SWSY_PJ1

SCALABILITY FOR THE BAZO BLOCKCHAIN WITH SHARDING

Term Project 1

Roman Blum

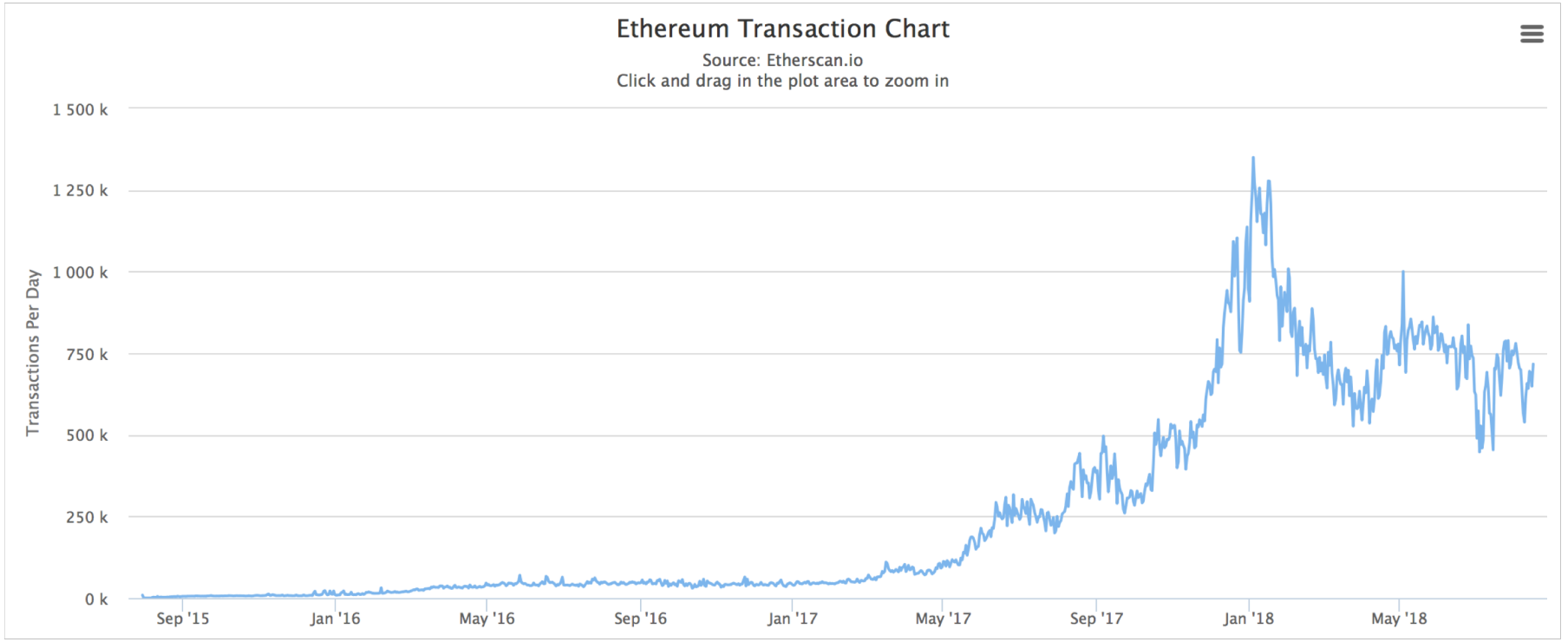
roman.blum@hsr.ch

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Scalability of Blockchain Consensus Protocols

- **Situation: Bitcoin can process roughly 8 transactions per second (tps), Ethereum around 15 tps**
  In comparison, Visa could process up to 65k tps

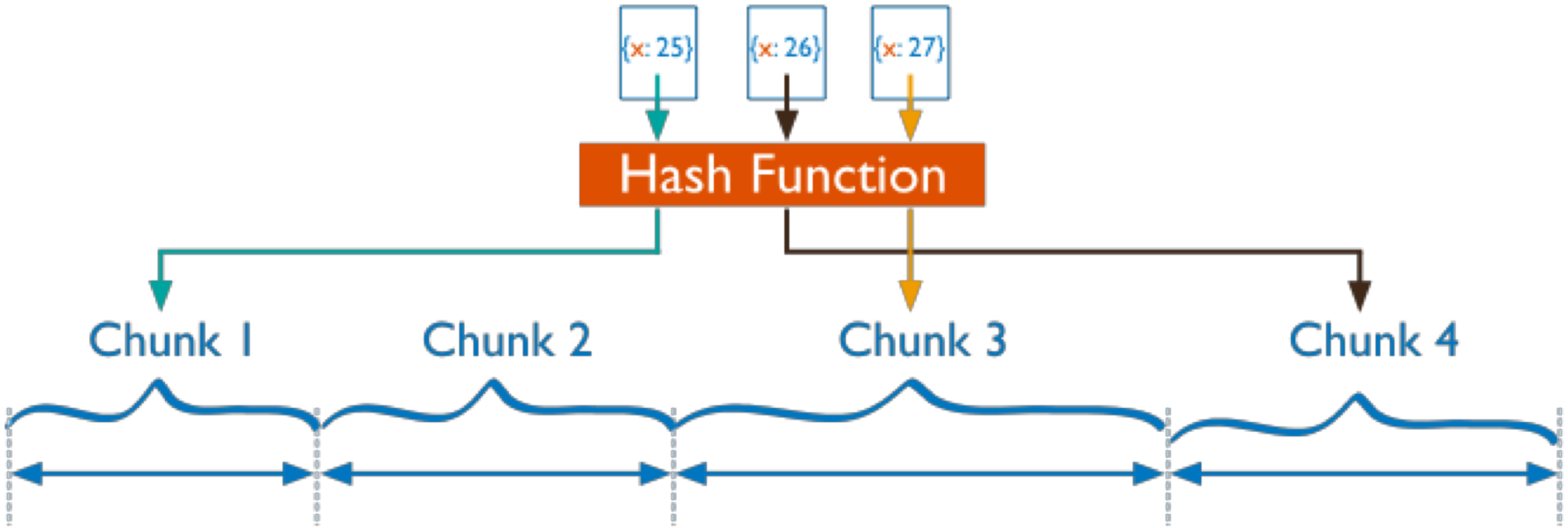


Blockchain Scalability Solutions

- **Plasma**
  Connects parent and child chains to the main blockchain and transforms it into a tree-structure
- **State Channels**
  A two-way communication channel to send transactions off-chain
- **Sharding**
  Divides the network state into partitions, where each node is responsible for a single partition

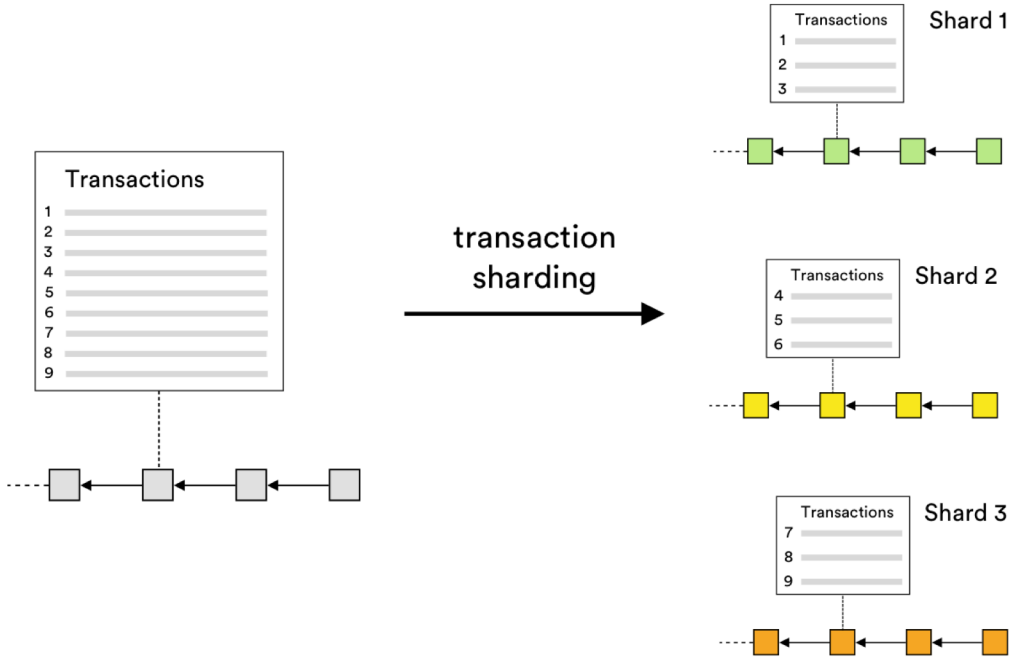
Database Sharding

- **Compute a hash of the shard key field's value**
- **Each chunk is assigned a range based on the hashed shard key values**
- **Each chunk is held on a separate database server instance**



Blockchain Sharding

- **The transaction load on the network is divided into different shards**
- **Only designated nodes validate the transaction, and not the entire network**
- **The blockchain should scale linearly with every new node added to the network.**
- **Higher scalability without sacrificing security or decentralization**



Sharding in Bazo

What is Bazo?

- **Blockchain developed by the University of Zurich and University of Applied Sciences Rapperswil**
- **Created for an external financial entity for the use of a customer bonus points program**
- **Developed as PoW-based blockchain in 2017, moved to chain-based PoS in 2018**
- **Seed exposure security vulnerability solved after the introduction of PoS**
- **Current and past feature developments**
  - Ledger Pruning (done)
  - Bazo Virtual Machine (done)
  - Scalability with Sharding (wip)
  - Smart Contract Language (next semester)

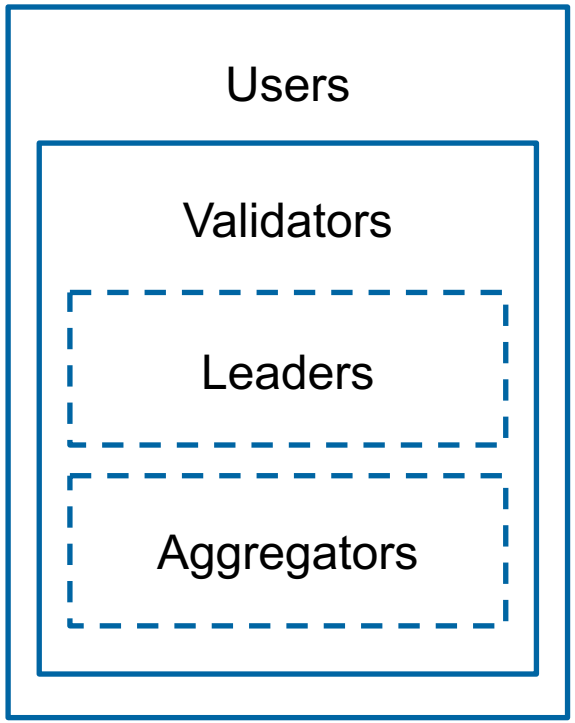
Sharding in Bazo

A conceptual overview on how to achieve scalability for the Bazo Blockchain with Sharding:

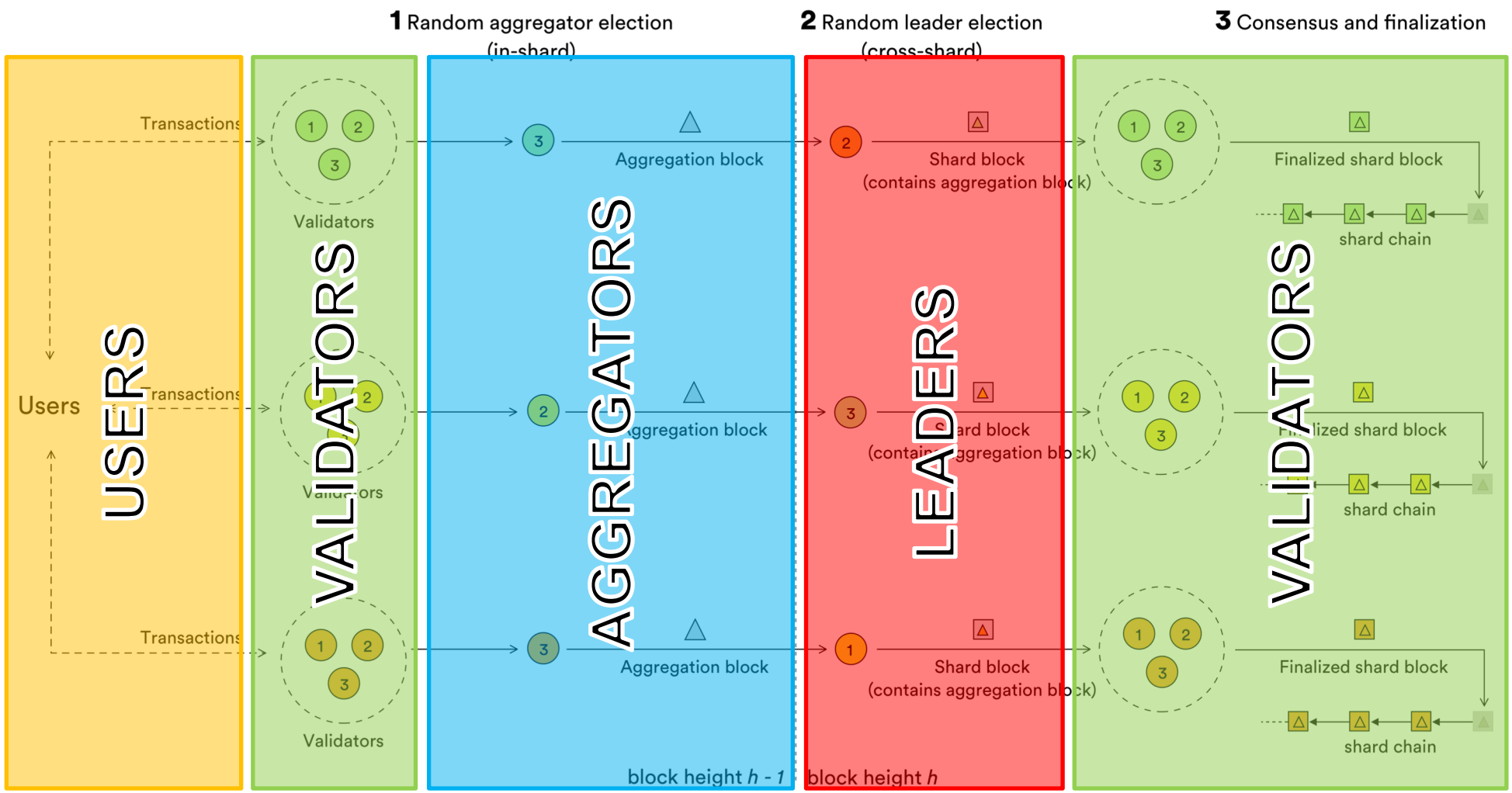
- **Dynamic Load-Balancing**
  Computational work is equally divided by dynamically adjusting the number of partitions (shards)
- **Self-Contained Proofs**
  Miners can verify the validity of transaction without the blockchain history
- **Transaction Aggregation**
  Automatically reduce the overall size of the blockchain over time

Overview of Entities

- **User**
  Uses Bazo's infrastructure to transfer funds and run smart contracts
- **Validator**
  Participates in Bazo's consensus protocol and validates blocks
- **Leader**
  A special type of validator who has the right to append the next block to a random shard
- **Aggregator**
  A special type of validator who aggregates transactions of a shard



Protocol Overview



Users

Assume user *U*.

- ***U* uses Bazo's infrastructure to send/receive coins and run smart contracts.**
- ***U* has a wallet with a public-private keypair (pk_{wall}, sk_{wall}).**
- ***U* creates a transaction by including a self-contained proof.**

Self-Contained Proofs (SCPs)

- **Traditionally, miners require the full blockchain to validate a transaction of a user**
  e.g. does the user have sufficient funds to execute the transaction
- **Instead of validators requiring the full blockchain to validate a transaction, a user has to provide all required proofs in the transaction in order for validators to verify a transaction, independent of the blockchain.**
- **A SCP is a list of Merkle proofs. A user has to provide one Merkle proof for each block that contains one or more transactions where he or she sent or received funds**

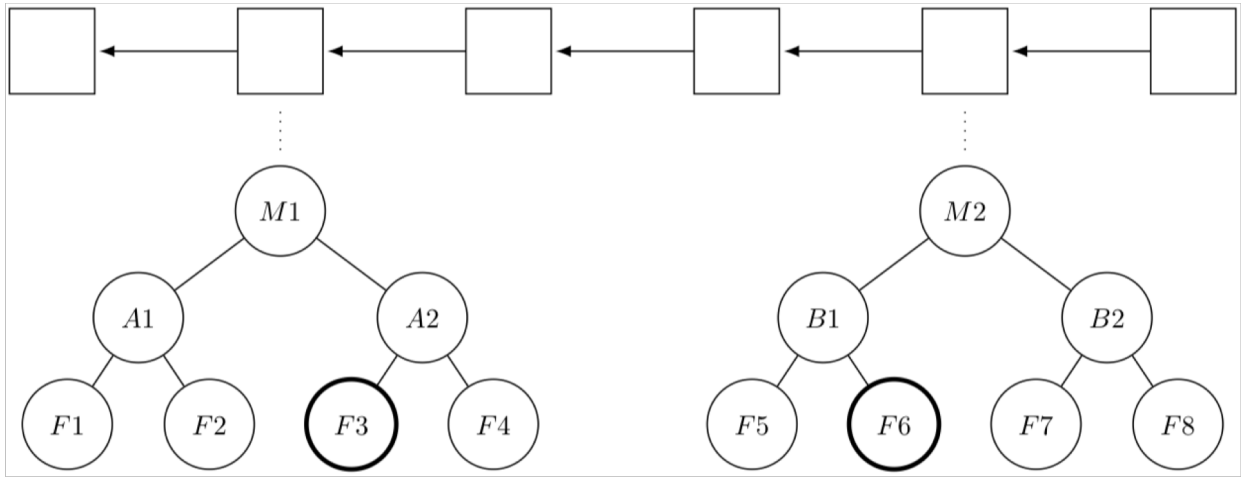
Example of a SCP

- **Assume a user *A* with past transactions *F3* and *F6***
- **A new transaction of user *A* has to include two Merkle proofs, i.e.,**

$$SCP = \{Proof_1, Proof_2\}$$

where

$$Proof_1 = \{F5, F6, B2\}$$
$$Proof_2 = \{F3, F4, A1\}$$

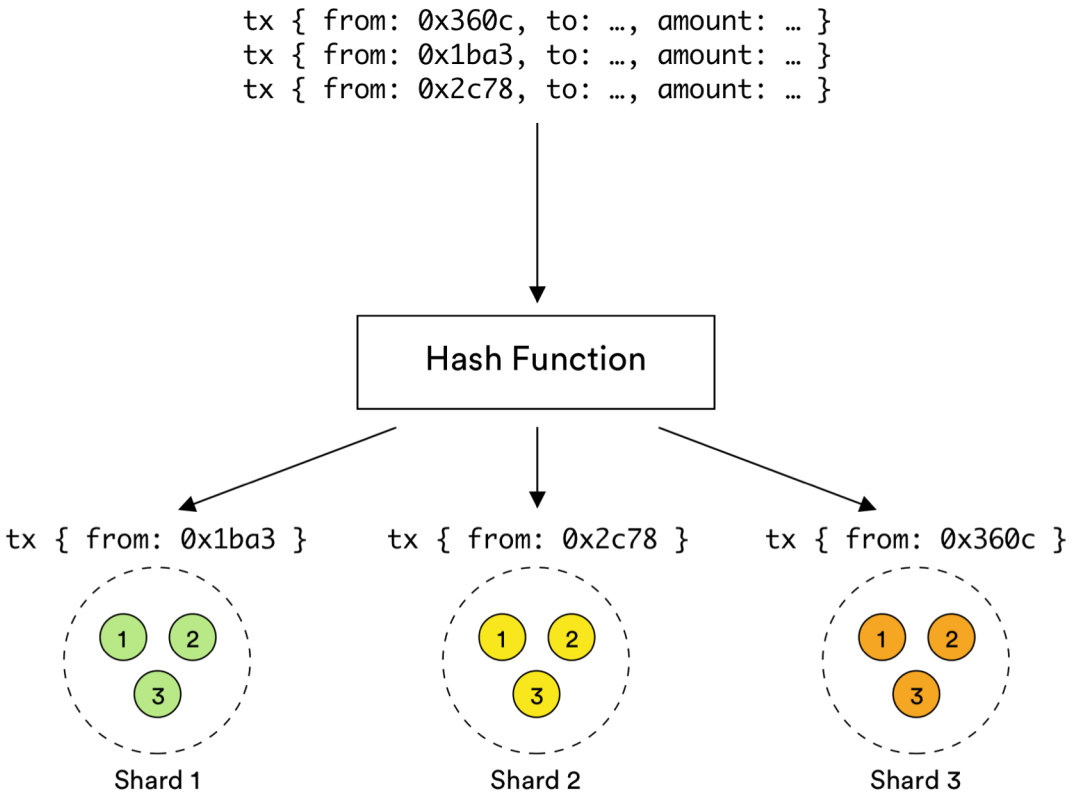


Validators

Assume validator *V*.

In addition to all properties of a user,

- ***V* participates in Bazo's consensus protocol and validates blocks.**
- ***V* has a commitment keypair (pk_{comm}, sk_{comm}) to join the set of validators.**
- ***V* validates blocks based on pk_{wall}.**
- ***V* stores block headers of all shards and block bodies based on pk_{wall}.**

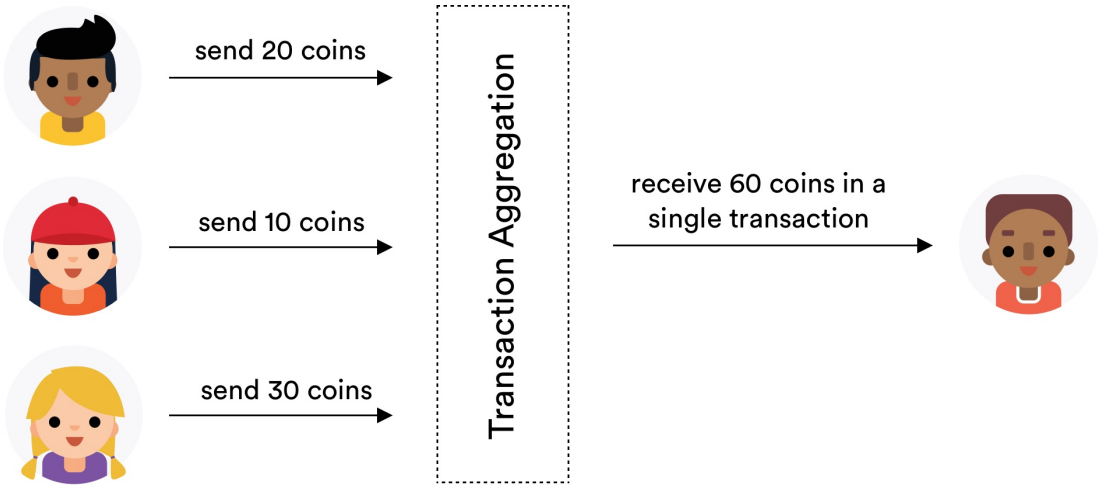


Aggregators

Assume aggregator *A*.

In addition to all properties of a validator,

- **an aggregator is a special type of validator who aggregates transactions of a shard.**
- ***A* creates a special type of transaction called "Aggregation Transaction" by summing up transferred amounts.**
- ***A* aggregates transactions based on pk_{wall}.**
- ***A* adds aggregation transactions to a special type of block called "Aggregation Block".**



Leaders

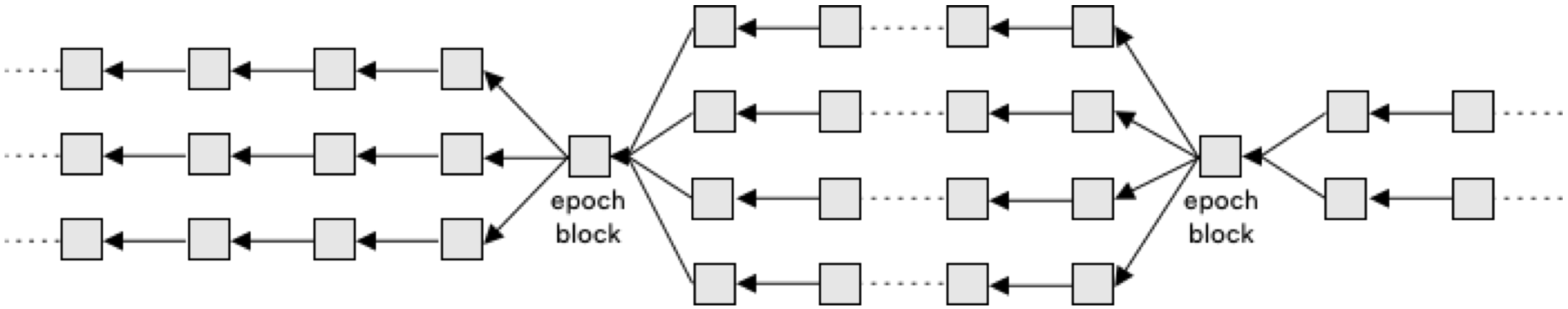
Assume leader *L*.

In addition to all properties of a validator,

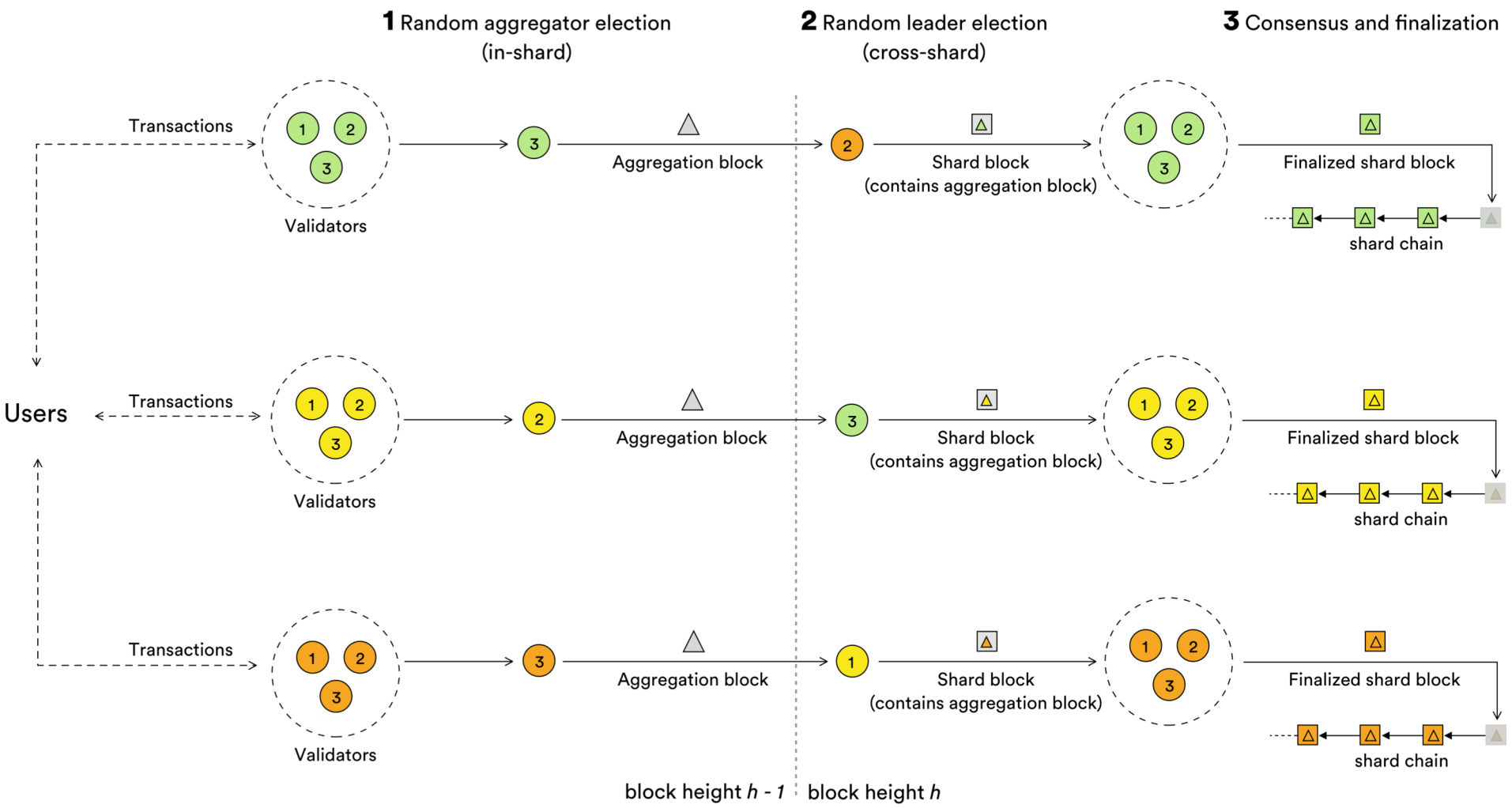
- **a leader is a special type of validator who has the right to append the next block to a random shard.**
- ***L* participates in the election process to become a leader of a random shard *S*.**
- ***L* validates funds transactions and adds them to a new block *B*.**
- ***L* further validates an aggregation block and includes them in block *B*.**
- ***L* proposes block *B* to validators of shard *S*.**

Epochs

- The algorithm proceeds in epochs.
- An epoch ends after a predefined number of blocks.
- An epoch finalizes with a special type of block called epoch block.
- An epoch block serves as a marker and is required for load-balancing.



Protocol Overview



But, does it scale?

- **It should.**

But, does it scale?

- ~~**It should.**~~
- **Transactions are distributed between shards**
  Increases scalability of the network.
- **Computational resources are distributed between nodes**
  Reduces system requirements and cost for running a node.
- **Leaders are elected randomly**
  Mitigate single shard attacks
- **Minimal cross-shard communication**
  Reduces overhead of communication and waiting times between nodes.

Questions?

Roman Blum
roman.blum@hsr.ch

References

- Scale in Distributed Systems
  B. Clifford Neuman, 1994
- Notes on Scalable Blockchain Protocols (version 0.3.2)
  Vitalik Buterin, 2015
- Introduction to Parallel Computing 2nd Edition
  A. *Grama, A. Gupta, G. Karypis, V. Kumar, 2003*
- Scalability for the Bazo Blockchain with Sharding
  R. Blum, 2018
- Sharding in MongoDB, Link: https://docs.mongodb.com/manual/sharding/
  MongoDB Documentation, visited 14th Aug 2018