BLOCKCHAIN-BASED SMART INDUSTRY INTEGRATION

DEMYSTIFYING BLOCKCHAIN

Ivan Gudymenko, ivan.gudymenko@t-systems.com
Marian Neubert, marian.neubert@t-systems.com
Introduction

The notion of a (distributed) ledger

Blockchain as a distributed ledger and a state machine

Aligning on what is valid: a variety of consensus algorithms

Putting it all into a real-world context: a smart industry example

Proof-of-concept demonstration

Questions and answers
IT-SECURITY@T-SYSTEMS MMS
WHO WE ARE

SECURITY

DATENSCHUTZ

GOVERNANCE, RISK AND COMPLIANCE

MANAGED SECURITY SERVICES

TRAINING AND AWARENESS

BLOCKCHAIN
DEEPLY INTERCONNECTED WORLD
FROM PLAIN CLIENT/SERVER TO “ANY-TO-ANY”

Global communication: a paradigm shift

18.12.2017
PARADIGM SHIFT: TRUST DECENTRALIZATION
DECENTRALIZING THE “ROOT OF TRUST”

Well-defined trust models and control

“50 shades of Trust”
THE NEW CHALLENGES AND OPPORTUNITIES
HOW TO DEAL WITH “ANY-TO-ANY” AND HETEROGENEITY

• We would like to allow as many parties to communicate as possible
  • ... as freely as possible
  • ... as secure as possible

• But how to avoid the anarchy, sabotage and mess?
  • While preserving privacy, security and correctness?

• A huge number of new-old challenges arise (yet again)
THE NOTION OF A LEDGER
KEEPING TRACK OF THE TRANSACTIONS

• Ledger “is the principal book or computer file for recording and totaling economic transactions” [1]

• Nowadays, ledgers have been largely digitalized

• Ledgers can been found everywhere where the (financial) transactions have to be kept track of

• However, every institution maintains its own ledger
  • How to share information between different ledgers
  • And align on what is valid and what is not?

• Correct, let’s introduce a SINGLE SHARED LEDGER!

A SHARED LEDGER
MAKING IT USABLE AND SECURE

• A distributed ledger shared between several instances/companies/institutions should be
  • Trusted by all parties
  • Immutable
  • Usable and maintainable by all parties
  • ... and so on, and so forth

• It has to be clear how the parties ALIGN on the valid state of the shared distributed ledger
  • Consensus should be reached
  • The procedure must be clear, transparent and applicable for all participants

• And many more ....
IMPLEMENTING IT WITH A BLOCKCHAIN

BLOCKCHAIN AS A DISTRIBUTED LEDGER

- The data structure essentially represents a linked hash chain
  - Therefore immutable (cannot be changed afterwards) by design

- With a set of rules and permissions to write the data into the chain and agreeing on transactions validity

- With a consensus algorithm regulating which transactions are valid

- Consensus algorithm defines which nodes are allowed to write/validate and how

---

Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System
BLOCKCHAIN AS A STATE MACHINE
MAINTAINING THE DISTRIBUTED STATE

• The state of blockchain transactions can be seen as an append-only log

• The state gets updated by appending a block of transactions to the log
  • $\text{State}_{\text{new}} = \text{UPDATE}(\text{State}_{\text{old}})$

• WHO is allowed to update the state and under which conditions is defined by a CONSENSUS ALGORITHM

Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System
CONSENSUS ALGORITHMS
A HEART OF EVERY BLOCKCHAIN SYSTEM

• The security and trust model of a blockchain system is largely defined by the underlying consensus algorithm

• The topic of consensus algorithms is in fact fairly well researched in the area of in distributed computing
  • Yes, some people are essentially re-inventing the wheel here, but it is out of scope of this talk 😊

• Consensus algorithms can be very roughly divided into permissionless and permissioned

• Permissionless can be also referred to as decentralized
  • e.g. as in Bitcoin, Ethereum

• Permissioned are also known as consortium
  • e.g. Byzantine Fault Tolerance (BFT) consensus

A blockchain system is as secure and robust, as its consensus algorithm!
THE NOTION OF A “SMART CONTRACT”
BUILDING IN BUSINESS LOGIC

• “Smart Contracts are self-executing contractual states, stored on the blockchain, which nobody controls and therefore everyone can trust” [1]

• Essentially, smart contract is a distributed application that represents business logic

• Smart contract properties:
  • Self-Imposable
  • Trustless
  • Fast
  • Cheap [2]

[1] https://www.smartcontract.com
## BLOCKCHAIN-BASED SYSTEMS LAYERING

<table>
<thead>
<tr>
<th>Type</th>
<th>DLT (Distributed Ledger Technology)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Area</td>
<td>CryptoCurrencies &amp; Transactions</td>
</tr>
<tr>
<td></td>
<td>Smart Contracts</td>
</tr>
<tr>
<td></td>
<td>others</td>
</tr>
<tr>
<td>Implementation</td>
<td>Bitcoin, Ethereum, HyperLedger, BigChainDB, etc.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Public vs. Private</td>
</tr>
<tr>
<td></td>
<td>Permissioned vs. Permissionless</td>
</tr>
<tr>
<td></td>
<td>Consensus-Mechanismus (Proof of X Konzepte)</td>
</tr>
<tr>
<td>Tools</td>
<td>Wallets</td>
</tr>
<tr>
<td></td>
<td>Clients</td>
</tr>
<tr>
<td></td>
<td>Mining</td>
</tr>
<tr>
<td></td>
<td>Dapps</td>
</tr>
<tr>
<td></td>
<td>Explorers</td>
</tr>
<tr>
<td></td>
<td>etc.</td>
</tr>
</tbody>
</table>
SMART FACTORY
DIGITALE VERTRÄGE FÜR INTELLIGENTE FABRIKEN
SMART FACTORY
KURZVORSTELLUNG
1. Kapazitätsanfrage über GUI mit Produktauswahl (Form, Logo etc.)
2. Kapazität bei Auftragnehmer verfügbar; Annahme der Anfrage; Vertragsabschluss
3. Subcontractor einladen (Bohren, Einlegen, Handarbeit)
4. Werkstück wird registriert
5. Statusmeldung für Bearbeitungsschritt 1
6. Statusmeldung für Bearbeitungsschritt 2
7. Statusmeldung für Bearbeitungsschritt 3
8. Fertigung Werkstück beendet
OPC UA
KURZÜBERBLICK
KOMMUNIKATION MIT OPC UA
SCHNITTSTELLEN ÜBERGREIFEND

LÖSUNGEN

- STANDARDISIERUNG DER TECHNOLOGIE
- OFFENE FORMATE UND SPEZIFIKATIONEN
- UNIVERSELLE KOMMUNIKATIONS-MÖGLICHKEITEN

---

ERP

MES

SCADA

SPS

OPC UA

UNTERNEHMENSEBENE

BETRIEBSEBENE
CMMS, Archiv und Analysen

PROZESSEBENE
HMI

STEUERUNGSEBENE
PLC

---

T-Systems

18.12.2017
ÜBERBLICK ÜBER OPC UA
ENTWICKLUNG

HAUPTNACHTEILE VON OPC WERDEN BEHOBEN:
• Starke Authentifizierung und Validierung eingebaut
• Kommunikation über Firewall/Domäengrenzen möglich
• Skalierbar und Redundant, Plattformunabhängig, Fehlertolerant
• Nicht nur der Transport sondern auch die Semantik wird definiert

• Spezifikation und Teile der OPC UA-Stacks seit 2015 Open-Source
• Implementierungen für .NET, ANSI C/C++ und Java verfügbar
• Verschiedene Hersteller bieten fertige SDK‘s
OPC UA IMPLEMENTIERUNG
ROBUST UND SKALIERBAR

Performance
- Hohe Geschwindigkeit in der Kommunikation
- Kleiner Memory-Footprint
- Geringe Last auf dem Zielsystem

Redundanz
- Konfigurierbare Timeouts und Retransmits
- Hochverfügbarkeit auf Applikations- und Netzwerkbasis

Skalierbar
- Vom Mikrocontroller bis zum Enterprise-Server
- Gateways zum Aggregieren implementierbar