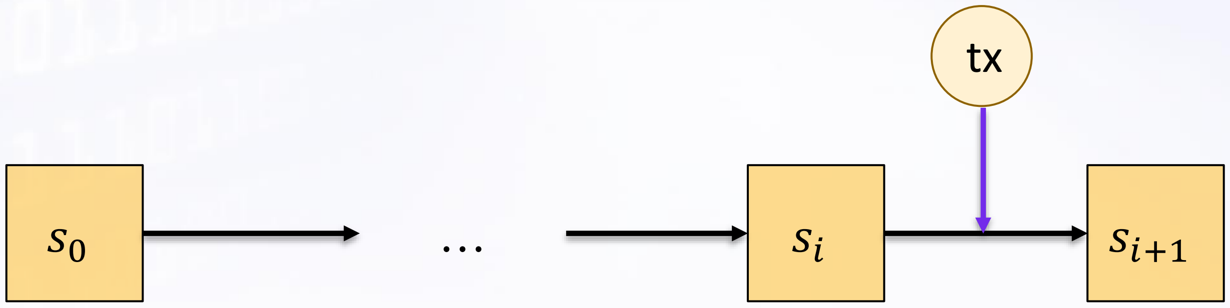
Integration of MPC into Besu through an extended private transaction model

Daniel Morales, Isaac Agudo, Javier Lopez
Network, Information and Computer Security (NICS) Lab
Universidad de Málaga, Spain

Introduction

- What is a Blockchain?
  - Decentralized network
  - Immutable storage
  - Data integrity
  - Byzantine Fault Tolerant Consensus protocols (PoW, PoS, PoA…)
- Use cases
  - Financial, e-government, voting, medical, metaverse…
- Generalization: State Transition System
  - $(S, \rightarrow)$, with S the set of all possible states and $\rightarrow$ the relation between states
  - $APPLY(s_i, tx) \rightarrow \{s_{i+1} \vee error\}$



NICS

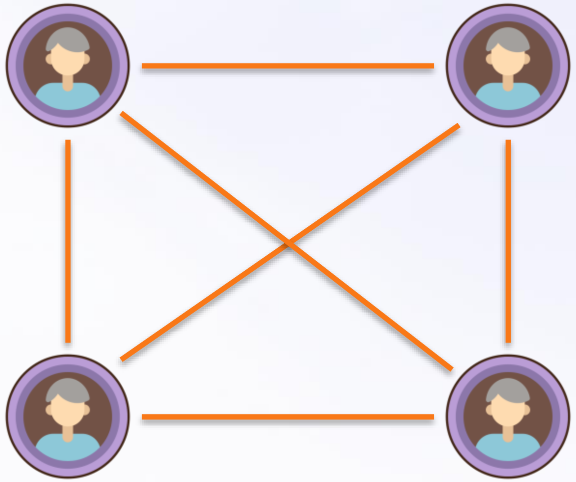
Privacy problem and related work

- Privacy/confidentiality problem
  - Every data must be public to enable verifiability
  - End-users can be profiled and deanonymized
  - Many use cases work with personal/sensitive data, e.g., biometrics
- Solutions
  - Cryptocurrencies
    - Mixers: break the link between sender and receiver
  - Smart contracts
    - Quorum / Besu: private transactions between a specific subgroup
  - Privacy by design
    - Arbitrum: off-chain computation of private data
    - Hawk: privacy based on Zero Knowledge Proofs
    - Oasis Network / Secret Network: privacy based on Trusted Execution Environments

NICS

MPC – Decentralized confidential computations

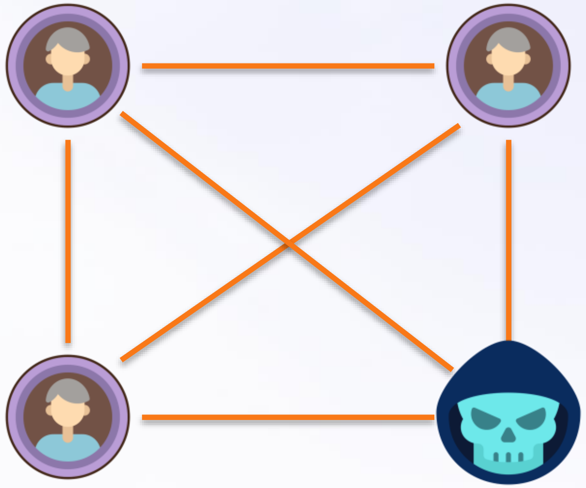
- Secure Multi-Party Computation (MPC)
  - Why MPC?
    - Decentralized cryptographic protocols
    - Enable computation on encrypted data
  - A set of parties $\{P_1, \dots, P_n\}$ jointly compute a function $f(\cdot)$ on private data $\{x_1, \dots, x_n\}$
  - **Correctness**
    - The parties involved in the computation obtain the desired output correctly, i.e., MPC computes $y \leftarrow f(x_1, \dots, x_n)$ and nothing else
  - **Privacy**
    - Party P_i does not learn anything but x_i and y



NICS

MPC – Decentralized confidential computations

- Secure Multi-Party Computation (MPC)
  - Why MPC?
    - Decentralized cryptographic protocols
    - Enable computation on encrypted data
  - A set of parties $\{P_1, \ldots, P_n\}$ jointly compute a function $f(\cdot)$ on private data $\{x_1, \ldots, x_n\}$
  - **Correctness**
    - The parties involved in the computation obtain the desired output correctly, i.e., MPC computes $y \leftarrow f(x_1, \ldots, x_n)$ and nothing else
  - **Privacy**
    - Party P_i does not learn anything but x_i and y
  - MPC is secure even if some parties are malicious



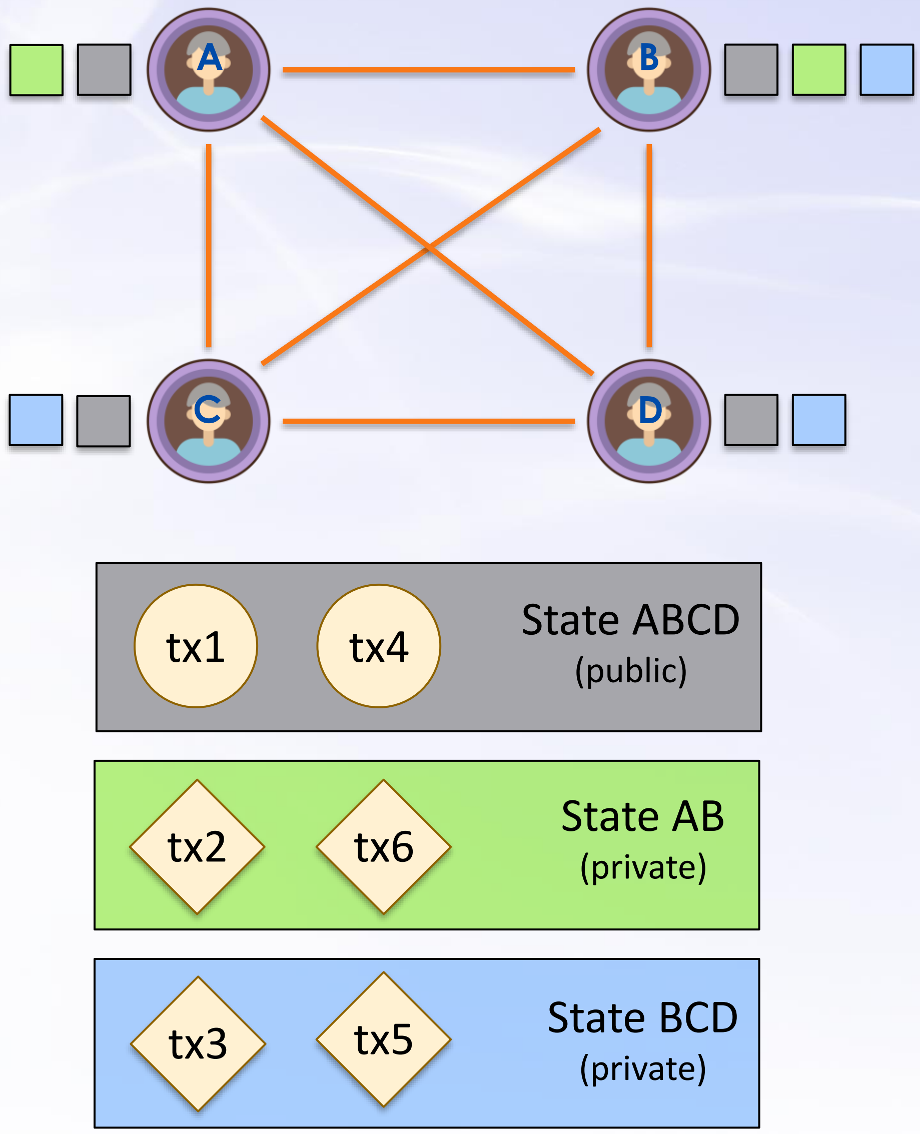
NICS

Hyperledger Besu - Definitions

- Why Hyperledger Besu?
  - An Ethereum blockchain client
  - Permissioned network
  - Supports private transactions

- Privacy group
  - A **subgroup of the blockchain** network whose nodes share the **same view of the state**
  - A private transaction is only visible to a specific privacy group
  - Extended state of one node:

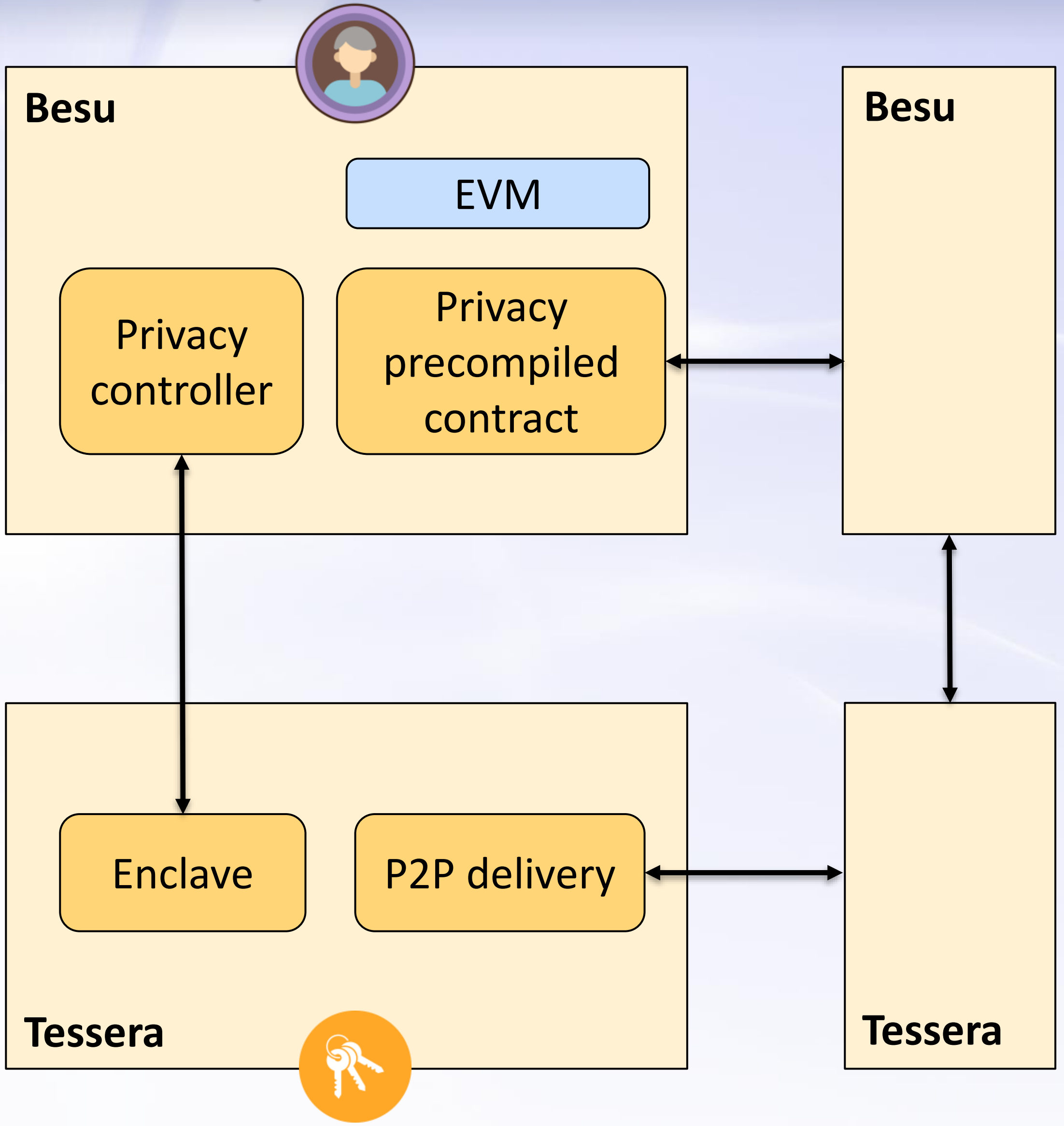
$$S_{ext} = S_{pub} \cup \left(\bigcup_{N \in P} S_N^P \right)$$



NICS

Hyperledger Besu – Main components

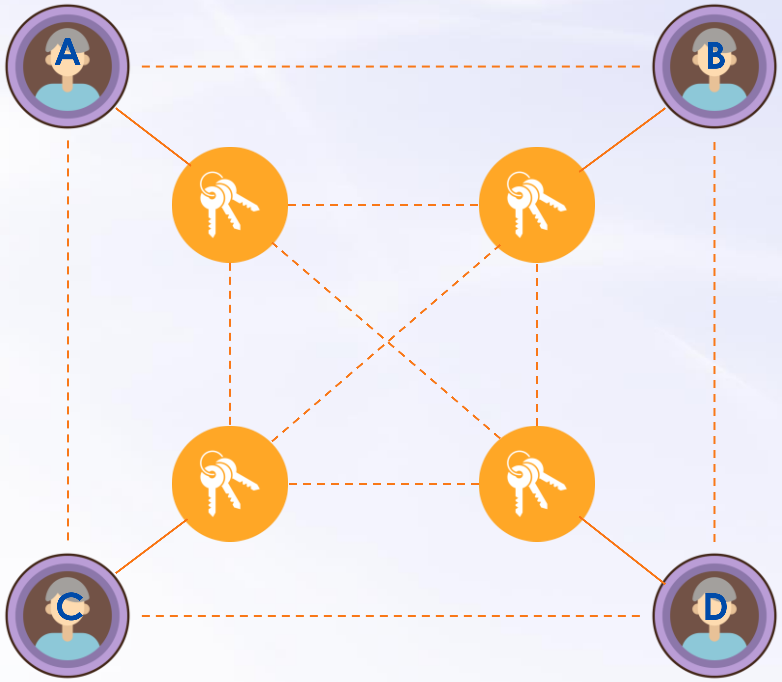
- Private transaction manager (Tessera)
  - Performs private transaction delivery
- Privacy controller
  - Manages private transactions in Besu and communicates with the private transaction manager
- Privacy precompiled contract
  - A special purpose Smart contract that resides inside Besu and manages private transaction execution



NICS

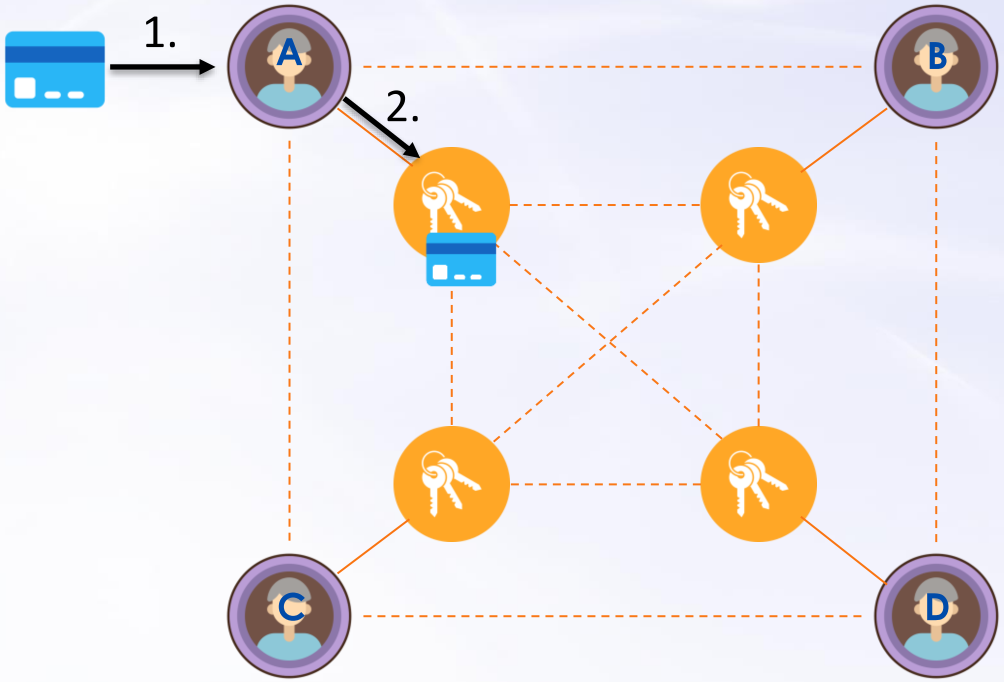
Hyperledger Besu - Example

- Sending private transactions in Hyperledger Besu



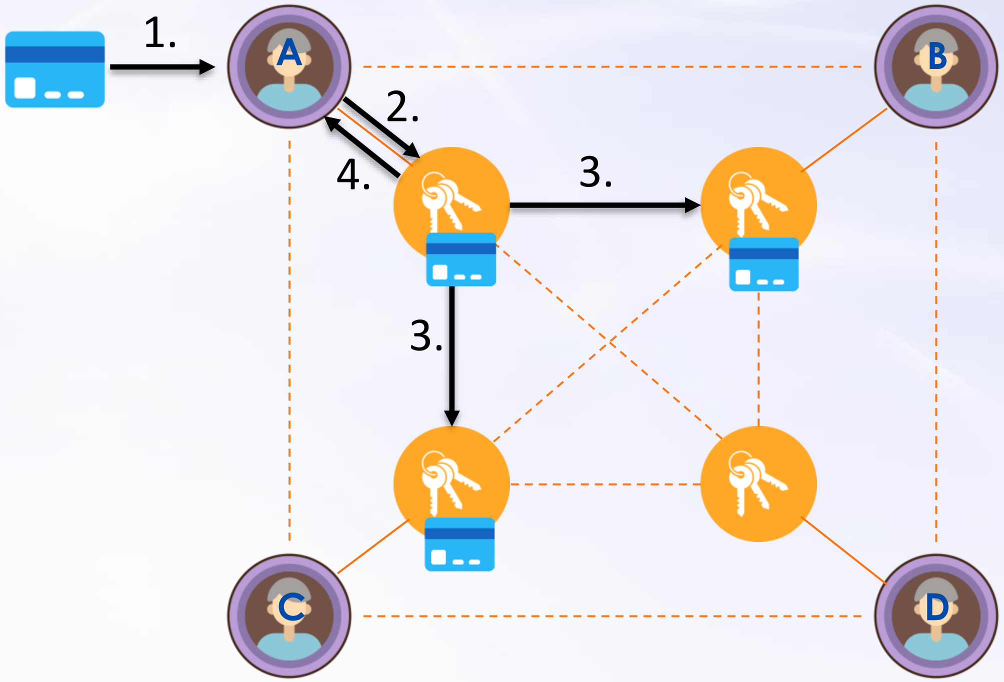
Hyperledger Besu - Example

- Sending private transactions in Hyperledger Besu
  - 1. New transaction
    - From: A
    - To: BC
  - 2. Send transaction



Hyperledger Besu - Example

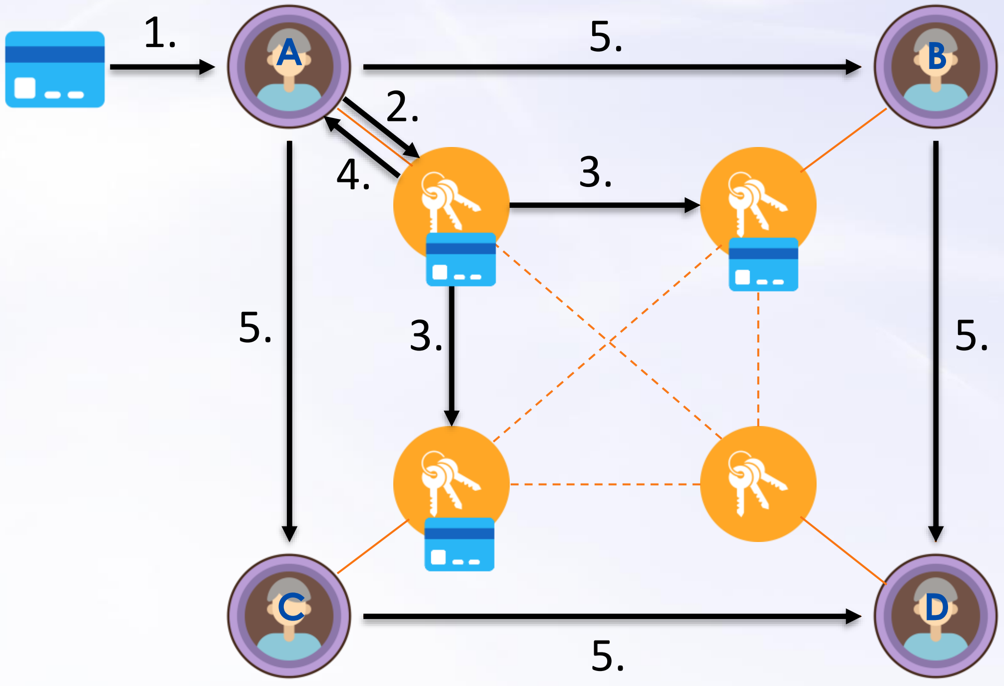
- Sending private transactions in Hyperledger Besu
  - 1. New transaction
    - From: A
    - To: BC
  - 2. Send transaction
  - 3. Push transaction
  - 4. Receive key (pseudorandom ID)



NICS

Hyperledger Besu - Example

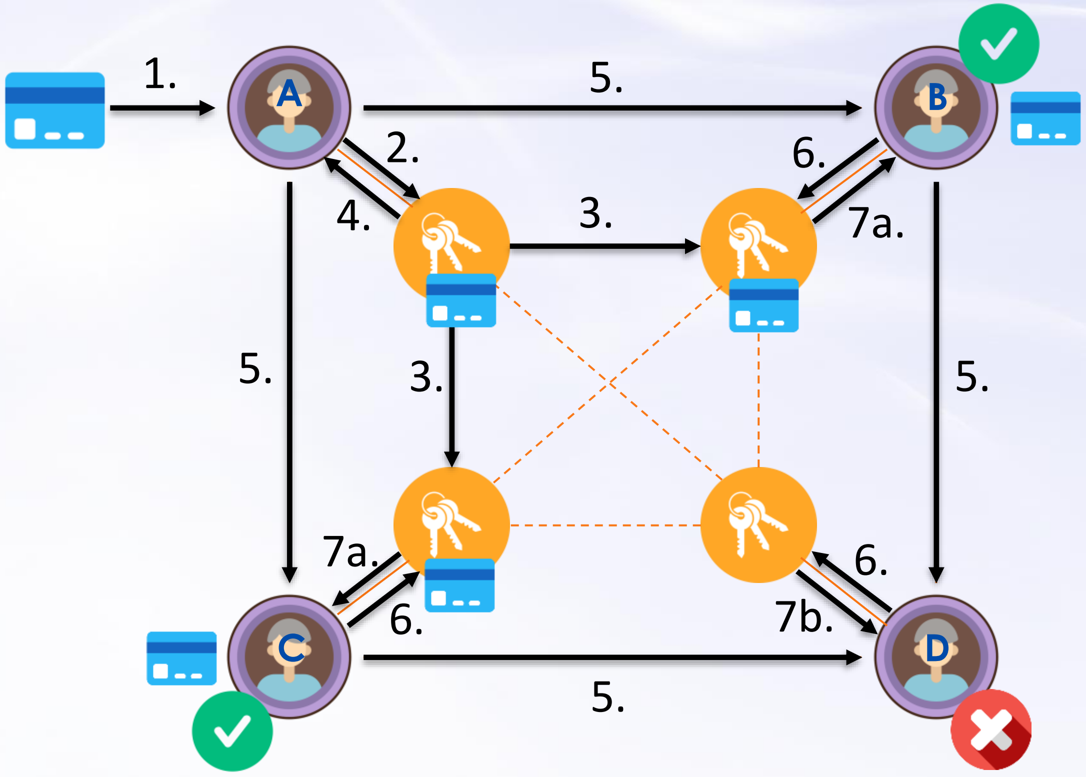
- Sending private transactions in Hyperledger Besu
  - 1. New transaction
    - From: A
    - To: BC
  - 2. Send transaction
  - 3. Push transaction
  - 4. Receive key (pseudorandom ID)
  - 5. Flood Privacy Marker Transaction



NICS

Hyperledger Besu - Example

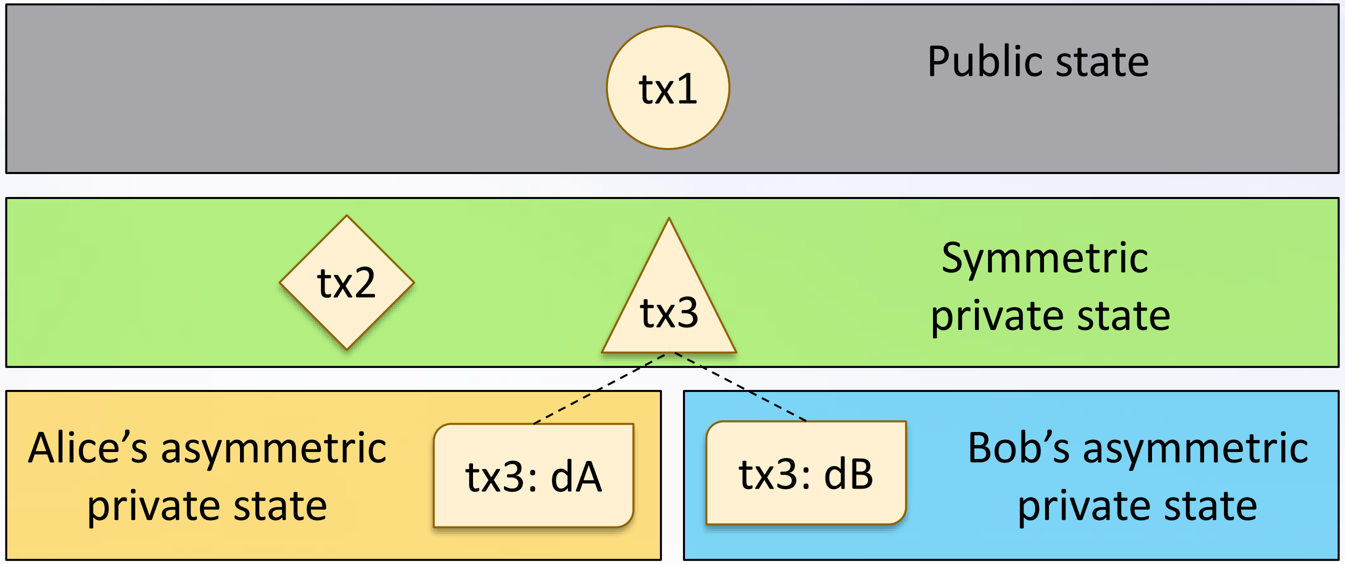
- Sending private transactions in Hyperledger Besu
  - 1. New transaction
    - From: A
    - To: BC
  - 2. Send transaction
  - 3. Push transaction
  - 4. Receive key (pseudorandom ID)
  - 5. Flood Privacy Marker Transaction
  - 6. Query Private Marker Transaction
  - 7a. OK: transaction available
  - 7b. ERROR: transaction not available



NICS

Extended private transaction model for MPC

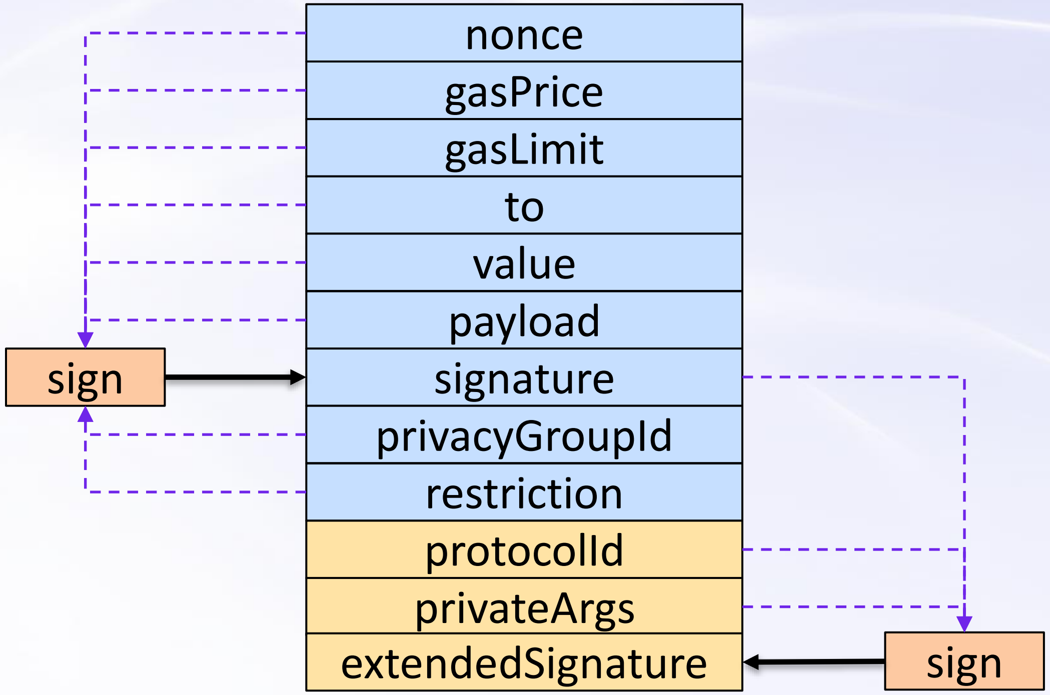
- MPC allows an asymmetric state model
  - Alice and Bob share a privacy group
  - They see the same view for symmetric private transactions
  - They see different data regarding an asymmetric private transaction
    - Reminder: In MPC, x_i is only seen (in plaintext) by the owner P_i



Extended private transaction model for MPC

- Extended private transactions
  - A new type of transaction supporting MPC executions
  - It extends the standard private transaction of Besu
    - **ProtocolId:** specifies which MPC protocol is executed
    - **PrivateArgs:** carries the private data of one user
      - Standard payload is filled with dummy data
    - **ExtendedSignature:** signs extended data
  - **Signature** is publicly verifiable by any node in the blockchain, but **ExtendedSignature** is only verifiable at user's Besu client

Extended private transaction's fields



NICS

Extended private transaction model for MPC

- Interface Smart Contract
  - It exposes the MPC methods to the application using a standard interface
  - Besu translates the contract methods to MPC messages

```
contract PSI {
    constructor(
        address _alice,
        address _bob,
        uint256 _universeLength) public;
    function load(
        uint256[] memory _aliceSet) public
        onlyAlice();
    function consume(
        uint256[] memory _bobSet) public
        onlyBob()
        returns(uint256[] memory);
}
```

```
contract BftRng {
    constructor(
        address[] _participants) public;
    function generate() public
        returns(uint256);
    function commit() public;
    function reveal() public;
    function getRandom() public
        returns(uint256);
}
```

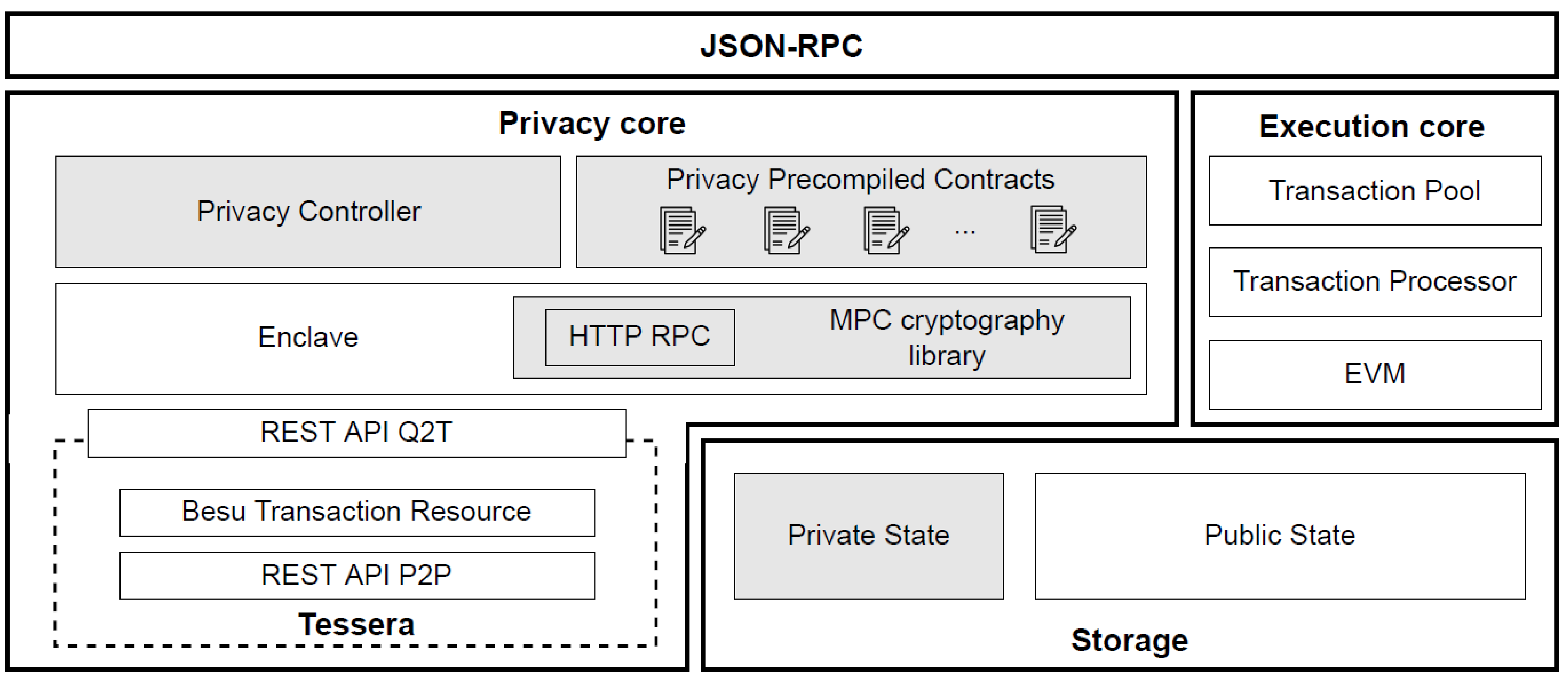
NICS

Extended private transaction model for MPC

- MPC execution
  - Interface Smart Contract ←→ Coordination layer ←→ Cryptographic MPC library
  - **Privacy controller:**
    - Extended to discriminate symmetric private transactions from those asymmetric, based on *protocolID*
    - Creates a new transaction without *privateArgs*, ready for delivery to other nodes
    - Sends the asymmetric private transaction to Tessera
  - PMT is flooded and involved nodes retrieve the private transaction from Tessera
  - **Privacy precompiled contract:**
    - One contract per each Interface Smart Contract
    - Instead of executing the transaction with the execution core (EVM), it coordinates the online execution of the MPC using the cryptographic library

NICS

Extended private transaction model for MPC



Extended private transaction model for MPC

- MPC message delivery
  - MPC protocols communicate off-chain
  - Generalize MPC to a tuple of messages $\left((h_1, m_1), \ldots, (h_n, m_n)\right)$
  - Standard Tessera does not understand MPC messages
  - 3 approaches
    - **Modify Tessera:** a new endpoint in Tessera does not store the message hash as identifier, but the header h_i
    - **Whisper protocol in Besu:** sending off-chain secure messages between Besu nodes
    - **Message Delivery Smart Contract:** a Message Delivery precompiled contract supports a message delivery Interface Smart Contract. Each MPC message triggers a new symmetric private transaction delivery, which contract contains the header h_i

Conclusions and future work

- Conclusions
  - Executing MPC protocols from blockchain applications
  - Standard interfaces: Smart contracts
  - Extended transaction model
    - On-chain coordination
    - Off-chain execution
- Future work
  - Add verifiability mechanisms for MPC
  - Dynamic models for MPC (hybrid compilers)
  - Benchmarks to test performance for different MPC protocols

NICS

Integration of MPC into Besu through an extended private transaction model

Daniel Morales, Isaac Agudo, Javier Lopez
Network, Information and Computer Security (NICS) Lab
Universidad de Málaga, Spain

NICS