Decentral Groupmanagement in Messengers with Delegated Proof of Stake

AndHell - 12.09.2019

Index

- Basics
- Idea
- Design
- Simulation
- Result

Basics

- Groupchats
    - Signal
    - WhatsApp
- Delegated Proof of Stake (DPoS)

Groupchats

Message thread with multiple participants

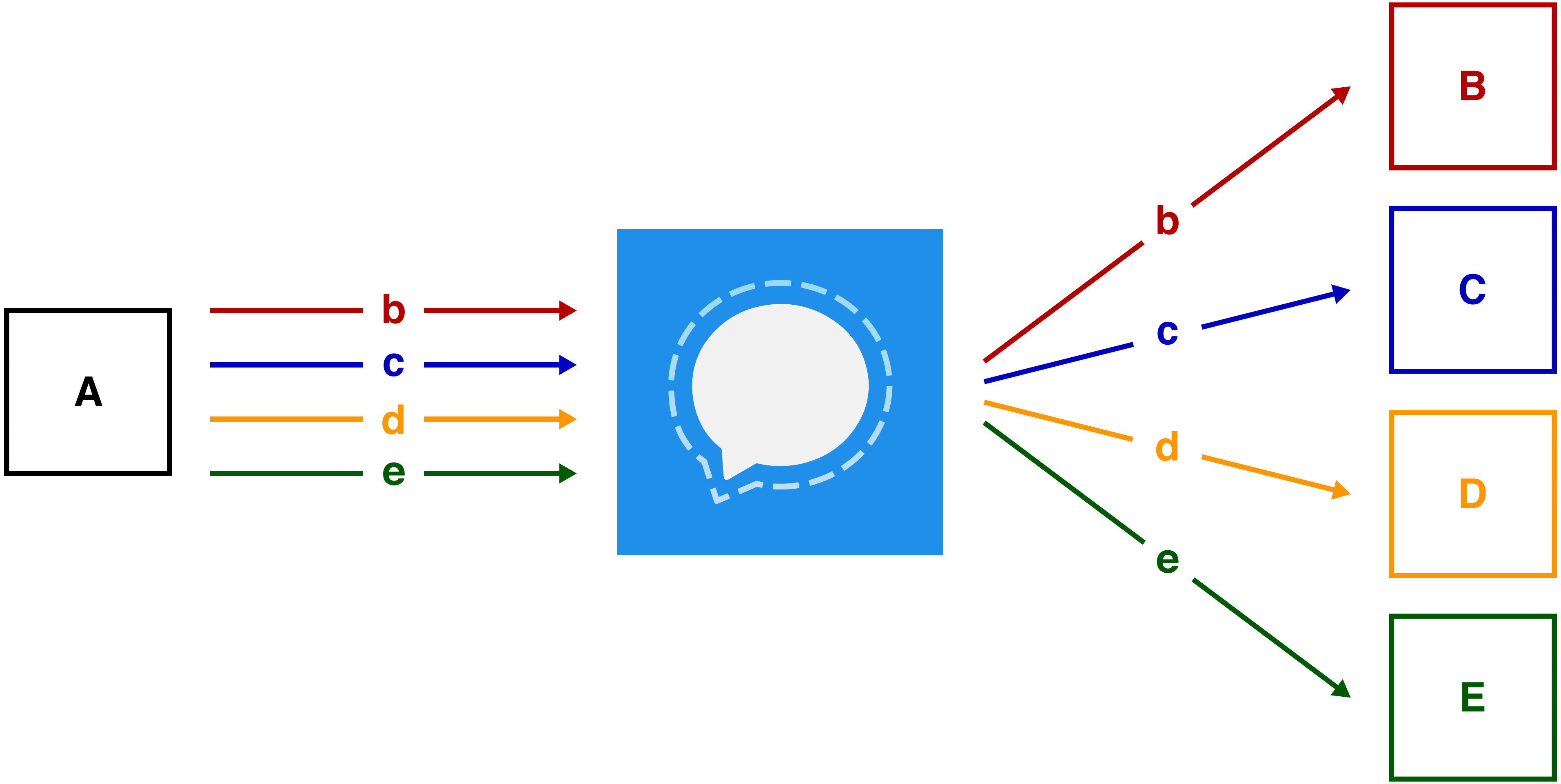
Asynchornous

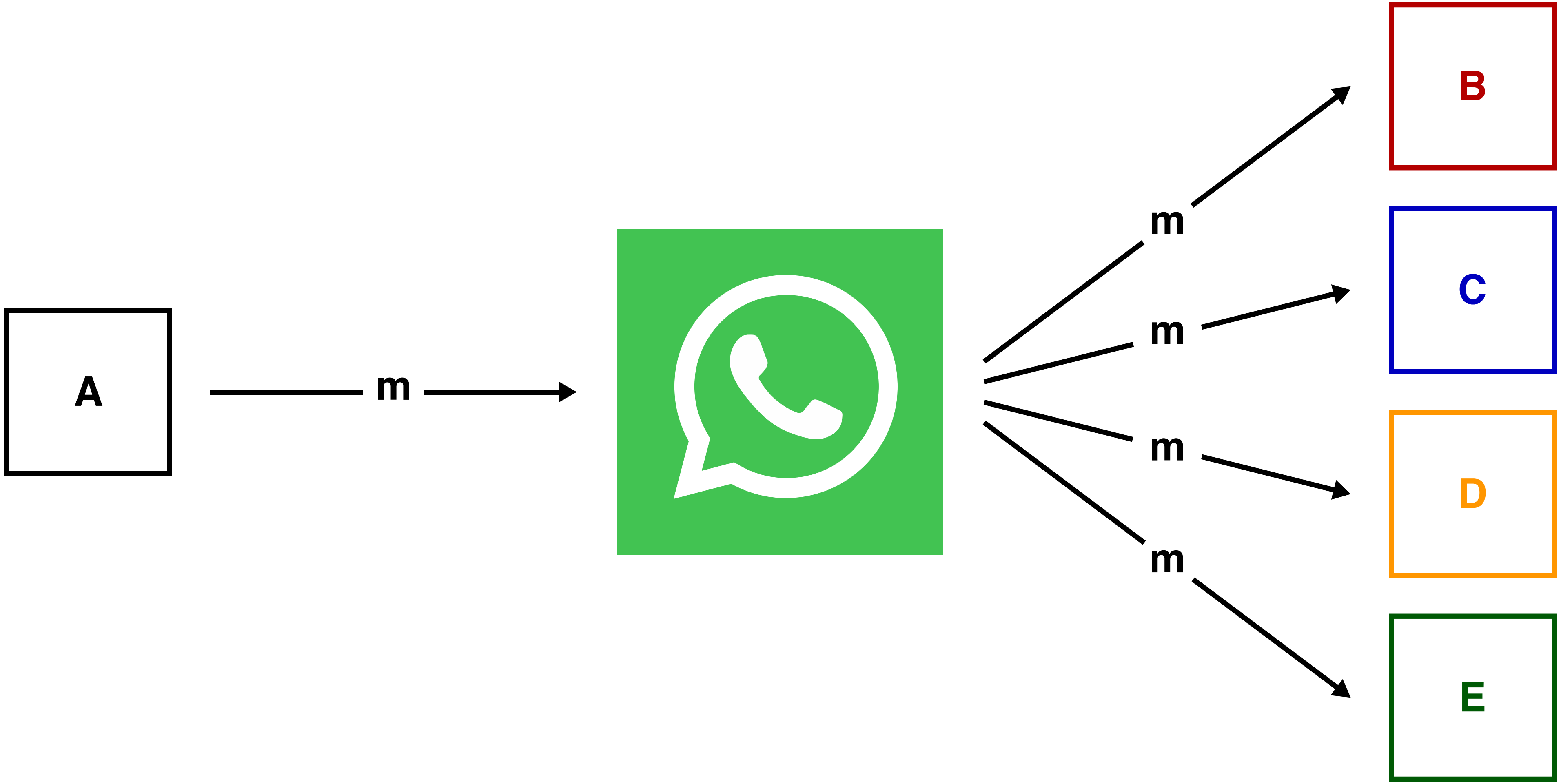
$$gr = (Id_{gr}, \mathcal{M}_{gr}, \mathcal{M}^{*}_{gr}, info_{gr})$$

Clossness

Privacy

Same State


A
b
c
d
e
b
c
d
e
B
C
D
E


A
m
m
m
m
m
B
C
D
E

Delegated Proof of Stake

Delegated Proof of Stake

- Members **elect** *Delegates*
  - Votes are attached to Transactions
  - Votes can be shared at any time
  - Count votes every N blocks
- *Delegates* build blocks
  - Signing of *Transactions*
  - Predefined order

Idea

Idea

Groupmanagment with Delegates Proof of Stake

- Decentral Management
  - No central Server
  - No *Admins*
  - Suggestion + Confirmation
- Groupstate secured by a Blockchain

Design

- Blockchain
- Voting

Basics

User

- unique ID $A, B, C, D, ...$
- Public Keys are known

Group

- Message delivery as in Signal
- Private

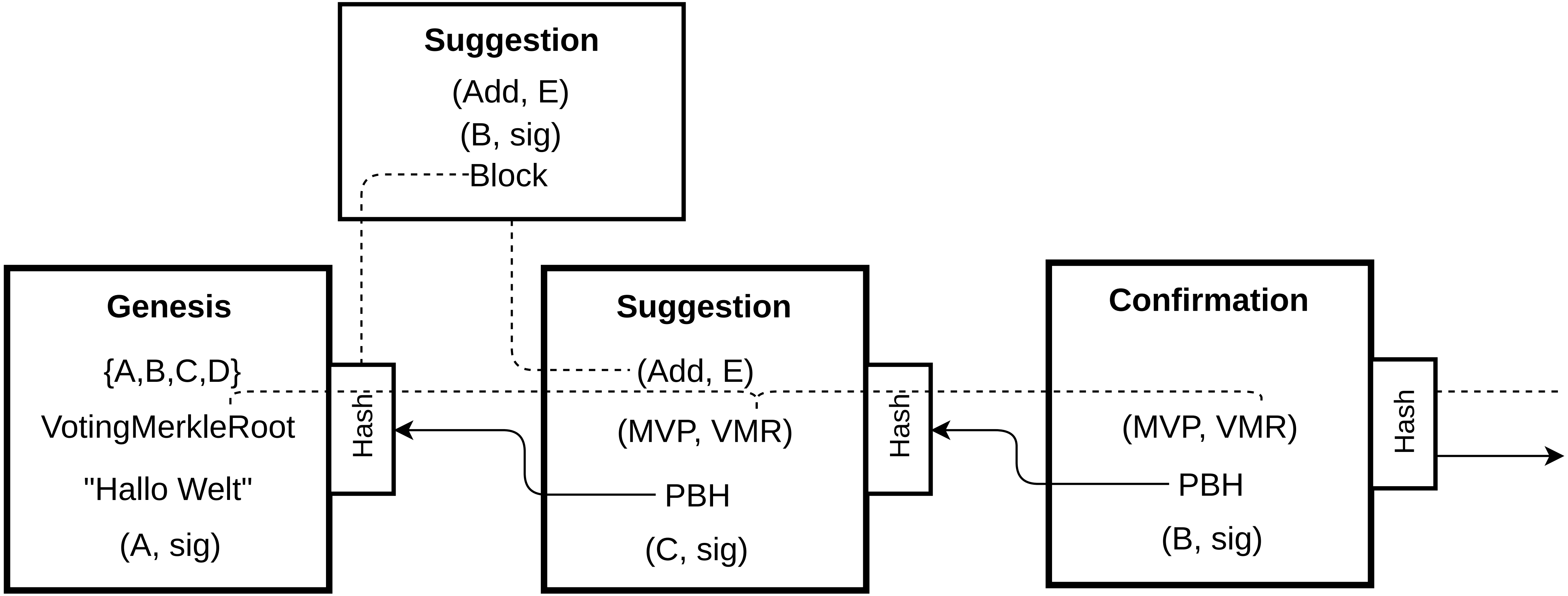
Blockchain

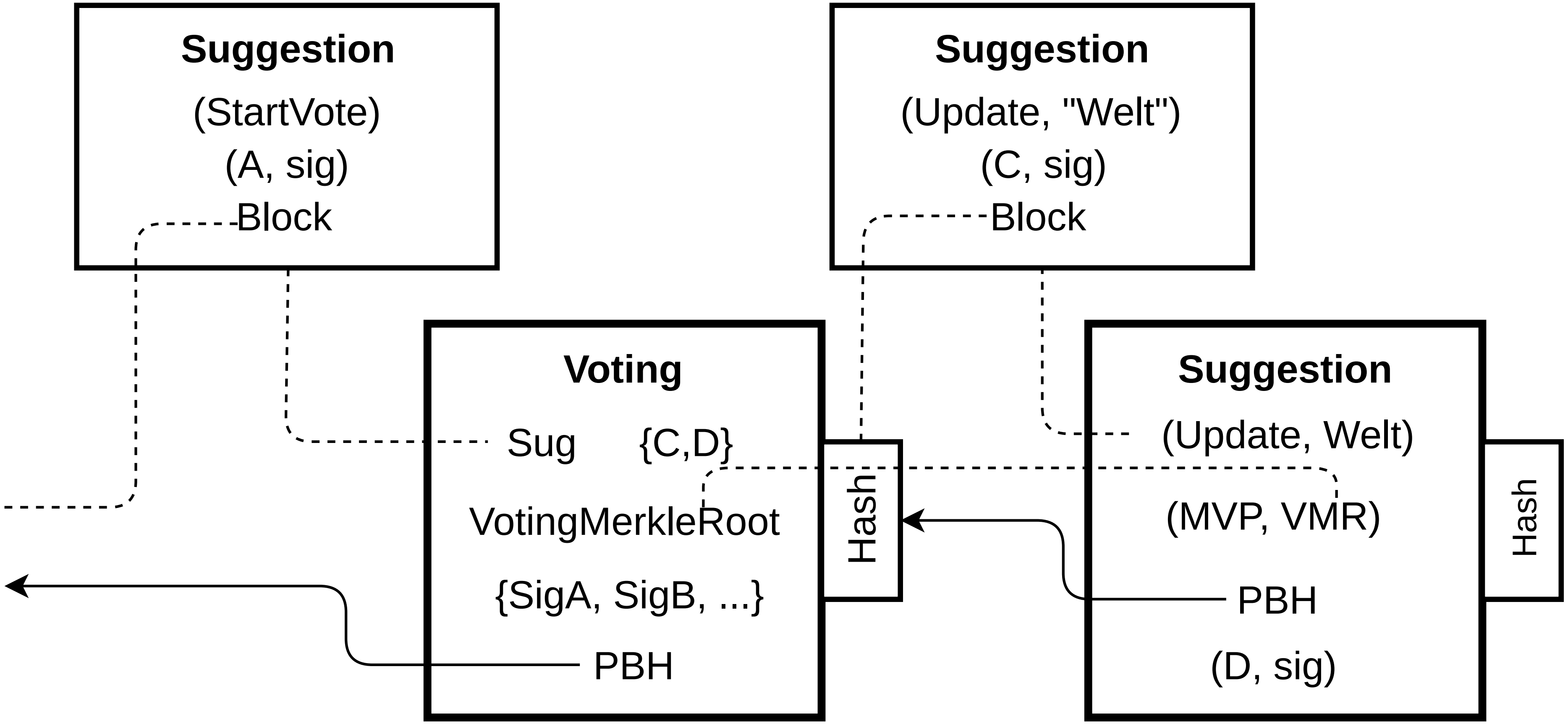
Datastructure for the histroy of a Group

- by hashes chained list of signed blocks
- Information
    - Members
    - Meta Information (Name, Picture, ...)
    - Delegates

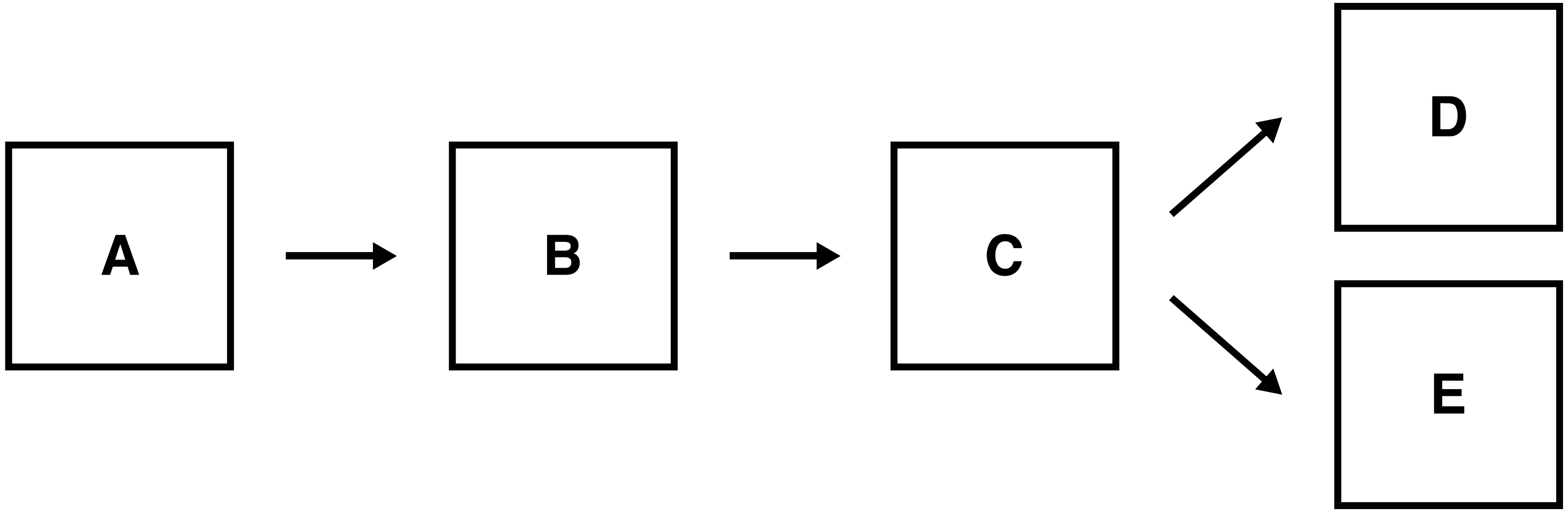
Blocks

- GenesisBlock
- SuggestionBlock
- ConfirmationBlock
- VotingBlock

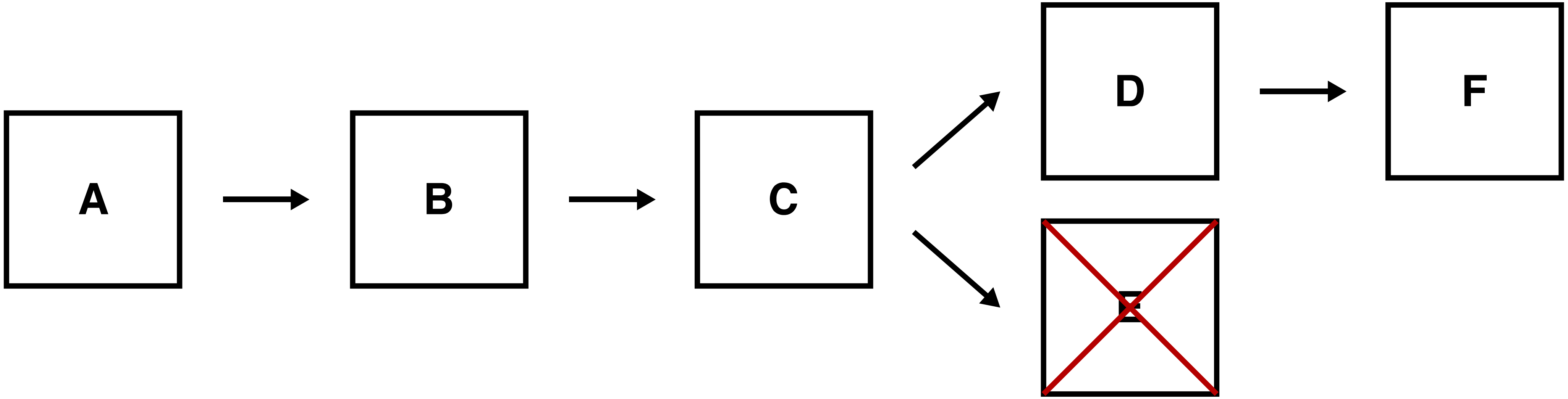

Suggestion
(Add, E)
(B, sig)
Block
Genesis
{A,B,C,D}
VotingMerkleRoot
"Hallo Welt"
(A, sig)
Hash
Suggestion
(Add, E)
(MVP, VMR)
PBH
(C, sig)
Hash
Confirmation
(MVP, VMR)
PBH
(B, sig)
Hash


Suggestion
(StartVote)
(A, sig)
Block
Suggestion
(Update, "Welt")
(C, sig)
Block
Voting
Sug {C,D}
VotingMerkleRoot
{SigA, SigB, ...}
PBH
Hash
Suggestion
(Update, Welt)
(MVP, VMR)
PBH
(D, sig)
Hash

Forks



Forks



Voting

- *Election Days*
- Can be started by everyone
- Two Timeframes
  - Voting Window
    $vote = ([M], mr, sig_{mr})$
  - Collection Window
    $vResult = ([M], mr, sig_{mr}, [vote])$

Simulation

Technologie

- Console application
- C# (dotnetcore)
- libsodium
- RabbitMQ

Clients

- sleep
- process new messages
- Action requiered?
  - Voting
  - Sign Suggestion
- [Optional] send random Suggestion

Result

Clossness

Privacy

Same State

Result - Voting

- scalability for large groups
- synchronous
- public votes

- asynchronous Voting

Result - Forks

- Information loss
- Can be exploited by delegates

- server-side-timestamp
- Fork unification

Conclusion

</Slides>