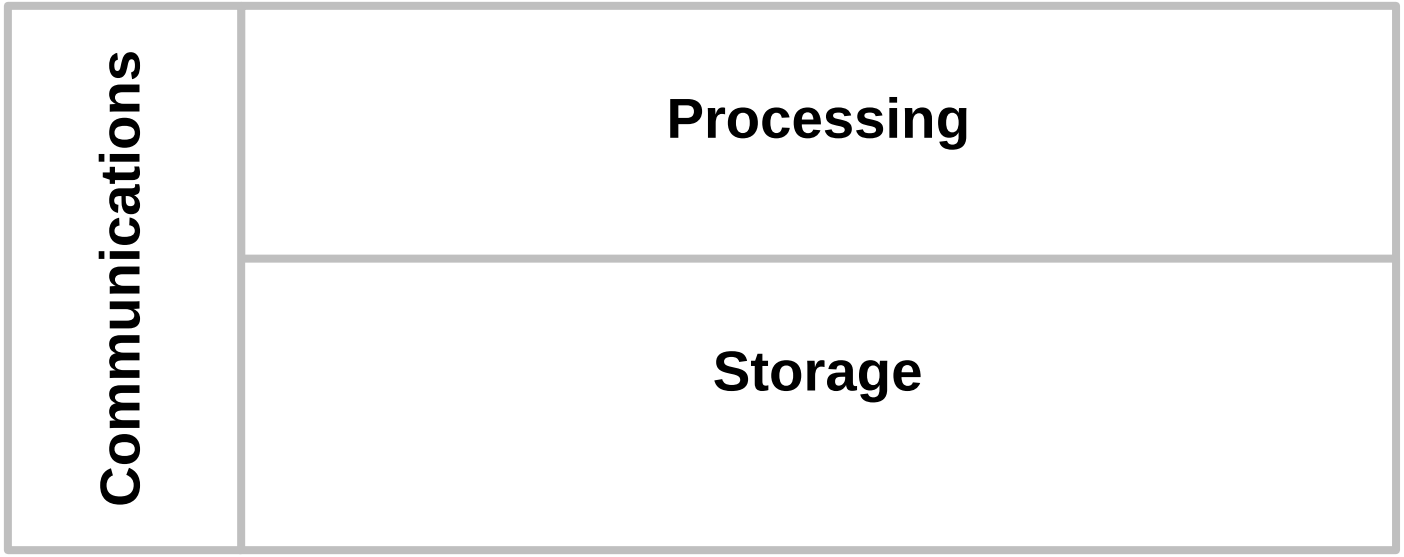
BigchainDB: A Scalable Blockchain Database, In Python

Trent McConaghy

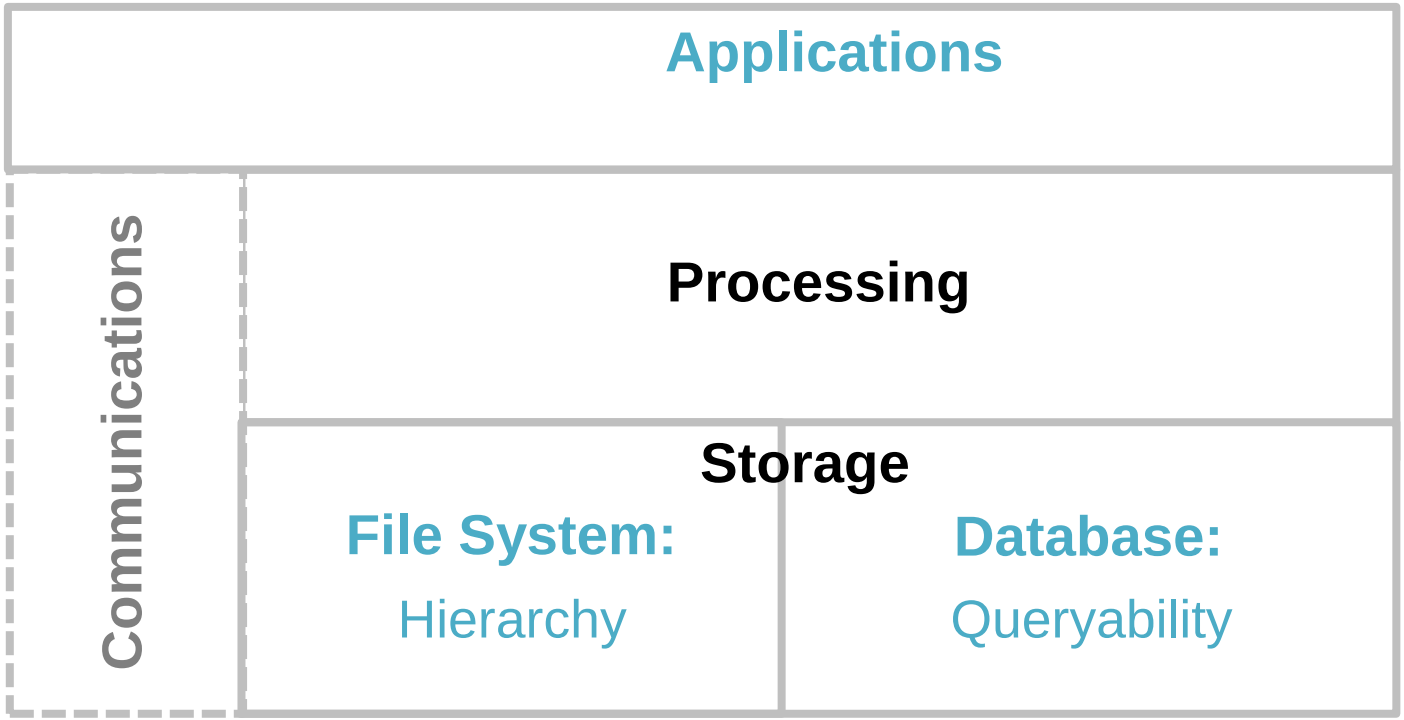
BIGCHAINDB

ascribe®

The Elements of Computing



Modern Application Stacks



The modern cloud application stack

Applications

Processing

EC2

File System

S3

Database

MySQL

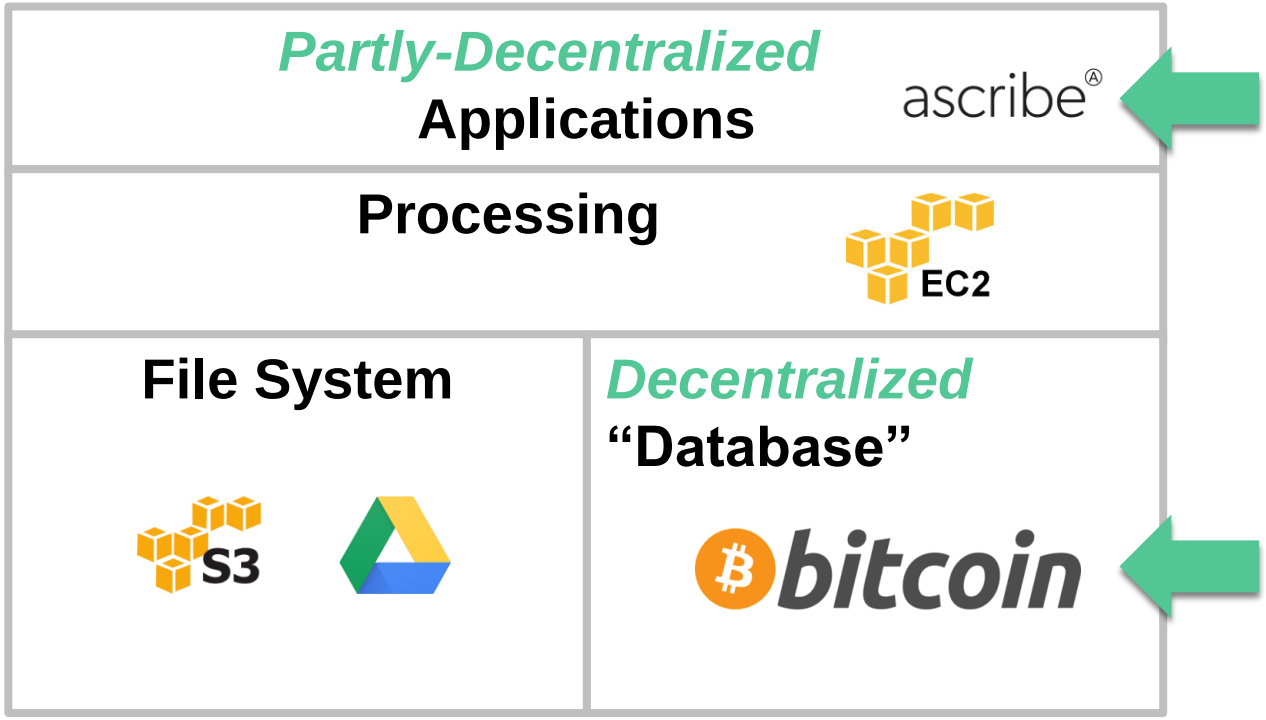
mongoDB

Along came Bitcoin…

“Magic Internet Money”

Bitcoin sparked a revolution

Truly own digital assets, supply chain visibility, ….

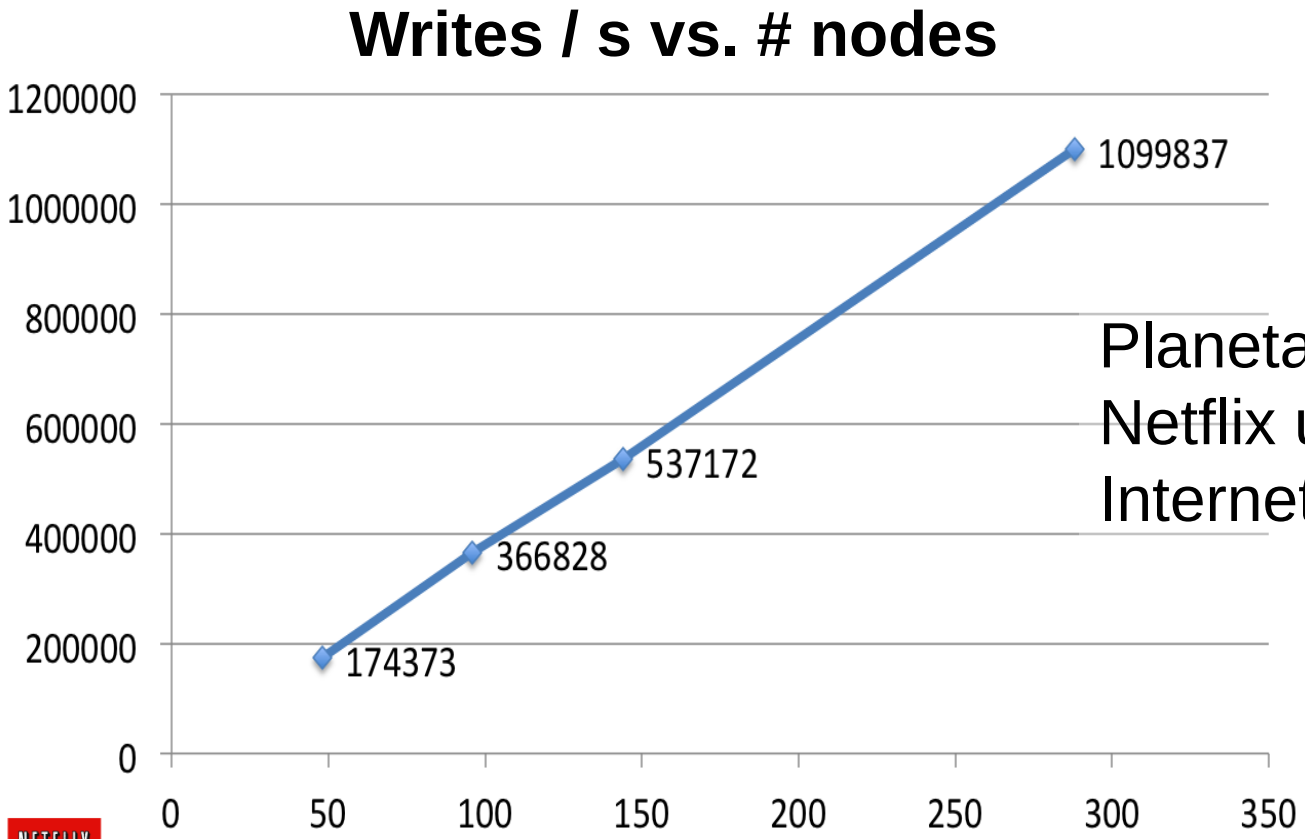


1.5 tx/s
50GB

What about **planetary** scale?

Planetary scale:
Netflix uses 37% of
Internet bandwidth

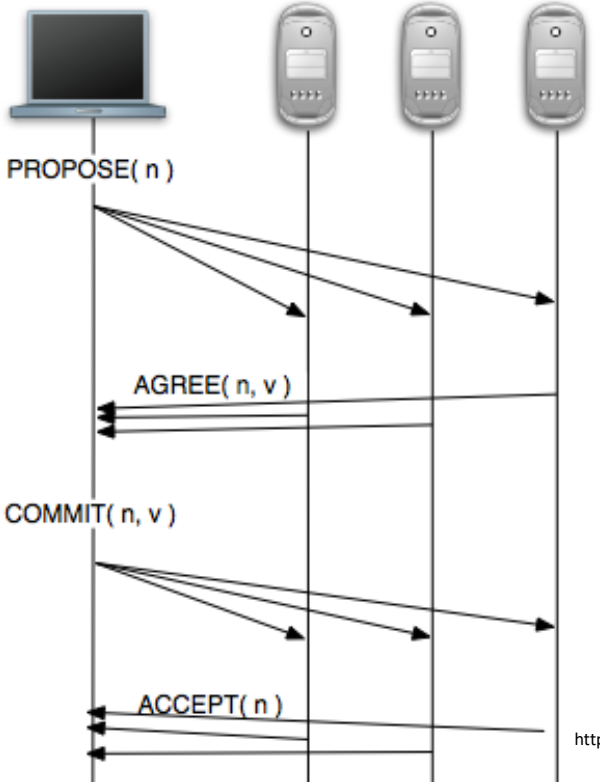
“Big data” Distributed DBs



Planetary scale:
Netflix uses 37% of
Internet bandwidth

http://1.bp.blogspot.com/-ZFtW7MFMqZQ/TrG5ujuDGdI/AAAAAAAAAWw/heceeMD50x4/s1600/scale.png

To be Distributed,
Big Data DBs Must Solve Consensus



Byzantine Consensus (1982)

Paxos (1990/1998)

https://medium.com/the-bigchaindb-blog/the-writings-of-leslie-lamport-abridged-a67df77f464#.1lr34qt6s
http://the-paper-trail.org/blog/consensus-protocols-paxos/

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

INTRODUCING BIGCHAINDB

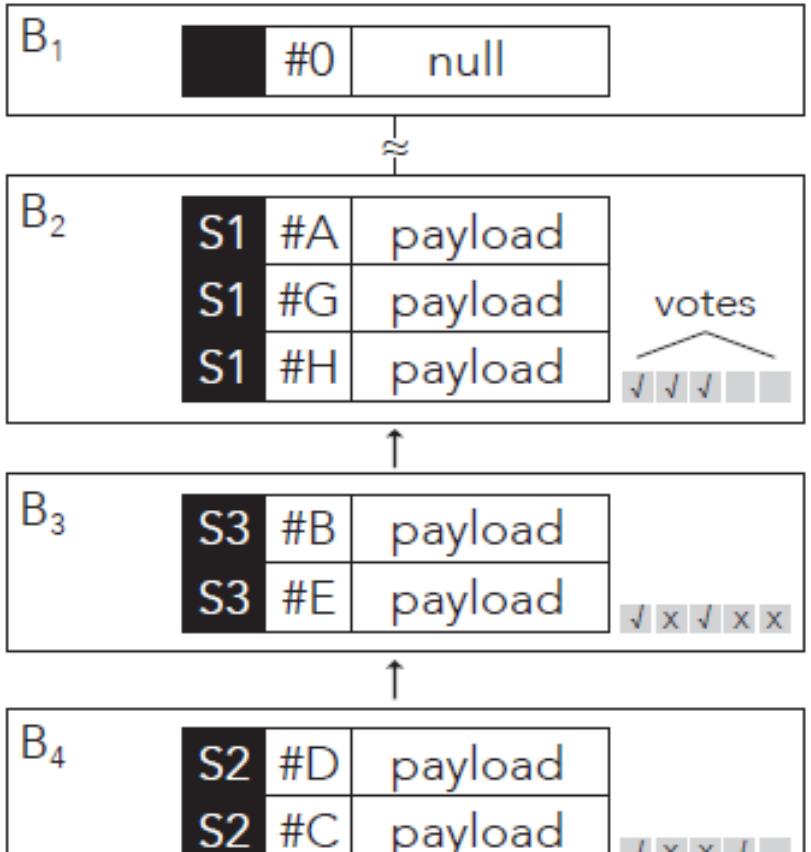
How to Blockchain-ify Big Data

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

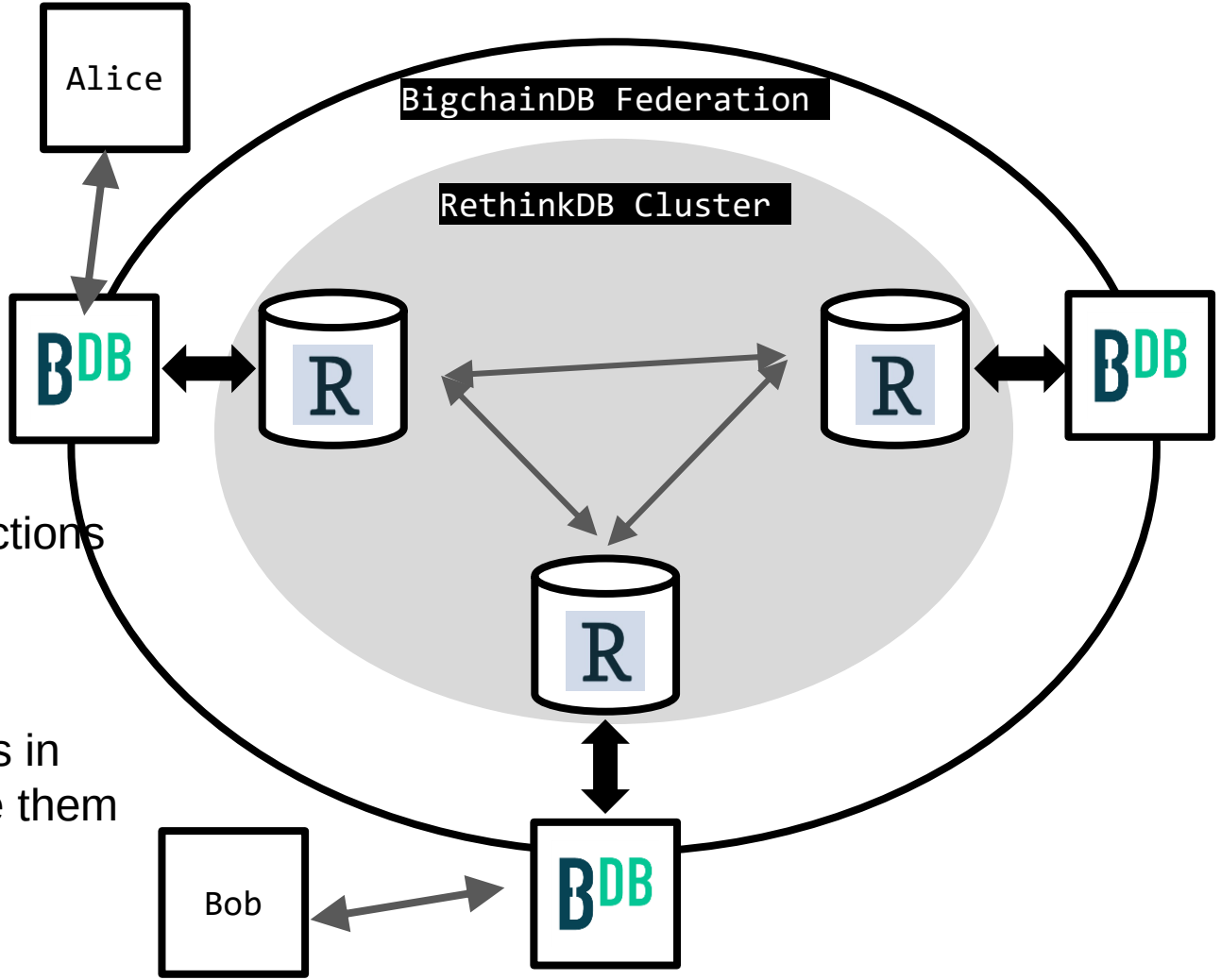
Add in blockchain characteristics

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

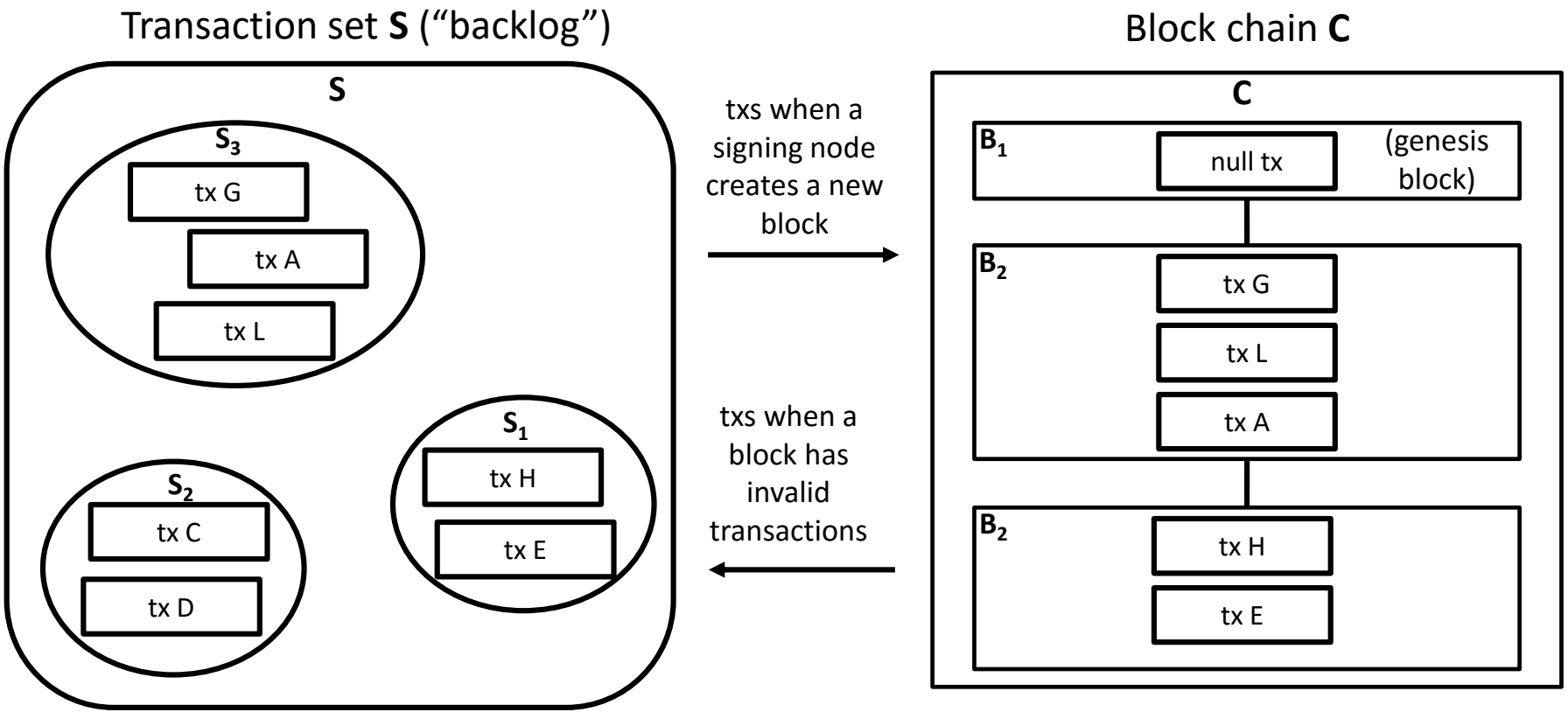


System Arch

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

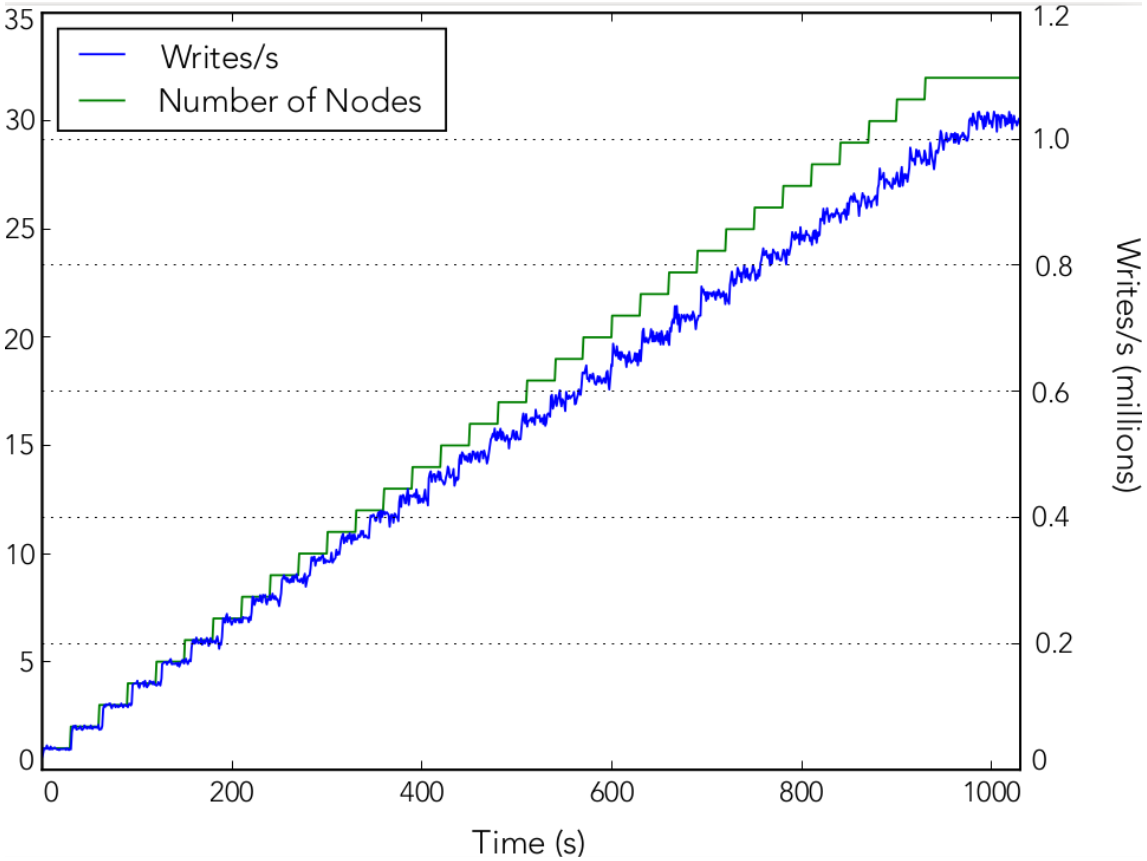


Two Tables


Transaction set **S** ("backlog")
S
S_3
tx G
tx A
tx L
S_1
tx H
tx E
S_2
tx C
tx D
txs when a signing node creates a new block
txs when a block has invalid transactions
Block chain **C**
C
B_1
null tx
(genesis block)
B_2
tx G
tx L
tx A
B_2
tx H
tx E

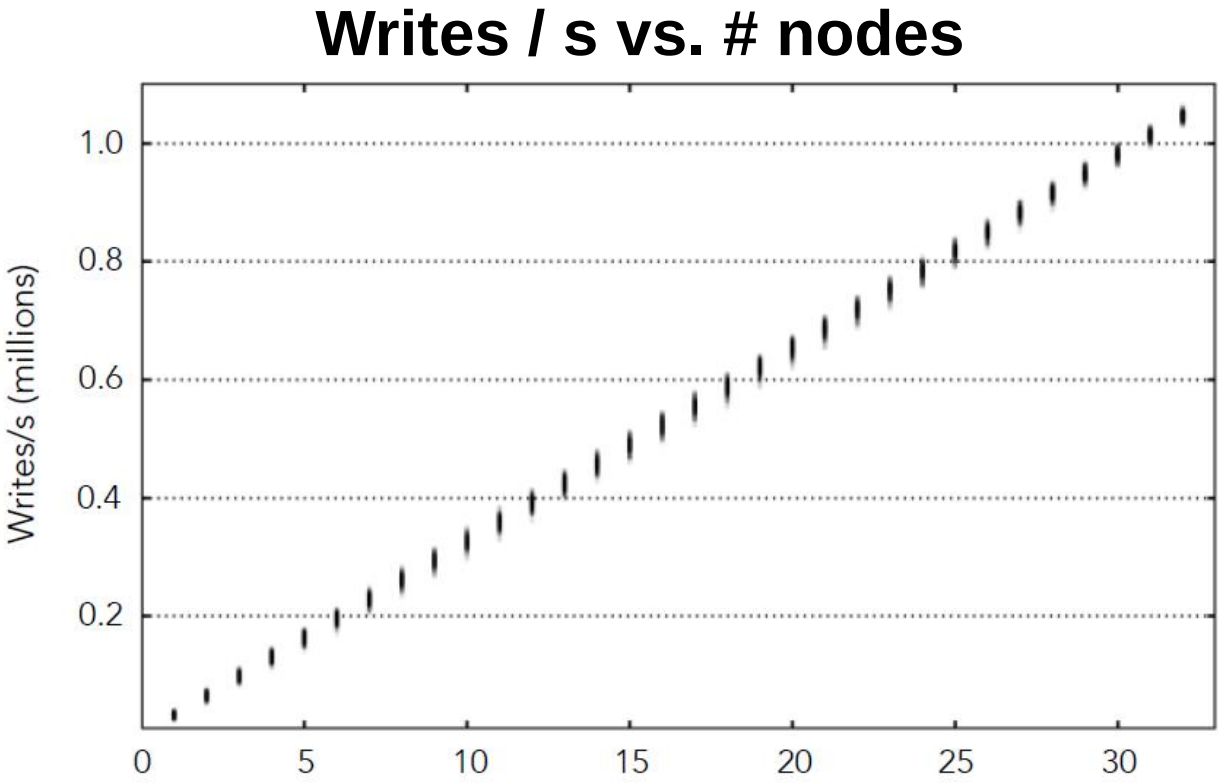
Benchmarks 1/2

Storage: SSD
Nodes: 32
EC2 instance: c3.8xlarge
Cores: 32
Network: 10Gbps



www.bigchaindb.com/whitepaper

Benchmarks 2/2



www.bigchaindb.com/whitepaper

https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf

bigchaindb-whitepaper.pdf 1 / 65

BigchainDB: A Scalable Blockchain Database (DRAFT)

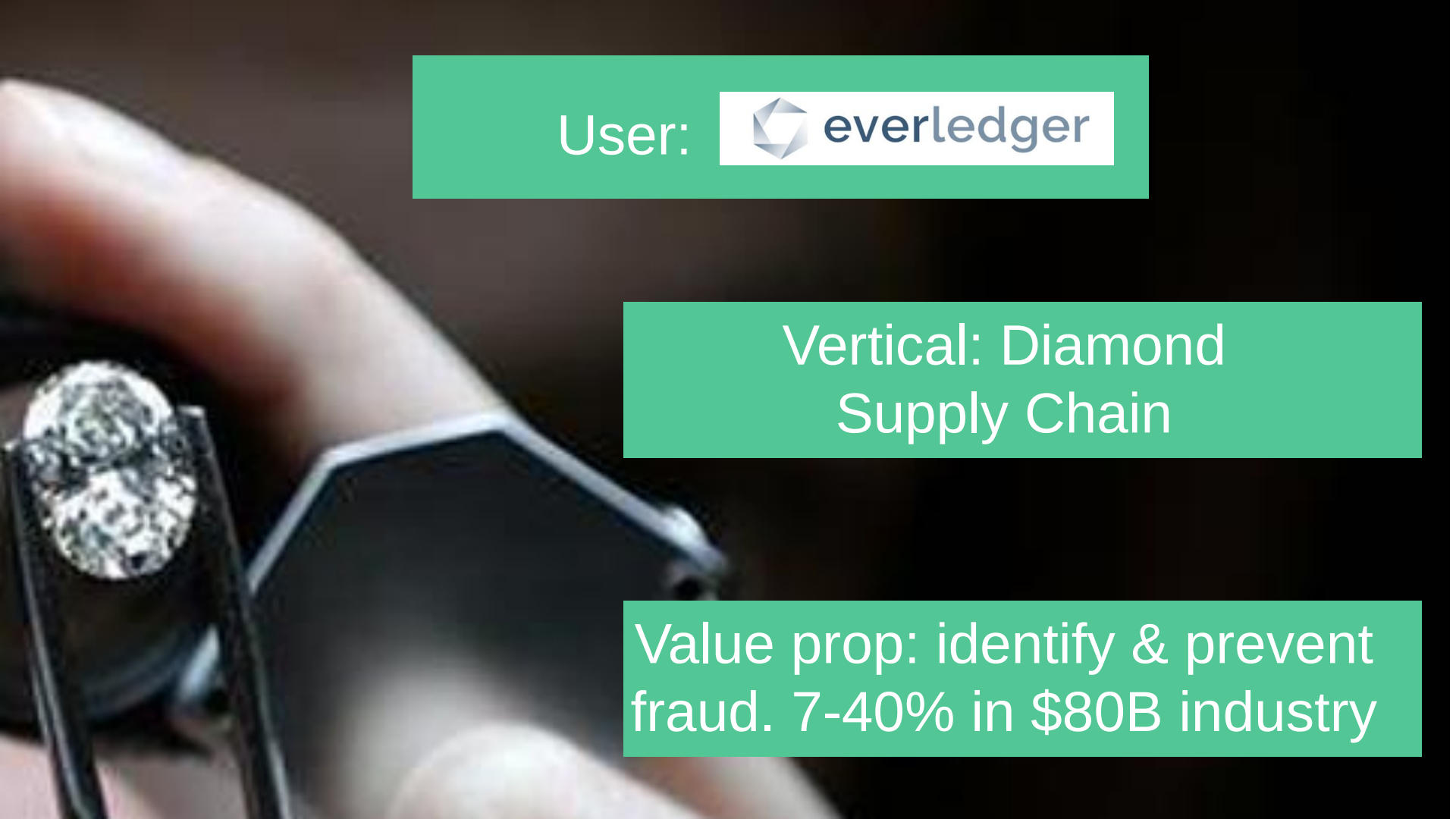
McConaghy, Trent Marques, Rodolphe Müller, Andreas
De Jonghe, Dimitri McConaghy, Troy McMullen, Greg
Henderson, Ryan Bellemare, Sylvain Granzotto, Alberto

April 7, 2016
ascribe GmbH, Berlin, Germany

www.bigchaindb.com/whitepaper

This paper desc
ization ecosystem
mance of 1 millio
and sub-second la
database (DB), and through a set of innovations adds blockchain characteristics: decentralized control, immutability, and creation & movement of digital assets. BigchainDB inherits characteristics of modern distributed databases:

	Traditional blockchains	Big Data	BIGCHAINDB
Immutability	✓		✓
Decentralized control	✓		✓
Assets	✓		✓
High Throughput		✓	✓
Low Latency		✓	✓
High Capacity		✓	✓
Rich Permissioning		✓	✓
Query Capabilities		✓	✓



User: everledger

Vertical: Diamond Supply Chain

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

User: RWE

Vertical: Energy Supply Chain

Value prop: manage $ flow in energy deregulation

User: Tangent90

Vertical: Medical Journals / Supply Chain

Value prop: government-mandated transparent $ flow

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

Verticals: Art Supply Chain, Intellectual Property

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

PYTHON & USAGE

https://www.bigchaindb.com

Quick Start Guide

Install and Run

BigchainDB provides a rich API to create, query and transfer digital assets.

DOCUMENTATION GITHUB

```
$ Dependencies: Loaded
$ (See bigchaindb.readthedocs.org)
$ Python: Python 3.4+
$ RethinkDB: Running
$ ---------------

  # install bigchaindb
$ pip install bigchaindb

  # start bigchaindb
$ bigchaindb start
```

Install and Run

BigchainDB provides a rich API to create, query and transfer digital assets.

DOCUMENTATION | GITHUB

```
$ Dependencies: Loaded
$ (See bigchaindb.readthedocs.org)
$ Python: Python 3.4+
$ RethinkDB: Running
$ ---------------

  # install bigchaindb
$ pip install bigchaindb

  # start bigchaindb
```

GitHub, Inc. [US] https://github.com/bigchaindb/bigchaindb

This repository | Search | Pull requests | Issues | Gist

bigchaindb / **bigchaindb**

Watch 55 | Unstar 464

Code | Issues 76 | Pull requests 6 | Wiki | Pulse | Graphs | Settings

A scalable blockchain database https://www.bigchaindb.com/ — Edit

793 commits | 71 branches | 8 releases | 15 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or dow

vrde Merge branch 'sbellem-docker-dev' — Latest commit 48325cc 2 h

benchmarking-tests	replace tx/s by `nan` for the first result	9
bigchaindb	Merge branch 'docker-dev' of git://github.com/sbellem/bigchaindb into...	2 h
deploy-cluster-aws	Use pip3 instead of pip to upgrade pip, wheel & setuptools	a
docs	Merge branch 'docker-dev' of git://github.com/sbellem/bigchaindb into...	2 h
speed-tests	replaced json with rapidjson	2

Install and Run

BigchainDB provides a rich API to create, query and transfer digital assets.

DOCUMENTATION | GITHUB

```
$ Dependencies: Loaded
$ (See bigchaindb.readthedocs.org)
$ Python: Python 3.4+
$ RethinkDB: Running
$ ---------------

  # install bigchaindb
$ pip install bigchaindb

  # start bigchaindb
```

https://bigchaindb.readthedocs.io/en/latest/

BigchainDB

latest

Search docs

Docs » BigchainDB Documentation

Edit on GitHub

BigchainDB Documentation

Table of Contents

Run

vides a rich API to create, query and
ssets.

TATION

GITHUB

```
Dependencies: Loaded
(See bigchaindb.readthedoc
Python: Python 3.4+
RethinkDB: Running
---------------

# install bigchaindb
pip install bigchaindb

# start bigchaindb
bigchaindb start
```

5. The Python Server API by Example

First, make sure you have RethinkDB and BigchainDB *installed and running*, i.e. you installed them and you ran:

```
$ rethinkdb
$ bigchaindb configure
$ bigchaindb start
```

Don't shut them down! In a new terminal, open a Python shell:

```
$ python
```

Now we can import the `Bigchain` class and create an instance:

```
from bigchaindb import Bigchain
b = Bigchain()
```

This instantiates an object `b` of class `Bigchain`. When instantiating a

5.2. Create a Digital Asset

```
from bigchaindb import crypto

# Create a test user
testuser1_priv, testuser1_pub = crypto.generate_key_pair()

# Define a digital asset data payload
digital_asset_payload = {'msg': 'Hello BigchainDB!'}

# A create transaction uses the operation `CREATE` and has no inputs
tx = b.create_transaction(b.me, testuser1_pub, None, 'CREATE', payload=digital_

# All transactions need to be signed by the user creating the transaction
tx_signed = b.sign_transaction(tx, b.me_private)

# Write the transaction to the bigchain.
# The transaction will be stored in a backlog where it will be validated,
# included in a block, and written to the bigchain
b.write_transaction(tx_signed)
```

5.3. Read the Creation Transaction from the DB

```
# Retrieve a transaction from the bigchain
tx_retrieved = b.get_transaction(tx_signed['id'])
tx_retrieved
```


```
{
    "id":"933cd83a419d2735822a2154c84176a2f419cbd449a74b94e592ab807af23861",
    "transaction":{
        "conditions":[
            {
                "cid":0,
                "condition":{
                    "details":{
                        "bitmask":32,
                        "public_key":"BwuhqQX8FPsmqYiRV2CSZYWWsSWgSSQQFHjqxKEuqh
                        "signature":None,
                        "type":"fulfillment",
                        "type_id":4
```

5.3. Read the Creation Transaction from the DB


```
        "data":{
            "hash":"872fa6e6f46246cd44afdb2ee9cfae0e72885fb0910e2bcf9a5a2a4eadb4
            "payload":{
                "msg":"Hello BigchainDB!"
            }
        },
        "fulfillments":[
            {
                "current_owners":[
                    "3LQ5dTiddXymDhNzETB1rEkp4mA7fEV1Qeiu5ghHiJm9"
                ],
                "fid":0,
                "fulfillment":"cf:4:Iq-BcczwraM2UpF-TDPdwK8fQ6IXkD_6uJaxBZd984y
                "input":None
            }
        ],
        "operation":"CREATE",
        "timestamp":"1460981667.449279"
    },
```

5.4. Transfer the Digital Asset

```
# Create a second testuser
testuser2_priv, testuser2_pub = crypto.generate_key_pair()
```

```
# Create a transfer transaction
tx_transfer = b.create_transaction(testuser1_pub, testuser2_pub, tx_

# Sign the transaction
tx_transfer_signed = b.sign_transaction(tx_transfer, testuser1_priv)

# Write the transaction
b.write_transaction(tx_transfer_signed)
```

5.5. Double Spends

BigchainDB makes sure that a user can't transfer the same digital asset two or more times (i.e. it prevents double spends).

If we try to create another transaction with the same input as before, the transaction will be marked invalid and the validation will throw a double spend exception:

```
# Create another transfer transaction with the same input
tx_transfer2 = b.create_transaction(testuser1_pub, testuser2_pub, tx_retrieved_i

# Sign the transaction
tx_transfer_signed2 = b.sign_transaction(tx_transfer2, testuser1_priv)

# Check if the transaction is valid
b.validate_transaction(tx_transfer_signed2)
```

```
DoubleSpend: input `{'cid': 0, 'txid': '933cd83a419d2735822a2154c84176a2f419cbd4
```

Decentralization of the Cloud

Centralized → Partly Decentralized → Fully Decentralized

Centralized	Partly Decentralized	Fully Decentralized
Apps	Partly Dec. Apps	Dec. Apps
Proc'ing	Proc'ing	Dec. Proc'ing
FS	FS	Dec. FS
DB	Dec. DB	Dec. DB

Apps: Facebook, NETFLIX, Google Maps

Proc'ing: EC2

FS: S3, Google Drive

DB: MySQL, mongoDB

Partly Dec. Apps: ascribe®

Proc'ing: EC2

FS: S3, Google Drive

Dec. DB: BIGCHAINDB

Dec. Apps: Dapp

Dec. Proc'ing: ethereum

Dec. FS: IPFS

Dec. DB: BIGCHAINDB

BigchainDB: A Scalable Blockchain Database, In Python

bigchaindb.com

bigchaindb.readthedocs.org

github.com/bigchaindb