Fast and secure global payments with Stellar

Marta Lokhava, Giuliano Losa (Galois), David Mazières, Graydon Hoare,
Nicolas Barry, Eli Gafni (UCLA), Jonathan Jove, Rafał Malinowsky, and Jed McCaleb

Monday, October 28, 2019

Things we take for granted

A bank account in a stable currency such as USD

Access to well-regulated investments

Cheap international money transfers

Globally accepted, fee-free credit cards

Things we take for granted

Vanguard®

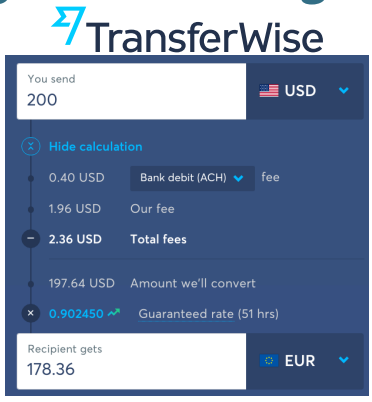
A bank account in a stable currency such as USD

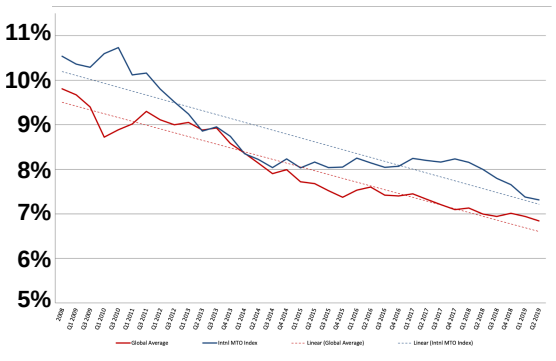
Access to well-regulated investments

Cheap international money transfers

Globally accepted, fee-free credit cards

Things we take for granted





A bank account in a stable currency such as USD

Access to well-regulated investments

Cheap international money transfers

Globally accepted, fee-free credit cards

Things we take for granted

All No Foreign Transaction Fee Cards

m-pesa

A bank account in a stable currency such as USD

Access to well-regulated investments

Cheap international money transfers

Globally accepted, fee-free credit cards

Stellar: equitable access to assets

1. **Open membership**
   - Anyone can issue, trade, and hold assets
   - All developers access the same API, from Ph.D. students to Franklin Templeton or IBM
2. **Issuer-enforced finality**
   - Security of issued tokens depends only on issuer (what we expect today)
   - Still need secure servers, but issuer owns or designates them
3. **Cross-issuer atomicity**
   - Trade any asset for any other (ensures you can bootstrap markets)
   - Get the best price on any trade without trusting your trading partner
   - Atomically trade through multiple assets w/o exchange-rate risk
     (E.g., trade NGN $\rightarrow$ Sketchy-Asset $\rightarrow$ PHP with no risk from Sketchy-Asset)

Non-solutions

Nacha

中国人民银行
THE PEOPLE'S BANK OF CHINA

Extend national payment network (ACH, SEPA, UPI) globally

- Requires compliance with national regulations, closed to new assets

Everyone just issues and manages their own assets

- Can't pay or trade across systems, closed to new assets

Move Paypal onto Ethereum as an ERC-20 token

- Double redemption risk not under issuer's control

Non-solutions

微信支付
WeChat Pay

Extend national payment network (ACH, SEPA, UPI) globally

- Requires compliance with national regulations, closed to new assets

Everyone just issues and manages their own assets

- Can't pay or trade across systems, closed to new assets

Move Paypal onto Ethereum as an ERC-20 token

- Double redemption risk not under issuer's control

Non-solutions

Name	Symbol	Market Cap	Algorithm	Hash Rate	1h Attack Cost	NiceHash-able
Bitcoin	BTC	$188.00 B	SHA-256	78,549 PH/s	*$765,484*	0%
Ethereum	ETH	$19.26 B	Ethash	169 TH/s	*$95,684*	2%
BitcoinCashABC	BCH	$5.47 B	SHA-256	2,301 PH/s	*$22,422*	1%
Litecoin	LTC	$4.38 B	Scrypt	303 TH/s	*$20,501*	2%
BitcoinSV	BSV	$2.41 B	SHA-256	958 PH/s	*$9,337*	3%
Monero	XMR	$1.34 B	CryptoNightR	304 MH/s	*$4,619*	2%
EthereumClassic	ETC	$766.86 M	Ethash	12 TH/s	*$6,823*	31%

Extend national payment network (ACH, SEPA, UPI) globally

- Requires compliance with national regulations, closed to new assets

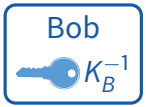
Everyone just issues and manages their own assets

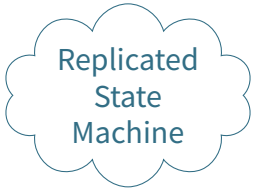
- Can’t pay or trade across systems, closed to new assets

Move Paypal onto Ethereum as an ERC-20 token

- Double redemption risk not under issuer’s control

Stellar transaction model





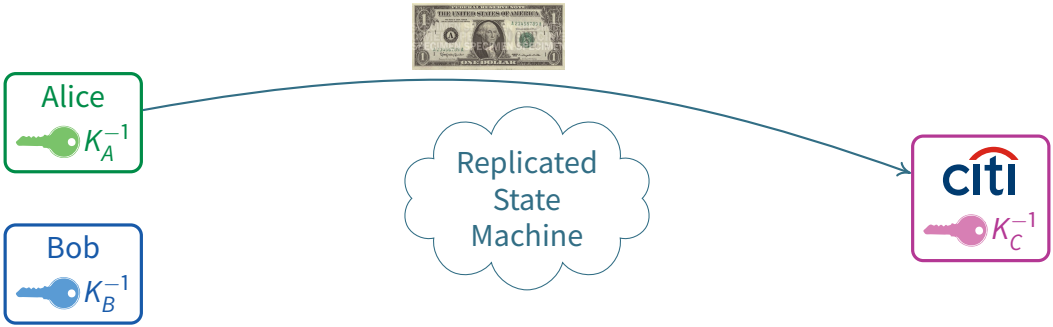
Global replicated state machine (RSM) executes transactions to keep ledger state

- Accounts named by public key authorizing operations on the account
- Accounts can issue assets; issuing account part of asset name

Transactions guarantee atomicity

- Multiple operations from multiple accounts with either all succeed or all fail
- *Path payments* atomically trade through multiple assets (e.g., $1\,K_D\$ \rightarrow 1\,K_C\$ \rightarrow 1\,K_B$ babysit)

Stellar transaction model



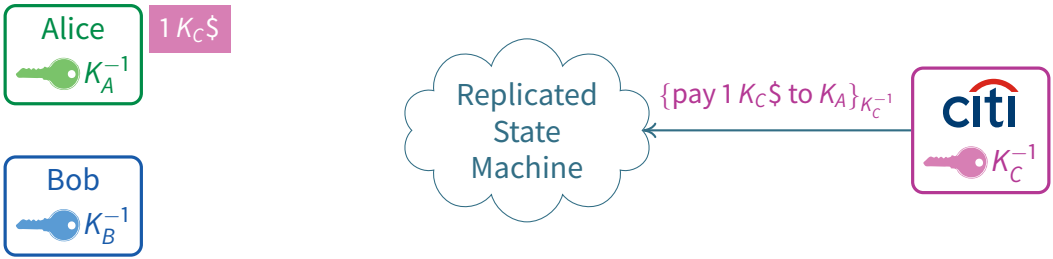
Global replicated state machine (RSM) executes transactions to keep ledger state

- Accounts named by public key authorizing operations on the account
- Accounts can issue assets; issuing account part of asset name

Transactions guarantee atomicity

- Multiple operations from multiple accounts with either all succeed or all fail
- *Path payments* atomically trade through multiple assets (e.g., $1\,K_D\$ \rightarrow 1\,K_C\$ \rightarrow 1\,K_B$ babysit)

Stellar transaction model



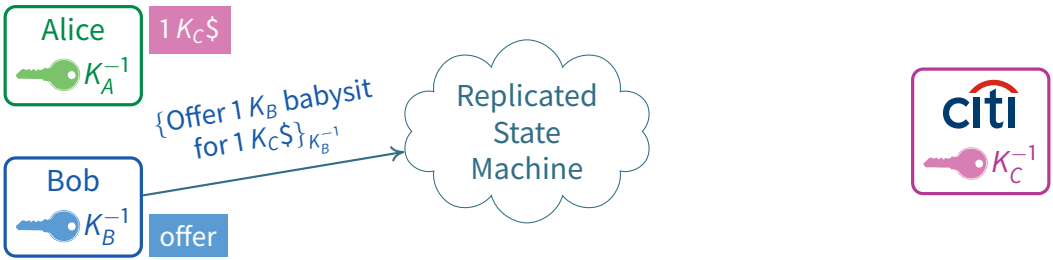
Global replicated state machine (RSM) executes transactions to keep ledger state

- Accounts named by public key authorizing operations on the account
- Accounts can issue assets; issuing account part of asset name

Transactions guarantee atomicity

- Multiple operations from multiple accounts with either all succeed or all fail
- *Path payments* atomically trade through multiple assets (e.g., $1\,K_D\$ \rightarrow 1\,K_C\$ \rightarrow 1\,K_B$ babysit)

Stellar transaction model



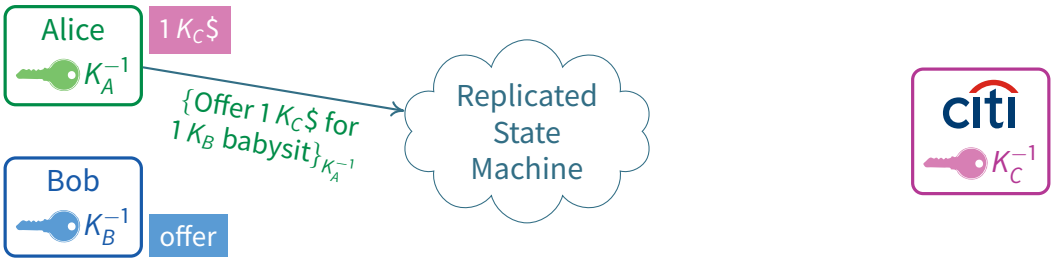
Global replicated state machine (RSM) executes transactions to keep ledger state

- Accounts named by public key authorizing operations on the account
- Accounts can issue assets; issuing account part of asset name

Transactions guarantee atomicity

- Multiple operations from multiple accounts with either all succeed or all fail
- *Path payments* atomically trade through multiple assets (e.g., $1\,K_D\$ \rightarrow 1\,K_C\$ \rightarrow 1\,K_B$ babysit)

Stellar transaction model



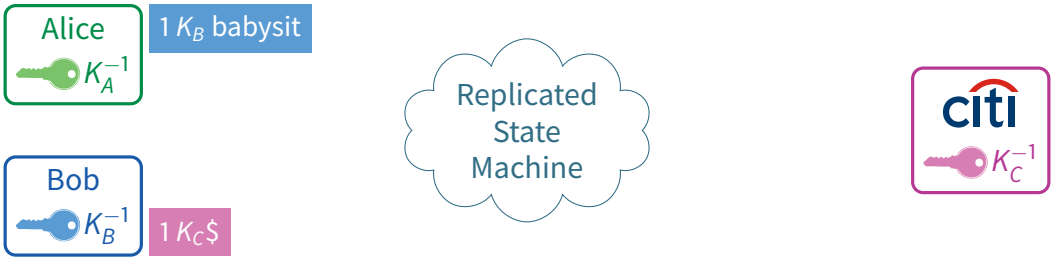
Global replicated state machine (RSM) executes transactions to keep ledger state

- Accounts named by public key authorizing operations on the account
- Accounts can issue assets; issuing account part of asset name

Transactions guarantee atomicity

- Multiple operations from multiple accounts with either all succeed or all fail
- *Path payments* atomically trade through multiple assets (e.g., 1 $K_D$$ $\rightarrow$ 1 $K_C$$ $\rightarrow$ 1 K_B babysit)

Stellar transaction model



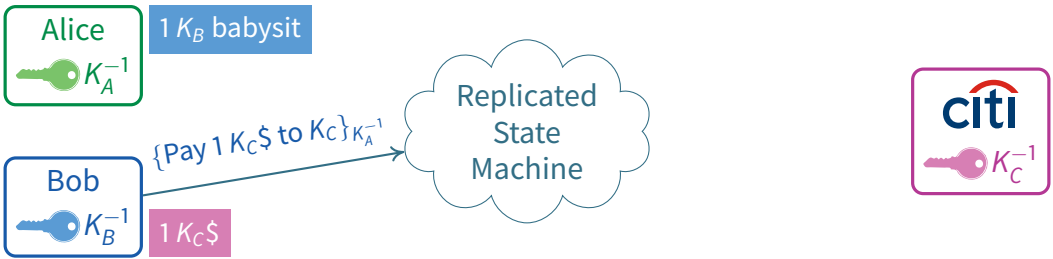
Global replicated state machine (RSM) executes transactions to keep ledger state

- Accounts named by public key authorizing operations on the account
- Accounts can issue assets; issuing account part of asset name

Transactions guarantee atomicity

- Multiple operations from multiple accounts with either all succeed or all fail
- *Path payments* atomically trade through multiple assets (e.g., $1\,K_D\$ \rightarrow 1\,K_C\$ \rightarrow 1\,K_B$ babysit)

Stellar transaction model



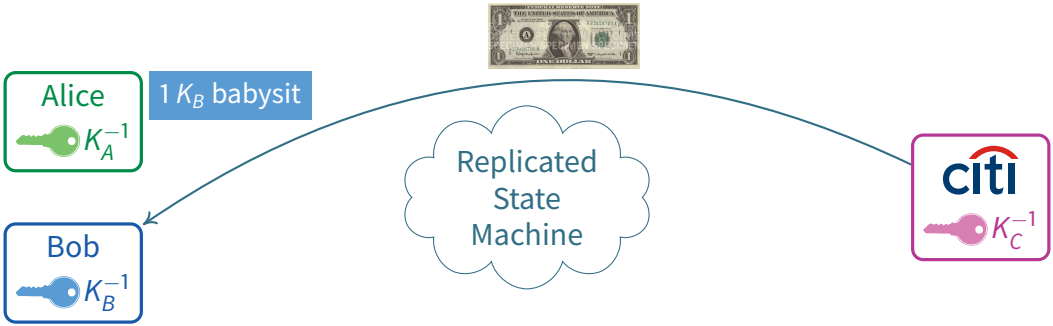
Global replicated state machine (RSM) executes transactions to keep ledger state

- Accounts named by public key authorizing operations on the account
- Accounts can issue assets; issuing account part of asset name

Transactions guarantee atomicity

- Multiple operations from multiple accounts with either all succeed or all fail
- *Path payments* atomically trade through multiple assets (e.g., $1\,K_D\$ \rightarrow 1\,K_C\$ \rightarrow 1\,K_B$ babysit)

Stellar transaction model



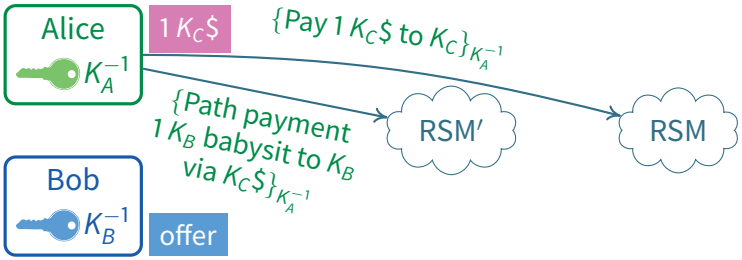
Global replicated state machine (RSM) executes transactions to keep ledger state

- Accounts named by public key authorizing operations on the account
- Accounts can issue assets; issuing account part of asset name

Transactions guarantee atomicity

- Multiple operations from multiple accounts with either all succeed or all fail
- *Path payments* atomically trade through multiple assets (e.g., $1\,K_D\$ \rightarrow 1\,K_C\$ \rightarrow 1\,K_B$ babysit)

How to guarantee ledger integrity?



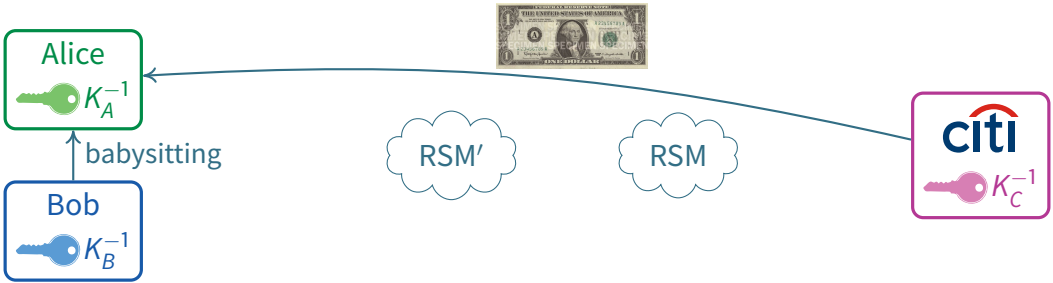
Model only works if everyone agrees on ledger state

- If ledger forks, system vulnerable to *double-spend attack*
- E.g., Alice gets both babysitting and $1, Bob can't redeem $K_C$$

Solution: Bob had better *follow* the server Citi uses to redeem $K_C$$

- Unless/until that server agrees, Bob shouldn't recognize Alice's babysitting credit

How to guarantee ledger integrity?



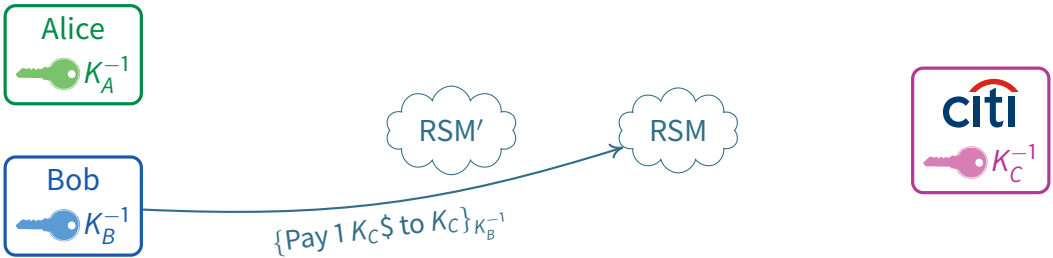
Model only works if everyone agrees on ledger state

- If ledger forks, system vulnerable to *double-spend attack*
- E.g., Alice gets both babysitting and $1, Bob can't redeem K_C\$

Solution: Bob had better *follow* the server Citi uses to redeem K_C\$

- Unless/until that server agrees, Bob shouldn't recognize Alice's babysitting credit

How to guarantee ledger integrity?



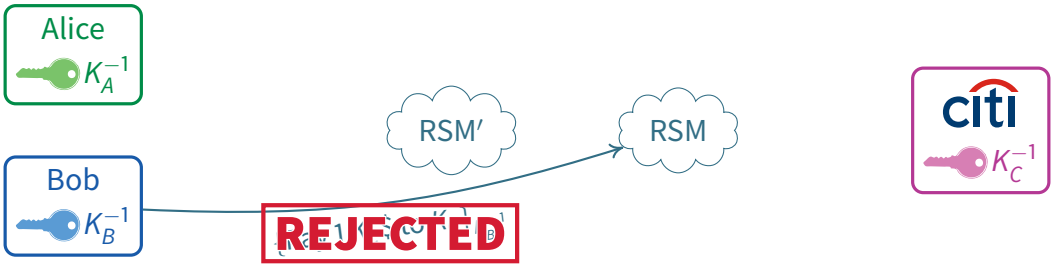
Model only works if everyone agrees on ledger state

- If ledger forks, system vulnerable to *double-spend attack*
- E.g., Alice gets both babysitting and $1, Bob can't redeem $K_C$$

Solution: Bob had better *follow* the server Citi uses to redeem $K_C$$

- Unless/until that server agrees, Bob shouldn't recognize Alice's babysitting credit

How to guarantee ledger integrity?



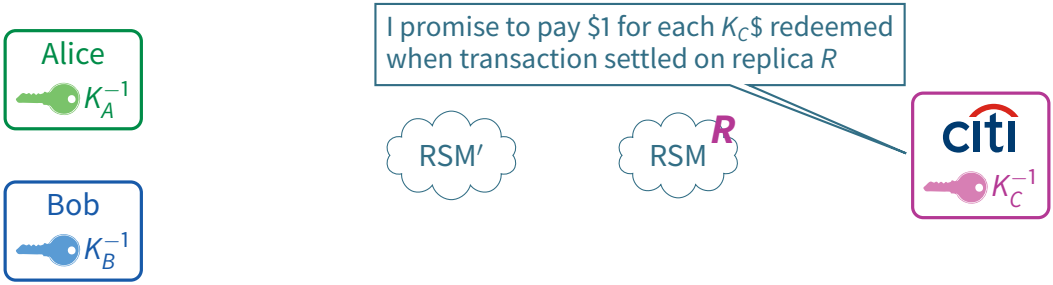
Model only works if everyone agrees on ledger state

- If ledger forks, system vulnerable to *double-spend attack*
- E.g., Alice gets both babysitting and $1, Bob can't redeem K_C\$

Solution: Bob had better *follow* the server Citi uses to redeem K_C\$

- Unless/until that server agrees, Bob shouldn't recognize Alice's babysitting credit

How to guarantee ledger integrity?



Model only works if everyone agrees on ledger state

- If ledger forks, system vulnerable to *double-spend attack*
- E.g., Alice gets both babysitting and $1, Bob can't redeem $K_C$$

Solution: Bob had better *follow* the server Citi uses to redeem $K_C$$

- Unless/until that server agrees, Bob shouldn't recognize Alice's babysitting credit

The Internet hypothesis

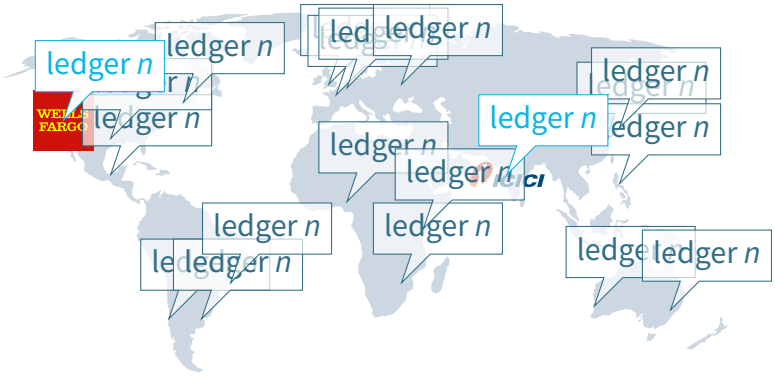
Will two organizations that don't follow each other agree on ledger state?

- Yes if the follow graph transitively converges

Hypothesis: any two nodes transitively follow a common node

- Empirically true of Internet (e.g., China⟷Stanford⟷Google) and legacy payments
- And if they don't, maybe a fork is okay (risk limited to in-flight transactions)

The Internet hypothesis



Will two organizations that don't follow each other agree on ledger state?

- Yes if the follow graph transitively converges

Hypothesis: any two nodes transitively follow a common node

- Empirically true of Internet (e.g., China⟷Stanford⟷Google) and legacy payments
- And if they don't, maybe a fork is okay (risk limited to in-flight transactions)

Byzantine agreement from the Internet hypothesis

Stellar consensus protocol (SCP) secures Stellar ledger

- Safety and liveness formally verified for arbitrary configurations

Key idea: broadcast protocol steps conditioned on other nodes' steps

- Take step if all nodes mutually satisfied

For availability, must generalize "follows" to sets of peers, called *quorum slices*

- Take step if any quorum slice unanimously willing
- E.g., slices(v_1) = alls set comprising a majority from each of 3 organizations

Definition (Quorum)

A *quorum* is a set of nodes containing at least one slice of each non-faulty member.

Byzantine agreement from the Internet hypothesis

Stellar consensus protocol (SCP) secures Stellar ledger

- Safety and liveness formally verified for arbitrary configurations

Key idea: broadcast protocol steps conditioned on other nodes' steps

- Take step if all nodes mutually satisfied

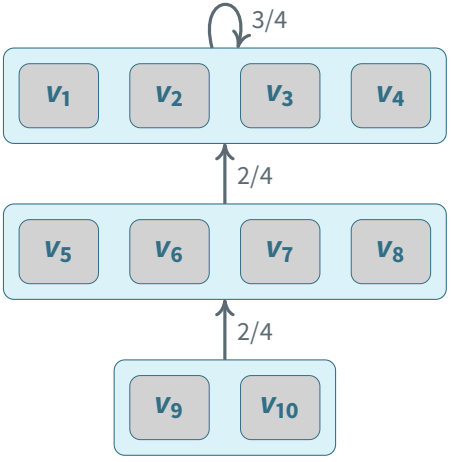
For availability, must generalize "follows" to sets of peers, called *quorum slices*

- Take step if any quorum slice unanimously willing
- E.g., slices(v_1) = alls set comprising a majority from each of 3 organizations

Definition (Quorum)

A *quorum* is a set of nodes containing at least one slice of each non-faulty member.

Quorum slices in action



Top tier: slice is three out of $\{v_1, v_2, v_3, v_4\}$ (including self)

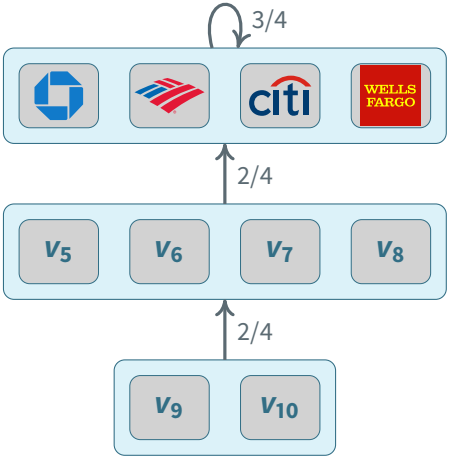
Middle tier: slice is self + any two top tier nodes

Leaf tier: slice is self + any two middle tier nodes

Like the Internet, no central authority appoints top tier

- But market can decide on *de facto* tier one organizations
- Don't even require exact agreement on who is a top tier node

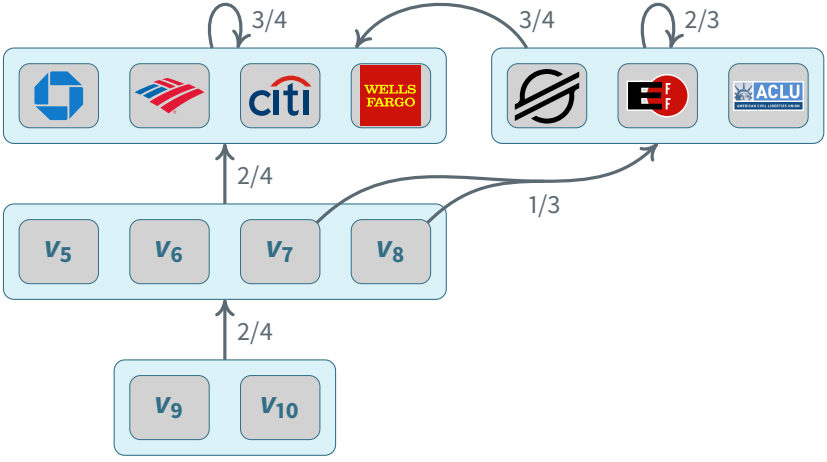
Quorum slices in action



Like the Internet, no central authority appoints top tier

- But market can decide on *de facto* tier one organizations
- Don’t even require exact agreement on who is a top tier node

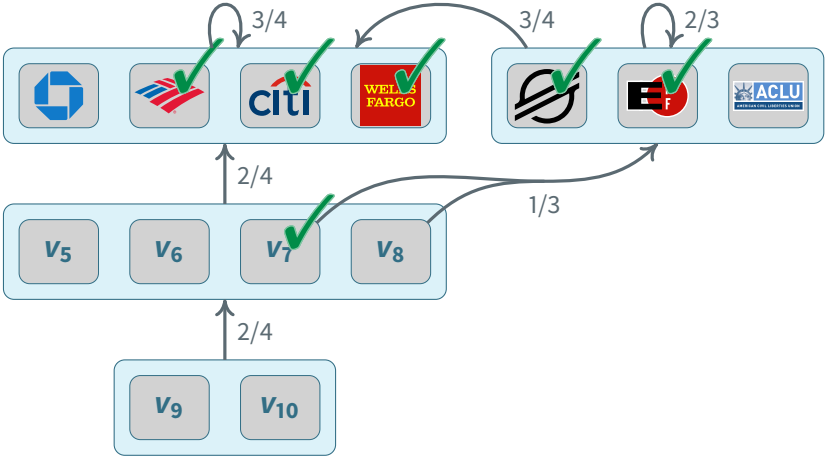
Quorum slices in action



Like the Internet, no central authority appoints top tier

- But market can decide on *de facto* tier one organizations
- Don't even require exact agreement on who is a top tier node

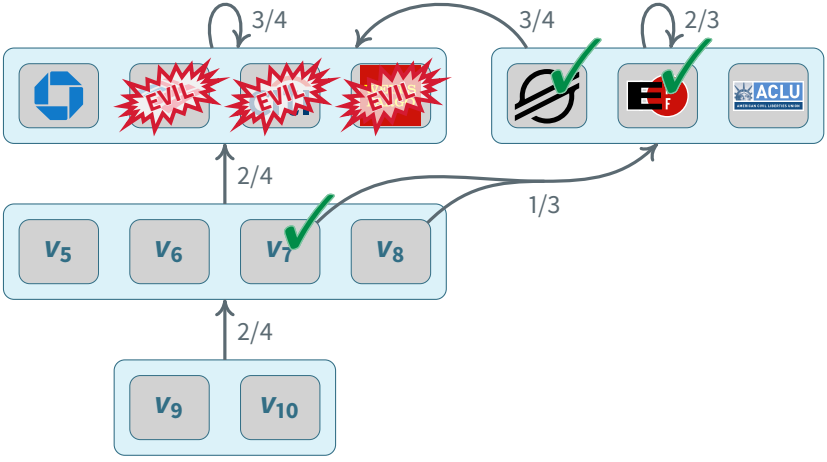
Quorum slices in action



Example: Citibank pays $1,000,000,000 to v_7

- Colludes to reverse transaction and double-spend same money to v_8
- Stellar & EFF won't revert, so ACLU cannot accept and v_8 won't either

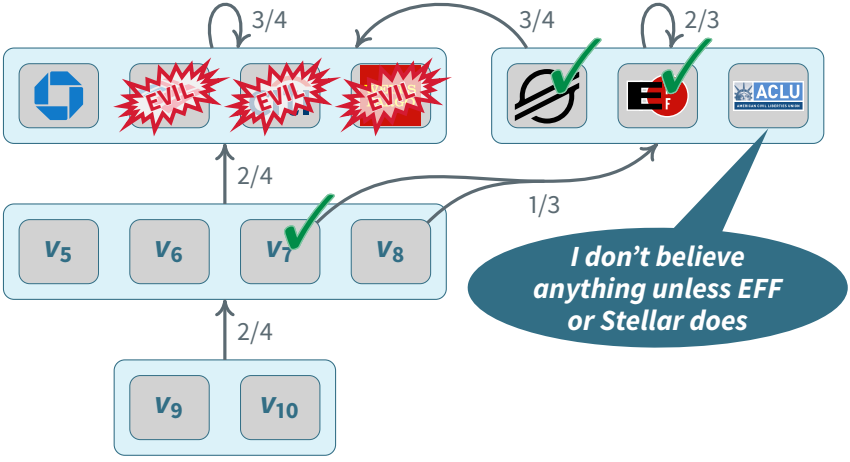
Quorum slices in action



Example: Citibank pays $1,000,000,000 to v_7

- Colludes to reverse transaction and double-spend same money to v_8
- Stellar & EFF won't revert, so ACLU cannot accept and v_8 won't either

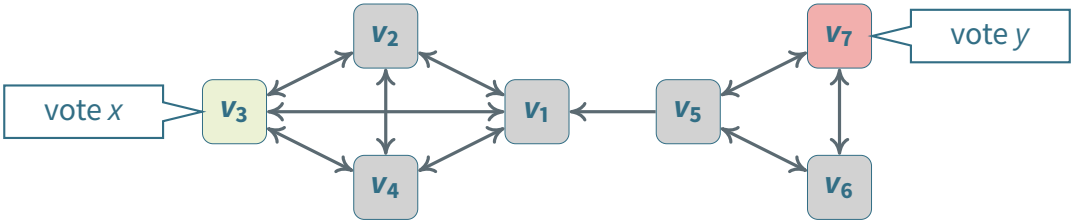
Quorum slices in action



Example: Citibank pays $1,000,000,000 to v_7

- Colludes to reverse transaction and double-spend same money to v_8
- Stellar & EFF won't revert, so ACLU cannot accept and v_8 won't either

Fundamental building block: federated voting



Vote for a statement if you believe it has a chance of prevailing

- E.g., $x =$ "Choose transaction set T for ledger n in ballot b"

Accept if you are in a quorum that unanimously votes for or accepts x

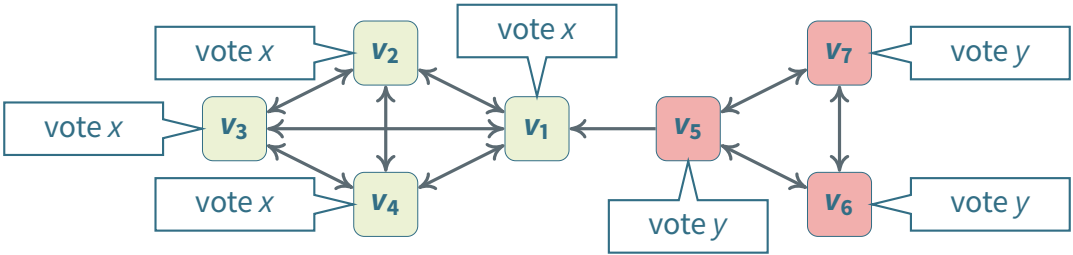
Also accept if each of your slices has accepting member

- Either it's true or you have lost liveness

Confirm (externalize) statement if you are in a quorum that unanimously accepts

Key property: if one "intact" node confirms a statement, all eventually will

Fundamental building block: federated voting



Vote for a statement if you believe it has a chance of prevailing

- E.g., $x =$ "Choose transaction set T for ledger n in ballot b"

Accept if you are in a quorum that unanimously votes for or accepts x

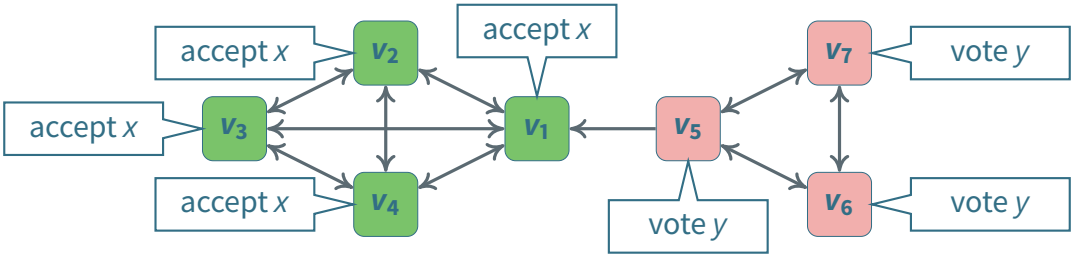
Also accept if each of your slices has accepting member

- Either it's true or you have lost liveness

Confirm (externalize) statement if you are in a quorum that unanimously accepts

Key property: if one "intact" node confirms a statement, all eventually will

Fundamental building block: federated voting



Vote for a statement if you believe it has a chance of prevailing

- E.g., $x =$ "Choose transaction set T for ledger n in ballot b"

Accept if you are in a quorum that unanimously votes for or accepts x

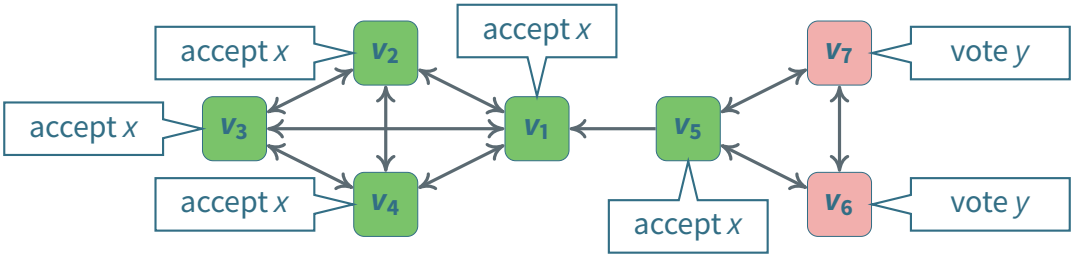
Also accept if each of your slices has accepting member

- Either it's true or you have lost liveness

Confirm (externalize) statement if you are in a quorum that unanimously accepts

Key property: if one "intact" node confirms a statement, all eventually will

Fundamental building block: federated voting



Vote for a statement if you believe it has a chance of prevailing

- E.g., $x =$ "Choose transaction set T for ledger n in ballot b"

Accept if you are in a quorum that unanimously votes for or accepts x

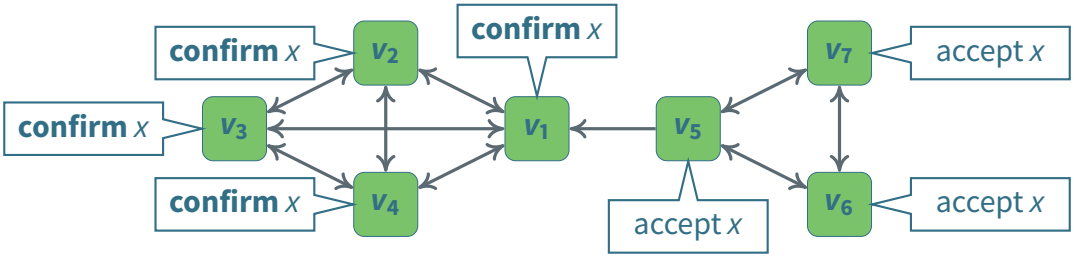
Also accept if each of your slices has accepting member

- Either it's true or you have lost liveness

Confirm (externalize) statement if you are in a quorum that unanimously accepts

Key property: if one "intact" node confirms a statement, all eventually will

Fundamental building block: federated voting



Vote for a statement if you believe it has a chance of prevailing

- E.g., $x =$ "Choose transaction set T for ledger n in ballot b"

Accept if you are in a quorum that unanimously votes for or accepts x

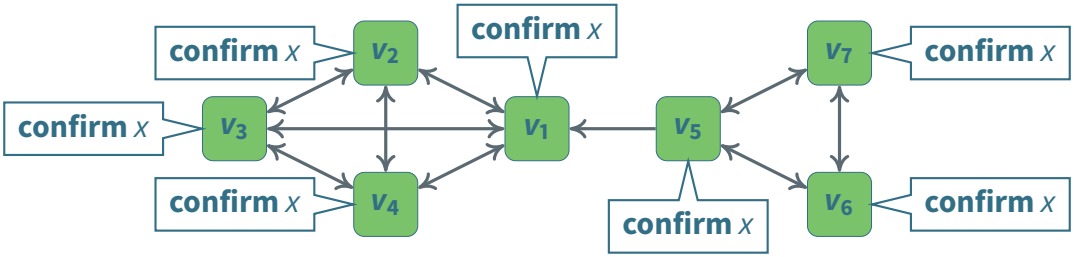
Also accept if each of your slices has accepting member

- Either it's true or you have lost liveness

Confirm (externalize) statement if you are in a quorum that unanimously accepts

Key property: if one "intact" node confirms a statement, all eventually will

Fundamental building block: federated voting



Vote for a statement if you believe it has a chance of prevailing

- E.g., $x =$ "Choose transaction set T for ledger n in ballot b"

Accept if you are in a quorum that unanimously votes for or accepts x

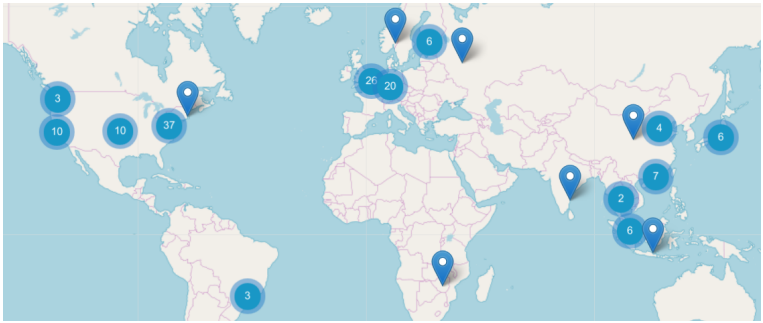
Also accept if each of your slices has accepting member

- Either it's true or you have lost liveness

Confirm (externalize) statement if you are in a quorum that unanimously accepts

Key property: if one "intact" node confirms a statement, all eventually will

Status



Production network has been running 4 years

- Ledger closes every 5 seconds, currently allows 1,000 operations/ledger
- Presently 133 nodes, 74 validators, 17 “tier-one” nodes run by 5 organizations

Shows open-membership Byzantine agreement is viable

30+ assets tracked on 3rd-party stellar.expert, about to be many more

First Stellar conference, Meridian, next week in Mexico city

Questions?

www.stellar.org