


**TAL
TECH**

 blockchain.taltech.ee

SCALABILITY

08.07.2021

AGENDA

- Motivation
- Scalable consensus protocols
- Offchain protocols
- Payments channels
- State channels
- Virtual channels

BLOCKCHAIN SCALABILITY: A REAL PROBLEM

- This is a graph of the number of daily bitcoin transactions tracked over the years:

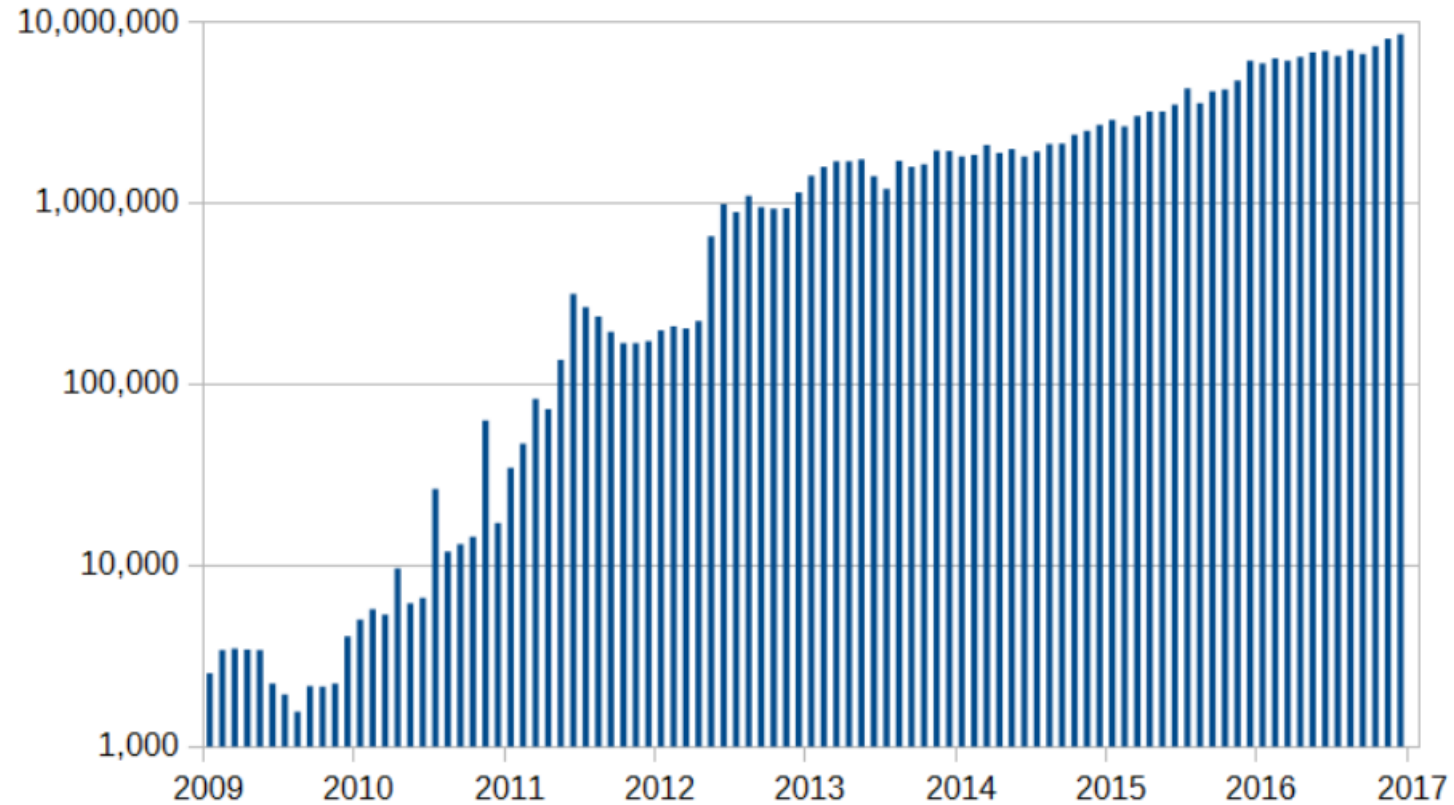


Image Courtesy: Wikipedia

BLOCKCHAIN SCALABILITY: A REAL PROBLEM

- Over time, the number of Ethereum transactions per month has been:

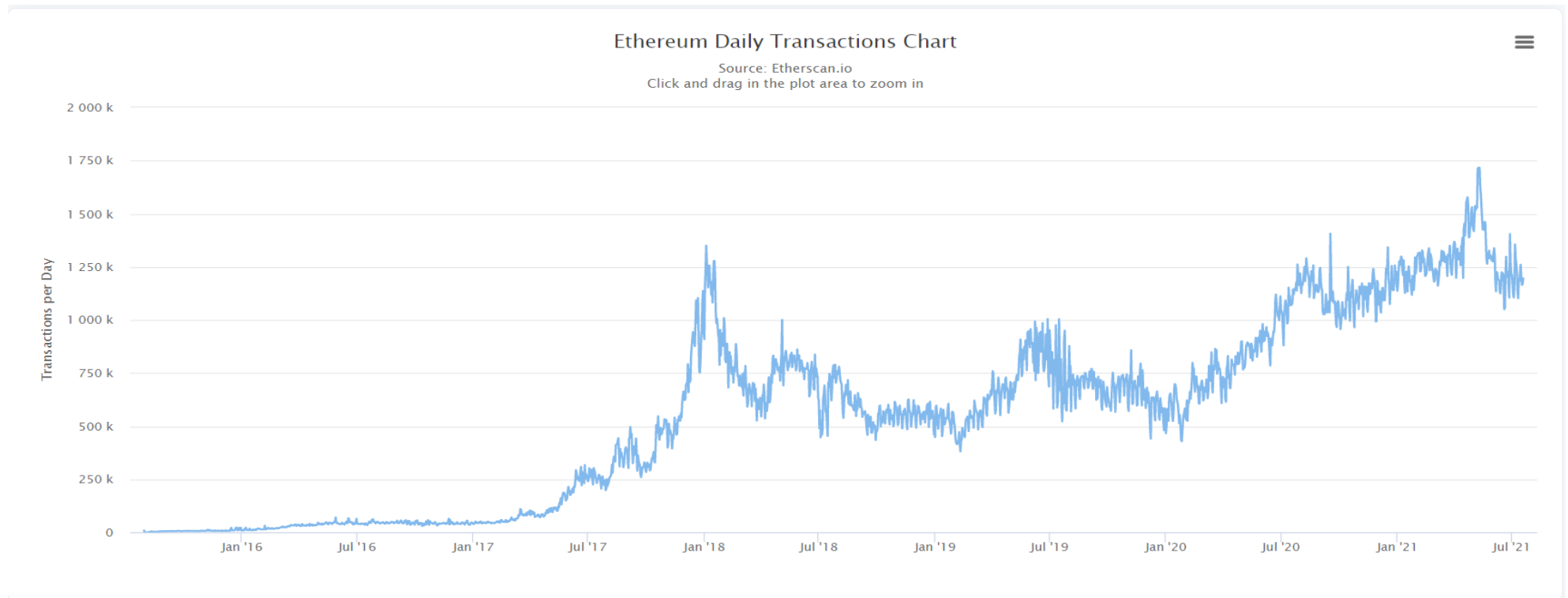


Image Courtesy: Etherscan

BLOCKCHAIN SCALABILITY: A REAL PROBLEM

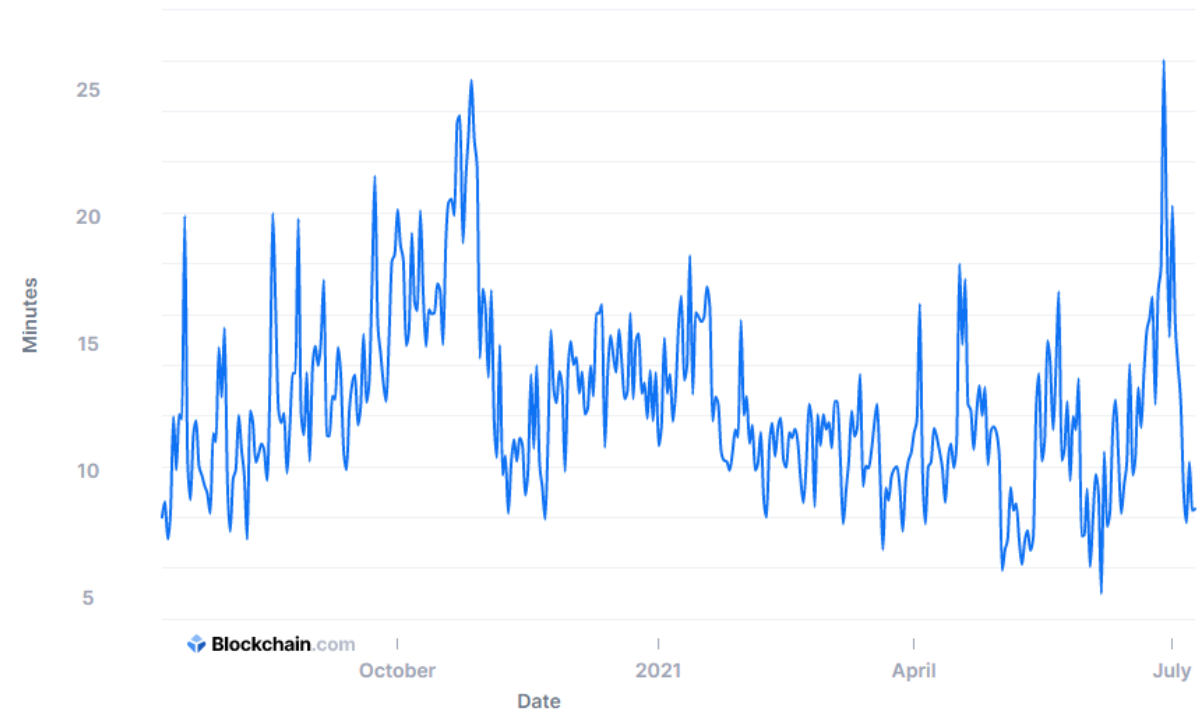
- Visa and PayPal perform transaction faster than bitcoin and ethereum.
- Ethereum processes 20 transactions per second, while PayPal processes 193 and Visa processes 1667 transactions per second.
- The following are the primary scalability issues in cryptocurrencies:
 - The time is taken to put a transaction in the block.
 - The time is taken to reach a consensus.

TIME SPENT ON BITCOIN TRANSACTION IN THE BLOCK

- In bitcoin and Ethereum, a transaction is validated when a miner validates the transaction data.
- Bitcoin requires more time and transaction fees.
- The faster miners will deposit a higher transaction fee into their block.
- Paying the lowest possible transaction fees will result in waiting for a median of 13 minutes for your transaction to process.

Median Confirmation Time

The median time for a transaction with miner fees to be included in a mined block and added to the public ledger.



TIME SPENT ON ETHEREUM TRANSACTION IN THE BLOCK

- Ethereum is designed to process 1000 transactions per second. however, It has a gas limit of 6. million.
- Miners cannot include transactic which add up to or less than the gas limit of the block.
- Once again, a number of transactions going through is limited.

Ethereum Average Gas Limit Chart

Source: Etherscan.io
Click and drag in the plot area to zoom in



Source: <https://etherscan.io/chart/gaslimit>

SCALABLE CONSENSUS PROTOCOL: BITCOIN-NG

- Bitcoin-NG blockchain protocol serializes transactions, much like Bitcoin, but allows for faster throughput and lower latency.
- A single leader handles serializing state machine transitions.
- This protocol introduced, **key blocks for leader election** and **microblocks** that contain ledger entries.

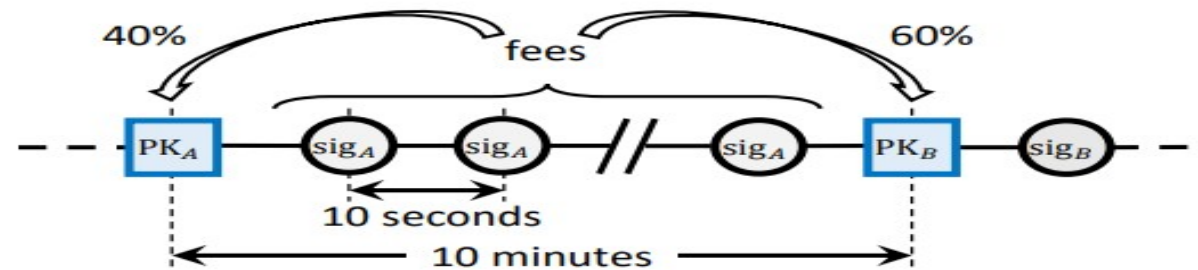


Figure 1: Structure of the Bitcoin-NG chain. Microblocks (circles) are signed with the private key matching the public key in the last key block (squares). Fee is distributed 40% to the leader and 60% to the next one.

KEY BLOCKS AND LEADER ELECTION

- Key blocks are used to choose a leader.
- As in Bitcoin, for a key block to be valid, the cryptographic hash of its header must be smaller than the target value.
- A key block contains a public key that will be used in subsequent microblocks.
- Nodes pick the branch with the most work, aggregated over all key blocks, with random tie breaking in case of fork.

MICROBLOCKS

- Once a node generates a key block it becomes the leader.
- The maximum size of microblocks is predefined.
- If the microblock's timestamp difference is smaller than the minimum, then the microblock is invalid.
- A microblock contains ledger entries and a header.

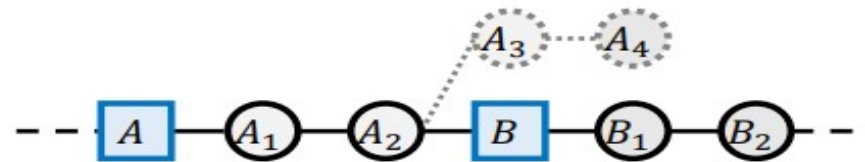


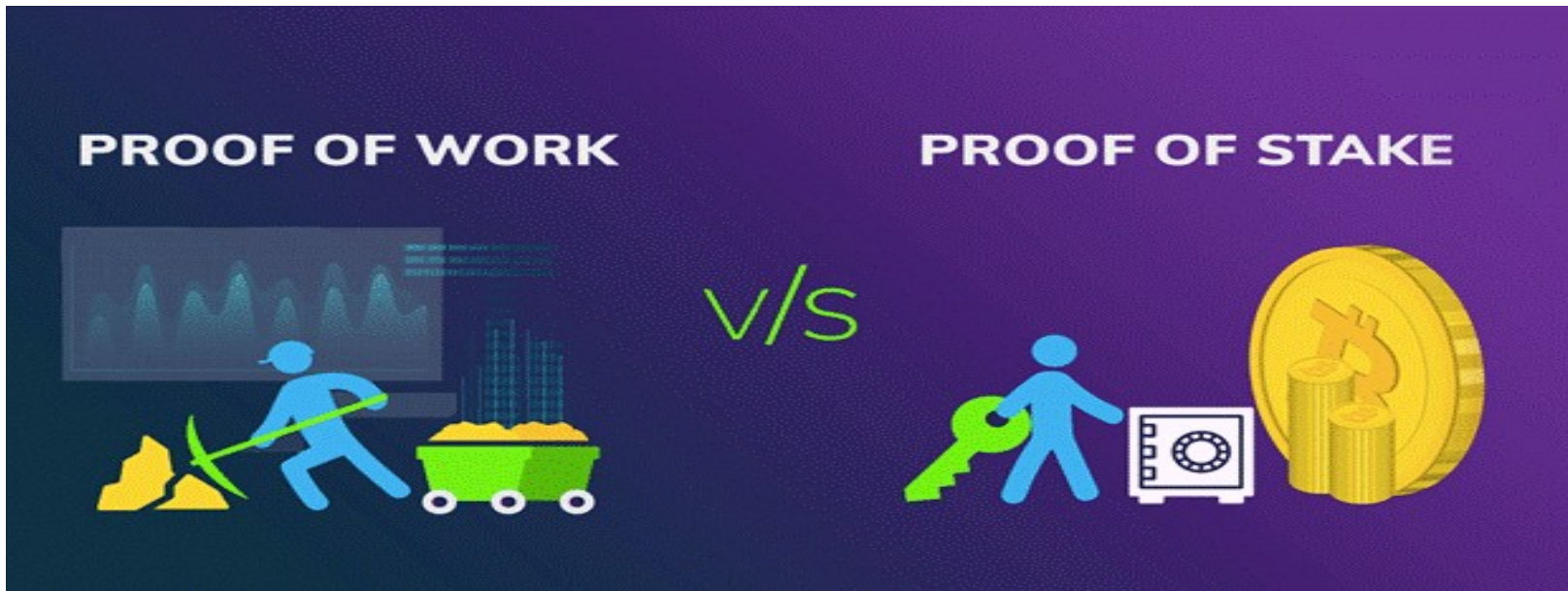
Figure 2: When microblocks are frequent, short forks occur on almost every leader switch.

Source: Bitcoin-NG: A Scalable Blockchain Protocol

ELIMINATING POW: PROOF OF STAKE

Justification: people who have the money are naturally interested in the stability of the currency.

Currencies: Steem(DPoS), NEO, Qtum, Ethereum 2.0 (planned)



Source: <https://www.c-sharpcorner.com/article/proof-of-work-vs-proof-of-stake/>

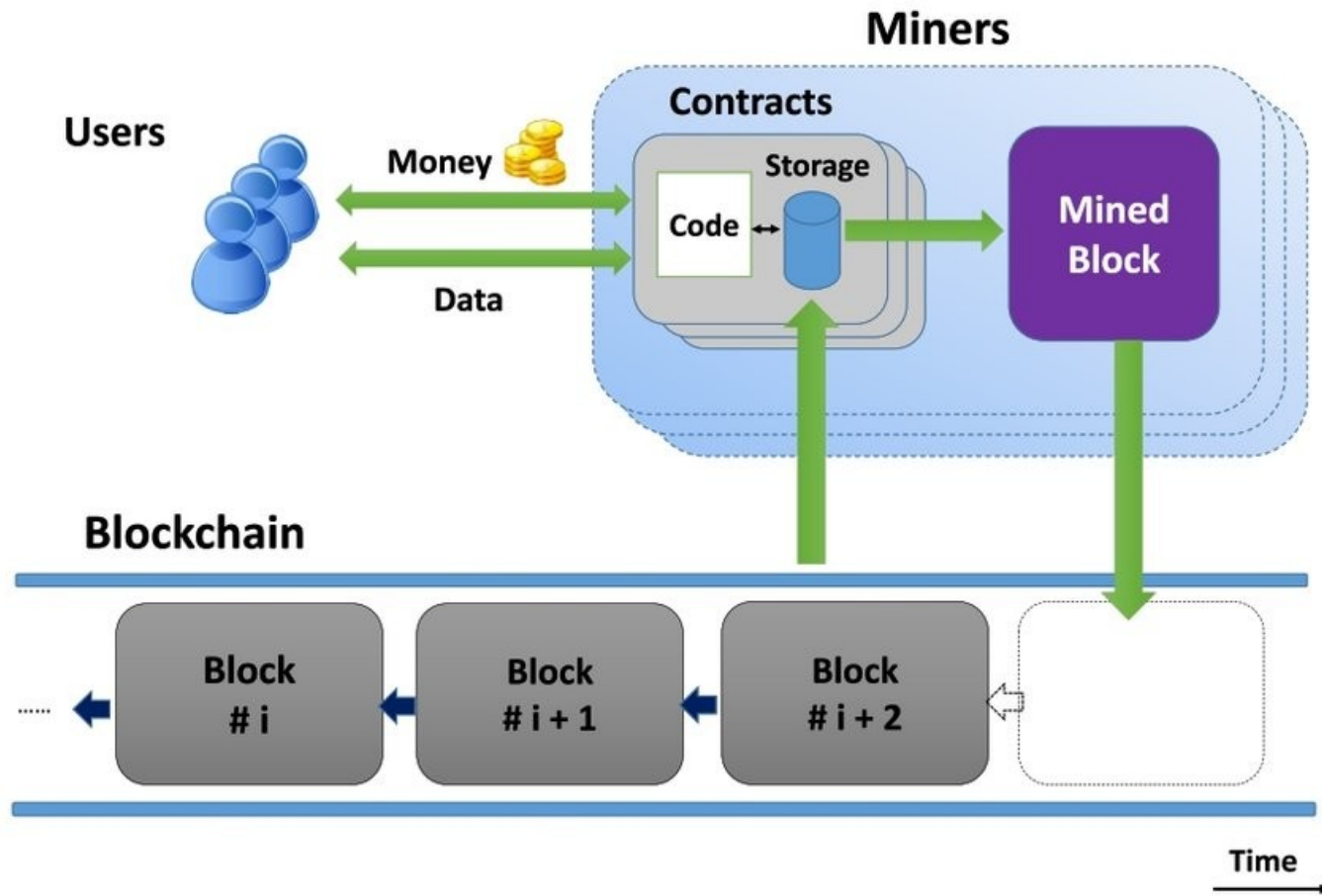
OFF-CHAIN PROTOCOLS

- It is difficult to make recurring micropayments on Blockchains.
- Transaction fees can be a significant cost for many users, particularly those transacting in small amounts.
- Getting your transaction confirmed can take from 10 minutes to several hours in Bitcoin transaction.
- Several protocols (Like lightning network) exist to enable off-chain transactions, which typically offer lower fees and faster settlement times.
- Off-chain: transactions are not recorded on the blockchain (only in case of dispute).

PAYMENT CHANNEL WITH SMART CONTRACTS

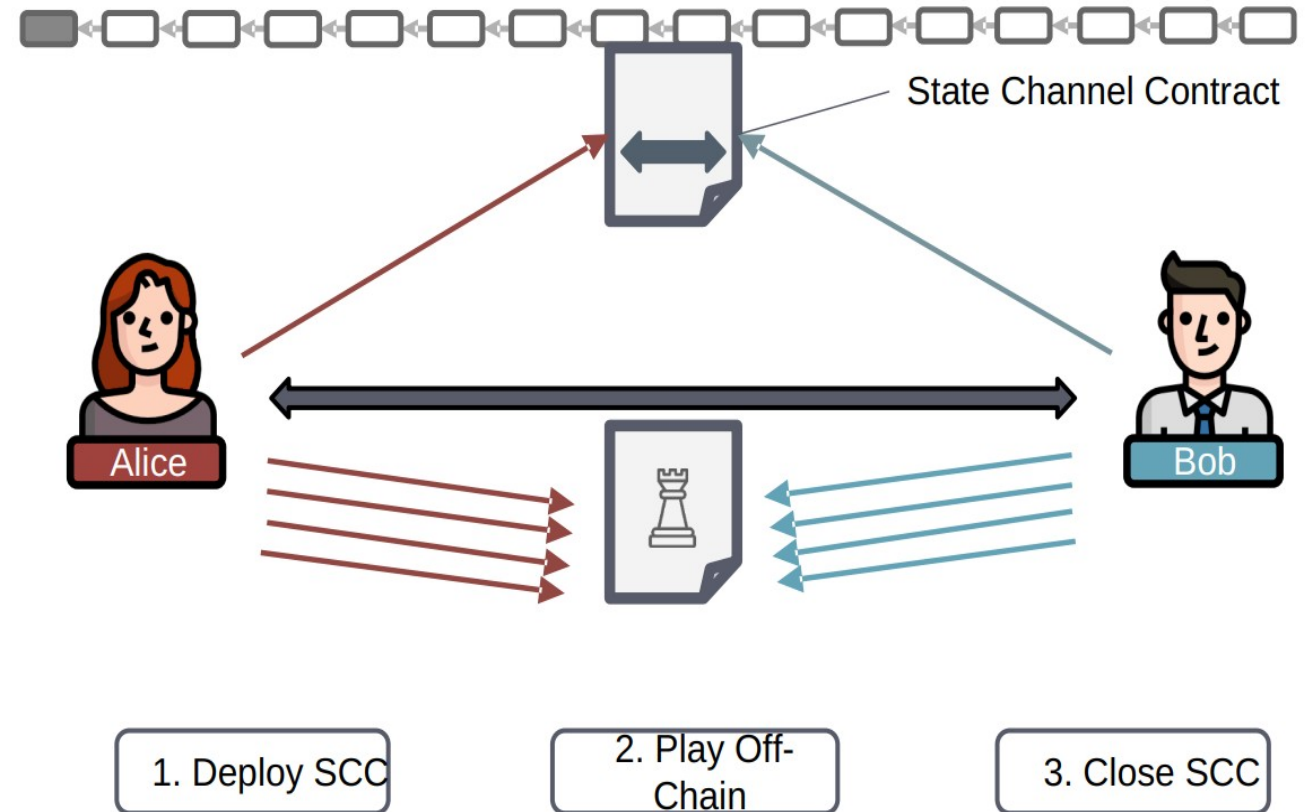
- Idea: use smart contract as judge in case of dispute.
- We call this contract the Adjudicator.
- Invalidation of old states with version counter
- Every on-chain action (dispute, close) has timeout
- Collaborative instant close is possible

GENERAL SMART CONTRACTS



LEDGER STATE CHANNEL

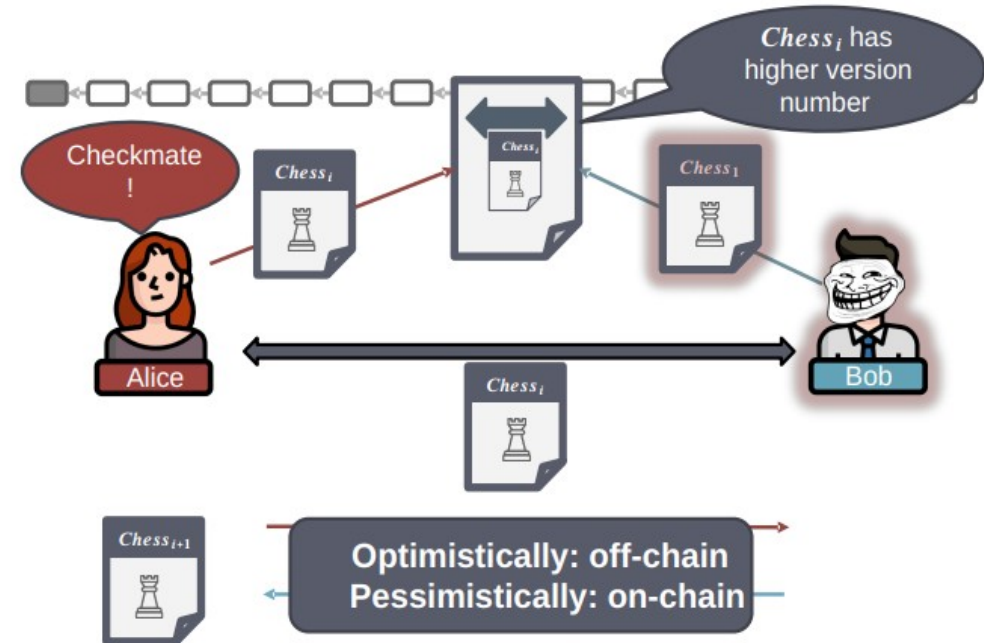
- Two parties deploy a State Channel Contract (SCC), denoted SCC, on the blockchain, in which each party locks some amount of coins.
- Once SCC is deployed and funded, the parties have the state channel γ .
- Upon completion of off-chain trading, the parties will inform the state channel contract SCC about the final coin distribution in γ .



LEDGER STATE CHANNEL: DISPUTE RESOLUTION

- If, one party cheats by, e.g., refusing to communicate, the other party can always ask the SCC smart contract to finish the contract.
- SCC must learn about the latest agreements parties reached about the game state.
- Thereafter, any of the two parties can ask SCC to execute the contract instance on any function f and any input z .

Dispute in a Ledger Channel (Contract Registration)



VIRTUAL STATE CHANNEL

- Alice and Bob already have a ledger state channel α with a third party Ingrid.
- Alice and Bob want to perform some contract code C (e.g., lottery game) off-chain.
- Alice and Bob can open a virtual state channel which functions the same as a ledger state channel between them.
- For the virtual state channel γ the role of such a judge is played by Ingrid.

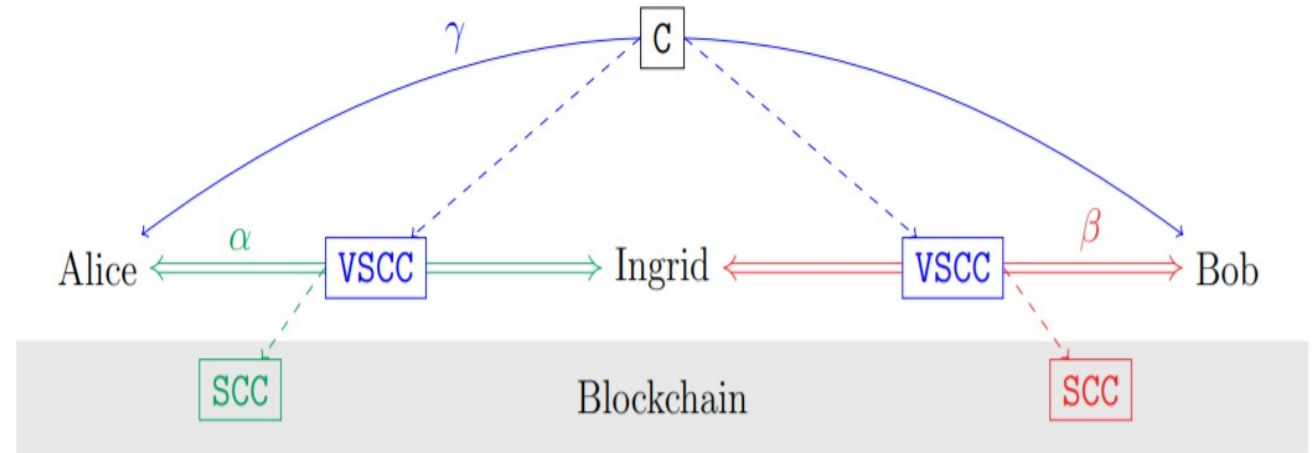


Figure 4.2.: Construction of a virtual state channel γ over two ledger channels α, β . The ledger state channels are supported by the SCC smart contracts on the blockchain. The virtual state channel γ is supported by two VSCC contract instances, one in each subchannel. A contract instance of C is created in γ .

Source:

https://tuprints.ulb.tu-darmstadt.de/17476/1/thesis_Final.pdf

VIRTUAL STATE CHANNEL CREATION

- Alice and Bob have decided to use Ingrid as a mediator for their virtual state channel.
- The proposed instance ν_β of VSCC is a “copy” of the virtual state channel γ where Ingrid plays the role of Alice.
- Once both subchannels, α and β , contain a VSCC contract instance, the virtual state channel γ is created.
- The off-chain contract execution occurs exactly the same in the virtual state channel as in the ledger state channels.

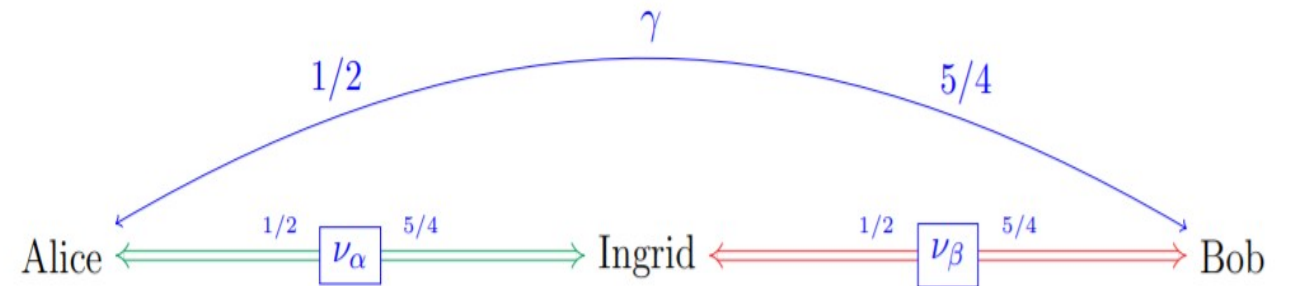


Figure 4.3.: The figure depicts the initial/final balances of parties in a virtual state channel γ and how they reflect the VSCC contract instance opened in the subchannels of γ .

Source:

https://tuprints.ulb.tu-darmstadt.de/17476/1/thesis_Final.pdf

VIRTUAL STATE CHANNEL: DISPUTE RESOLUTION

- Off-chain contract execution is conducted in exactly the same manner as it is in case of ledger state channels.
- For example, consider that Bob is malicious and fails to sign-off on the new state proposed by Alice.
- Alice's goal is to tell ν_α and encourage ν_β to believe that G_w is the current state of the contract instance G that they agreed on.
- she sends the state (G_w, w, s_B) to ν_α , where s_B is Bob's signature on (G_w, w) . She does it by calling a function "Reg(G_w, w, s_B)" (see Step 1).

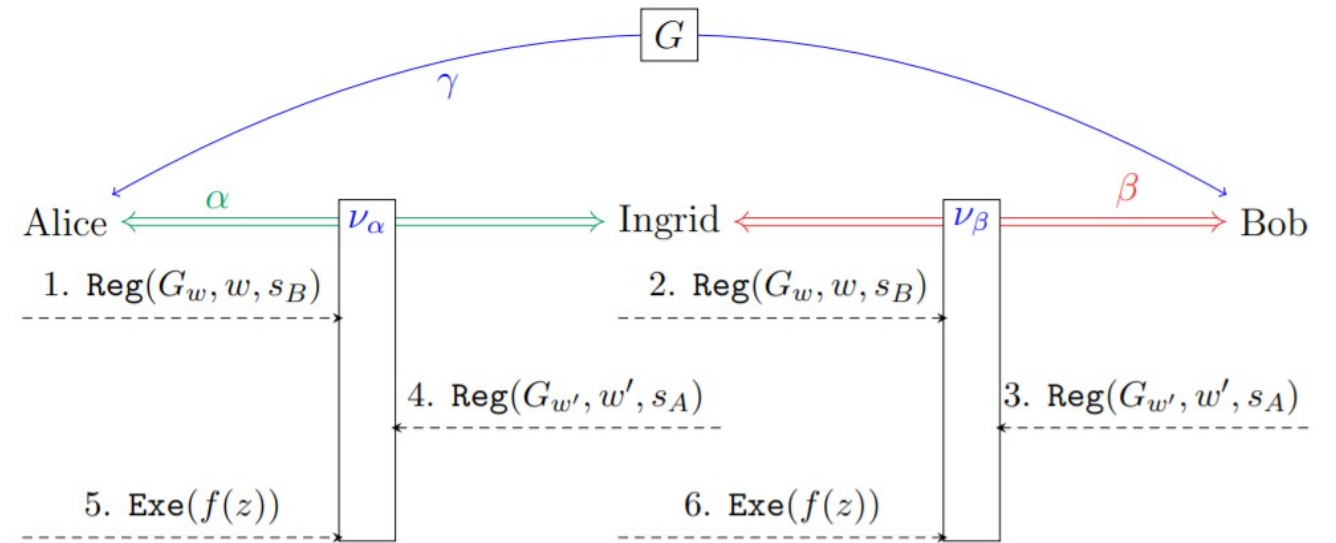


Figure 4.4.: Illustration of the forced execution process. Only the function calls are shown (the messages sent by the contracts are omitted).

Source:

https://tuprints.ulb.tu-darmstadt.de/17476/1/thesis_Final.pdf

VIRTUAL STATE CHANNEL: DISPUTE RESOLUTION

- Message is forwarded to Ingrid who calls a function "Reg(G_w, w, s_B)" in channel β (see Step 2).
- In Step 3, Bob answers back to v_β with his latest contract instance (i.e., he says "Reg($G_{w'}, w', s_A$)").
- When Ingrid learns about Bob's version from v_β , she forwards this information to v_α (see Step 4).
- After state registration is over, Alice calls (in Step 5) a function "Exe($f(z)$)" of v_α , thereby asking v_α to execute $f(z)$ on the registered state (G_w, w).

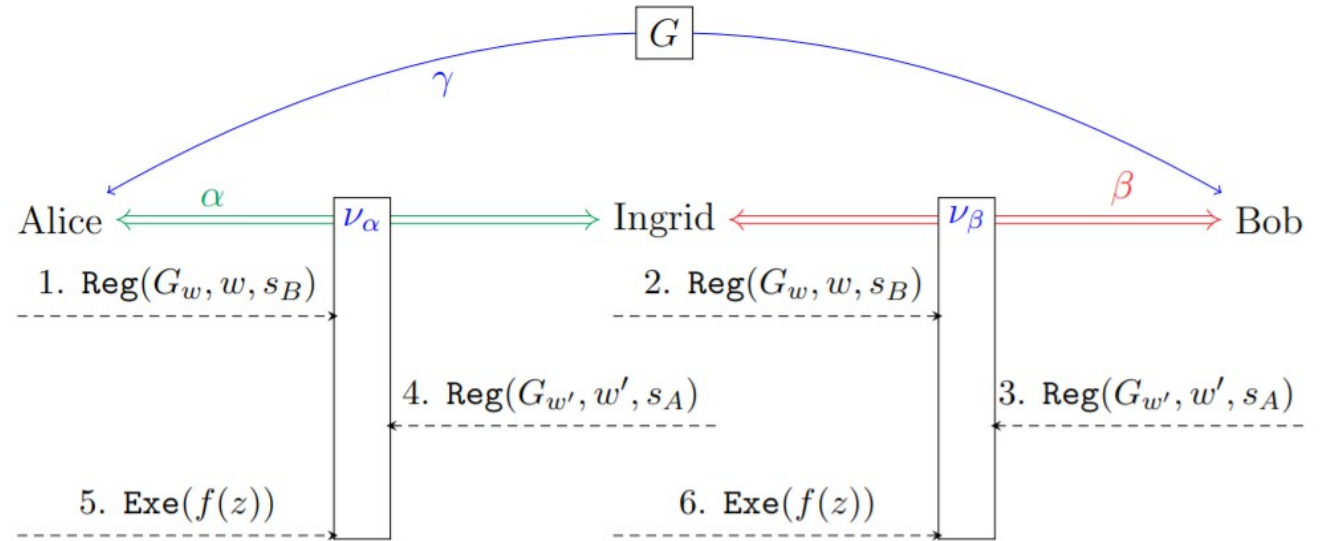


Figure 4.4.: Illustration of the forced execution process. Only the function calls are shown (the messages sent by the contracts are omitted).

Source:

https://tuprints.ulb.tu-darmstadt.de/17476/1/thesis_Final.pdf

**TAL
TECH**

Thank you very much for your attention!

Q & A?

Reference: Arumaithurai M., Introduction to Blockchains, Tallinn, Estonia 2019, <https://tinyurl.com/n2y3k5pu>