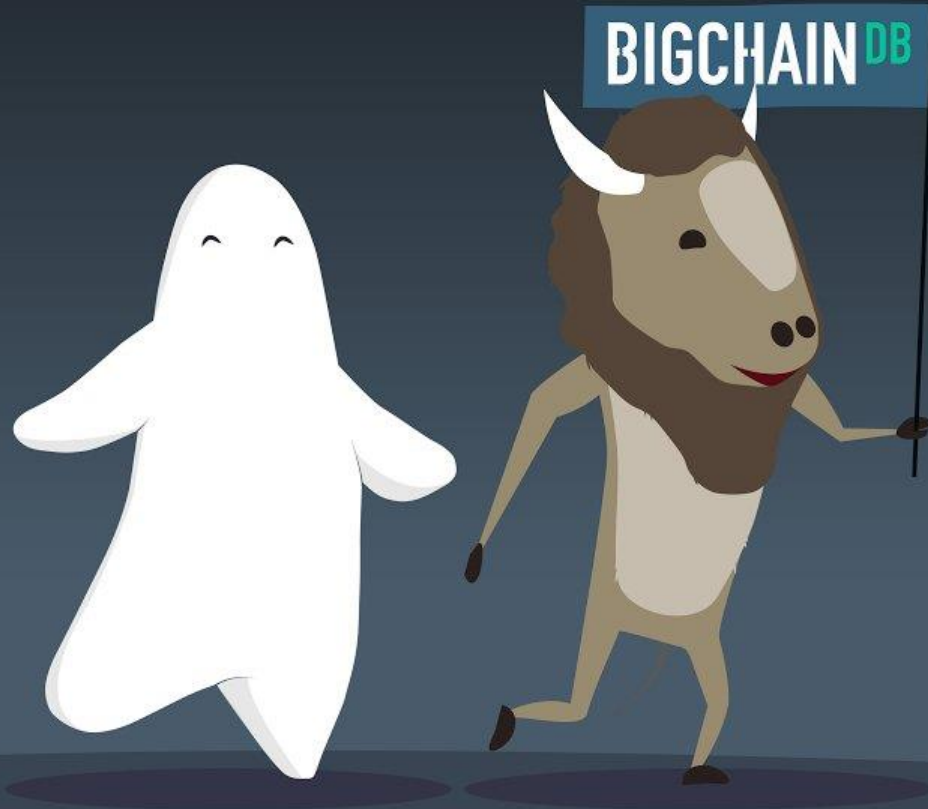
Lineage: Amazon's [Dynamo paper](#) and [Distributed HashTables](#).

Data model: A global collection of KV pairs.

Tested: Redis, Riak

Good at: Handles size well. Processing a constant stream of small reads and writes. Fast. Programmer friendly.

BIGCHAINDB IS RETHINKING BLOCKCHAIN DATABASES



Why RethinkDB?

Strong consistency guarantees **linearizability** and **atomicity**

Real time notifications of any change in the data

ReQL

- ★ Powerful query language
- ★ Flexible indexing

JSON

- ★ standard
- ★ high support (language independent interchange format)
- ★ easy to serialize

Raft consensus protocol equivalent in fault-tolerance and performance to **Paxos**

Easy configuration (sharding, replication, cluster)

Open source and **great community**

ARCHI-
TECTURE

Arch: review

Retain Big Data DB's Performance

Let the Paxos derivative *solve order*. Get out of its way!

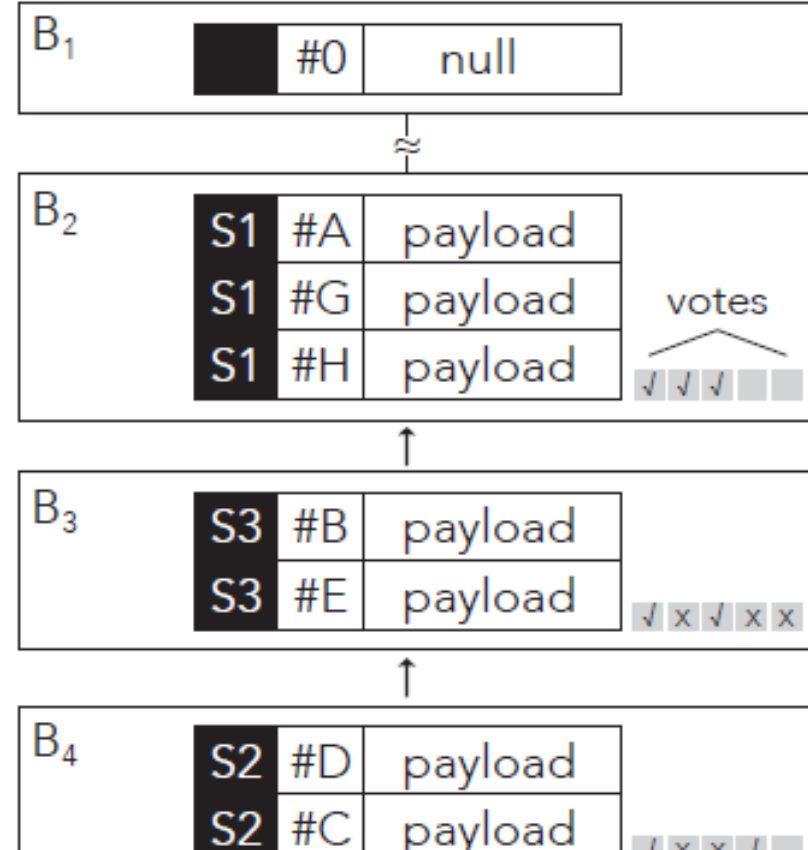
It naturally builds a log of *all* txs

Add in blockchain characteristics

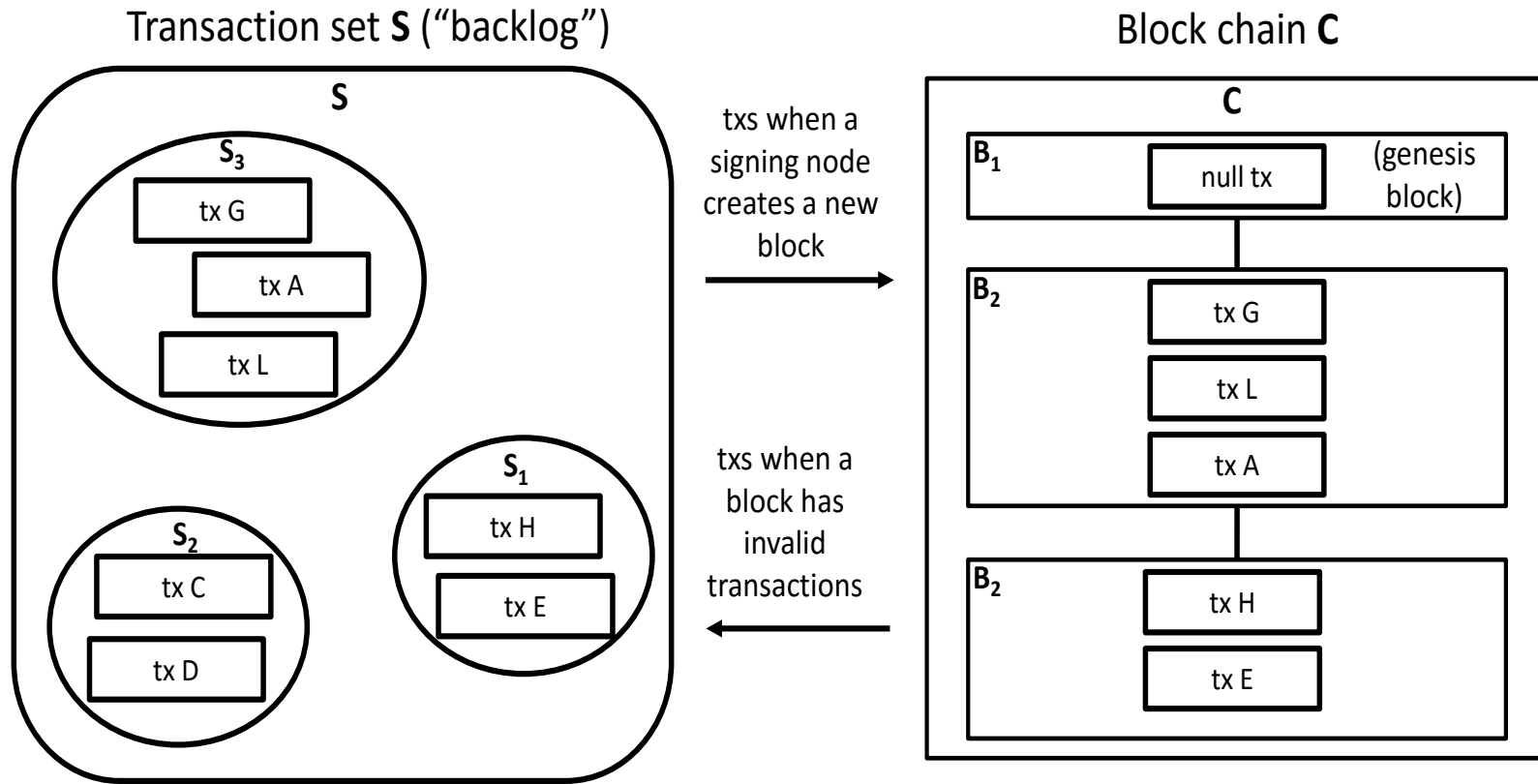
Decentralization: federation voting on txs. Group into blocks for speed.

Immutability: hash on prev. blocks

Assets: Digital signatures etc.

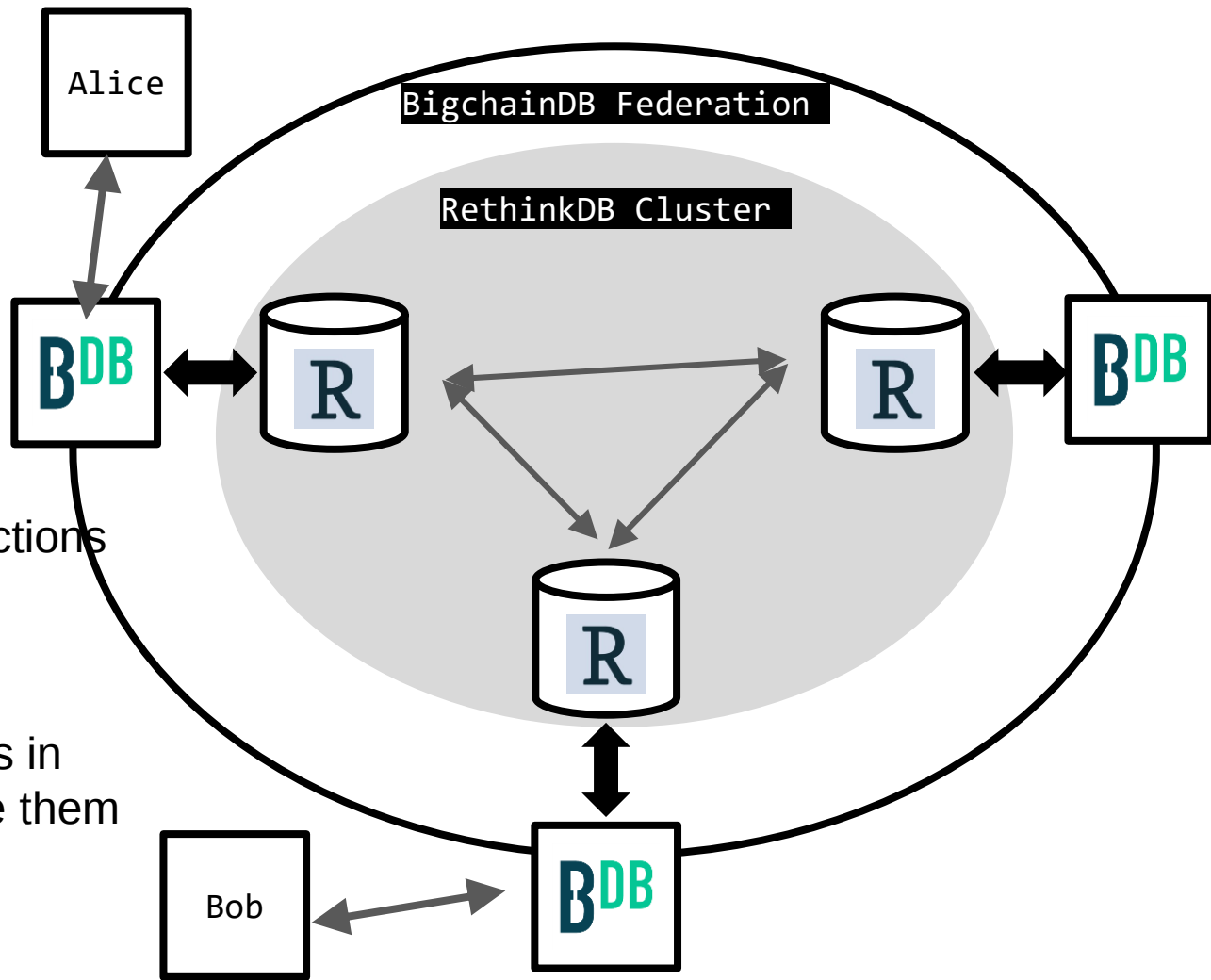


Two Tables



System Arch

- ★ **RethinkDB**
handles intra-cluster communication
- ★ **BigchainDB Nodes**
accept new transactions via an API
- ★ **BigchainDB Nodes**
bundle transactions in blocks and validate them



TXS

Transaction

hash <i>key</i>	from <i>key</i>	to <i>key</i>	input <i>hash</i>	payload <i>json data</i>	signature <i>w/priv_key</i>
---------------------------	---------------------------	-------------------------	-----------------------------	------------------------------------	---------------------------------------

Transaction example

hash 123	from <i>eventim</i>	to <i>eventim</i>	input -	payload {ticket_id: 2661}	signature #####
--------------------	-------------------------------	-----------------------------	-------------------	--	---------------------------

hash 456	from <i>eventim</i>	to <i>Alice</i>	input 123	payload {ticket_id: 2661}	signature #####
--------------------	-------------------------------	---------------------------	---------------------	--	---------------------------

hash 789	from <i>Alice</i>	to <i>Bob</i>	input 456	payload {ticket_id: 2661}	signature #####
--------------------	-----------------------------	-------------------------	---------------------	--	---------------------------

Malicious transaction example

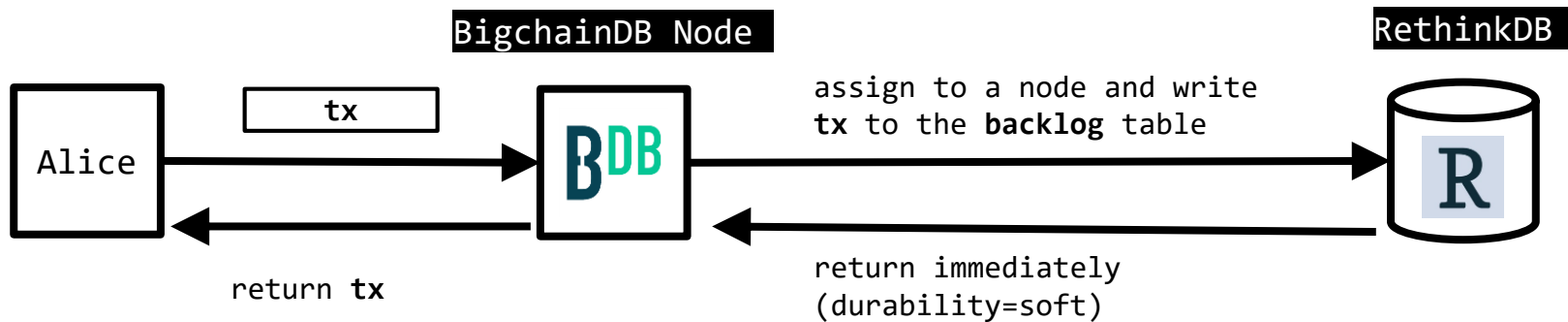
hash 123	from <i>eventim</i>	to <i>eventim</i>	input -	payload {ticket_id: 2661}	signature #####
-------------	------------------------	----------------------	------------	---------------------------------	--------------------

hash 456	from <i>eventim</i>	to <i>Carly</i>	input 123	payload {ticket_id: 2661}	signature #####
-------------	------------------------	--------------------	--------------	---------------------------------	--------------------

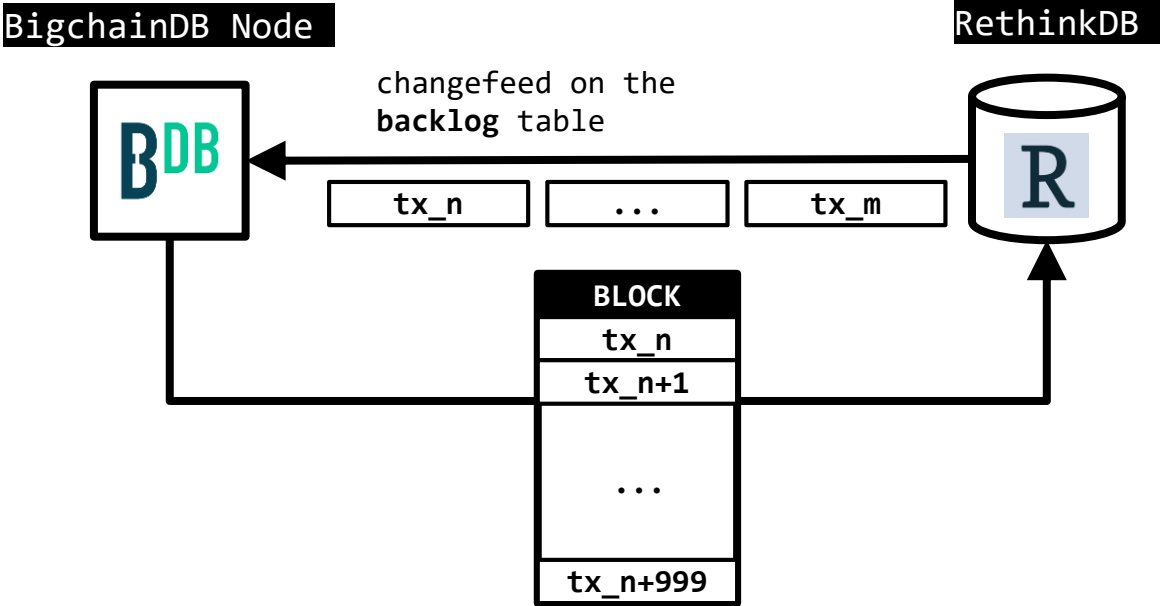
hash 789	from <i>Carly</i>	to <i>Alice</i>	input 456	payload {ticket_id: 2661}	signature #####
-------------	----------------------	--------------------	--------------	---------------------------------	--------------------

hash <i>0ab</i>	from <i>Carly</i>	to <i>Bob</i>	input 456	payload {ticket_id: 2661}	signature #####
--------------------	----------------------	------------------	--------------	---------------------------------	--------------------

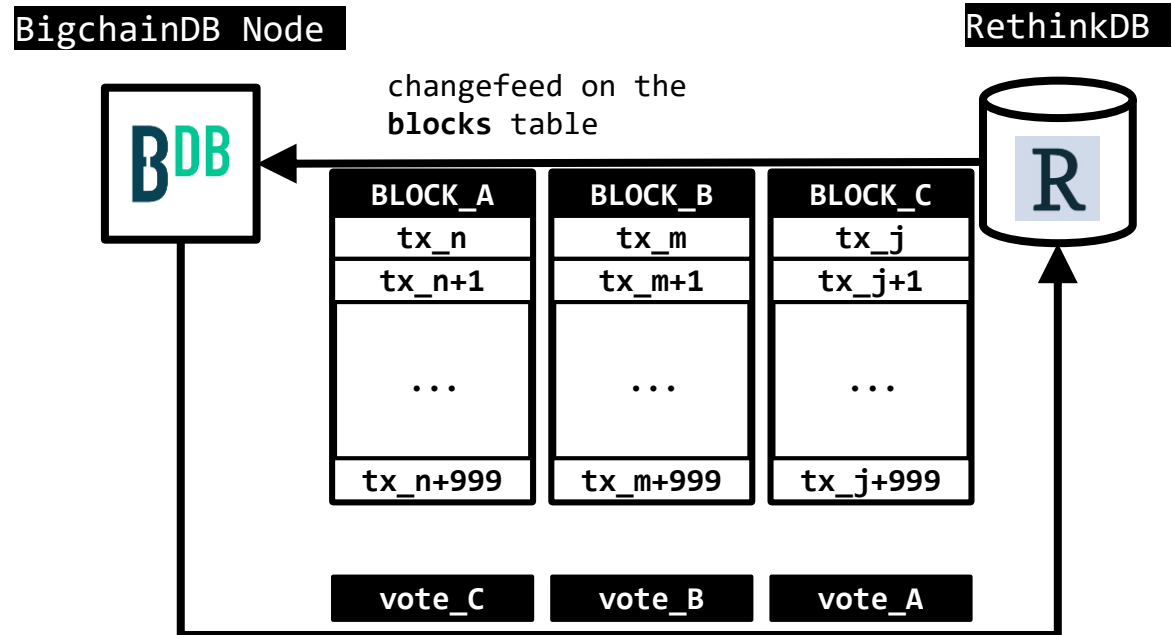
Lifecycle of a transaction



Making blocks

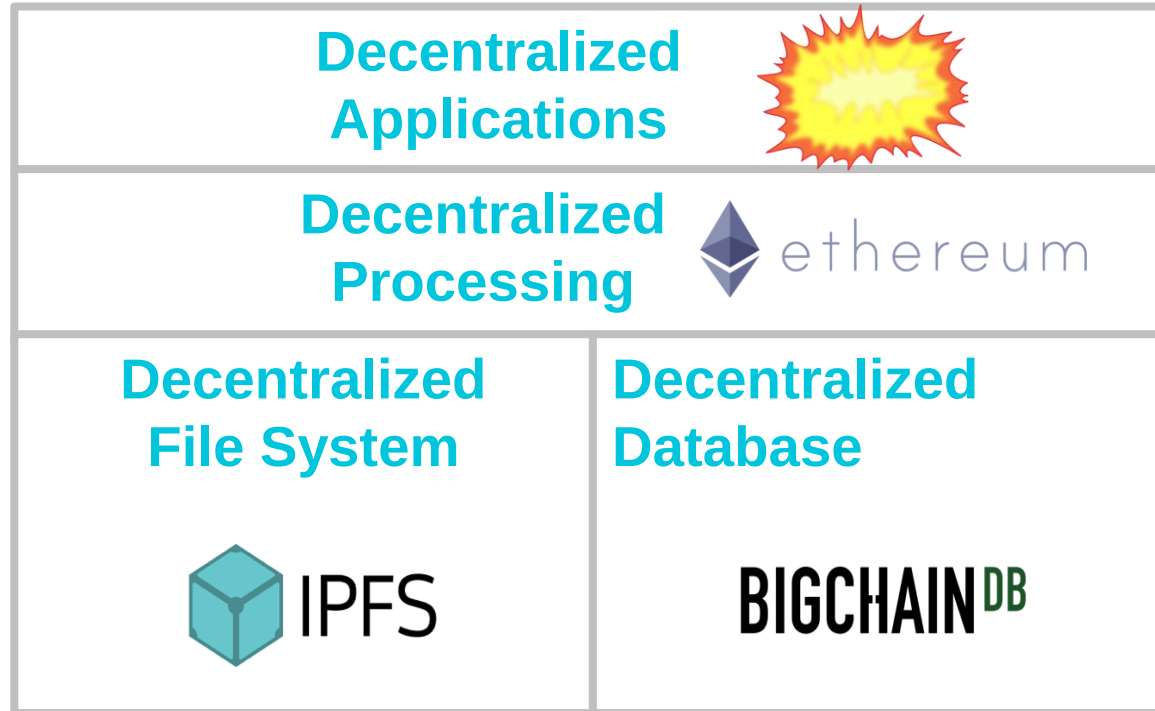


Validating blocks



RECAP

Vision: A DB for the Emerging Planetary-Scale *Decentralized* Cloud



BigchainDB: A Scalable Blockchain Database For the Planet & The Enterprise (& Wrigley)

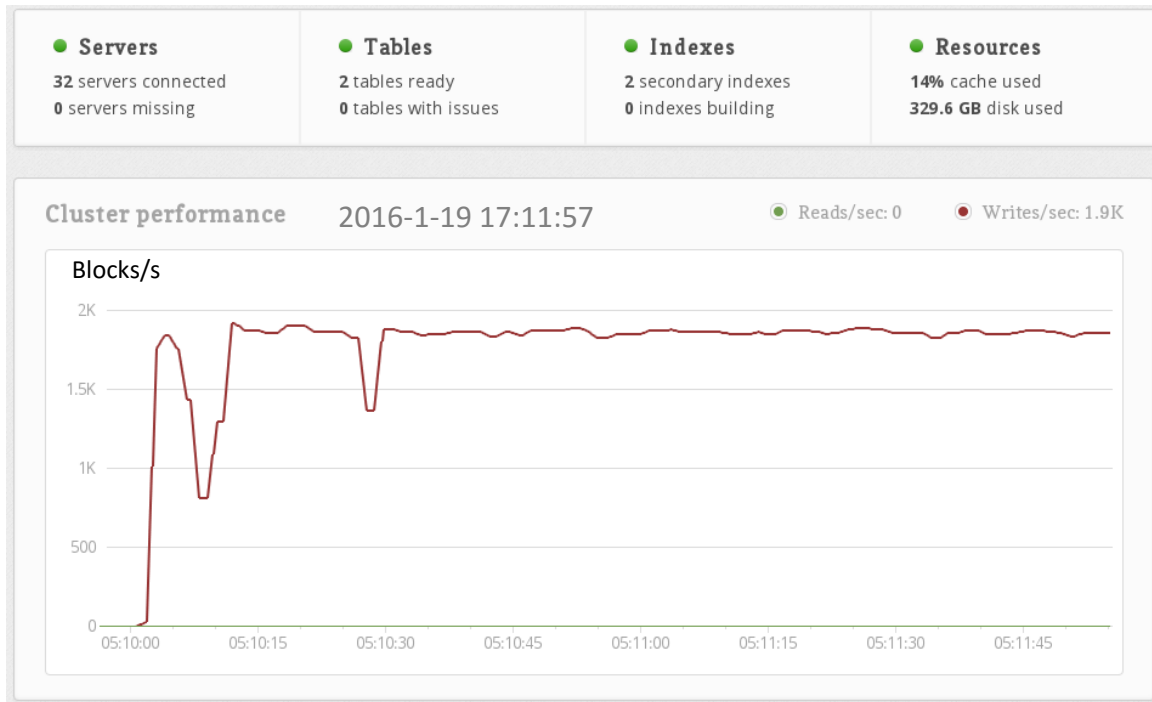


github.com/bigchaindb
bigchaindb.com/whitepaper



APPENDIX

BIGCHAIN^{DB} > Million Writes per Second



>1 Million

BigchainDB handled nearly 2 Mio Writes/s

- 1,900 blocks written per second
- Each Block can store 1,000 transactions
- BigchainDB **scales linearly** as nodes are added
- Limiting constraint is I/O, not transactions or validations
- Each Node adds 48 TB in storage capacity

Decentralized Applications			
Decentralized Processing			 ethereum
Decentralized File System	Decentralized Database	Decentralized Ledger	
 IPFS	BIGCHAIN^{DB}		