Software Architecture and Engineering for Blockchain-based Applications

Ingo Weber | Principal Research Scientist & Team Leader
ingo.weber@data61.csiro.au

Conj. Assoc. Professor, UNSW Australia | Adj. Assoc. Professor, Swinburne University
www.csiro.au
What does a Blockchain look like?
http://ethviewer.live
What blockchain do you need (if any)?

1. Can a traditional database technology meet your needs?
   - YES: Does more than one participant need to be able to update the data?
     - YES: Does the data need to be kept private?
       - YES: Do you need to control who can make changes to the blockchain software?
       - NO: Would all the participants trust a third party?
         - YES: You might need a permissioned blockchain (medium transaction speed).
         - NO: You don’t need a blockchain (fast transaction speed).
     - NO: Is this database likely to be attacked or censored? Do you need redundant copies in multiple distributed computers?
       - YES: You might need a public blockchain (slow transaction speed).
       - NO: You might need a permissioned blockchain (medium transaction speed).
   - NO: Do you and all those updaters trust one another?
     - YES: You might need a public blockchain (slow transaction speed).
     - NO: You don’t need a blockchain (fast transaction speed).

Source: IEEE Spectrum Oct 2017
Blockchain Research at Data61
Blockchain Research at Data61

• Designing Systems with Blockchain
  - Design Trade-offs
  - Model-driven development
  - Governance and risk management

• Trustworthy Blockchain Systems
  - Formal
  - Empirical

• Defining and Using Smart Contracts
  - As Legal Contracts
  - Business Process
Designing Systems with Blockchain

• Design Process
  • The blockchain as a software connector, X. Xu, C. Pautasso, L. Zhu et al., WICSA2016.

• Quality Analysis
  • Quantifying the cost of distrust: Comparing blockchain and cloud services for business process execution. P. Rimba, A. B. Tran, I. Weber et al., accepted by Scalable Computing and Communications (SCAC) journal, 2017

• Model-Driven
  • From business process models, see next slide

• Integration with other systems

• Governance and risk management
  • Risks and Opportunities for Systems Using Blockchain and Smart Contracts, Treasury report
Defining and Using Smart Contracts

• Business Process
  • Untrusted business process monitoring and execution using blockchain,
  • Optimized Execution of Business Processes on Blockchain,
  • Caterpillar: A blockchain-based business process management system,
  • Runtime verification for business processes utilizing the Bitcoin blockchain,
    C. Prybila, S. Schulte, C. Hochreiner, and I. Weber,
    Future Generation Computer Systems (FGCS), accepted August 2017

• Legal vs. Smart Contracts
  • Evaluation of Logic-Based Smart Contracts for Blockchain Systems,
    F. Idelberg, G. Governatori, R. Riveret et al., RuleML2016
Trustworthy Blockchain Systems

• Formal
  • The Blockchain Anomaly, C. Natoli, V. Gramoli, NCA2016
  • On the Danger of Private Blockchains, V. Gramoli, DCCL2016

• Empirical
  • On availability for blockchain-based systems, I. Weber, V. Gramoli et al., SRDS 2017
Projects with Australian Treasury

- Funded by Australian National Innovation Science Agenda
- Two reports, launched 6 June 2017
- See www.data61.csiro.au/blockchain

- DLT Foresight
  - What might plausibly happen, across society & economy?

- Technical Risks & Opportunities
  - How do needs in various use cases fit blockchain’s capabilities?
Red Belly Blockchain (with USyd)

• New technology particularly for private / consortium blockchain
• In lab experiment achieved 660,000 tps with 300 nodes in one DC, and > 50,000 tps with globally distributed nodes
Architecting Applications on Blockchain

On availability for blockchain-based systems, I. Weber, V. Gramoli et al., SRDS 2017
Overview

• Many **interesting applications** for Blockchain
  • Basically of interest in most lack-of-trust settings where a distributed application can coordinate multiple parties
  • Examples:
    – Supply chains
    – Handling of titles, e.g., land, water, vehicles
    – Identity
  • Many startups and initiatives from enterprises / governments

• ... but also many **challenges**
  • When to use blockchain
  • Trade-offs in architecture
    – Downsides: cost, latency, confidentiality
    – What to handle on-chain, what off-chain?
Our work – AAP team

• Architecting applications on Blockchain:
  • Taxonomy and design process
  • “Cost of Distrust”: how much more expensive is blockchain?
  • Availability analysis from viewpoint of DApss
  • Latency: simulation under changes
• Model-driven development of smart contracts
  • Business process execution
  • Model-based generation of registries and UIs
Blockchains are Not Stand-Alone Systems

UI for humans

Key management

IoT integration

Blockchain is a component

Legacy systems

BIG DATA

private data

Auxiliary databases
Non-Functional Trade-Offs

• Compared to conventional database & script engines, blockchains have:
  
  (-) Confidentiality, Privacy
  (+) Integrity, Non-repudiation
  (+ read/ - write) Availability
  (-) Modifiability
  (-) Throughput / Scalability / Big Data
  (+ read/ - write) Latency

  Security: combination of CIA properties
### Taxonomy

Blockchain-related design decisions regarding (de)centralisation, with an indication of their relative impact on quality properties

Legend: ⊕: Less favourable, ⊕⊕: Neutral, ⊕⊕⊕: More favourable

<table>
<thead>
<tr>
<th>Design Decision</th>
<th>Option</th>
<th>Fundamental properties</th>
<th>Cost efficiency</th>
<th>Performance</th>
<th>#Failure points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully Centralised</td>
<td>Services with a single provider (e.g., governments, courts)</td>
<td>⊕⊕⊕</td>
<td>⊕⊕⊕</td>
<td>⊕⊕</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Services with alternative providers (e.g., banking, online payments,</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>cloud services)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partially Centralised &amp;</td>
<td>Permissioned blockchain with permissions for fine-grained operations</td>
<td>⊕⊕⊕</td>
<td>⊕⊕⊕</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td>Partially Decentralised</td>
<td>on the transaction level (e.g., permission to create assets)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Permissioned blockchain with permission-less normal nodes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fully Decentralised</td>
<td>Permission-less blockchain</td>
<td>⊕⊕⊕</td>
<td>⊕⊕</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td>Verifier</td>
<td>Single verifier trusted by the network (external verifier signs valid</td>
<td>⊕⊕⊕</td>
<td>⊕⊕</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td></td>
<td>transactions; internal verifier uses previously-injected external</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>state)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>M-of-N verifier trusted by the network</td>
<td>⊕⊕⊕</td>
<td>⊕⊕</td>
<td>⊕</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ad hoc verifier trusted by the participants involved</td>
<td>⊕</td>
<td>⊕⊕</td>
<td>⊕</td>
<td></td>
</tr>
</tbody>
</table>

Also consider skills available for a specific platform

Majority (nodes, power, stake)

M (per ad hoc choice)
Blockchain-related design decisions regarding storage and computation, with an indication of their relative impact on quality properties

<table>
<thead>
<tr>
<th>Design Decision</th>
<th>Option</th>
<th>Fundamental properties</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item data</td>
<td>On-chain</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Embedded in transaction (Bitcoin)</td>
<td>⊕⊕⊕⊕</td>
<td>⊕</td>
</tr>
<tr>
<td></td>
<td>Embedded in transaction (Public Ethereum)</td>
<td>⊕⊕⊕⊕</td>
<td>⊕</td>
</tr>
<tr>
<td></td>
<td>Smart contract variable (Public Ethereum)</td>
<td>⊕⊕⊕⊕</td>
<td>⊕</td>
</tr>
<tr>
<td></td>
<td>Smart contract log event (Public Ethereum)</td>
<td>⊕⊕⊕⊕</td>
<td>⊕</td>
</tr>
<tr>
<td></td>
<td>Private / Third party cloud</td>
<td>⊕⊕⊕⊕</td>
<td>⊕</td>
</tr>
<tr>
<td></td>
<td>Peer-to-Peer system</td>
<td>⊕⊕⊕⊕</td>
<td>⊕</td>
</tr>
<tr>
<td></td>
<td>Off-chain</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Smart contract</td>
<td>⊕⊕⊕⊕</td>
<td>⊕</td>
</tr>
<tr>
<td></td>
<td>Separate chain</td>
<td>⊕⊕⊕⊕ (public) ⊕ (public)</td>
<td>⊕</td>
</tr>
<tr>
<td>Computation</td>
<td>On-chain</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transaction constraints</td>
<td>⊕⊕⊕⊕</td>
<td>⊕</td>
</tr>
<tr>
<td></td>
<td>Smart contract</td>
<td>⊕⊕⊕⊕</td>
<td>⊕</td>
</tr>
<tr>
<td></td>
<td>Off-chain</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Private / Third party cloud</td>
<td>⊕⊕⊕⊕</td>
<td>⊕</td>
</tr>
<tr>
<td>Design Decision</td>
<td>Option</td>
<td>Fundamental properties</td>
<td>Cost efficiency</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------</td>
<td>------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Blockchain scope</td>
<td>Public blockchain</td>
<td>★★★★</td>
<td>★</td>
</tr>
<tr>
<td></td>
<td>Consortium/community blockchain</td>
<td>★★</td>
<td>★★</td>
</tr>
<tr>
<td></td>
<td>Private blockchain</td>
<td>★</td>
<td>★★</td>
</tr>
<tr>
<td>Data structure</td>
<td>Blockchain</td>
<td>★★★★</td>
<td>★</td>
</tr>
<tr>
<td></td>
<td>GHOST</td>
<td>★★</td>
<td>★★</td>
</tr>
<tr>
<td></td>
<td>BlockDAG</td>
<td>★</td>
<td>★★</td>
</tr>
<tr>
<td></td>
<td>Segregated witness</td>
<td>★★★★</td>
<td>★</td>
</tr>
<tr>
<td>Consensus Protocol</td>
<td>Security-wise</td>
<td>Proof-of-work</td>
<td>★★★</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proof-of-retrievability</td>
<td>★★★</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proof-of-stake</td>
<td>★★</td>
</tr>
<tr>
<td></td>
<td>BFT (Byzantine Fault Tolerance)</td>
<td>★</td>
<td>★★</td>
</tr>
<tr>
<td></td>
<td>Scalability-wise</td>
<td>Bitcoin-NG</td>
<td>★★★</td>
</tr>
<tr>
<td></td>
<td>Off-chain transaction protocol</td>
<td>★</td>
<td>★★</td>
</tr>
<tr>
<td></td>
<td>Mini-blockchain</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Protocol Configuration</td>
<td>Security-wise</td>
<td>X-block confirmation</td>
<td>★</td>
</tr>
<tr>
<td></td>
<td>Checkpointing</td>
<td>★★★★</td>
<td>★★</td>
</tr>
<tr>
<td></td>
<td>Scalability-wise</td>
<td>Original block size and frequency</td>
<td>★★★</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Increase block size / Decrease mining time</td>
<td>★</td>
</tr>
<tr>
<td>New blockchain</td>
<td>Security-wise</td>
<td>Merged mining</td>
<td>★★★★</td>
</tr>
<tr>
<td></td>
<td>Hook popular blockchain at transaction level</td>
<td>★★</td>
<td>★</td>
</tr>
<tr>
<td></td>
<td>Proof-of-burn</td>
<td>★</td>
<td>★★</td>
</tr>
<tr>
<td></td>
<td>Scalability-wise</td>
<td>Side-chains</td>
<td>★★★★</td>
</tr>
<tr>
<td></td>
<td>Multiple private blockchains</td>
<td>★</td>
<td>★</td>
</tr>
</tbody>
</table>
Cost of Distrust

• RQ: How much more expensive is blockchain over Cloud services?
  • Lens: business process execution
  • AWS SWF vs. Ethereum public blockchain
    – In both cases: pay per instruction
  • Experiments on two use cases:
    – Incident management (literature)
      – 32 instances on public Ethereum vs. 1000 runs on SWF
    – Invoicing (industry, 5316 log traces, 65K events)
      – Full log replayed on SWF and private Ethereum

• Result:
  • 2 orders of magnitude more expensive to use blockchain
    – exchange rate: 1 Ether = US$ 11.32  (26/8/2016)
  • ~US$ 0.35 per process instance on public blockchain
    – outweighed by cost of escrow (if needed) for about US$ 10 of value
Availability

- Availability: the system’s readiness for correct service
  - Blockchain as component: read: +++ ; write: ?
- Ethereum: 12 blocks confirmation is accepted, said to take 3 mins
  - Measurements:
Availability

- Potential issue: block gas limit
  - Gas limit is set by miners through “voting”
  - The sum of Gas of all transactions in a block must be less than the limit
- Response to DDoS attack: lower block gas limit
Availability

• Potential issue: block gas limit
  • Gas limit is set by miners through “voting”
  • The sum of Gas of all transactions in a block must be less than the limit
• Response to DDoS attack: lower block gas limit
• Who would be affected by that?
Latency simulation

• Goal: predict latency for blockchain-based application before building it
  • Challenge specifically for latency: mean and variation
• Means: Architecture performance modeling
  • Paladio Component Models with individual latency distributions + connections + probability of branching
  • Allows changing the models for What-If analysis
  • For instance: change inter-block time on private blockchain – what does that mean for overall application latency?
Latency simulation

• Goal: predict latency for blockchain-based application before building it
  • Challenge specifically for latency: mean and variation
• Means: Arch
  • Paladio Component Models with individual latency distributions + connections + probability of branching
  • Allows changing the models for what-if analysis
  • For instance: change inter-block time on private blockchain – what does that mean for overall application latency?
Latency: what if we change required number of confirmation blocks?

1 block

6 blocks

12 blocks
Using Smart Contracts for Business Process Monitoring and Execution

Ingo Weber, Sherry Xu, Regis Riveret, Guido Governatori, Alexander Ponomarev and Jan Mendling

Untrusted business process monitoring and execution using blockchain. BPM 2016

Luciano García-Bañuelos, Alexander Ponomarev, Marlon Dumas, and Ingo Weber

Optimized Execution of Business Processes on Blockchain. BPM 2017

Orlenys López-Pintado, Luciano García-Bañuelos, Marlon Dumas, and Ingo Weber.

Caterpillar: A blockchain-based business process management system. BPM Demo 2017
Motivation

• Integration of business processes across organizations: a key driver of productivity gains.

• Collaborative process execution
  • Doable when there is trust – supply chains can be tightly integrated
  • Problematic when involved organizations have a lack of trust in each other
    → if 3+ parties should collaborate, where to execute the process that ties them together?
  • Common situation in “coopetition”
Motivation: example

Issues:
- Knowing the status, tracking correct execution → Trusted 3rd party?  
- Handling payments  
- Resolving conflicts → Blockchain!
Approach in a nutshell

- Goal: execute collaborative business processes as smart contracts
  - Translate (enriched) BPMN to smart contract code
  - Triggers act as bridge between Enterprise world and blockchain
- Smart contract does:
  - Independent, global process monitoring
  - Conformance checking: only expected messages are accepted, only from the respective role
  - Automatic payments & escrow
  - Data transformation
  - Encryption
Architecture

Design Time

- BPMN model
- Translator
- Process implementation

Factory contract

Run Time

- Process instance contract (Mediator/Monitor)
  - Escrow
  - Verification
  - Partial data payload
  - Execution state
  - Deposit collector
  - Business logic (Active/passive)
  - Participants account

On blockchain

- Factory contract
- Trigger
- Data payload
- Key distributor

Off blockchain

Interface
- Internal process
Runtime

• Instantiation:
  • New *instance contract* per process instance
  • Assign accounts to roles during initialization
  • Exchange keys and create secret key for the instance

• Messaging:
  • Instead of sending direct WS calls: send through triggers & smart contract
  • Instance contract handles:
    – Global monitoring
    – Conformance checking
    – Automated payments*
    – Data transformation*
Runtime

• Instantiation:
  • New *instance contract* per process instance
Demonstration
Summary

• Architecting and developing applications on Blockchain is challenging
  • Our research:
    – Designing applications with Blockchain
    – Software Architecture methods for Blockchain-based applications
    – Empirical and formal research on Blockchains
    – How to use smart contracts
    – Model-driven development of smart contracts

• Using Blockchain for process monitoring and execution
  • Applicable in lack-of-trust settings for collaborative process execution
  • Our approach:
    – translate from process models to Solidity
    – use triggers to connect Blockchain and Enterprise systems
  • Evaluation results: latency and cost should be acceptable in many cases
  • Screencast video and more details – see papers
Thank you

Ingo Weber | Principal Research Scientist & Team Leader
ingo.weber@data61.csiro.au
Conj. Assoc. Professor, UNSW Australia | Adj. Assoc. Professor, Swinburne University

AAP team Blockchain research: https://research.csiro.au/data61/blockchain/
Blockchain reports: https://www.data61.csiro.au/blockchain/

www.csiro.au


7. EthDrive: A Peer-to-Peer Data Storage with Provenance, Xiao Liang Yu, Xiwei Xu and Bin Liu. 29th International Conference on Advanced Information Systems Engineering (CAISE’17).

8. The Balance Attack or Why Forkable Blockchains are Ill-Suited for Consortium, Christopher Natoli and Vincent Gramoli. 2017 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN’17).