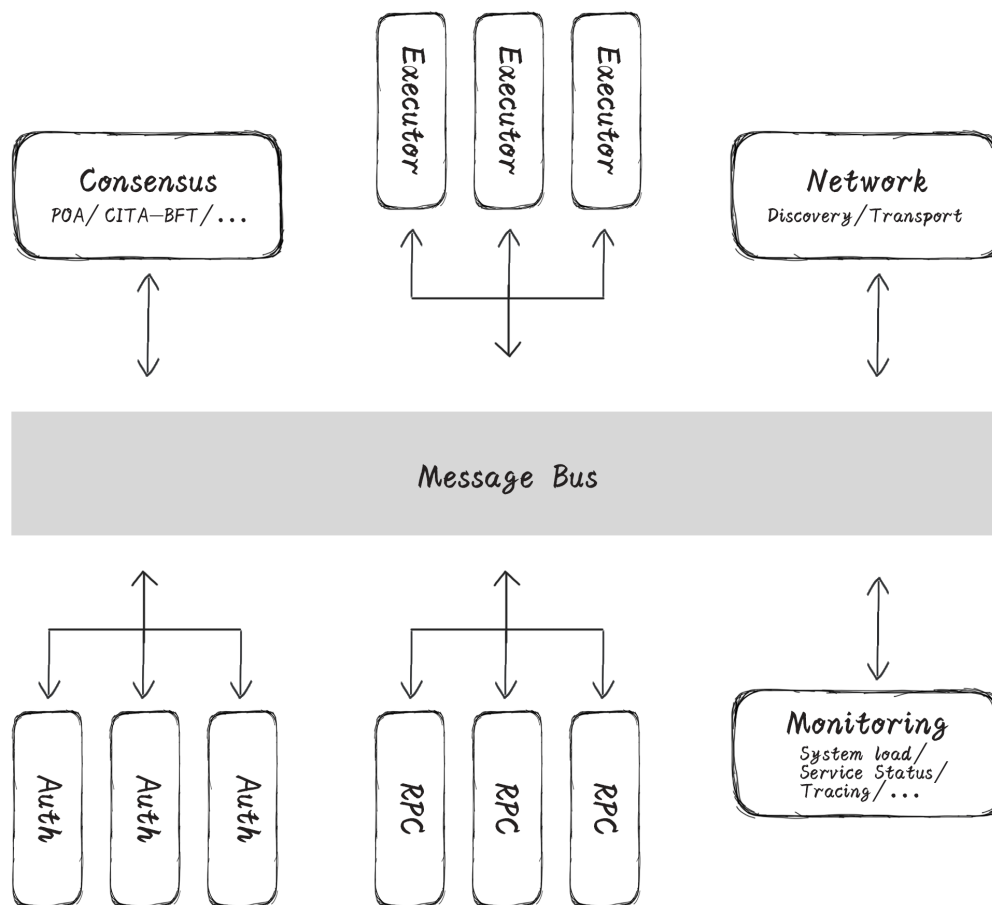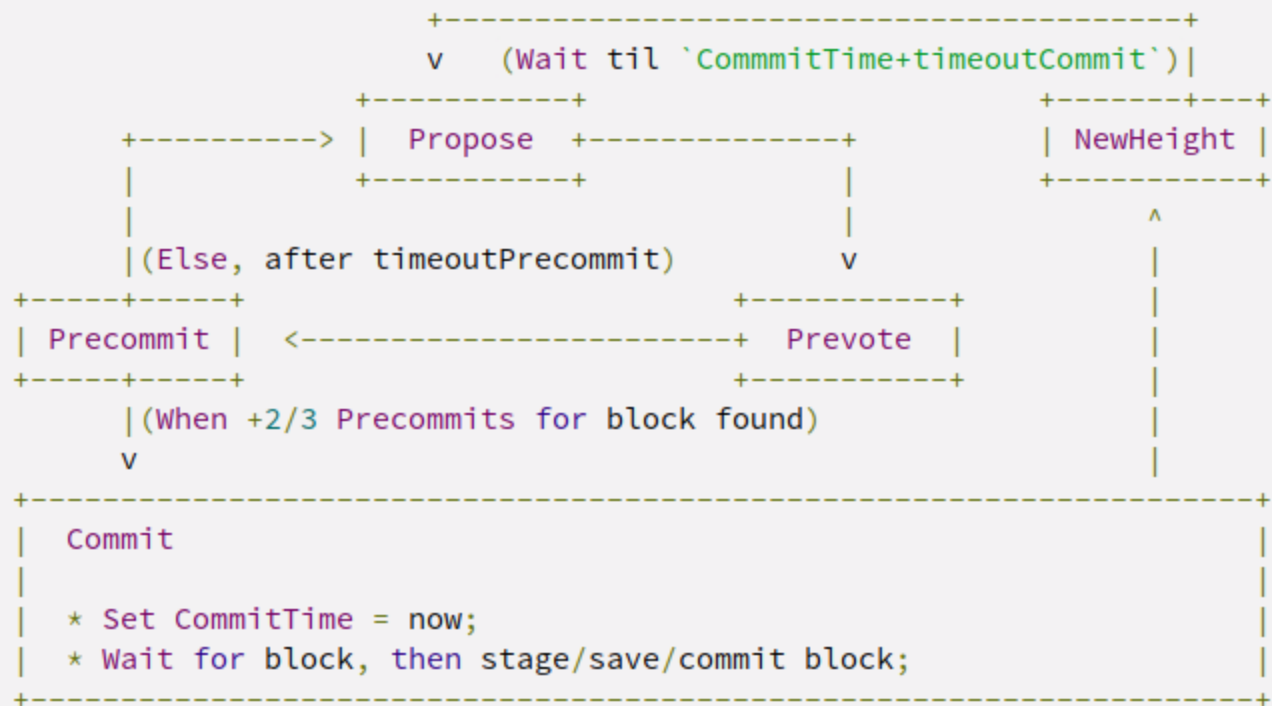Cryptape

Inter-enterprise
Trust Automation

Scalability

scale-out by sharding?

A even better way to scale
Microservices

CITA-BFT

```
                              +------------------------------------------+
                          v     (Wait til `CommmitTime+timeoutCommit`)|
                      +----------+                        +-------+---+
          +---------> |  Propose +---------------+        | NewHeight |
          |           +----------+               |        +----------+
          |                                      |             ^
          |                                      |             |
          |(Else, after timeoutPrecommit)        v             |
    +-----+-----+                          +----------+         |
    | Precommit |  <-----------------------+  Prevote |         |
    +-----+-----+                          +----------+         |
          |(When +2/3 Precommits for block found)               |
          v                                                     |
    +-----------------------------------------------------------+
    |   Commit                                                  |
    |                                                           |
    |   * Set CommitTime = now;                                 |
    |   * Wait for block, then stage/save/commit block;         |
    +-----------------------------------------------------------+
```
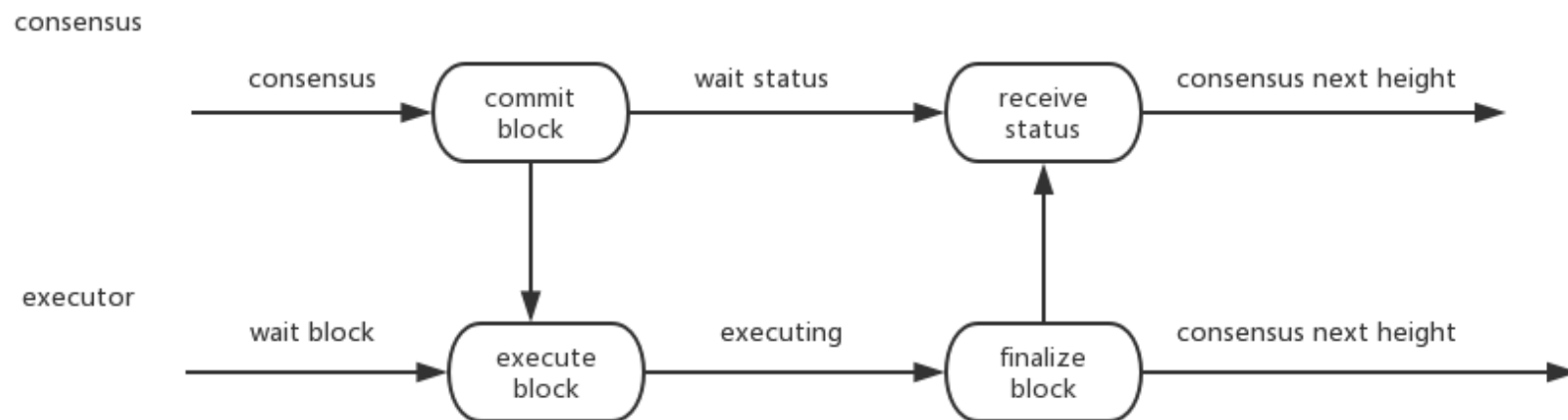
```
                    +---------------------------------------------+
                    v  (Wait til `CommmitTime+timeoutCommit`)  |
                +----------+                      +---+-------+
    +---------> |  Propose  +--------------+       | NewHeight |
    |           +----------+               |       +----------+
    |                                      |             ^
    |                                      |             |
    |(Else, after timeoutPrecommit)        v             |
+----+-----+                      +----------+           |
| Precommit |  <------------------------+  Prevote  |    |
+----+-----+                      +----------+           |
    |(When +2/3 Precommits for block found)              |
    v                                                    |
+------------------------------------------------------------+
|  Commit                                                    |
|                                                            |
|  * Wait for block, stage/save/commit block;               |
+------------------------------------------------------------+
```

Pre-processing

NewHeight -> (Propose -> Prevote -> Precommit)+ -> Commit -> NewHeight ->...

consensus

consensus → prevote → consensus → commit block → wait status → receive status → consensus next height

executor

wait proposal → pre-execute → pre-executing → check → Yes → finalize block → consensus next height

No → execute
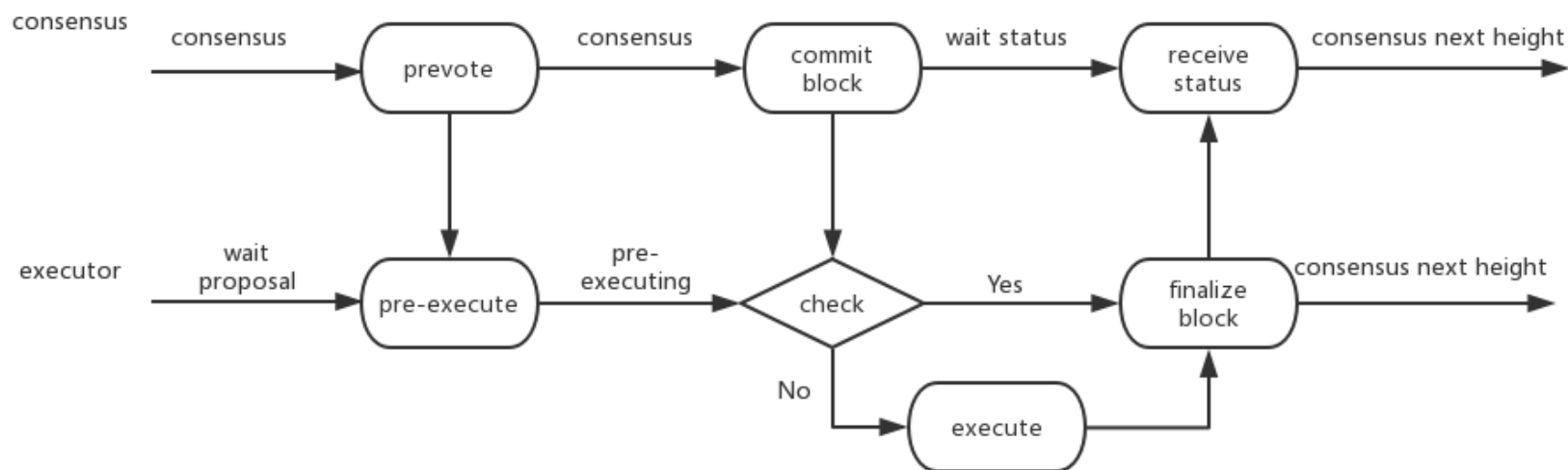
Cache

Verify Signature / Block / State ......

Message communication

Compression / package

Receipt Root

H_ABC
Hash(H_AB + H_C)

H_AB
Hash(H_A + H_B)

H_C
Hash(Tx C)

H_A
Hash(Tx A)

H_B
Hash(Tx B)
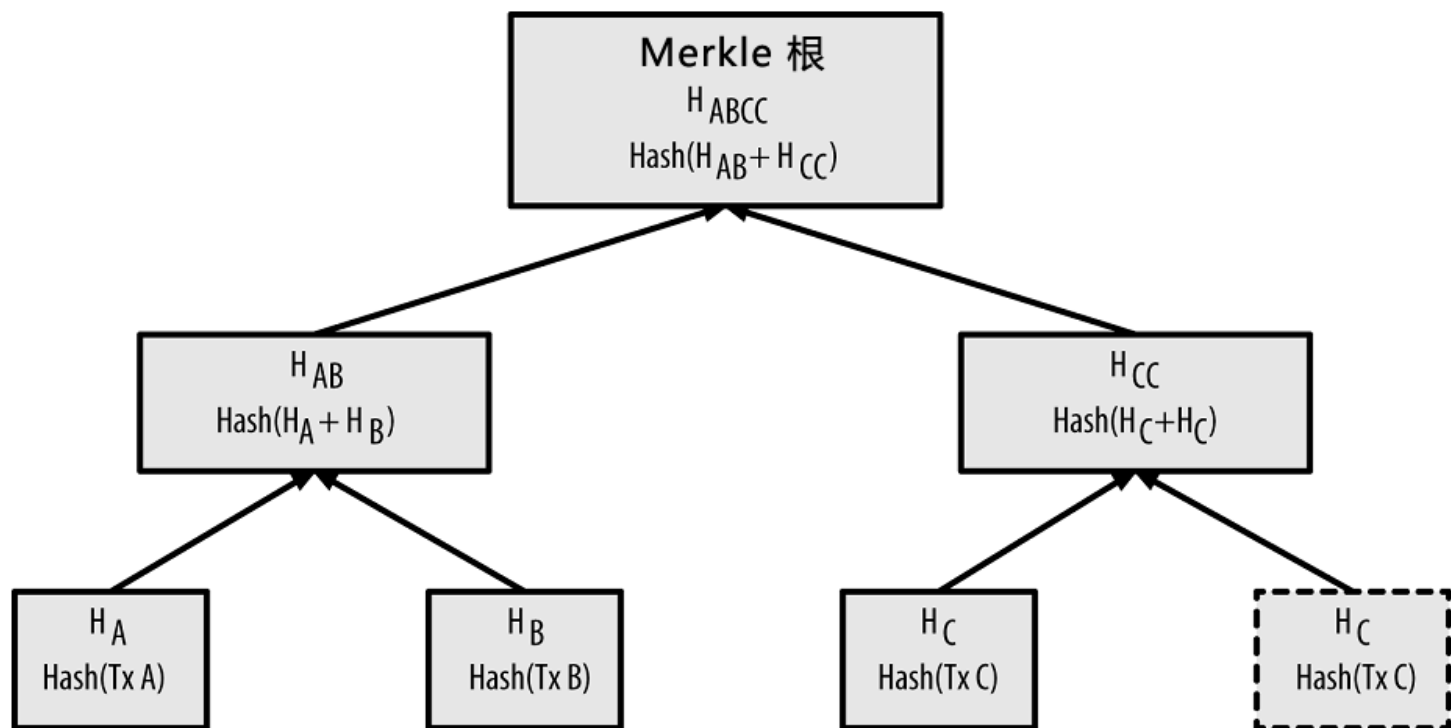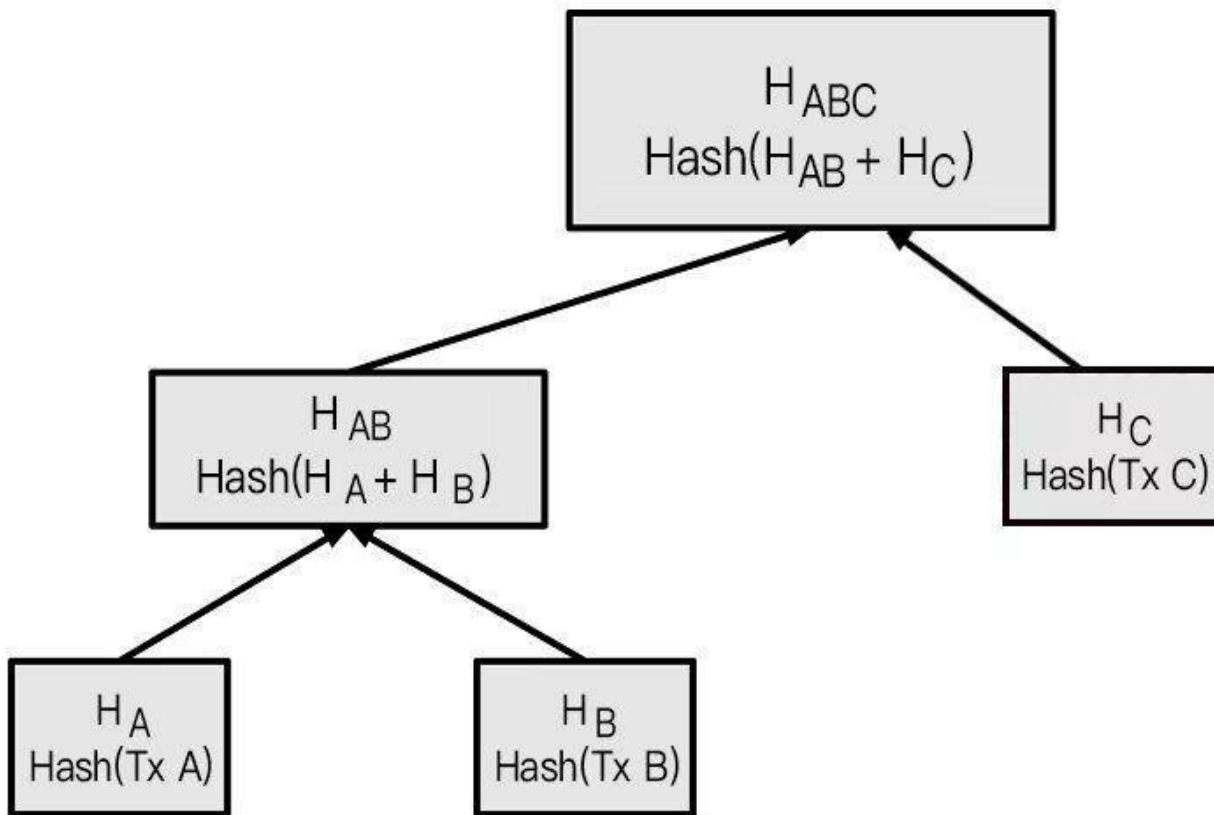
Crypto

secp256k1
keccak
ed25519
blake2b

Batch
Transactions

Rust

Performance:
Rust is blazingly fast and memory-efficient: with
no runtime or garbage collector

Reliability:
Rust's rich type system and ownership model
guarantee memory-safety and thread-safety

program correctness was
checked by tests
VS
logically proven

The role of tests

Db / Compact block ……

https://github.com/cryptape/cita

https://www.oschina.net/p/cita