BLOCKCHAINS IN A NUTSHELL: CONCEPTS AND APPLICATIONS

Wilson S. Melo Jr., DSc.
Researcher at Inmetro - Dimci/Dmtic/Lainf
wsjunior@inmetro.gov.br

In the next 40 minutes...

- What is a blockchain and how it works?
- The main elementary concepts
  - Trustworthiness, consensus and replication
  - Blocks & chains, ledger and smart contracts
  - Public *versus* permissioned blockchains
- Applications
  - Metrology and conformity assessment
  - A case study with Hyperledger Fabric

Blockchains: a disruptive concept

- Trust among people who do not trust each other
  - Trust is a very old problem
  - People pay dearly by trust
  - Security attacks usually try to compromise trust
- But who needs a blockchain?
- Two basic questions:
  - You do not have a trustworthy third party; or
  - Trusting a third part is too much expensive.

The trusted third party dilema

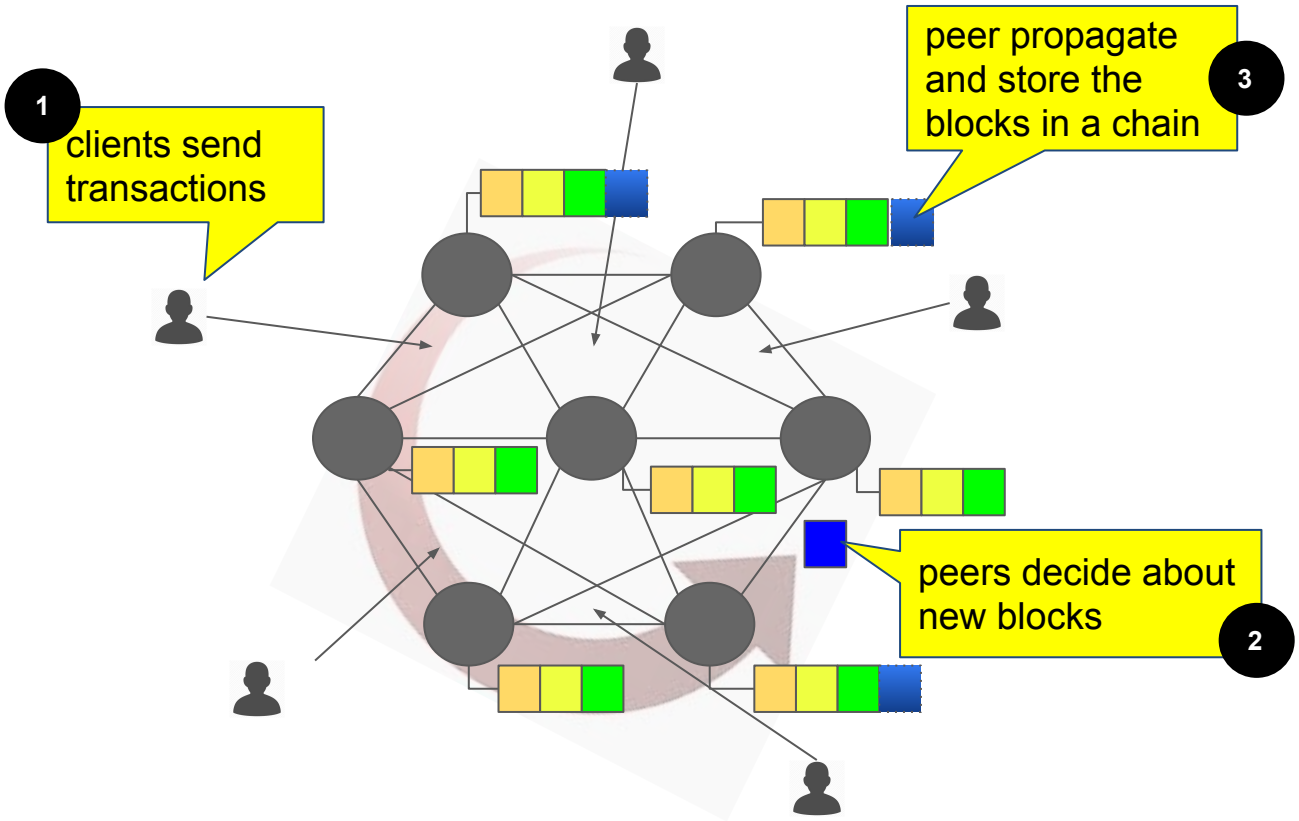
- “Trusted third parties (TTP) are security holes”
  - Nick Szabo, 2001
  - Existing TTP are valuable… and expensive
  - New TTP are costly and risky
- What about they could be removed?
  - A transportation service like Uber, without Uber?
  - A house rent service, without Airbnb?
  - A certified document, without a notary’s office?
  - A reliable measurement, without notified bodies?

Tricks behind the “magic” show...

Lessons from the past...

- The undefeatable power of the **gossip**!
- Blockchain: machines doing gossip...
  - Everybody knows about any transaction
  - Anyone who does not walk the line is exposed
  - Technology prevents overstatement

The basic mechanism



The four elements of a blockchain

Replicated ledger

- History of all transactions
- Append-only immutable past
- Distributed and replicated

Cryptography

- Integrity of the ledger
- Authenticity of transactions
- Privacy of transactions
- Identity of participants

Consensus

- Decentralized protocol
- Shared control tolerating disruption
- Transactions validated

Business logic

- Logic embedded in the ledger
- Executed together with transactions
- From simple "coins" to self-enforcing "smart contracts"

Reference: Cachin et al. (2017)

Replicating the ledger

- Ledger: append only data storage
  - Organized in blocks (granularity)
  - Blocks indicate a temporal order
  - Not only information, but also meta-information
- Every peer has its own ledger copy
  - But all the copies are expected to be the same
  - Replication does that
- But how do we assure authenticity/integrity?

Extensive use of cryptography

- Clients and peers need cryptographic keys
  - Every transaction is signed
  - Peers validate transactions against public keys
  - Ledger can be audited
- Blocks are linked each other by hash (chain)
  - Peers also sign new blocks
- Changes in blocks (order or content) and transactions are exposed
- But, how do we decide “what” is a block?

Consensus is the key

- Consensus: agreement about something
  - A canonical problem in distributed systems
  - “Save consensus, you saved the world”!
- Peers get consensus about blocks
  - Different protocols
  - Different security assumptions for each protocol
  - Permissioned *versus* public blockchains
- And what does motivate the consensus?

We need a business model

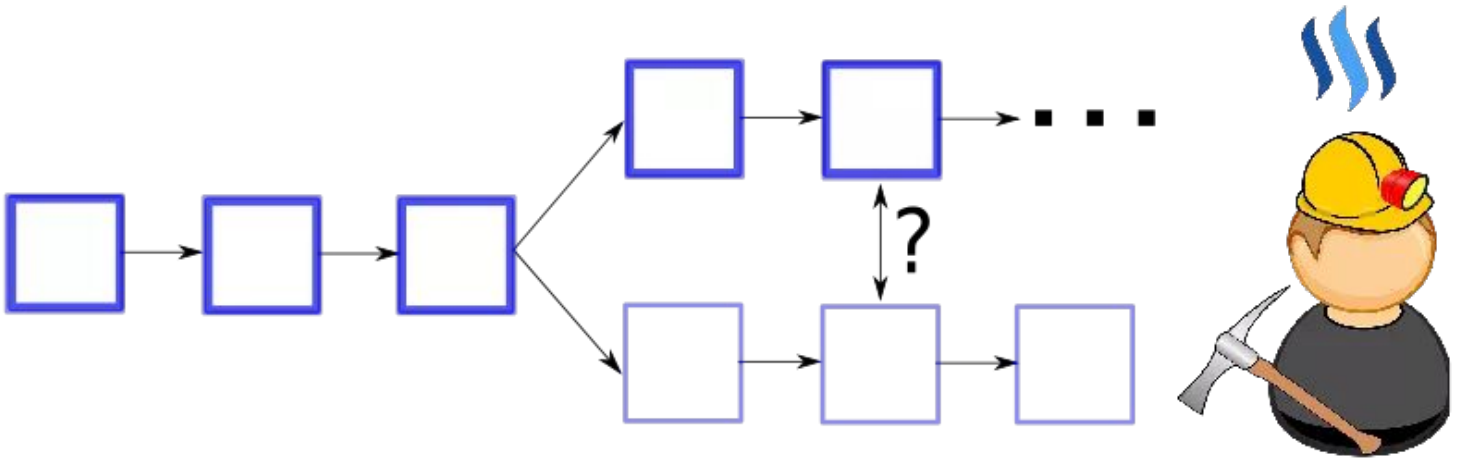
- Blockchain requires a business model
  - Peers need an incentive to work for consensus
  - Bitcoin → consensus is mining new coins
- Business logic can be embedded in the ledger
  - Meta information → self executable software
  - Transactions can invoke smart controls
- Not only an immutable data storage, but also a platform for running business ideas

Learning by examples

- Bitcoin ⟶ transactions are payments
- Public, but anonymous
  - Each user = a private key
- Consensus ⟶ Proof-of-work protocol
  - Peers compete each other for proposing a block
  - Cryptographic puzzle: computational resources
  - The winner earns bitcoins (mining)
- More than one block ⟶ creates a “fork”
- Low throughput, wait to confirm transactions

Solving a fork

- Usually, the longest branch wins
- Bitcoin rules: wait before confirm transaction.
  - 6 blocks, ~12 min per block ⟶ 1 hour!



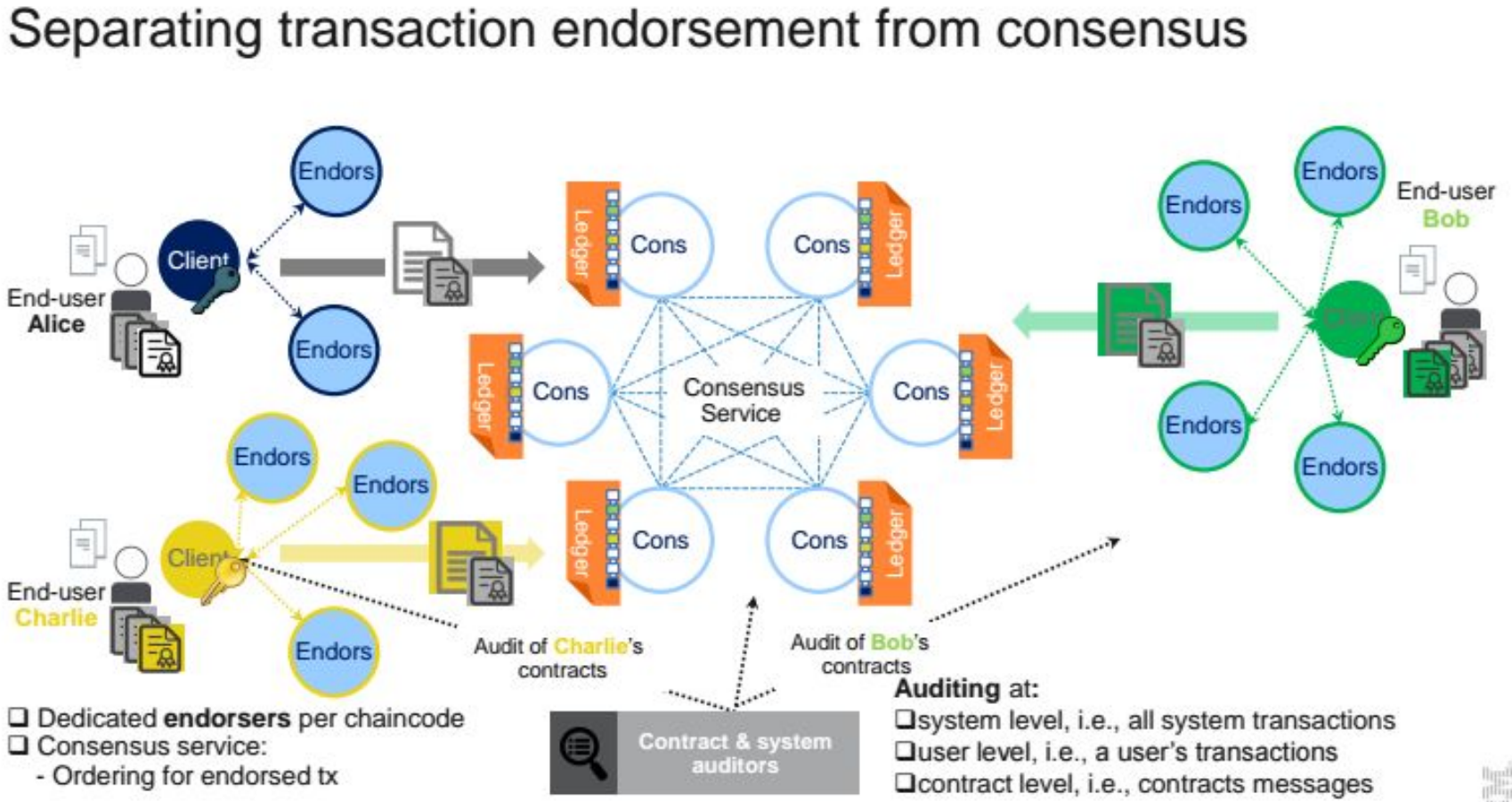
Public *versus* Permissioned

- Public blockchains does not require identities
  - Any peer can participate
  - Decentralized consensus
  - Performance is the main challenge
- Permissioned blockchains identifies the peers
  - … and enables Byzantine consensus
  - Tolerates faults to F in N = 3F + 1 nodes
  - No forks, more suitable for organizations solutions
  - It also improves throughput
  - One of the main areas in research nowadays

An example: Hyperledger Fabric

- Permissioned blockchain framework
  - Hosted by The Linux Foundation
  - Support multiple ledgers (channels)
  - Smart contracts in Go language (chaincode)
- Enables different consensus as plugins
  - Kafka+Zookeeper consensus
  - Byzantine consensus with BFT-SMaRT
  - Consensus is called *orderer service*
- Other concepts
  - Endorser peers and security policies

An example: Hyperledger Fabric

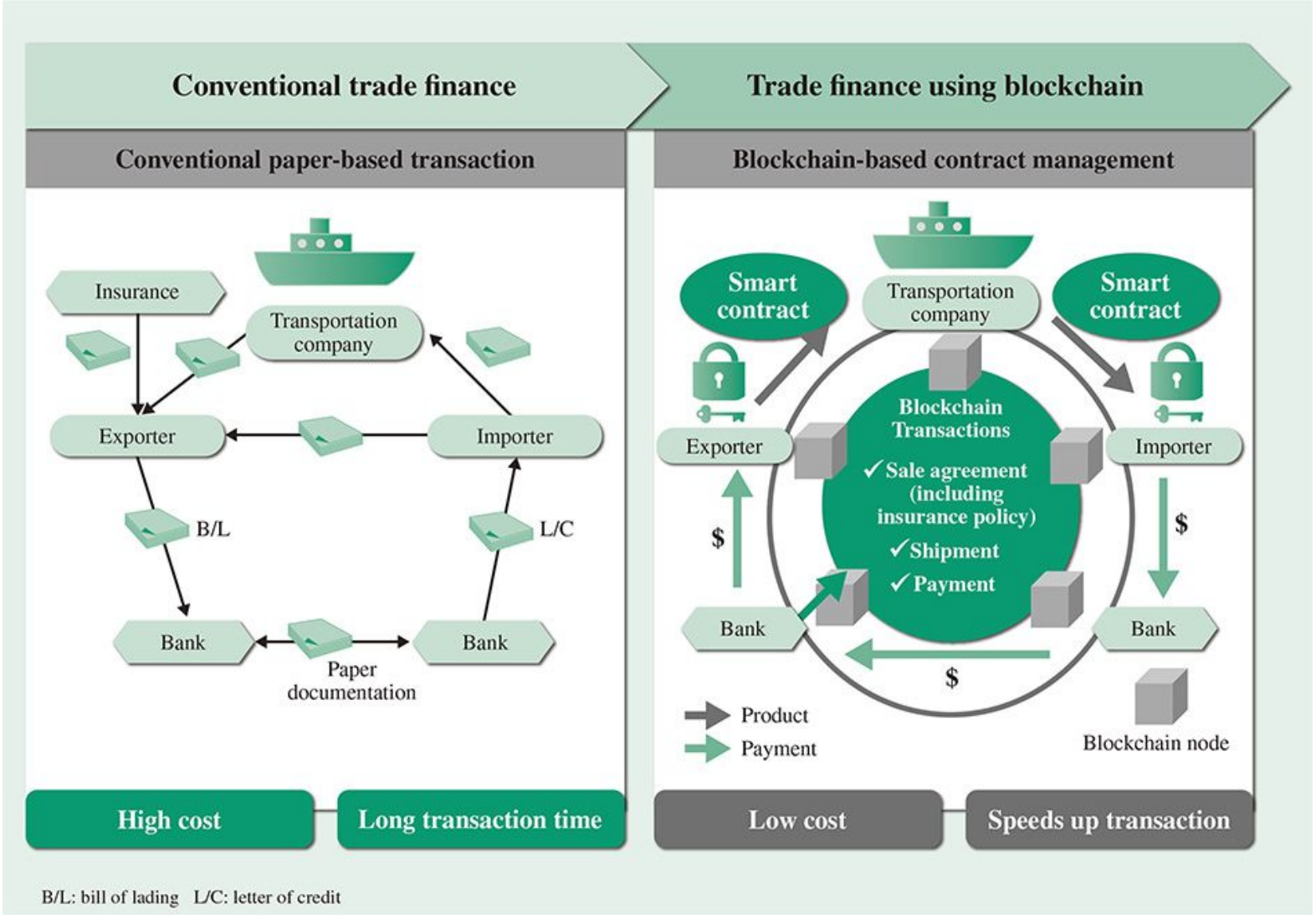


Ideas, examples and cases

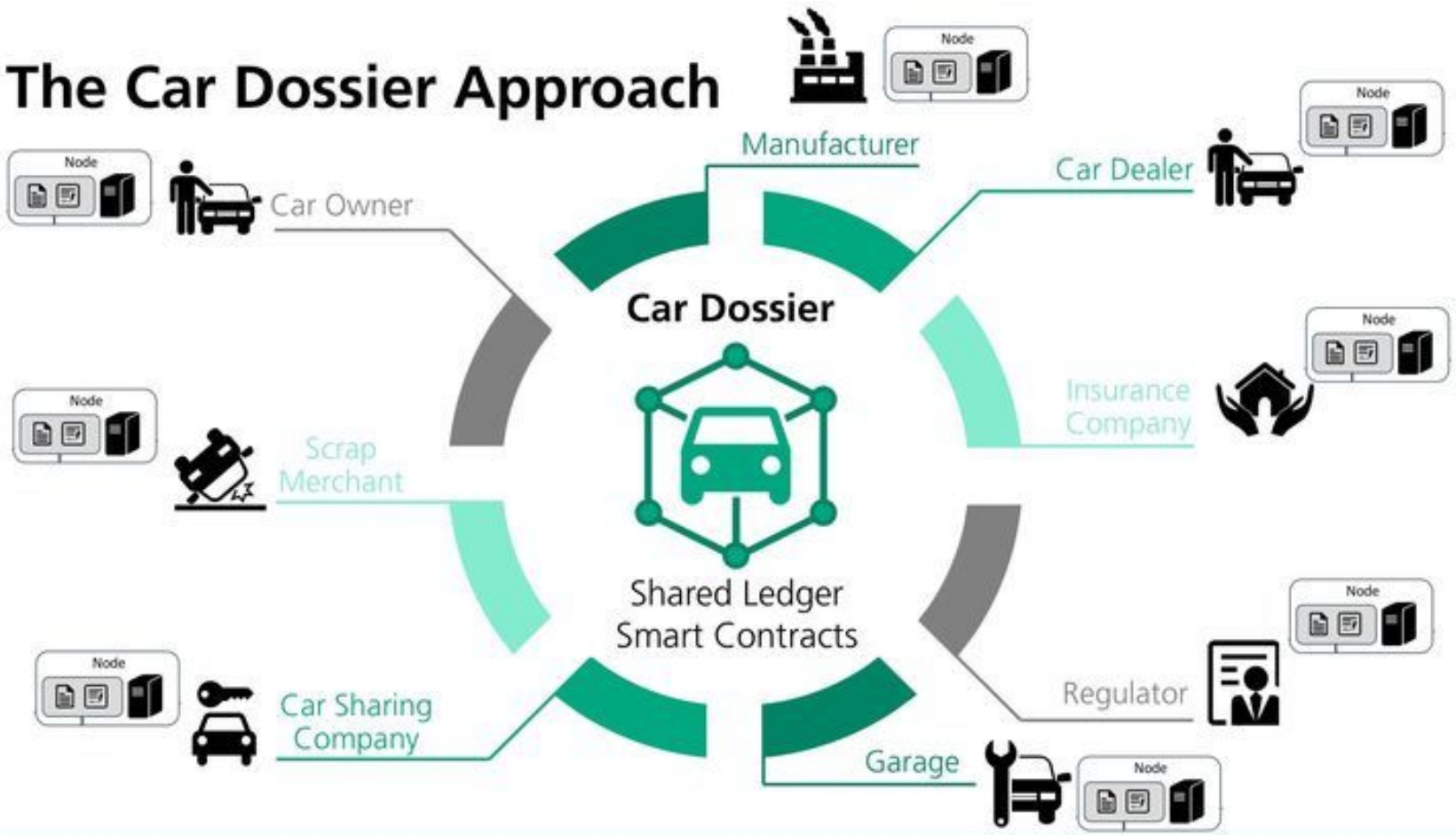
A diversity of approaches

- The pioneers
  - Cryptocurrency
  - Bitcoin, Ethereum, ...
- The intuitive ones
  - Distributed append-only database
  - Trading, docs storage, automatized workflow
- The specific ones
  - It depends on the application area
  - Metrology and conformity assessment

Complex trading ecosystems



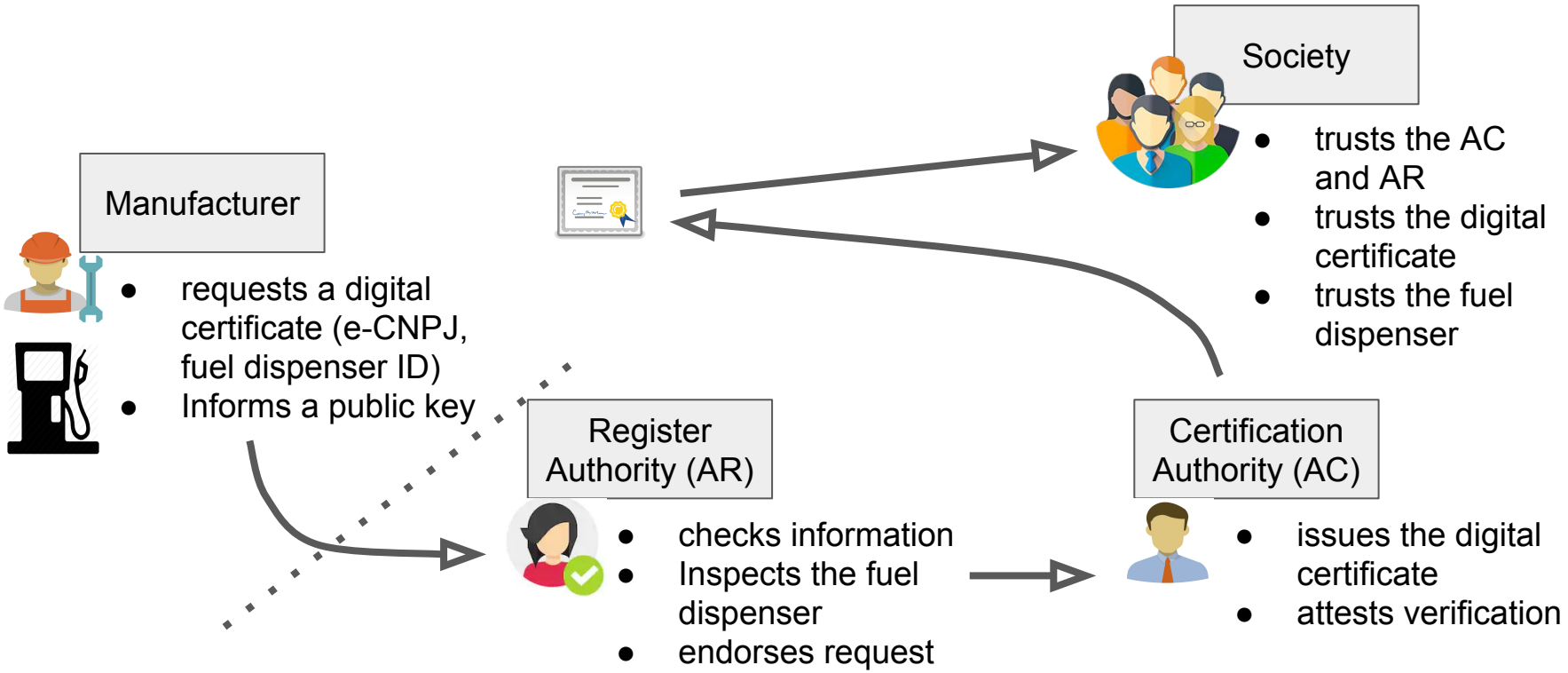
Insurance applications



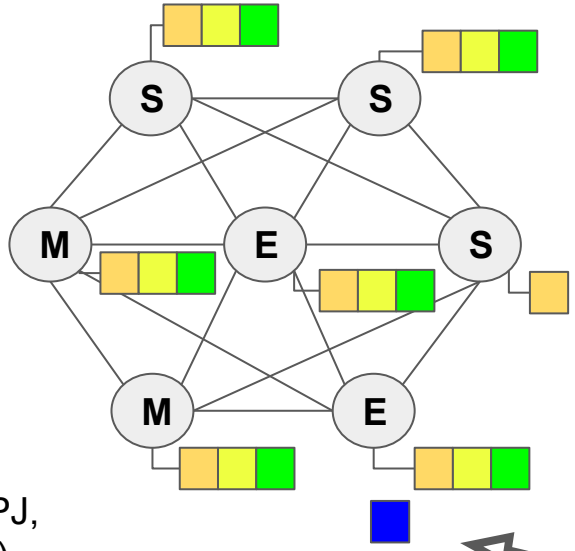
Measuring instruments PKI

- Sometimes, measuring instruments need to sign the measurements
  - Authenticity, integrity and non repudiation
  - Public key cryptography
- When a PKI becomes necessary
  - The instrument embeds a digital certificate
  - Someone must to manage the mess
    - Registry Authority (RA)
    - Certificate Authority (CA)
  - Complexity and costs

Measuring instruments PKI

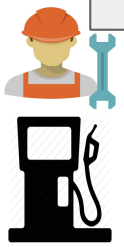


Blockchain as a PKI



Society

- checks the blockchain
- can also take part on it
- there is not collusion...
- … trusts the fuel dispenser



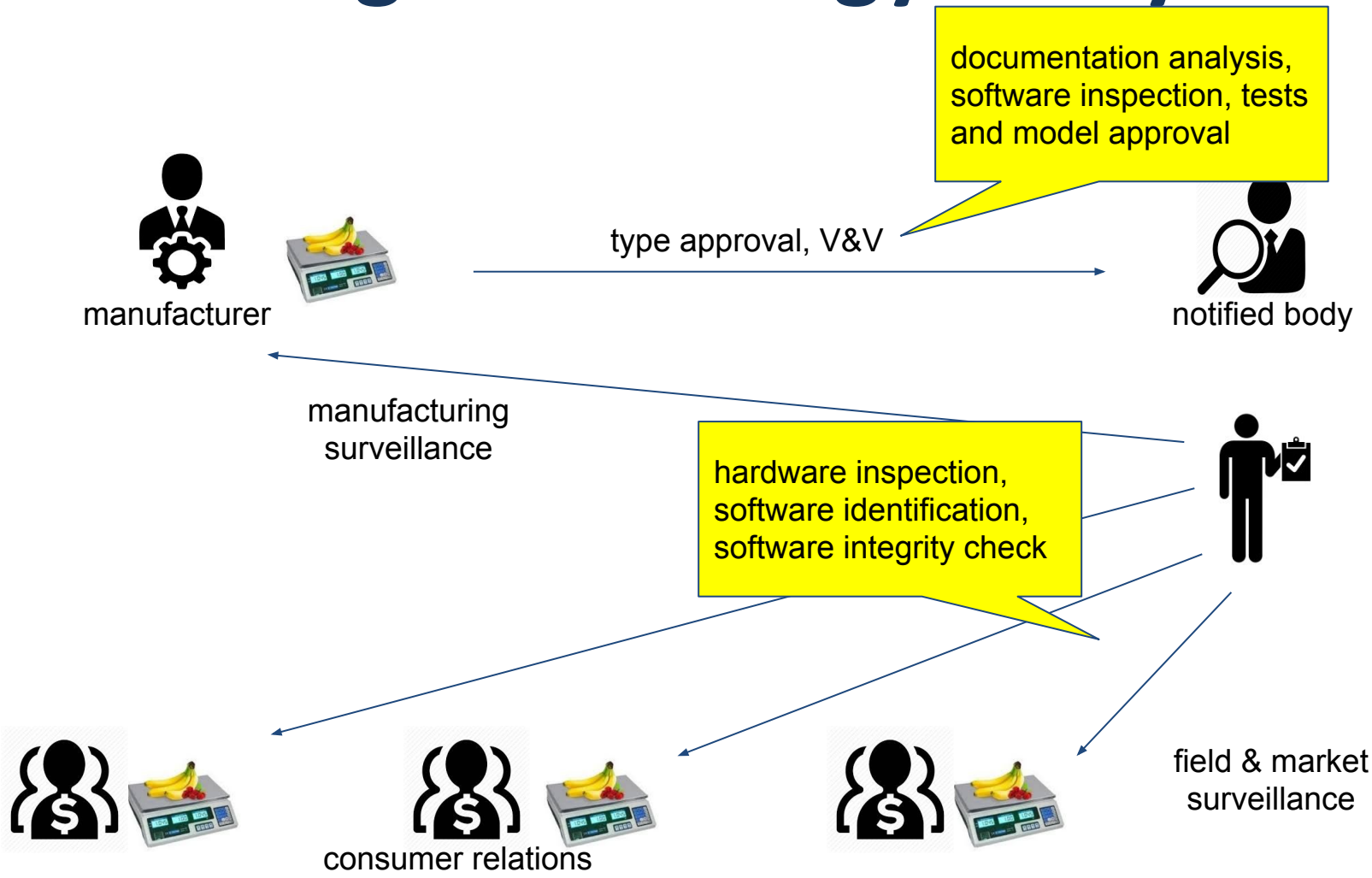
Manufacturer

- requests a digital certificate (e-CNPJ, fuel dispenser ID)
- signs its requirement
- asks for endorser

Permissioned Endorsers

- checks information
- inspects the fuel dispenser
- endorses transactions

Case: Legal metrology ecosystem

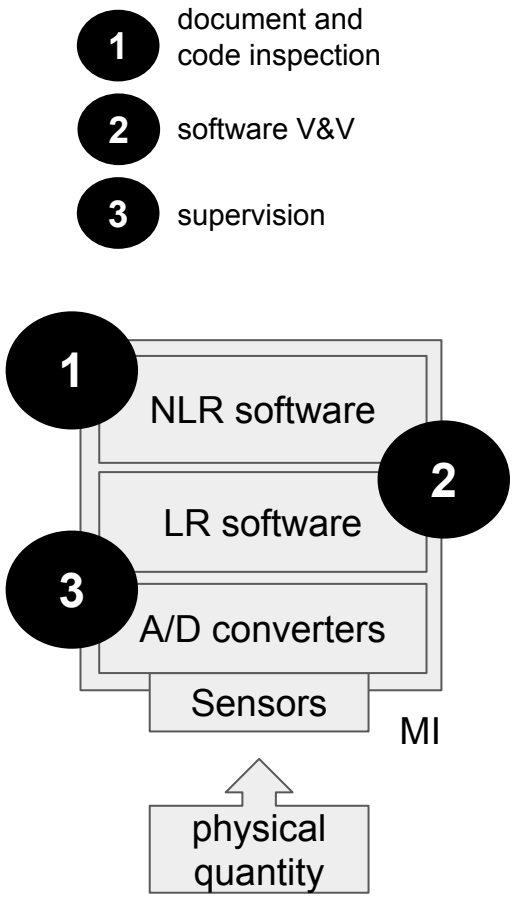


What is the problem?

- MI have become complex
- Regulation and control are expensive
  - **type approval**: complete MI hardware/software
  - **validation & verification**: complex use cases
  - **supervision**: too many instruments, capillarity, specialized knowledge
- Not affordable resources
- Idea: a **decentralized solution**

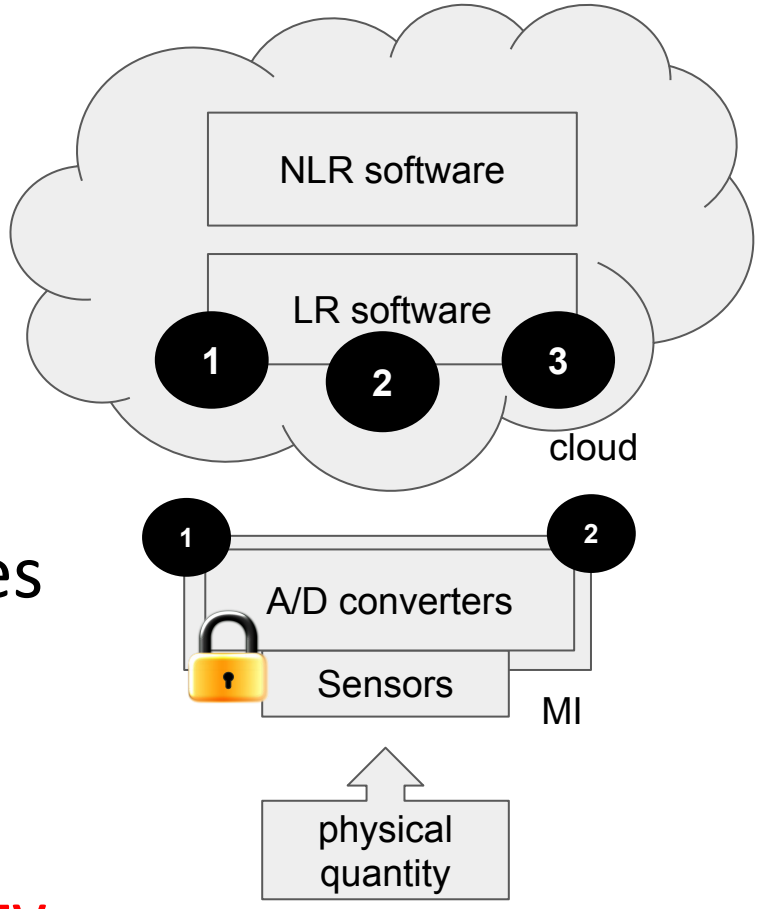
Understanding the models

- The traditional MI
  - Sensors and A/D converters
  - **Legally relevant (LR)** and **Non-Legally relevant (NLR)** software modules
- Usually strongly coupled
- Too many people can access
- Require intensive supervision
- Belong to an **interested part**



Understanding the models

- The cloud model
  - Oppermann et al. (2018)
  - LR and NLR in the cloud
  - Distributed measuring
- Reduces coupling and tests
- Reduces access to LR modules
- Optimizes supervision
  - Far less checkpoints
  - Tamper-proof hardware
- Belongs to an interested party

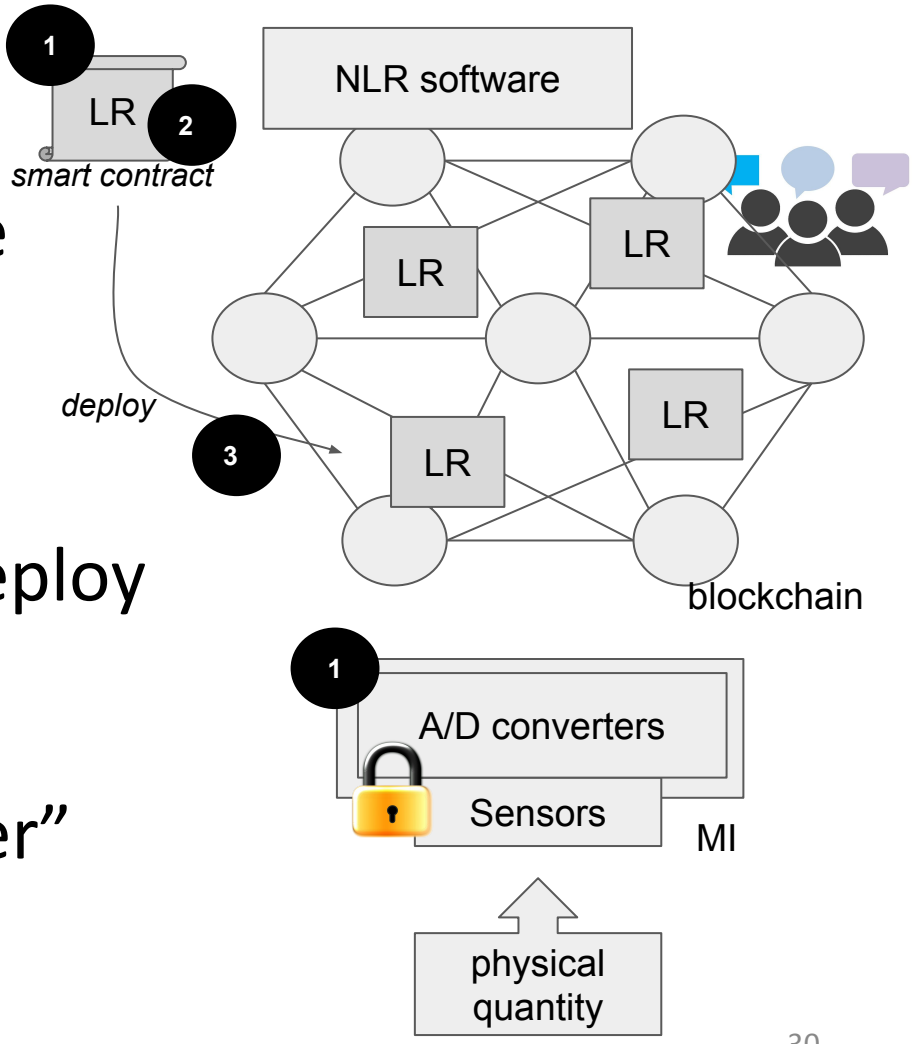


The blockchain model

- Distributed measuring using blockchains
  - Network peers are provided for different parties
  - Measuring is offered as a service
  - Measurements are stored in the ledger
- Decentralized surveillance
  - Independent parties can implement supervision
  - Commercial relations become transparent
  - Frauds can be detected consulting ledger

The blockchain model

- How does it works?
  - Measuring as a service
  - LR ⟶ *smart contract*
  - No coupling
  - Easy to test
  - Supervision only on deploy
- Software integrity
  - Assured by blockchain
- There is not an MI “owner”

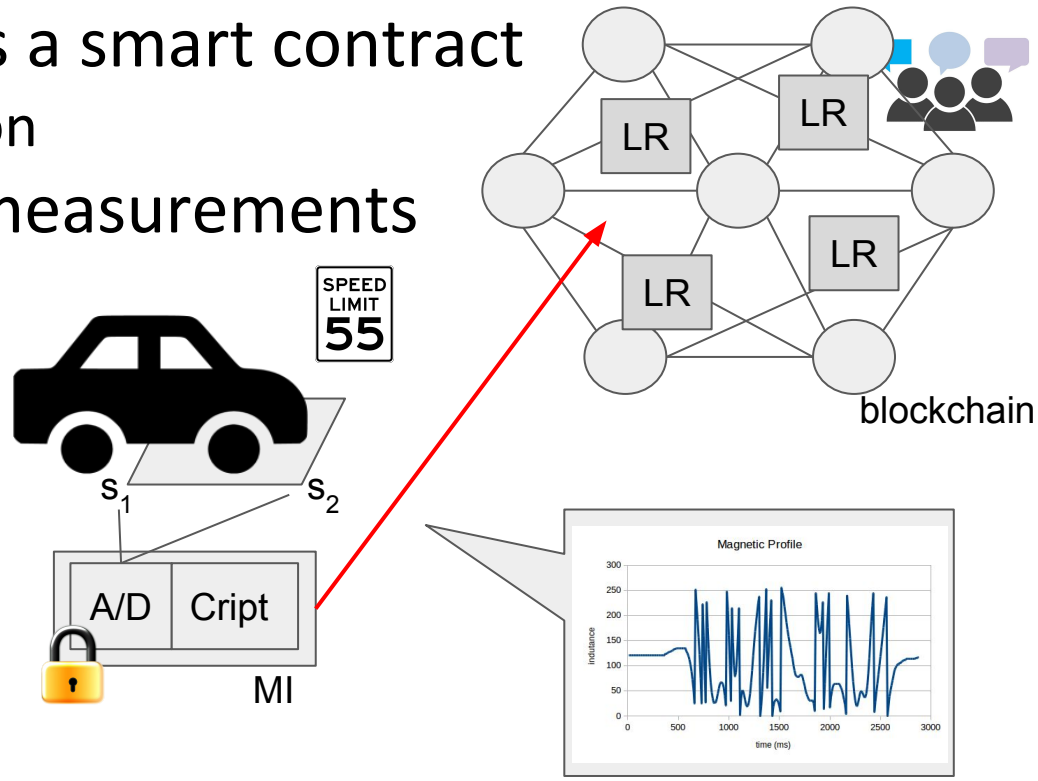


Security Analysis

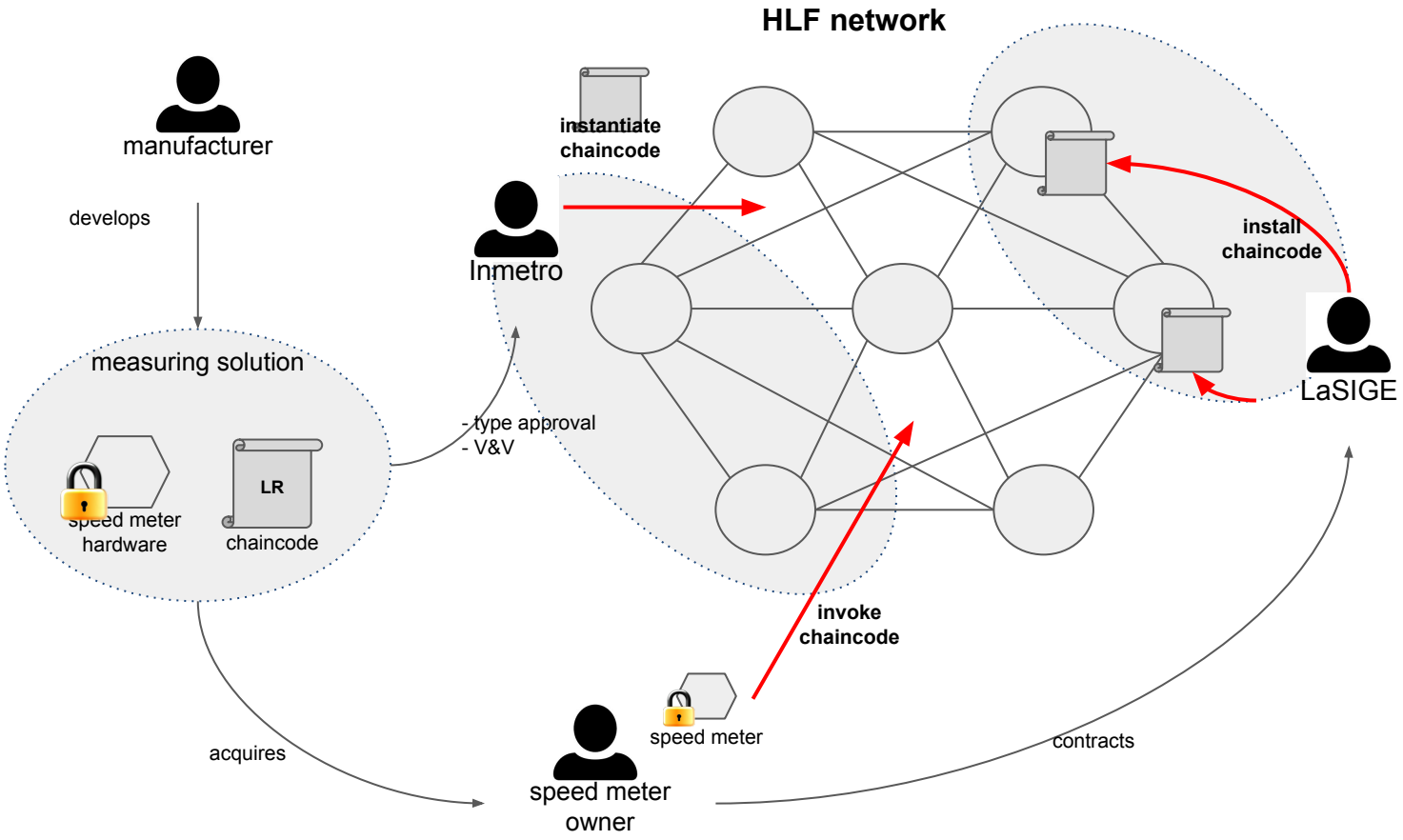
Requirement/Activity	Cloud	Blockchain
R1 - MI have physical sealing (physical components)	Required	Required
R2 - LR software identification and integrity checking	Required	Unnecessary
R3 - Secure LR software loading	Required	Unnecessary
A1 - MI document and code inspection	LR software	Smart contract
A2 - MI validation and verification (V&V)	Necessary	Necessary
A3 - MI metrological supervision (software inspection)	Partial	Unnecessary

Speed meter DMS in a blockchain

- A simple hardware send transactions
- LR software runs as a smart contract
  - Software protection
- Blockchain stores measurements
  - Authenticity
  - Integrity
  - Auditability

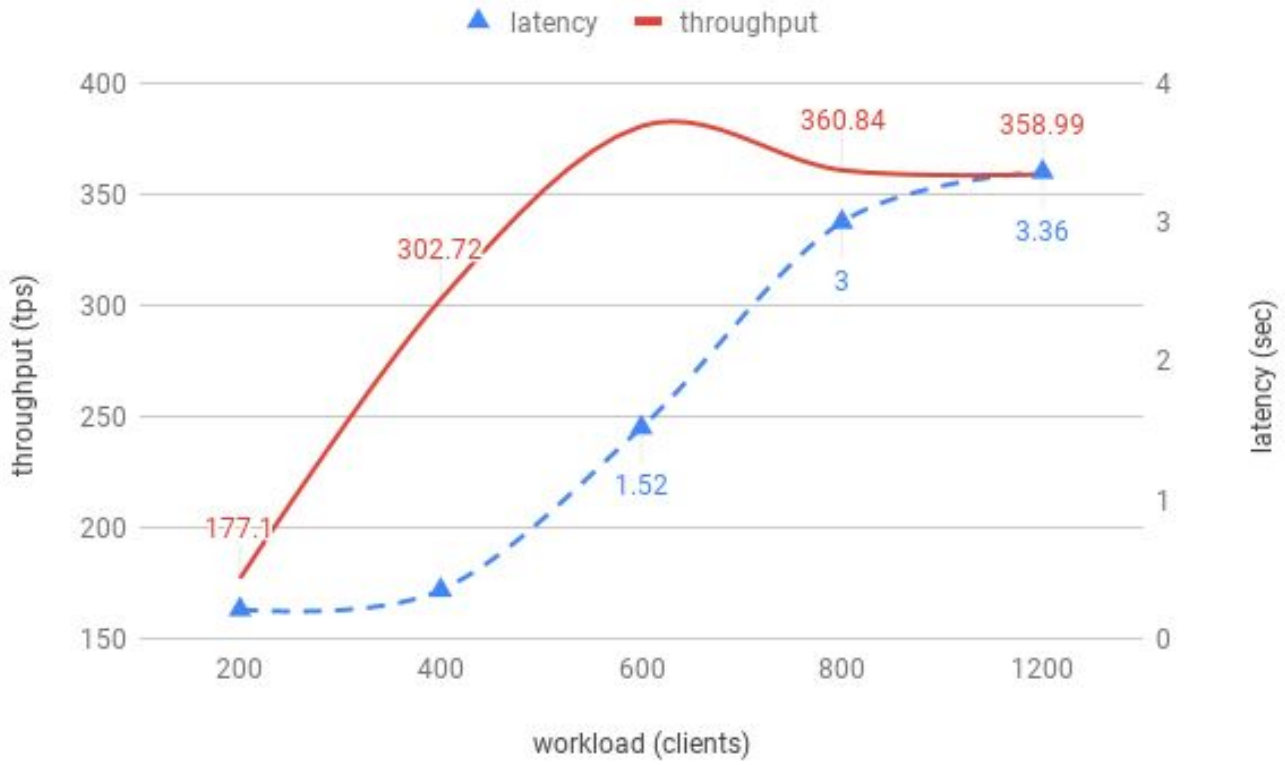


Our blockchain implementation



Case study results

- Experiment with data from Sao Paulo (rush hour)
  - vehicular fleet: more than 8 million
  - ~1,000 speed meters, 2,772 vehicles/h per meter



Conclusion

- A promising technology
  - It simplifies questions related to trust
  - Extensive use of cryptography and application of well-know concepts in distributed systems
- A world of new possibilities
- Including metrology…
  - Blockchains seems to be a suitable technology for distributed measuring systems
  - Other applications as conformity, auditing, PKI for manufacturers, LR software distribution, etc.

So far so good...

Questions, suggestions and discussions are always welcome!