About Some Of The Blockchain Problems

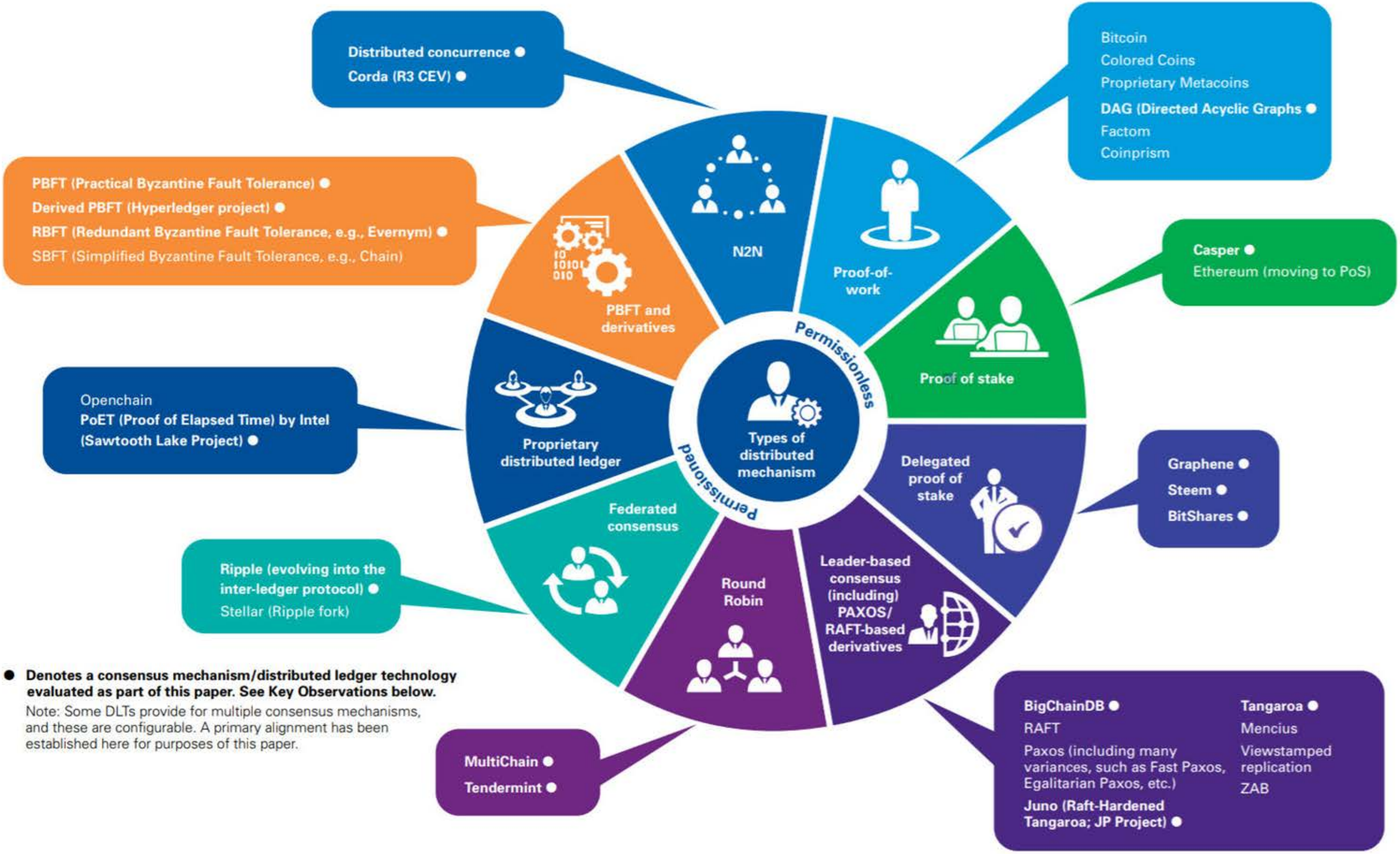
Alexander Bogdanov, Alexander Degtyarev, Magdalyne Kamande, Oleg Iakushkin, Vladimir Korkhov

Problem Types

- Distributed Information sharing
  - Privacy
  - Consistency
  - Volume

- Transactions exchange
  - Speed
  - Flexibility
  - Cost

- Mode of operation
  - Reversibility
  - Checkpoints
  - Fraud detection



Types of distributed mechanism

Permissionless

Permissioned

N2N

Distributed concurrence ●
Corda (R3 CEV) ●

Proof-of-work

Bitcoin
Colored Coins
Proprietary Metacoins
DAG (Directed Acyclic Graphs ●
Factom
Coinprism

Proof of stake

Casper ●
Ethereum (moving to PoS)

Delegated proof of stake

Graphene ●
Steem ●
BitShares ●

Leader-based consensus (including) PAXOS/ RAFT-based derivatives

BigChainDB ●
RAFT
Paxos (including many variances, such as Fast Paxos, Egalitarian Paxos, etc.)
Juno (Raft-Hardened Tangaroa; JP Project) ●

Tangaroa ●
Mencius
Viewstamped replication
ZAB

Round Robin

MultiChain ●
Tendermint ●

Federated consensus

Ripple (evolving into the inter-ledger protocol) ●
Stellar (Ripple fork)

Proprietary distributed ledger

Openchain
PoET (Proof of Elapsed Time) by Intel (Sawtooth Lake Project) ●

PBFT and derivatives

PBFT (Practical Byzantine Fault Tolerance) ●
Derived PBFT (Hyperledger project) ●
RBFT (Redundant Byzantine Fault Tolerance, e.g., Evernym) ●
SBFT (Simplified Byzantine Fault Tolerance, e.g., Chain)

● **Denotes a consensus mechanism/distributed ledger technology evaluated as part of this paper. See Key Observations below.**

Note: Some DLTs provide for multiple consensus mechanisms, and these are configurable. A primary alignment has been established here for purposes of this paper.

Internet of Value

https://bgx.ai/

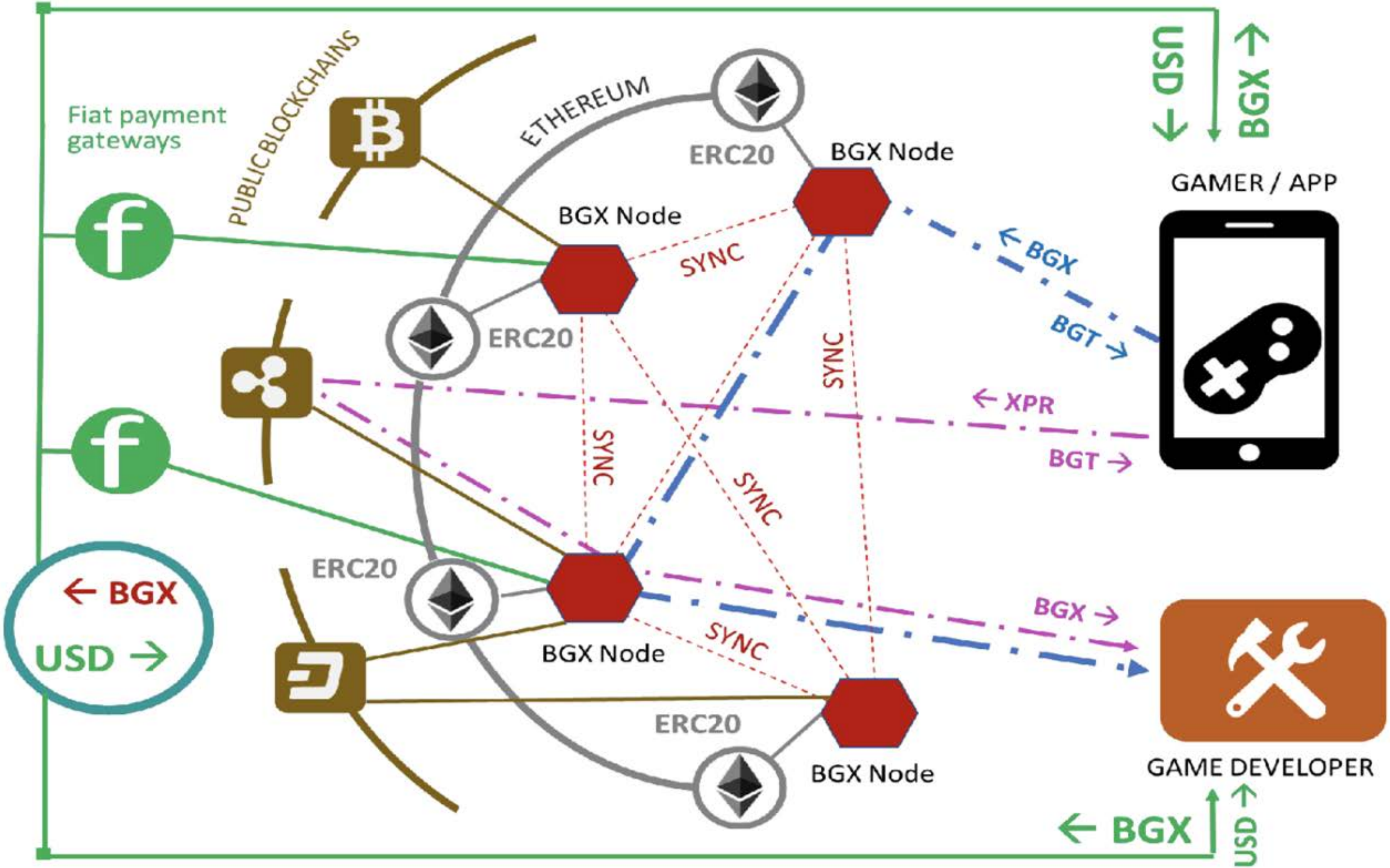
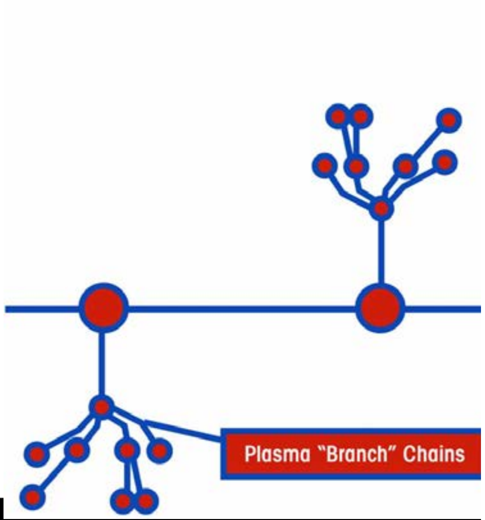
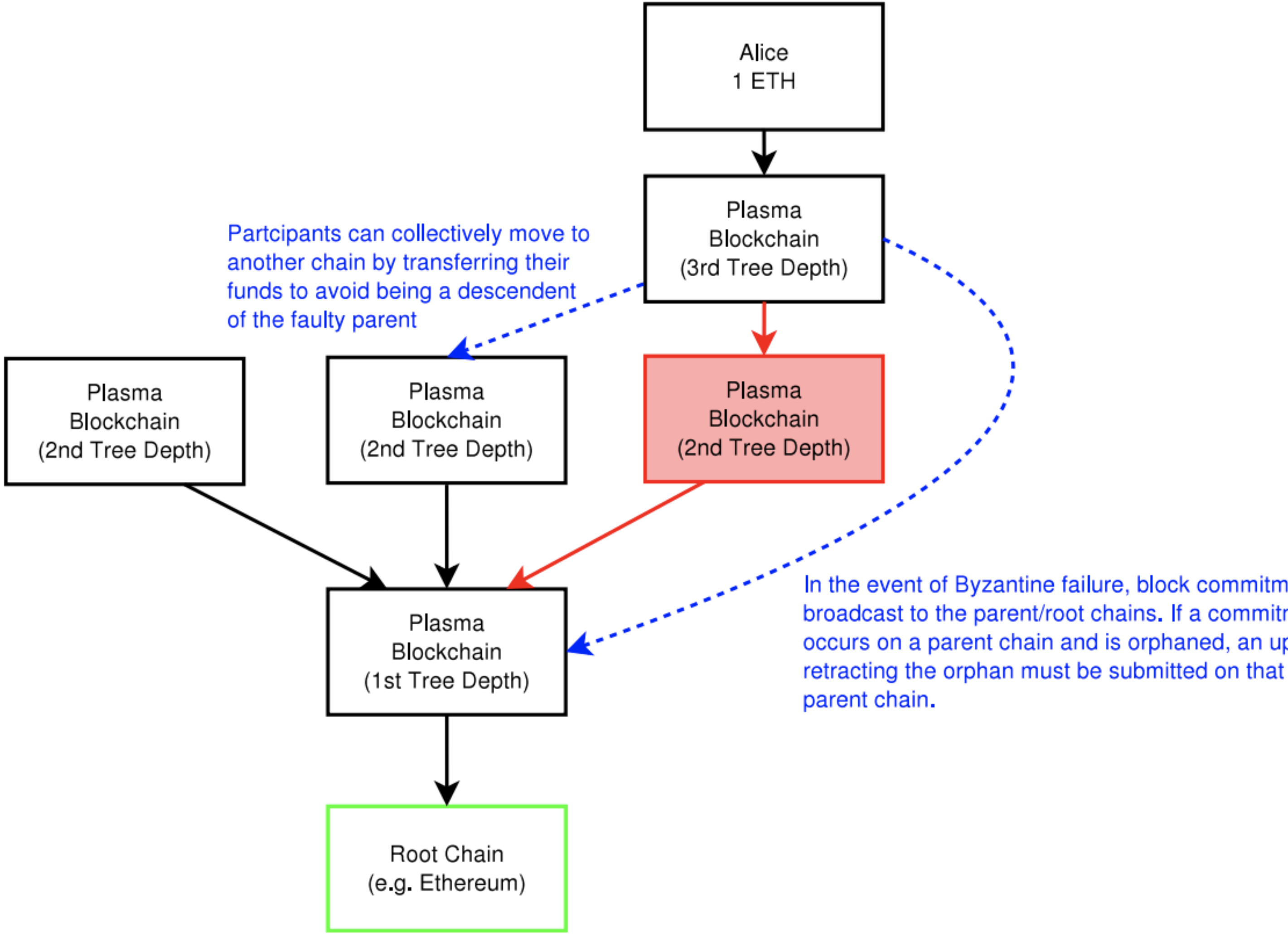


Figure 3 Transaction Flow

Chain Trees

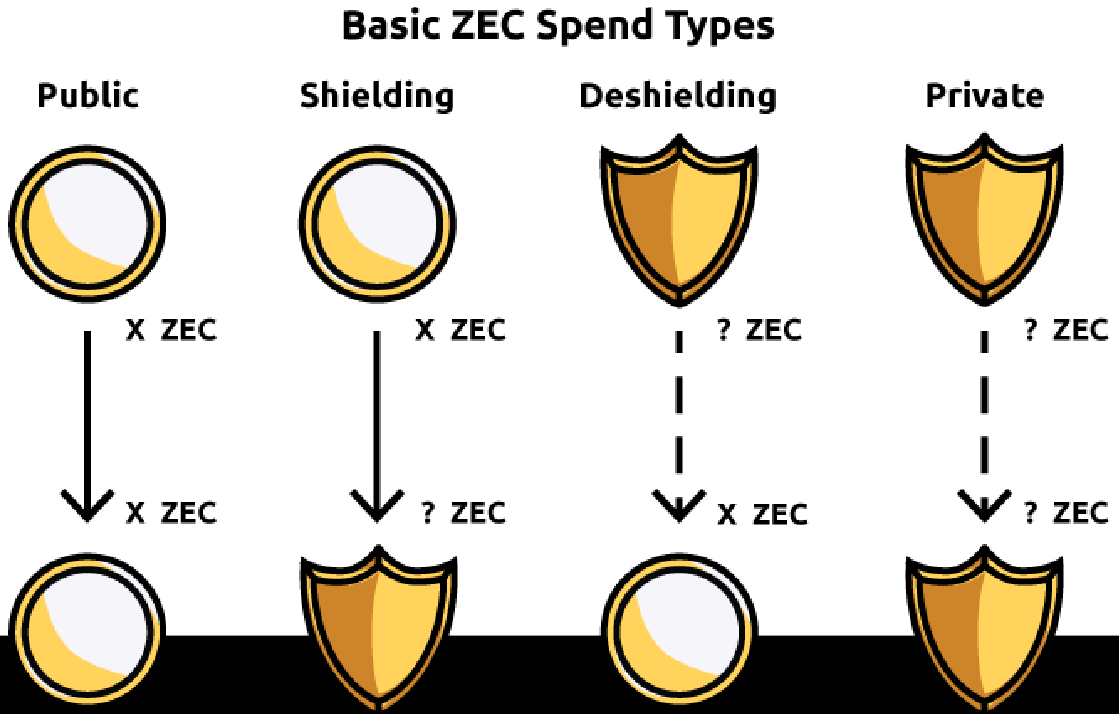




Partcipants can collectively move to another chain by transferring their funds to avoid being a descendent of the faulty parent

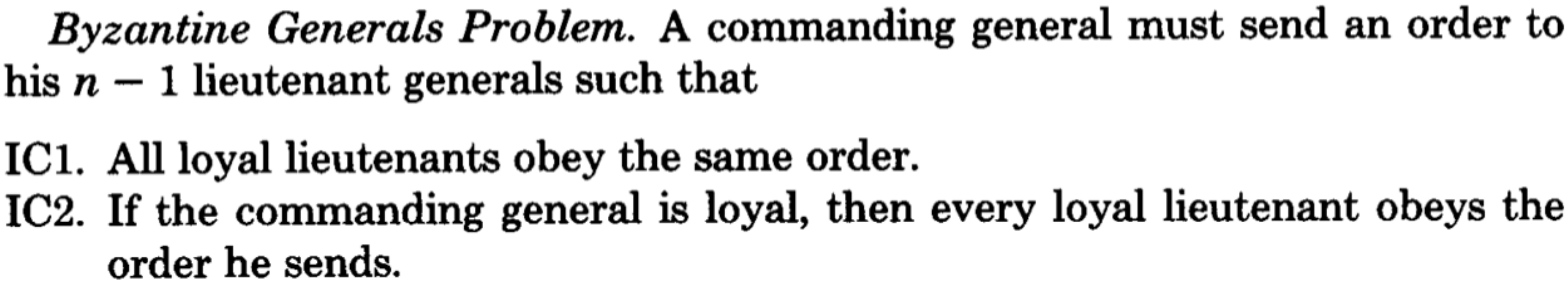
In the event of Byzantine failure, block commitm broadcast to the parent/root chains. If a commitr occurs on a parent chain and is orphaned, an up retracting the orphan must be submitted on that parent chain.

Privacy - Shielding



Basic ZEC Spend Types

Byzantine failures

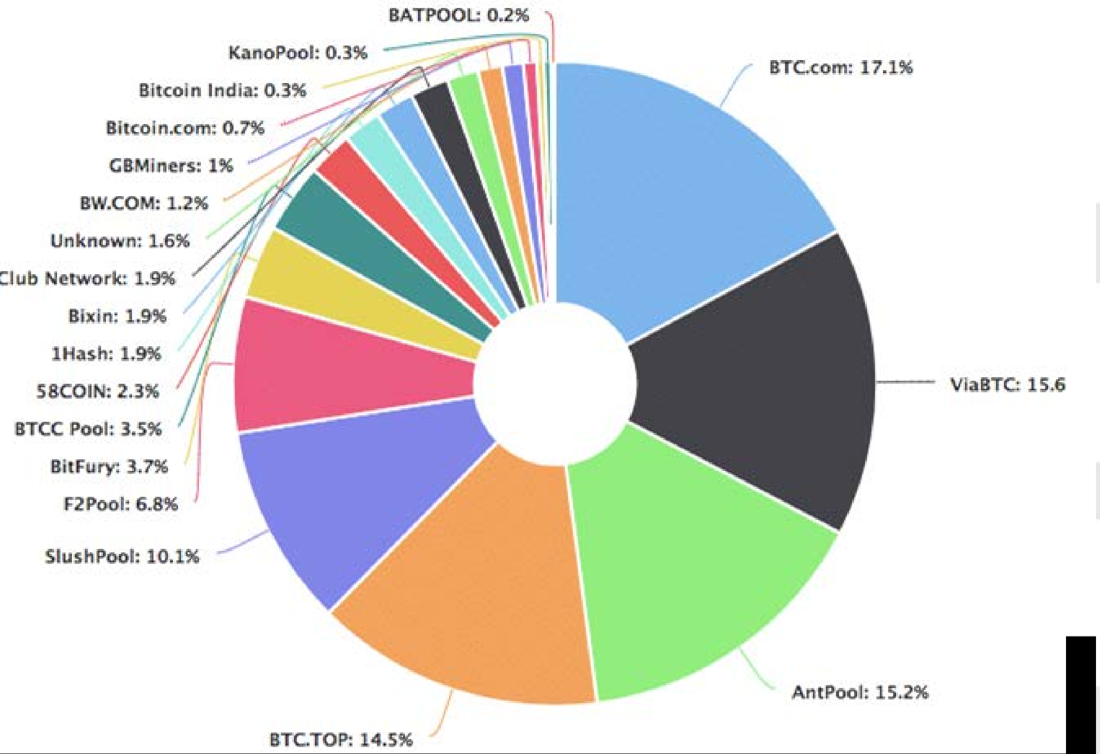


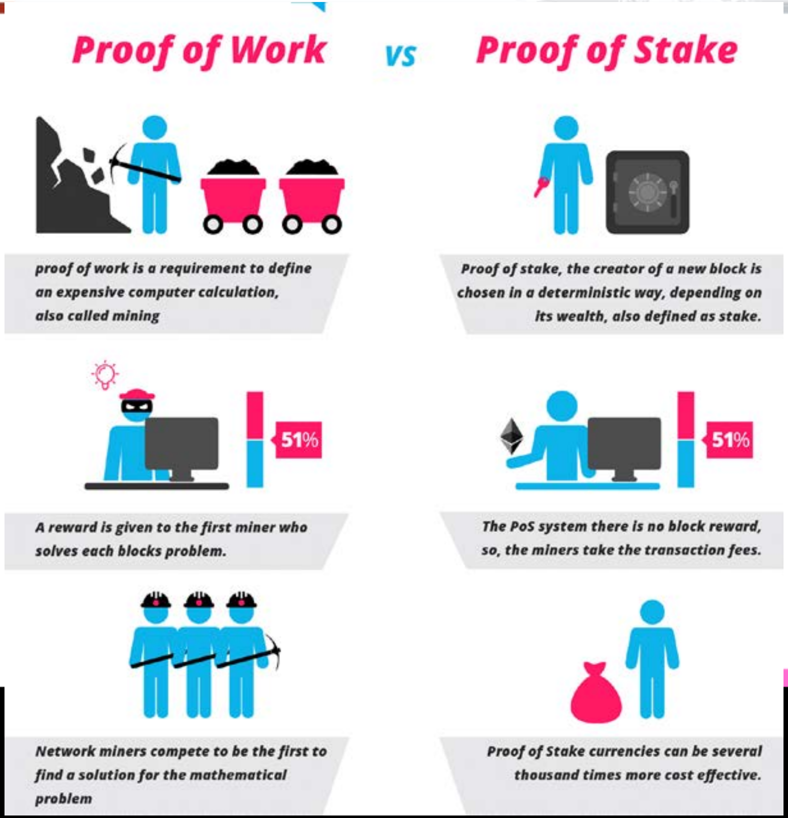
Byzantine Generals Problem. A commanding general must send an order to his $n - 1$ lieutenant generals such that

IC1. All loyal lieutenants obey the same order.
IC2. If the commanding general is loyal, then every loyal lieutenant obeys the order he sends.

- **Byzantine fault tolerance model**: a certain percentage of all miners are attackers, and the rest are honest altruistic people.
- **Economic model**: there is an attacker with a budget of $X which the attacker can spend to either purchase their own hardware or bribe other users, who are rational.

PoW vs PoS





proof of work is a requirement to define an expensive computer calculation, also called mining

Proof of stake, the creator of a new block is chosen in a deterministic way, depending on its wealth, also defined as stake.

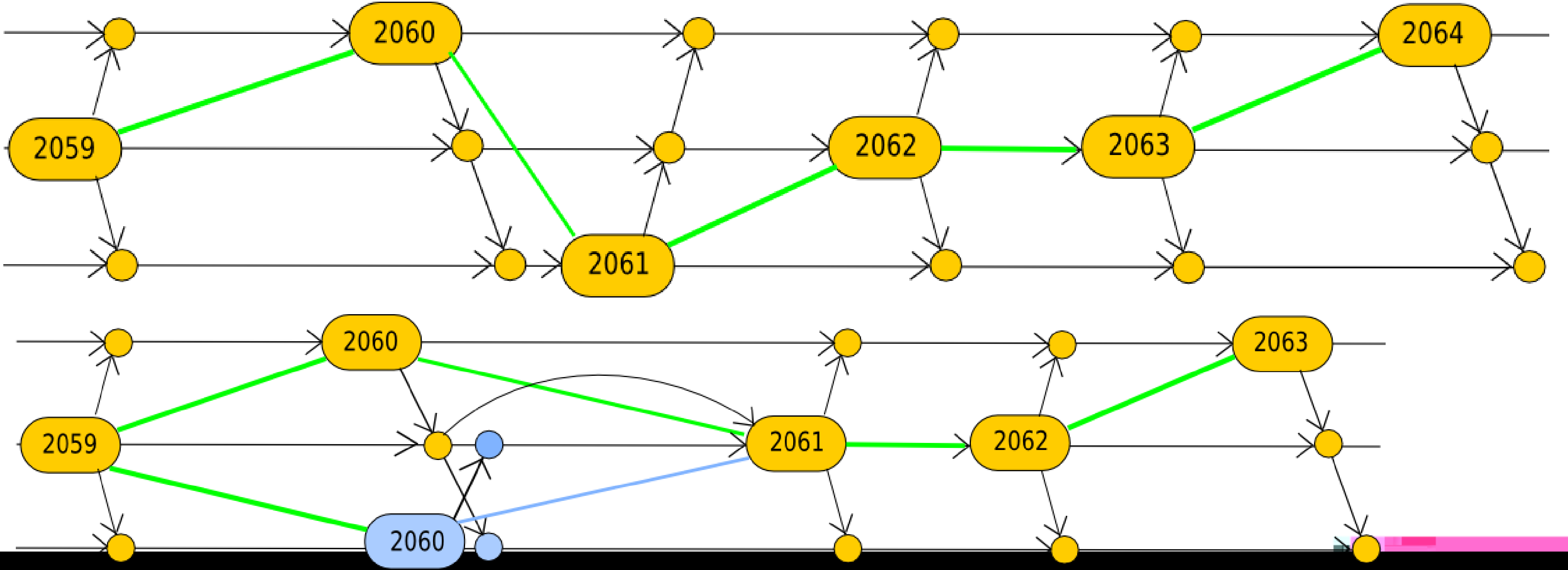
A reward is given to the first miner who solves each blocks problem.

The PoS system there is no block reward, so, the miners take the transaction fees.

Network miners compete to be the first to find a solution for the mathematical problem

Proof of Stake currencies can be several thousand times more cost effective.

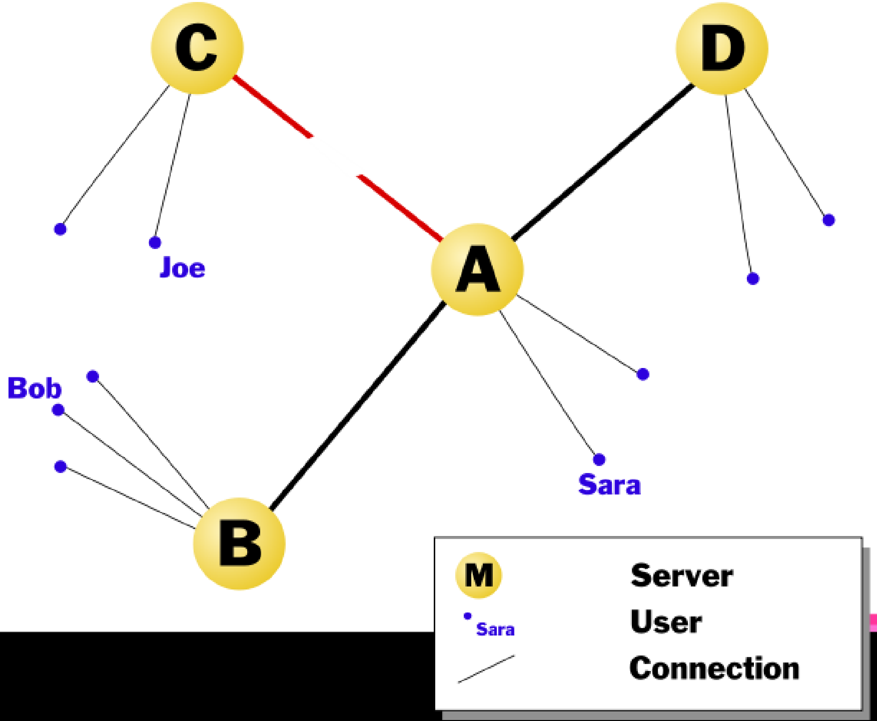
Uncle



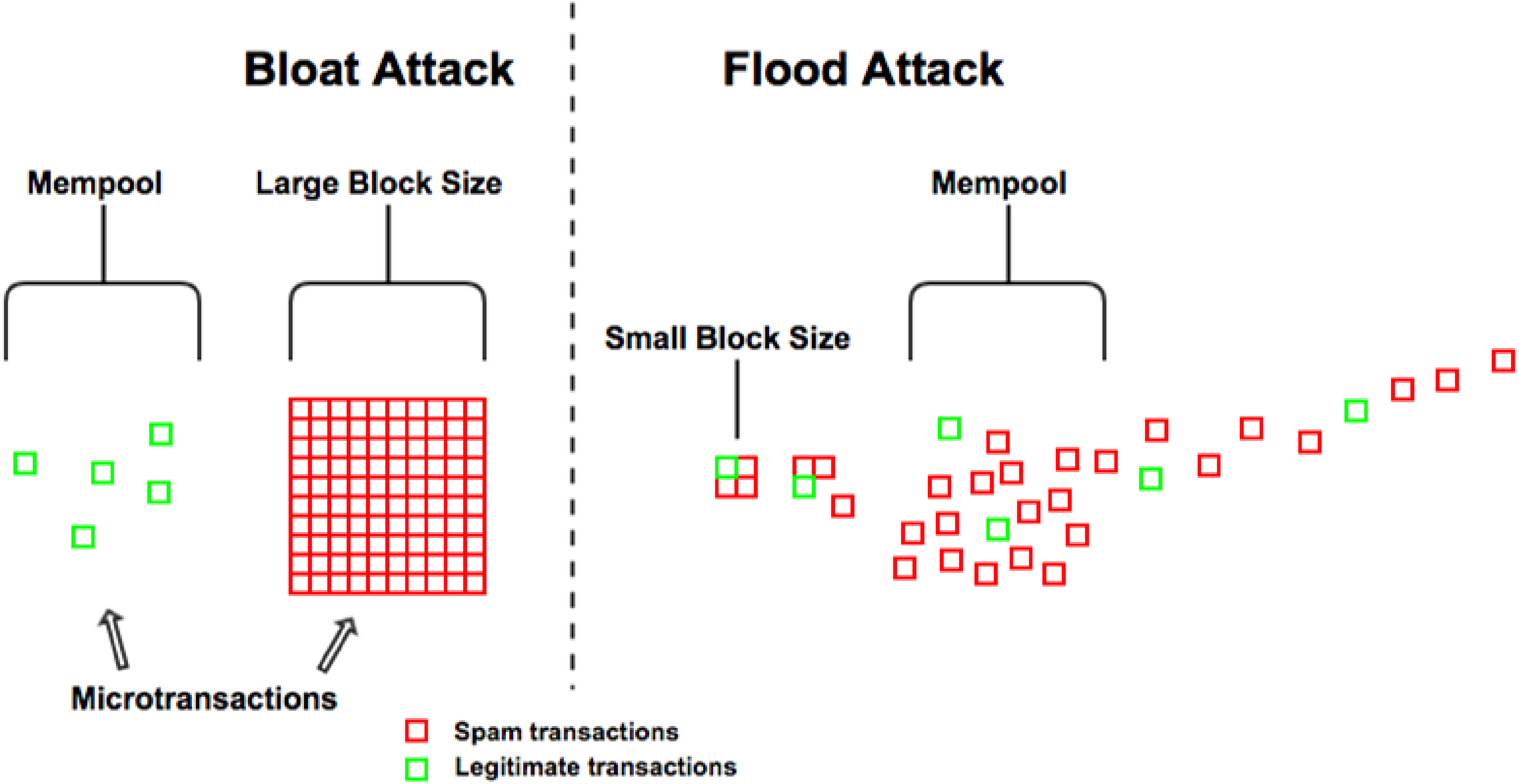
"uncle" is defined as a block with a valid header (the block itself need not be valid, since we only care about its proof-of-work) which is the child of the parent of the parent of the block but not the parent

Network Split Attack

when someone broadcasts a transaction using one of the networks, there is a risk that that transaction gets included in all “forked” blockchains

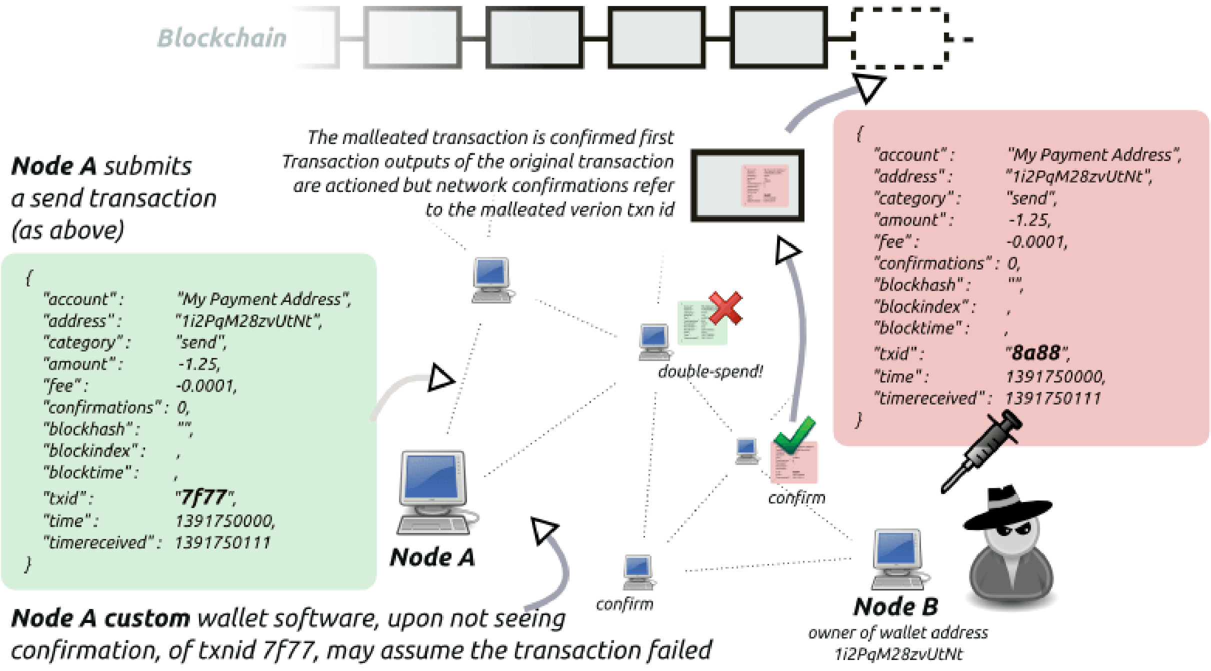


Denial-Of-Service Attack



Injection

Malleated Transaction ID injection



***Node A** submits
a send transaction
(as above)*

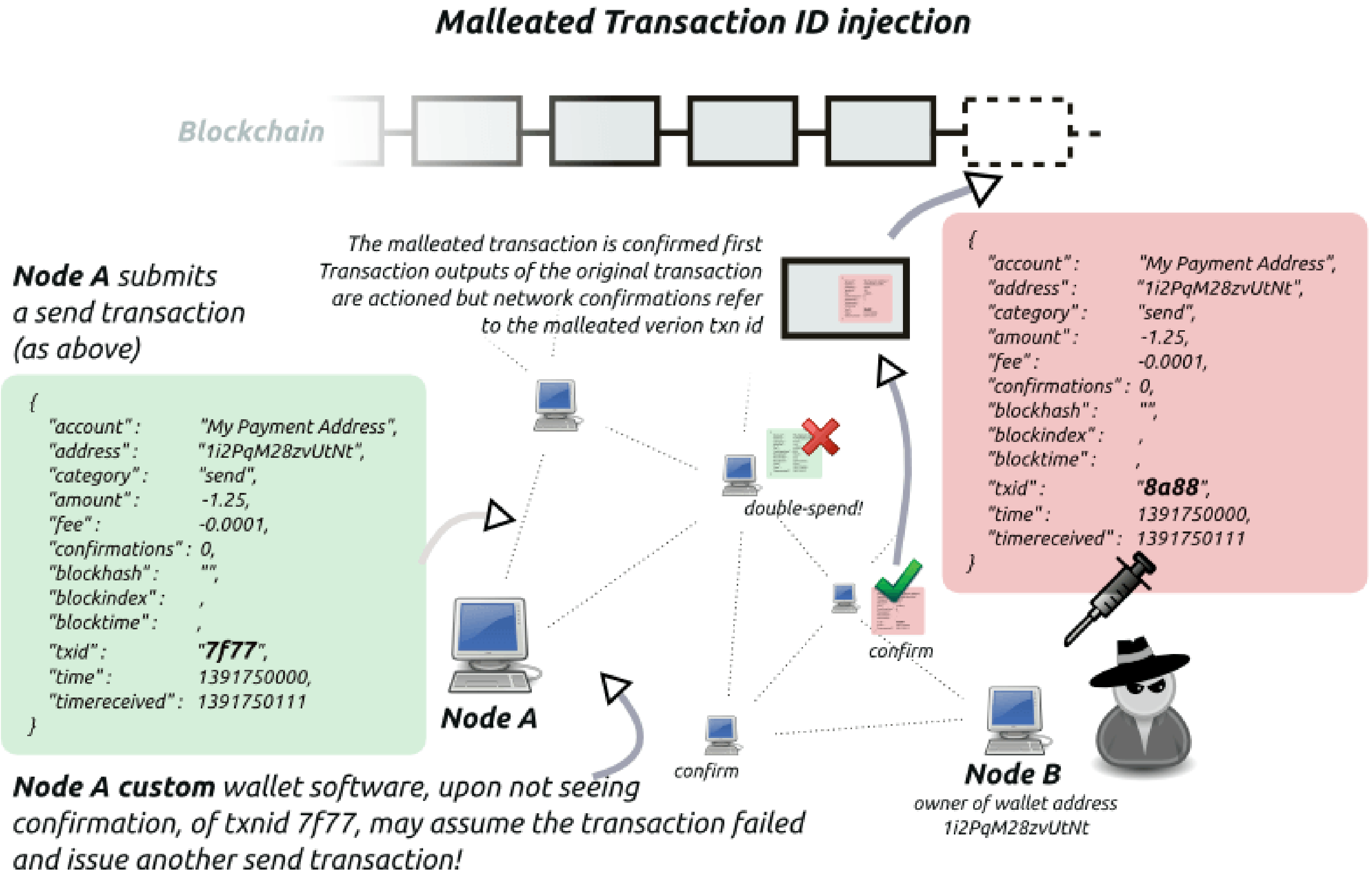
```
{
    "account" :          "My Payment Address",
    "address" :          "1i2PqM28zvUtNt",
    "category" :         "send",
    "amount" :           -1.25,
    "fee" :              -0.0001,
    "confirmations" :  0,
    "blockhash" :        "",
    "blockindex" :       ,
    "blocktime" :        ,
    "txid" :             "7f77",
    "time" :             1391750000,
    "timereceived" :  1391750111
}
```

*The malleated transaction is confirmed first
Transaction outputs of the original transaction
are actioned but network confirmations refer
to the malleated verion txn id*

```
{
    "account" :          "My Payment Address",
    "address" :          "1i2PqM28zvUtNt",
    "category" :         "send",
    "amount" :           -1.25,
    "fee" :              -0.0001,
    "confirmations" :  0,
    "blockhash" :        "",
    "blockindex" :       ,
    "blocktime" :        ,
    "txid" :             "8a88",
    "time" :             1391750000,
    "timereceived" :  1391750111
}
```

***Node A custom** wallet software, upon not seeing
confirmation, of txnid 7f77, may assume the transaction failed
and issue another send transaction!*

Injection



Malleated Transaction ID injection

Node A submits
a send transaction
(as above)

```
{
  "account" :         "My Payment Address",
  "address" :         "1i2PqM28zvUtNt",
  "category" :        "send",
  "amount" :          -1.25,
  "fee" :             -0.0001,
  "confirmations" :   0,
  "blockhash" :       "",
  "blockindex" :      ,
  "blocktime" :       ,
  "txid" :            "7f77",
  "time" :            1391750000,
  "timereceived" :    1391750111
}
```

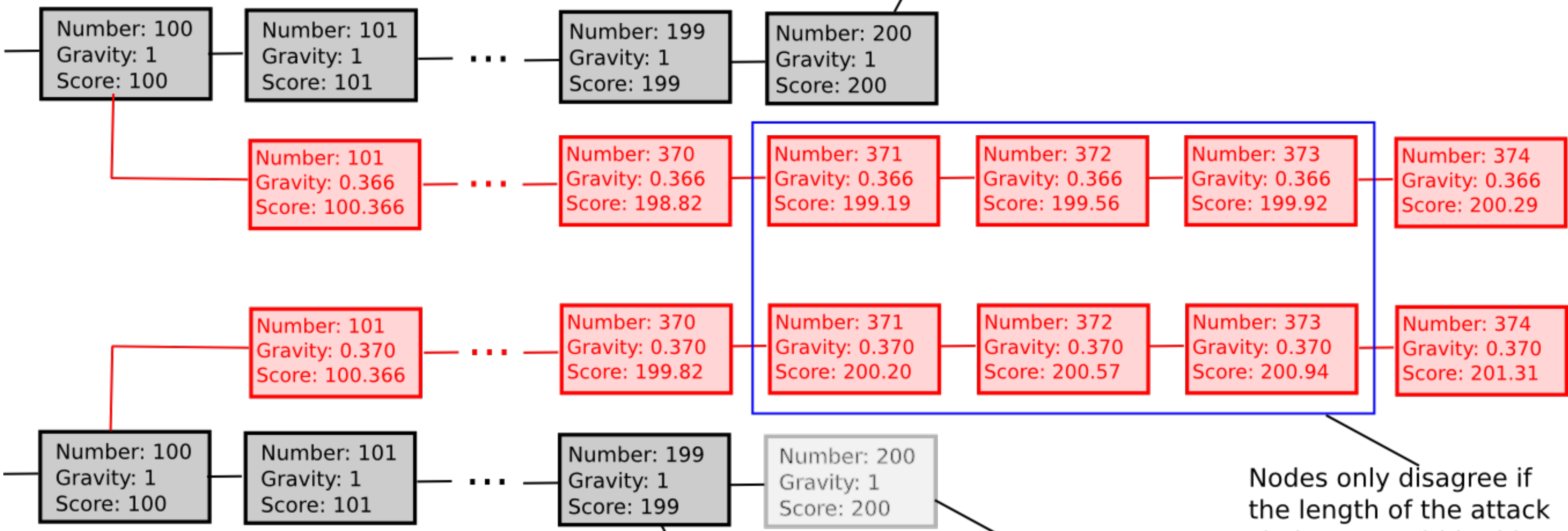
The malleated transaction is confirmed first
Transaction outputs of the original transaction
are actioned but network confirmations refer
to the malleated verion txn id

```
{
  "account" :         "My Payment Address",
  "address" :         "1i2PqM28zvUtNt",
  "category" :        "send",
  "amount" :          -1.25,
  "fee" :             -0.0001,
  "confirmations" :   0,
  "blockhash" :       "",
  "blockindex" :      ,
  "blocktime" :       ,
  "txid" :            "8a88",
  "time" :            1391750000,
  "timereceived" :    1391750111
}
```

Node B
owner of wallet address
1i2PqM28zvUtNt

Node A custom wallet software, upon not seeing confirmation, of txnid 7f77, may assume the transaction failed and issue another send transaction!

Distributed Concensus Problem



Original chain, node 1

Number: 100
Gravity: 1
Score: 100

Number: 101
Gravity: 1
Score: 101

...

Number: 199
Gravity: 1
Score: 199

Number: 200
Gravity: 1
Score: 200

Number: 101
Gravity: 0.366
Score: 100.366

...

Number: 370
Gravity: 0.366
Score: 198.82

Number: 371
Gravity: 0.366
Score: 199.19

Number: 372
Gravity: 0.366
Score: 199.56

Number: 373
Gravity: 0.366
Score: 199.92

Number: 374
Gravity: 0.366
Score: 200.29

Number: 101
Gravity: 0.370
Score: 100.366

...

Number: 370
Gravity: 0.370
Score: 199.82

Number: 371
Gravity: 0.370
Score: 200.20

Number: 372
Gravity: 0.370
Score: 200.57

Number: 373
Gravity: 0.370
Score: 200.94

Number: 374
Gravity: 0.370
Score: 201.31

Number: 100
Gravity: 1
Score: 100

Number: 101
Gravity: 1
Score: 101

...

Number: 199
Gravity: 1
Score: 199

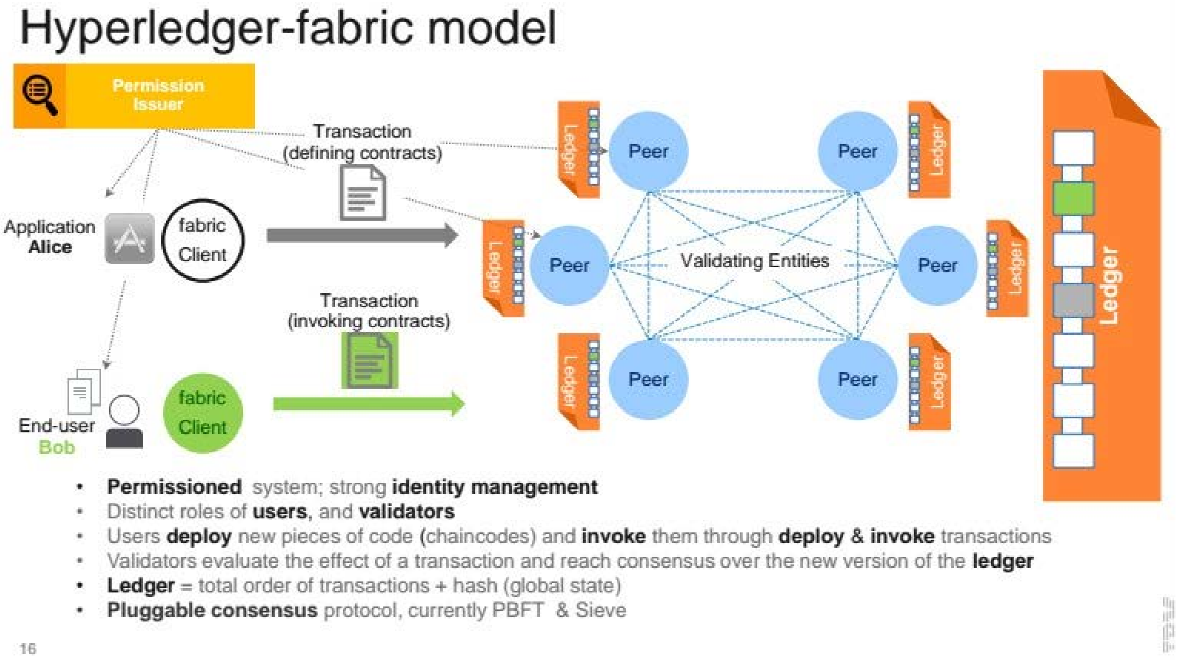
Number: 200
Gravity: 1
Score: 200

Original chain, node 2

Node 2 hears about this block later

Nodes only disagree if the length of the attack chain stays within this window (which slides right as the main chain lengthens)

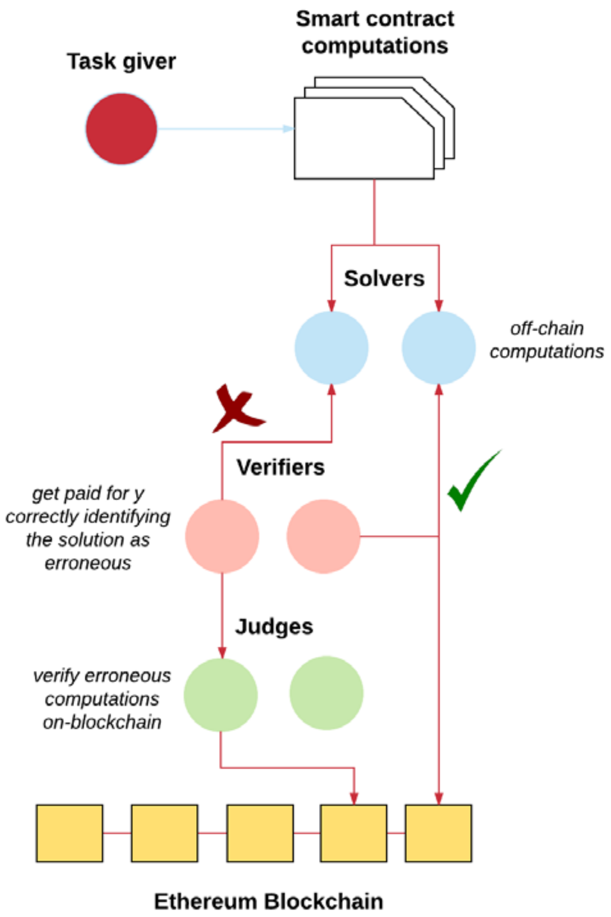
Off-Chain Computation



Hyperledger-fabric model

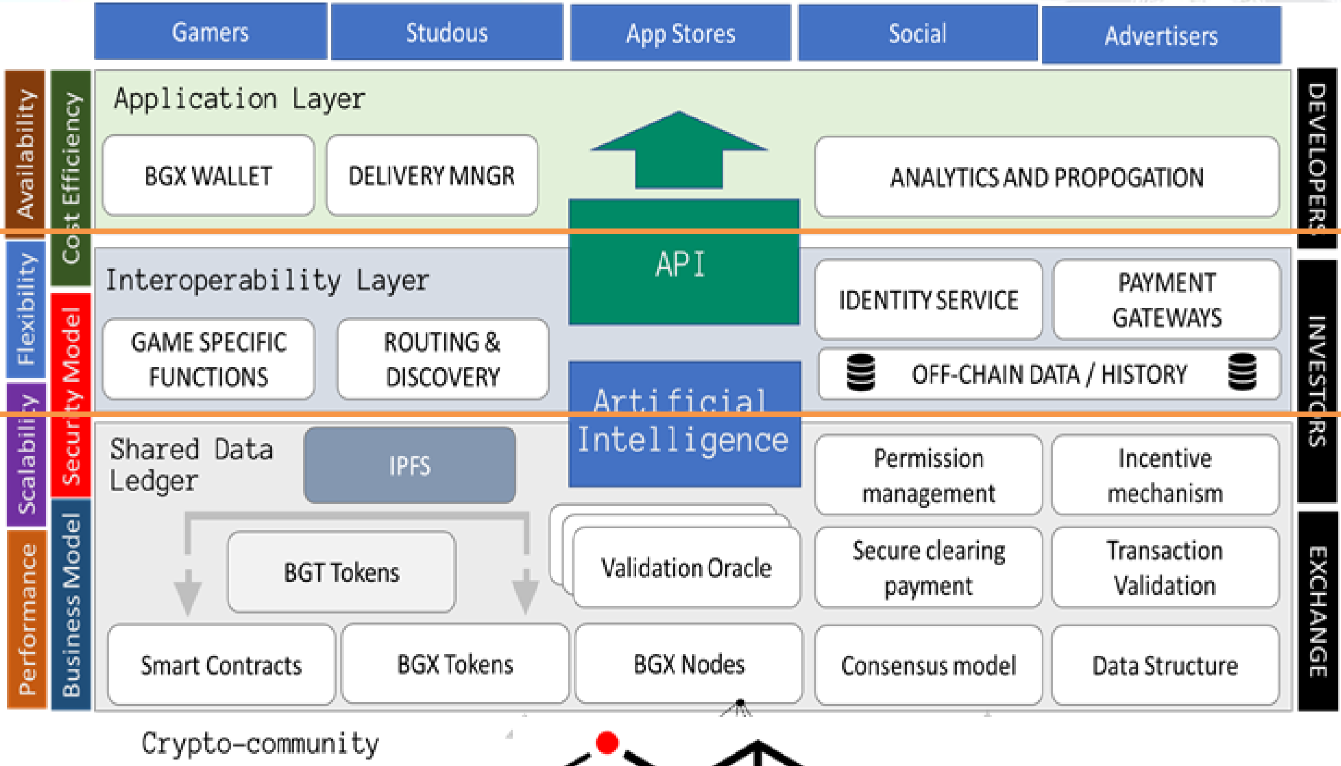
- **Permissioned** system; strong **identity management**
- Distinct roles of **users**, and **validators**
- Users **deploy** new pieces of code (chaincodes) and **invoke** them through **deploy** & **invoke** transactions
- Validators evaluate the effect of a transaction and reach consensus over the new version of the **ledger**
- **Ledger** = total order of transactions + hash (global state)
- **Pluggable consensus** protocol, currently PBFT & Sieve

16



Virtualization For Scalability

Virtual supercomputer



BGX

THANK YOU FOR ATTENTION!

Alexander Bogdanov, Alexander Degtyarev, Magdalyne Kamande, Oleg Iakushkin,
Vladimir Korkhov