Blockchain tradeoffs and scaling

Digital Assets - Week 6 (Lecture)

Rhys Bidder
rhys.m.bidder@kcl.ac.uk

KBS, QCGBF

Autumn 2024

Disclaimer 1: Any opinions expressed, errors or omissions should be regarded as those of the author and not necessarily those of KCL, KBS, QCGBF, QCB, CBI, BoE, or Chainklink Labs.

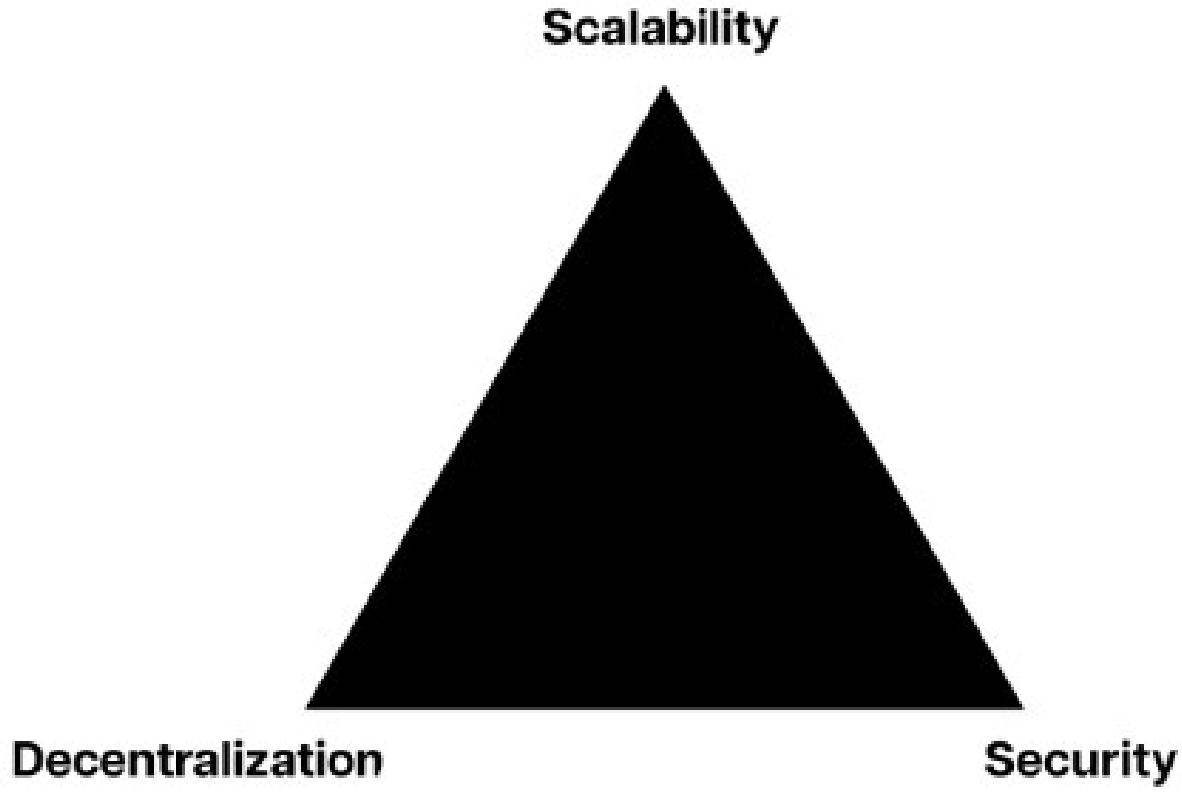
Disclaimer 2: Any references to cryptography are heavily simplified and leave out many important details. For any real-world security applications these notes should not be relied upon. Instead you should consult appropriate security resources and reliable security professionals.

Disclaimer 3: Cryptoasset transactions are illegal in some jurisdictions. Do not violate these or any other laws. I am not promoting the use of crypto in countries where it is illegal in any form and these slides are not a promotion of crypto or an invitation to participate in crypto-related activities in such countries. They are purely for educational purposes.

Outline

- The blockchain trilemma
- Node capability
- Validators in Ethereum
- Sharding
- State channels
- Sidechains
- Rollups

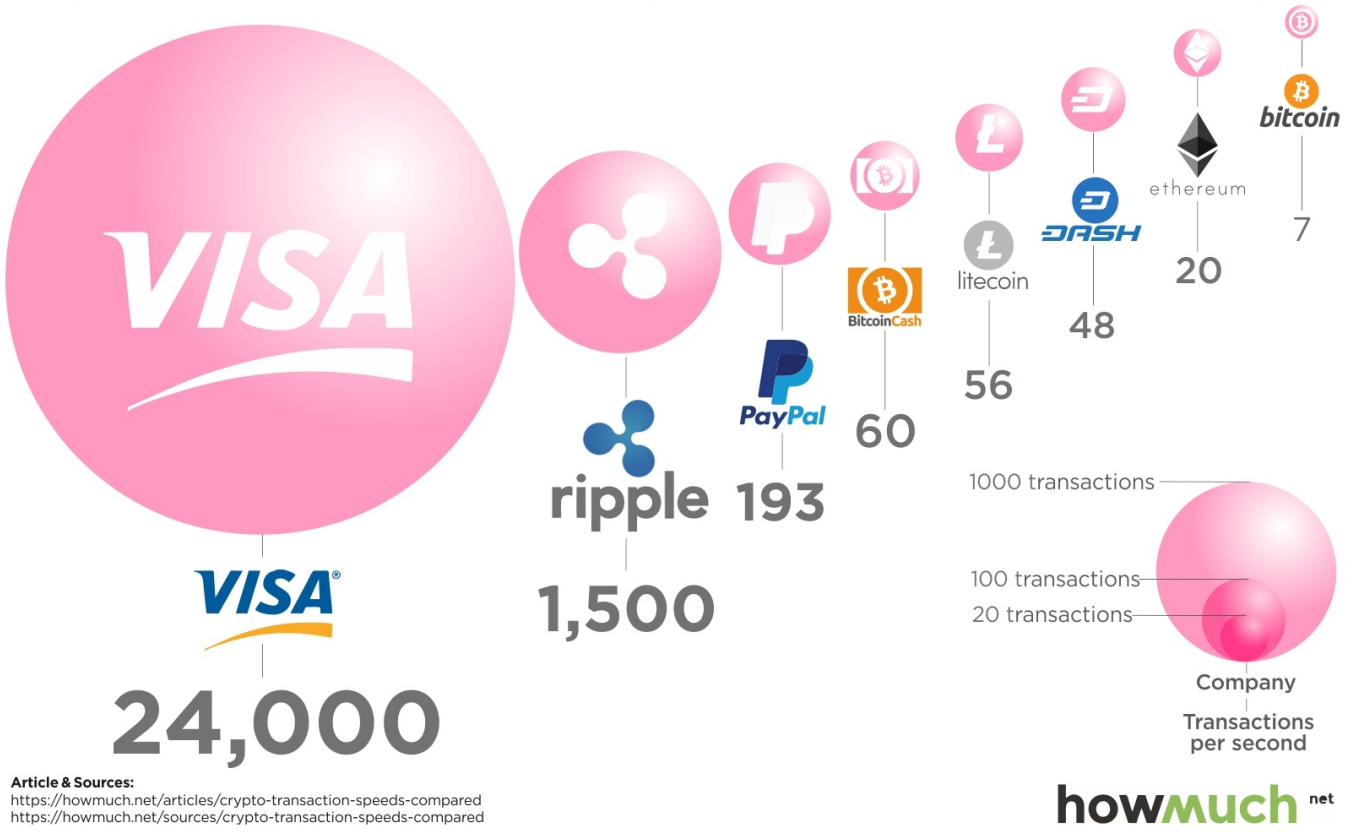
The blockchain trilemma



The blockchain ‘trilemma’ *Source:* *Nervos*

The blockchain trilemma

- Vitalik Buterin identified a particular set of tradeoffs for blockchain development in a blog post in 2017
- At the time it was 'impossible' to achieve all of the following three desiderata simultaneously:
  1. *Scalability:* Rapid and high volume throughput of transactions (frequently defined in terms of 'transactions per second' and compared to the high rates of CC networks - e.g. Visa offers 24k TPS)
  2. *Decentralization:* Dependence on a large network of multiple nodes, operated without a high degree of concentration, resistant to censorship, no single points of failure, and 'open to all'
  3. *Security:* The blockchain records are immutable, effectively impossible to manipulate, and only legitimate transactions are added to the chain



Transactions per second of various networks *Source: Web3 University, Sep 2022*

The blockchain trilemma

- Is this a genuine constraint?
  - No formal ‘proof’ that no blockchain design satisfies all three
  - Many people think that all three can be achieved
  - But so far has required difficult tradeoffs

Currently, in all blockchain protocols each node stores the entire state (account balances, contract code and storage, etc.) and processes all transactions. This provides a large amount of security, but greatly limits scalability: a blockchain cannot process more transactions than a single node can. In large part because of this, Bitcoin is limited to 3–7 transactions per second, Ethereum to 7–15, etc.

- Vitalik Buterin, Sharding FAQ, Dec 31, 2017

Some tradeoffs: Increase capability of nodes?

- Why not simply *increase block size* or *reduce block time* and demand that nodes have fast processors, massive memory and quick network connections?
  - Will make running a node very expensive
  - If it can't be run on a standard laptop, the network will shrink and become more concentrated
  - Decentralization and thus security deteriorates, even if TPS↑
  - Centralization makes it more likely that a node operator or a group of node operators can collude to manipulate the chain

In the quest to enhance decentralization and security, scalability often becomes a difficult hurdle. For instance, the Bitcoin network can only process around seven transactions per second, a number that pales in comparison to centralized payment systems like Visa, which can handle 24,000 transactions per second. This stark contrast is due to the inherent design of blockchain networks, where information needs to be processed by multiple participants, and consensus mechanisms like proof-of-work, which are secure but slow.

- What is the blockchain trilemma?, The Block, Sep 2023

Some tradeoffs: Increase capability of nodes?

- What does a node do?
  - Run specialized software (e.g. Geth as an execution client for Ethereum or Prysm as a consensus client) that validate transactions/build blocks, store/communicate/sync blockchain data with other nodes
  - All transactions are checked (the nodes check they satisfy basic correctness, and calculate how they would change the 'state')
  - Also checks versions of the blockchain that other nodes have sent around the network (recall, tampering with the database can quickly be revealed, thanks to hashes)
- Guides to Ethereum nodes/clients can be found here and here (and see this article contains info on Bitcoin nodes
  - Most blockchains (even Bitcoin and Ethereum) will have different types of nodes, executing somewhat different roles
  - For example, not all will validate, and not all will store all the transaction data of the blockchain

Some tradeoffs: Increase capability of nodes?

- Running a node at the moment requires moderate hardware
  - See here and here (8-16GB RAM, ok processor, 25MBit/s bandwidth, $\geq 1TB$ disk space)
  - But still can be difficult/risky
  - And blockchain is growing larger and larger, increasing computation demands and communication overhead
  - Network bandwidth, security/update, power demands are ongoing - in addition to initial setup and maintenance costs
  - Possible there may be (or soon be) regulatory requirements to monitor and satisfy
- Because of the demands on node operators there is a growing number of companies offering node hosting
  - Examples are Quicknode, NOWNodes, Alchemy, Infura
  - Usage perhaps implies a subtle form of centralization (as does dominance of particular clients - esp. Geth)

Manifestations of scalability problems

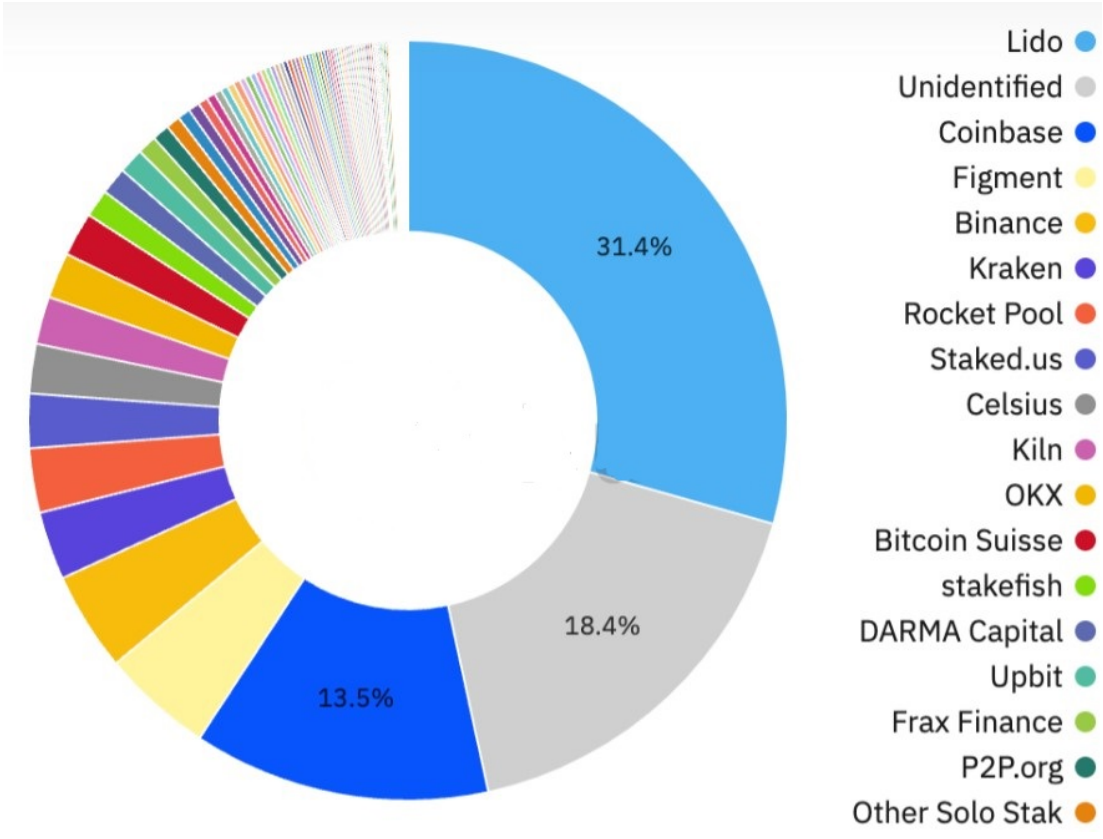
- What does it mean for a blockchain to have limited scalability?
  - Low throughput
  - Long delays for confirmation/finality
  - High gas fees

Validators in Ethereum PoS

- In the Ethereum protocol, validators are core to the Proof of Stake (PoS) consensus mechanism
  - Must stake (at least) 32 ETH to become a validator
- They play various roles (this is simplified):
  - Verify transactions and blocks for correctness (e.g. transactions in the block are executed to check proposed state changes are valid)
  - May be randomly allocated to a 'committee' that will determine a block
  - May be randomly selected to propose a block, otherwise (along with the other non-proposing committee members) will be called on to attest to the proposed block
  - Will (later) attest again to multiple blocks within the same 'epoch' (there are 32 slots in an epoch and a block is - or should be - added in each slot)

Debates around Ethereum validators

- Can speed up validation and finality by requiring smaller numbers of validators to agree?
  - At various points, majorities and super-majorities of validators are required to sign off on blocks, or the finalization of blocks
  - Reducing the number will affect security
- Some are concerned that 32 ETH requirement undermines decentralization
  - Possibly mitigated by the presence of staking pools
  - But staking services (e.g. Lido) charge a fee, and themselves may contribute to centralization
  - Buterin has recently written about how to reach a point where only 1 ETH is required of a validator



Distribution of staked ETH *Source: Hord*

Sharding

Are there ways to create a new mechanism, where only a small subset of nodes verifies each transaction? As long as there are sufficiently many nodes verifying each transaction that the system is still highly secure, but a sufficiently small percentage of the total validator set so that the system can process many transactions in parallel, could we not split up transaction processing between smaller groups of nodes to greatly increase a blockchain's total throughput?

- Vitalik Buterin, Sharding FAQ, Dec 31, 2017

Sharding

- Recall, in its original form, the Ethereum mainnet requires every node to process every transaction
  - A moderately specced computer will struggle to allow some on-chain methods of increasing throughput (larger blocks, faster block times)

  $\Rightarrow$ Decentralization/security will suffer
- Sharding divides network into interconnected subnets - referred to as 'shards'
- Each shard operates like a 'mini blockchain', processing a subset of transactions
- This reduces computation, storage, communication requirements (good for decentralization) and allows parallel processing across the network (good for scalability, gas fees)

Sharding

- Every shard is associated with a unique transaction group, which comprises
  - A header
  - A body of transactions
- The header comprises
  - A shard ID (the shard the group belongs to)
  - The pre-state of the shard (i.e. its part of the blockchain, before new transactions change the state)
  - The post-state of the shard (i.e. its part of the blockchain, after new transactions are processed)
  - Addresses of the (randomly chosen) set of validators who are validating the transactions *for this shard*

Sharding

- The overall blockchain possesses a ‘global state’, which is made up by the combination of the shards (and this by their transaction groups ‘mini’ states)
  - Only once a transaction group is recorded in the global state can a shard’s own state transition to ‘post-state’
  - Clearly, there needs to be coordination across shards
  - For example, users may wish to transact across different shards, shards need to ensure they don’t process the same transaction,
- This requires clever processes to retain the security of the network and to avoid communication slowing everything down
  - Sharding reduces the number of nodes dealing directly with a given set of transactions
  - Greater risk of security breaches
  - Offset by random allocation of validators and by overall reduction in computation load for a node
- More detailed explanations can be found here and here

State channels

- Rather than improving a given 'layer 1' blockchain, another option is to use 'off-chain' devices to process some of the tasks nodes would otherwise have to do
  - We know that blockchain computation can be expensive, and that large storage requirements and communication overhead undermine decentralization
  - What if some of these could be separated from the core blockchain, leaving it to focus on ensuring consensus and correctness?
- We already saw some reference to this in the CBDC lecture
  - Storing confidential data off chain helped with privacy
  - Here we are more focused on performance - don't need the full blockchain apparatus (which is 'slow') to process transactions or store all data that is used in transactions

State channels

- Users of a blockchain may open a separate 'state channel' of communication to engage in a high volume or rapid transactions
  - This is one example of a 'layer 2' solution
- State channel keeps track of the 'net position'
  - If we are talking about value transfers (as in the Bitcoin Lightning network) then the net position is in terms of value
  - If we are talking more generally, the net position is the overall evolution of the component of the blockchain state that is relevant to them the channel participants
- Once they have completed their business and agree to close the channel, the net position is conveyed back to the blockchain
  - Process only requires 'two' transactions on the layer-1 (opening and closing the channel)

State channels - Lightning network

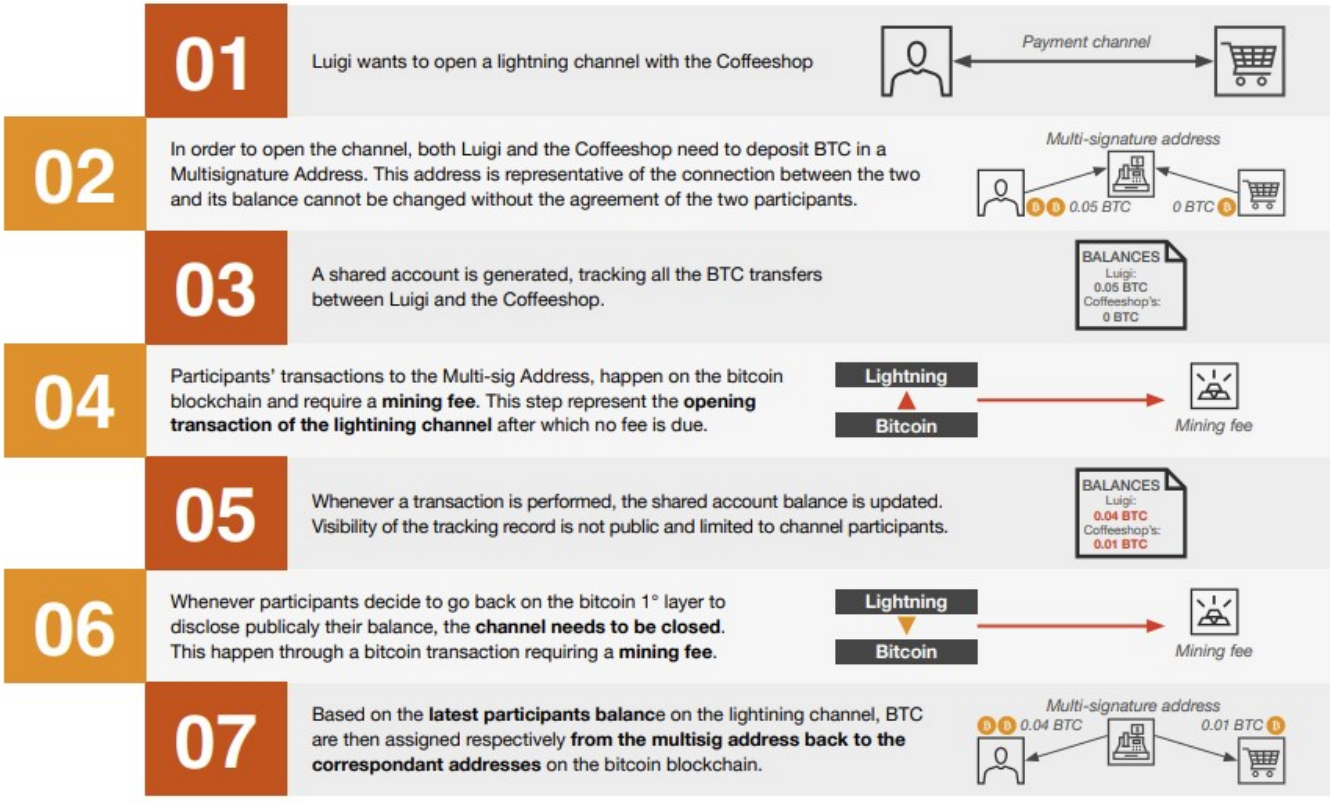
If Mike goes to a local coffee shop every day and wants to pay in Bitcoin, he could choose to make a small transaction for each coffee cup, but due to Bitcoin's scalability issues, the transaction can take over an hour to validate. Mike will also have to pay the Bitcoin network's high fees, even though he's making a tiny transaction.

With the Lightning Network, Mike can open up a payment channel with the coffee shop. Each coffee purchase is recorded within that channel, and the shop still gets paid. The transaction is cheap or possibly even free, as well as instant. When Mike's Bitcoin deposit in the channel runs out, he can choose to either close the channel or top it up. When a channel is closed, all of its transactions will then be recorded to the main Bitcoin blockchain.

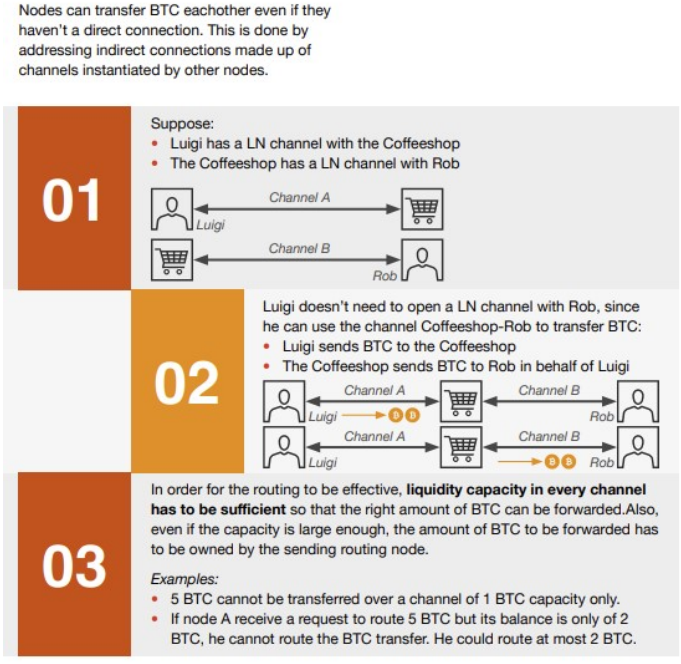
- What is the lightning network, CoinTelegraph, 2024

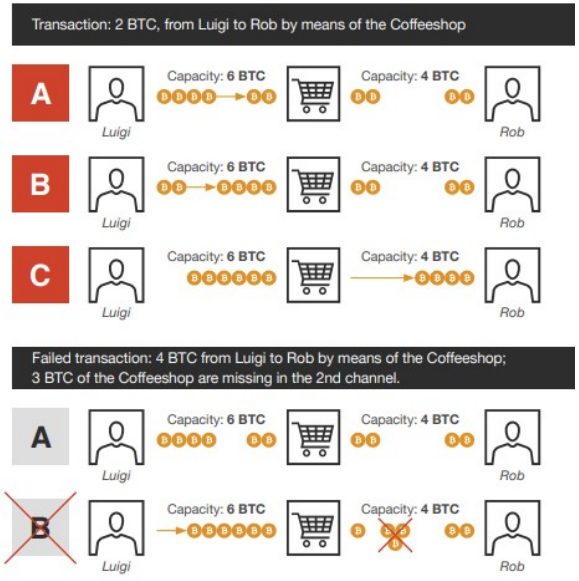
State channels - Lightning network

- A prominent example of state channels is the Lightning network that allows rapid transactions in Bitcoin
  - Upwards of 40k TPS (and theoretically 1m)
- The participants in the layer-2 first need to lock up Bitcoin on the mainnet
  - Send funds to an address that has multisig control
  - Multisig $\Rightarrow$ both parties will ultimately need to agree on the outcomes of their layer-2 transactions)
- It is not strictly necessary that counterparties need to open a direct channel with eachother - they can chain connections
  - This exploited by commercial solutions who offer 'intermediation' (e.g. Strike in El Savador - and see this map)
  - Note that this raises concerns of centralization/return to traditional payments models (which often come with competitive distortions)



Lightning network - basic example *Source: PWC*





Lightning network - indirect payments *Source: PWC*

State channels - Possible concerns

- Previously mentioned - possible centralization
  - And return to intermediated model
- Complexity
  - Different wallet types, APIs, communication systems required
  - Provides a larger attack surface (e.g. DDoS on the communication network) - and harder to run
- 'Closed channel fraud' (can be alleviated by 'watchtowers')
  - Various other attacks possible too

State channels - A possible attack

- Bob and Dave each fund a channel with 100k SAT each
- Bob pays Dave 50k SAT. Dave now has a claim to 150k SAT
- **Bob closes channel after the payment without consulting Dave, and submits the original balance of 100k SAT held by him and Dave as final to the Bitcoin blockchain.**
- Suppose Dave's node doesn't detect this (e.g. because he's gone offline) quickly enough (within 2,016 blocks on the Bitcoin blockchain - roughly two weeks), then Dave risks losing the 50k extra SAT Bob 'paid' him and both receive 100k SAT
- If within that 2,016 block period Dave's node notices that Bob tried to cheat, Dave will get his 150k SAT plus, as a penalty, Bob's 50k remaining SAT in the channel.

Sidechains

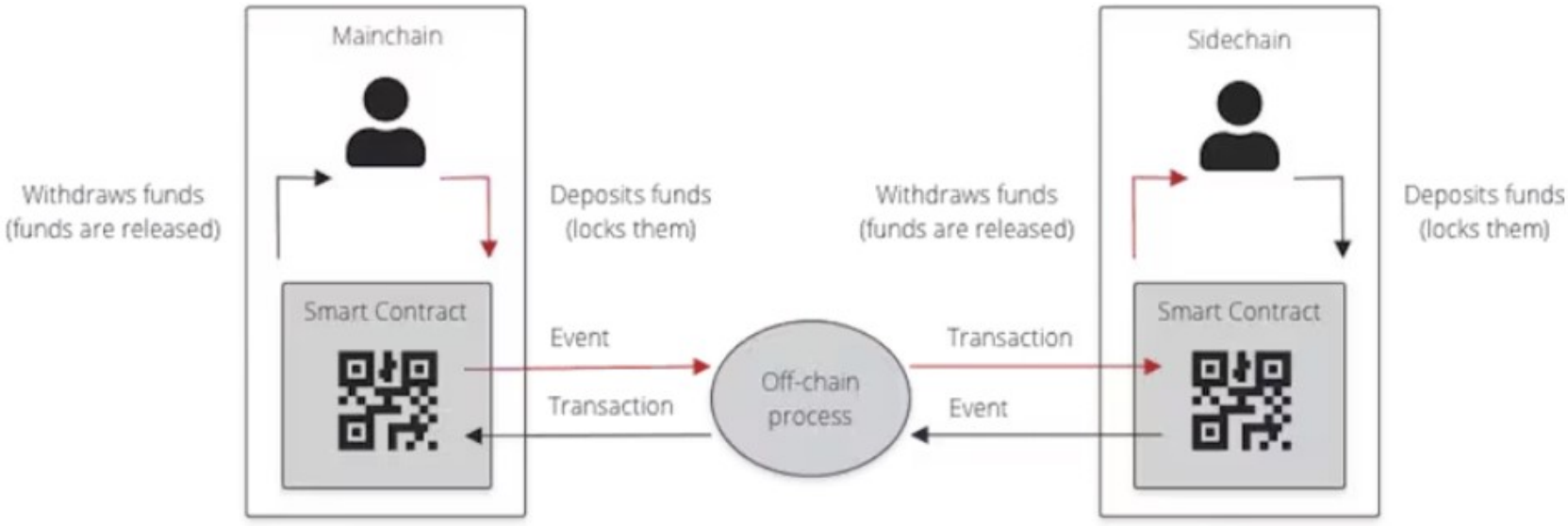
- Side chains are blockchains that connect to a 'parent' (typically layer-1) blockchain
  - Interact via a 'two-way peg' operating through smart contracts deployed on both chains
- Side chains have their own consensus mechanisms, which may be different from that used by the mainnet/layer-1
  - The side-chain may have its characteristics (e.g. consensus) tuned to emphasize particular use cases
  - Then applications that require, say, higher TPS, and (perhaps) less security can make use of the side chain
  - Note: may not have privacy and immediatie finality properties of state channels (but doesn't rely on participants always being on line and monitoring)
- Example: Polygon network (Ethereum sidechain)
  - It has its own validator network
  - Relies on PoS using POL (was MATIC)
  - See also Liquid network and Rootstock for Bitcoin

Sidechain - layer-1 interaction

- The ability to move value/tokens and information between layer-1 and the side chain is key
  - Allows app developers to leverage the large user-base of the layer-1
  - But also provides a smooth experience with rapid execution of the app on the sidechain
- A two-way peg allows the appearance of assets being transferred from one chain to the other
  - But in fact, assets are only locked on one chain, while equivalent amounts are then unlocked on the other chain
  - Note that this is an example of a broader pattern of chain interoperability
  - Smart contracts on each chain communicate (cryptographic proods) with eachother and lock/unlock assets

Sidechain - layer-1 interaction

- Process of 'pegging'
  - Transaction occurs on one blockchain (token sent to one of the peg smart contracts, say)
  - Smart contract locks the token and notifies the network
  - Communication channel used to inform the other blockchain
  - Information reaches the other smart contract, which unlocks the token on the other chain
- Relies on contracts being correct, consensus on the two chains functioning correctly, and the comms channel being secure
  - The off-chain comms channel (or 'bridge') may be implemented in various ways
  - Could use an oracle service (e.g. Chainlink)



An introduction to sidechains *Source:* *Coindesk/Ulam.io*

Rollups

- Rollups are another type of 'layer-2'
- They process transactions off chain, but then batch ('roll up') the settled transactions data to submit to the layer-1 for addition to the blockchain
- Do not operate their own validation/consensus (different from sidechains, which are themselves 'full' blockchains)
  - Sequencers build blocks of TXs to be processed on rollups
  - Once processed, the transaction results are pushed to L1
  - Piggy-backs on L1 consensus mechanism (and other mechanisms) to preserve security
  - Note: there is an open question about centralization of sequencers and their role in MEV!

Rollups

- Rollups help remove a huge computational load from the L1
  - Regarded as an important way to scale Ethereum - and recent protocol changes have promoted them
  - The intent is that they retain the L1's security while massively enhancing TPS
  - More than 50% of Ethereum transactions in June 2024 were processed on layer-2s (with rollups being dominant)

Types of rollups

There are two commonly used types of rollup:

1. *Optimistic rollups:* Quick processing but delayed finality
   - Arbitrum, Optimism, Base
2. *Zero-Knowledge (ZK) rollups:* More complicated/slower to execute, but more private/secure (and can reach finality quicker)
   - zkSync, Starkware

Optimistic rollup

- All transactions processed on the L2 are assumed to be valid
  - Saves time on checks
  - Helps TPS
- But after the results (implying evolution of the blockchain state) are pushed back to the L1, nodes can challenge any of the transactions
  - Finality is delayed for a certain period - to allow these challenges
  - The challenges are expressed using 'fraud proofs' that are sent to the L1
  - If the proof is correct (ie. that a transaction's implied result is incorrect/invalid) the transaction is reverted and the state returns to its original value
- Implicit that will only be applied to a small fraction of suspect transactions (otherwise lose the scaling benefit)

ZK rollup

- All transactions processed on the L2 are proved to be valid
  - Only the (zero knowledge) proofs are sent back to the L1
  - Proves validity of a transactions *without revealing any other details of the transaction*
- Immediate finality (no fraud proofs/dispute periods), less data transferred, inherently more 'private' but...
  - More difficult to implement and slower to run the transactions
  - zkEVM may be required to interact with them
  - Subtle issues around data availability arise (see also validiums)
- **Lots of work** being done to simplify and robustify these
  - Many people think ZK methods are the next big thing

Modular vs monolithic blockchains

- Beyond the scope of this course, but an interesting current debate is over monolithic vs modular blockchains
  - Natural continuation of shifting particular roles/activities off L1s
- Layer 2s can be thought of in terms of what roles they might take-on, and optimize for:
  - Execution: Process transactions
  - Settlement: Dispute resolution, bridge/pegging
  - Consensus: Order transactions, validate, build
  - Data availability: Where is transactinos data stored, how can it be retrieved or checked

Thanks for listening