Modelling Read-Cache Solutions

for Blockchains

•••

December 15th 2017

Jonas Vanden Branden

Overview

- Introduction
- The Problem
- A Solution
- Why Modelling?
- Conclusion

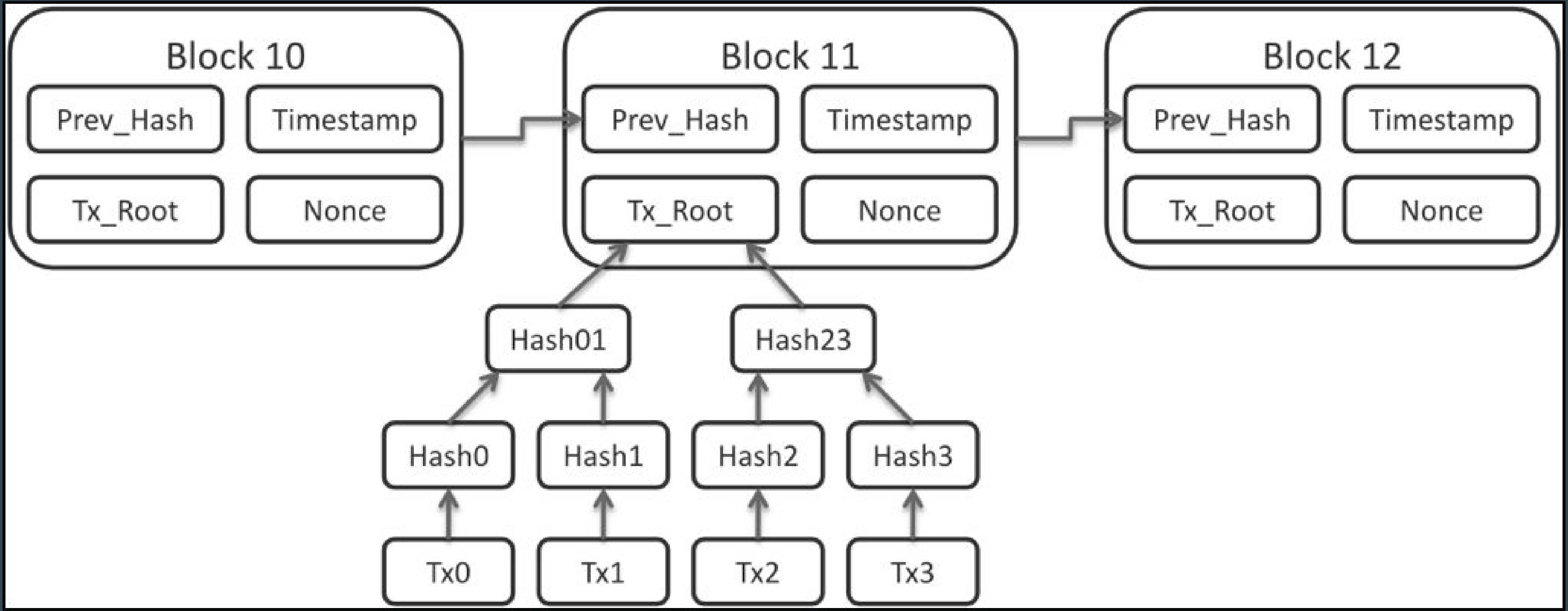
Introduction

Introduction

Blockchains:

- A solution for ***distributed systems*** to achieve ***consensus***
- Introducing Delegated Byzantine Fault Tolerance
- A shared ledger to keep track of transactions
- Multiple stakeholders working with common data (e.g. chain of logistics)
- Cryptographical foundation (hashing, signatures, …)

Blockchain Basics



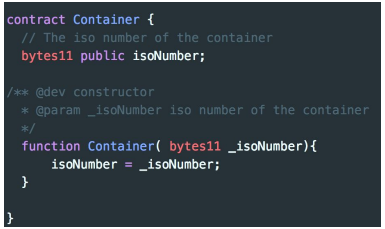
Introducing Smart Contracts...

Smart Contract:

A piece of code which is stored in the blockchain network

Smart Contracts

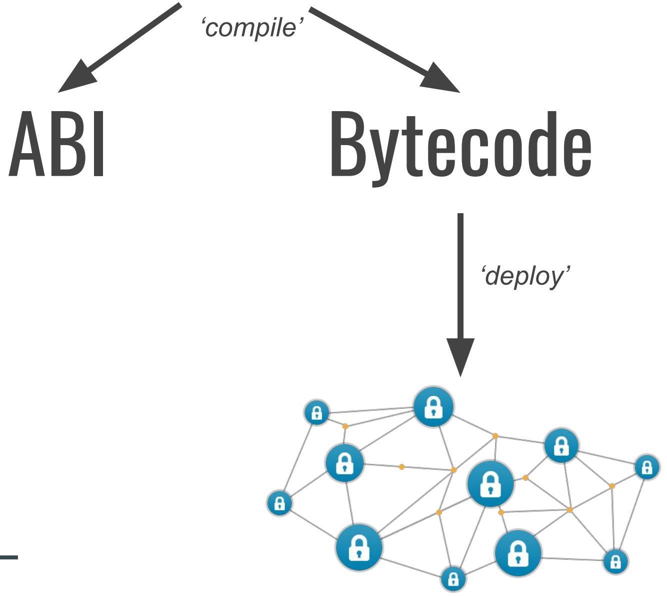
- Deterministic State Machines
- Deterministic?
  - Replication needed
- Which state?
  - State of contract
  - State of blockchain



```
contract Container {
  // The iso number of the container
  bytes11 public isoNumber;

/** @dev constructor
  * @param _isoNumber iso number of the container
  */
  function Container( bytes11 _isoNumber){
      isoNumber = _isoNumber;
  }

}
```



‘compile’

ABI

Bytecode

‘deploy’

Smart Contract Blockchains

Public:

- **Ethereum** (EVM, Solidity)
- **NEO (NeoVM, C#, Java, …)**

Private / Permissioned:

- **HyperLedger Fabric** (Go)
- **Tendermint** *("Byzantine fault-tolerant replicated state machines in any programming language")*

→ ***Each node needs to execute each transaction***

The Problem

Data Limitations

Blockchain as a (intelligent) database?

- No Querying Language
- Low throughput (Read & Write)
- High latency

Maybe not such a good idea....

A Solution

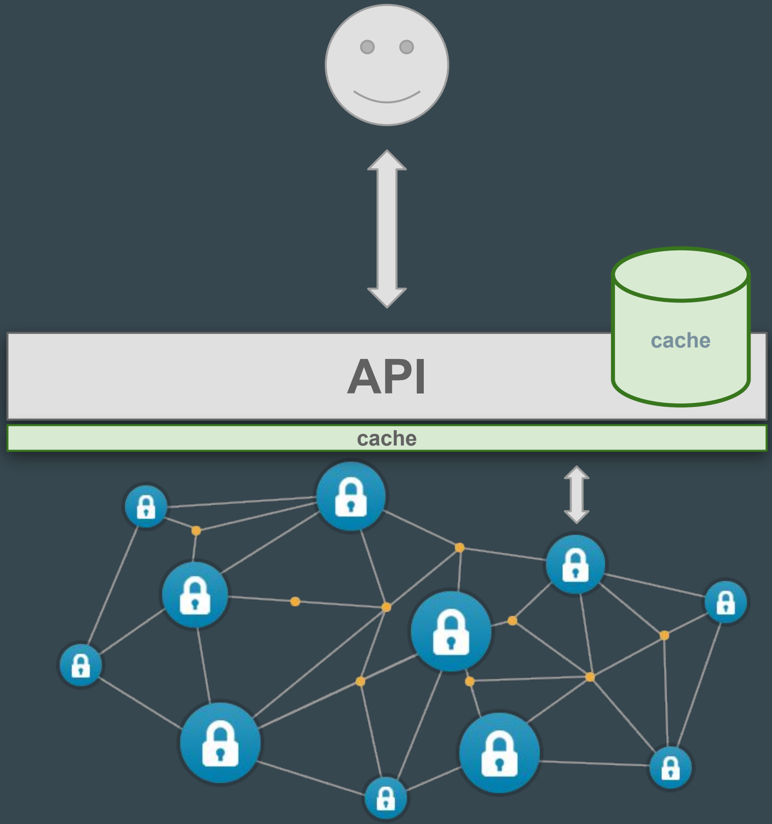
Current Workaround

Keep a local cache

- Not structural
- No guarantees about performance & consistency
- Very Centralized

Improved solution:

Incorporate the cache in the read protocol



Explore Solutions

Different distributed caching solutions provide **different properties**:

- Consistency
- Latency
- Throughput
- …(?)

How to compare them?

→ Construct a model / DSL

Why Modelling?

Why Modelling?

Save Time

- …by reducing implementation cost
- …by simulating time-intensive processes

Save Resources

- …by reducing infrastructure cost
- …by simulating resource-intensive processes

Improve Control

- …by working with simulation models
- …to model complex environments

More Abstraction

- …to control the networking environment
- …to omit irrelevant details

Conclusions

- Co-simulation for network influence?
- **Petri Net model** for consensus algorithm?
- How to model **distributed caching solutions**
  - ‘Top-down’: by analyzing possible algorithms
  - ‘Bottom-up’: by synthesizing possible properties

Any suggestions?

Questions