Besu PKI Support

Peer and Block Creation permissioning

Lucas Saldanha @ ConsenSys

Why add PKI support?

- PKI model is used by countless companies
- Easier integration with existing infrastructure
- Recent use-case: Compliance and Liability

Peer Permissioning

- **Requirements**
    - Only **authorized** peers can communicate with other peers
    - Communication between peers must be encrypted
- Enforce TLS on DevP2P communication
    - Including client-side verification so both sides validate the counterpart certificate
- Use independent *truststore* for Peer Permissioning
- Implementation is using Netty channel handlers for seamless and transparent integration

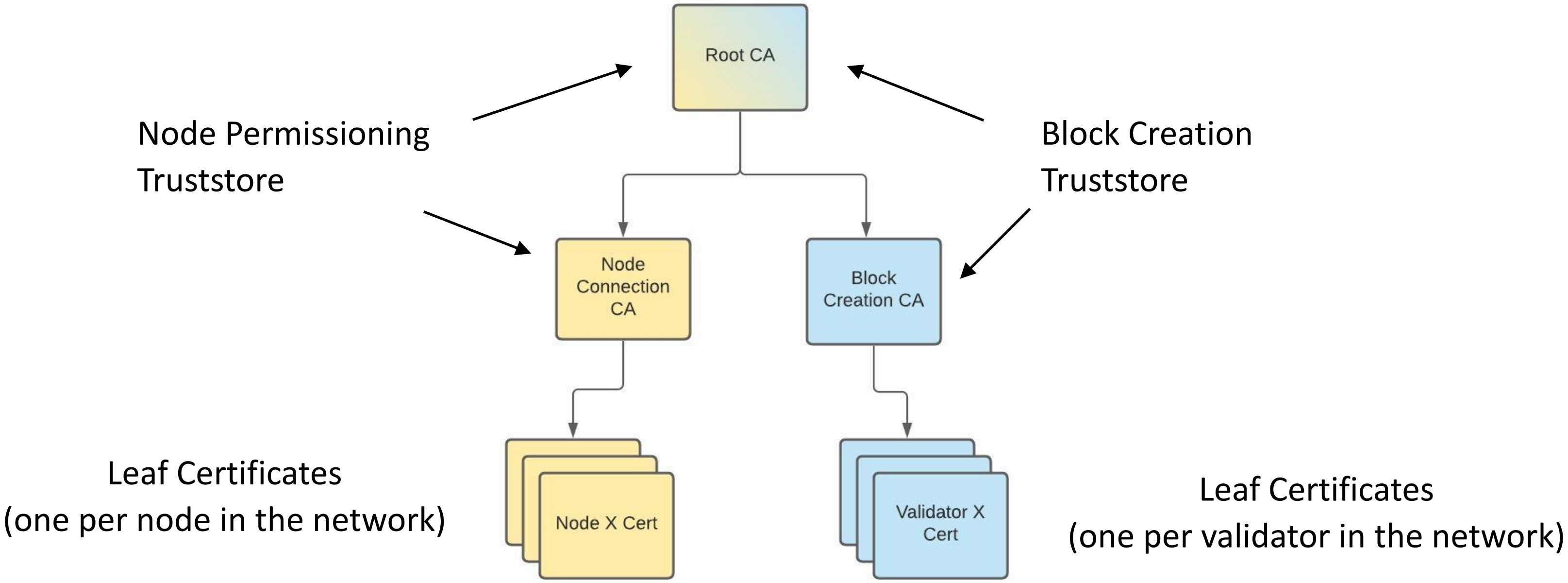
Block Creation Permissioning

- **Requirements**
    - Only blocks proposed by **authorized** validators are considered valid
    - Every block must contain a “stamp” identifying its proposer
- Adding a CMS field to the extra data in the block header
    - Only when running in this “PKI mode”
- Using independent *truststore* for Block Permissioning
- Implementation is using Bouncy Castle library for most crypto/certificate functionality

Block Creation Permissioning (cont.)

- **Proposing a block**
    - Validator creates CMS message with:
        - Signed payload (using its certificate’s private key)
        - Intermediate certificate chain (used for validation)
    - Signed payload is the proposed block hash (prevent it being reused in other blocks).
- **Validating a proposed block**
    - Verify that signed block hash matches proposed block hash
    - Validate signer’s certificate chain

Example Certificate Authority Tree



Node Permissioning Truststore

Block Creation Truststore

Leaf Certificates
(one per node in the network)

Leaf Certificates
(one per validator in the network)

Code Changes

- PKI module (https://github.com/hyperledger/besu/pull/2298)
    - Added initial code for dealing with certificates and keystores
- CMS creation and validation logic (https://github.com/hyperledger/besu/pull/2340)
    - All about CMS :)
- Netty “DeFramer” logic refactoring (https://github.com/hyperledger/besu/pull/2391)
    - Small change to enable adding extra channel handlers without breaking anything
- TODO (WIP - don’t have PRs yet)
    - Extra data changes and QBFT logic for stamping and validating proposed blocks
        - https://github.com/lucassaldanha/besu/tree/extra-data
    - DevP2P Over TLS changes
        - https://github.com/perusworld/besu/tree/p2p-over-ssl