블록체인

NEXT GENERATION

한국블록체인학회 학술대회 및 워크샵

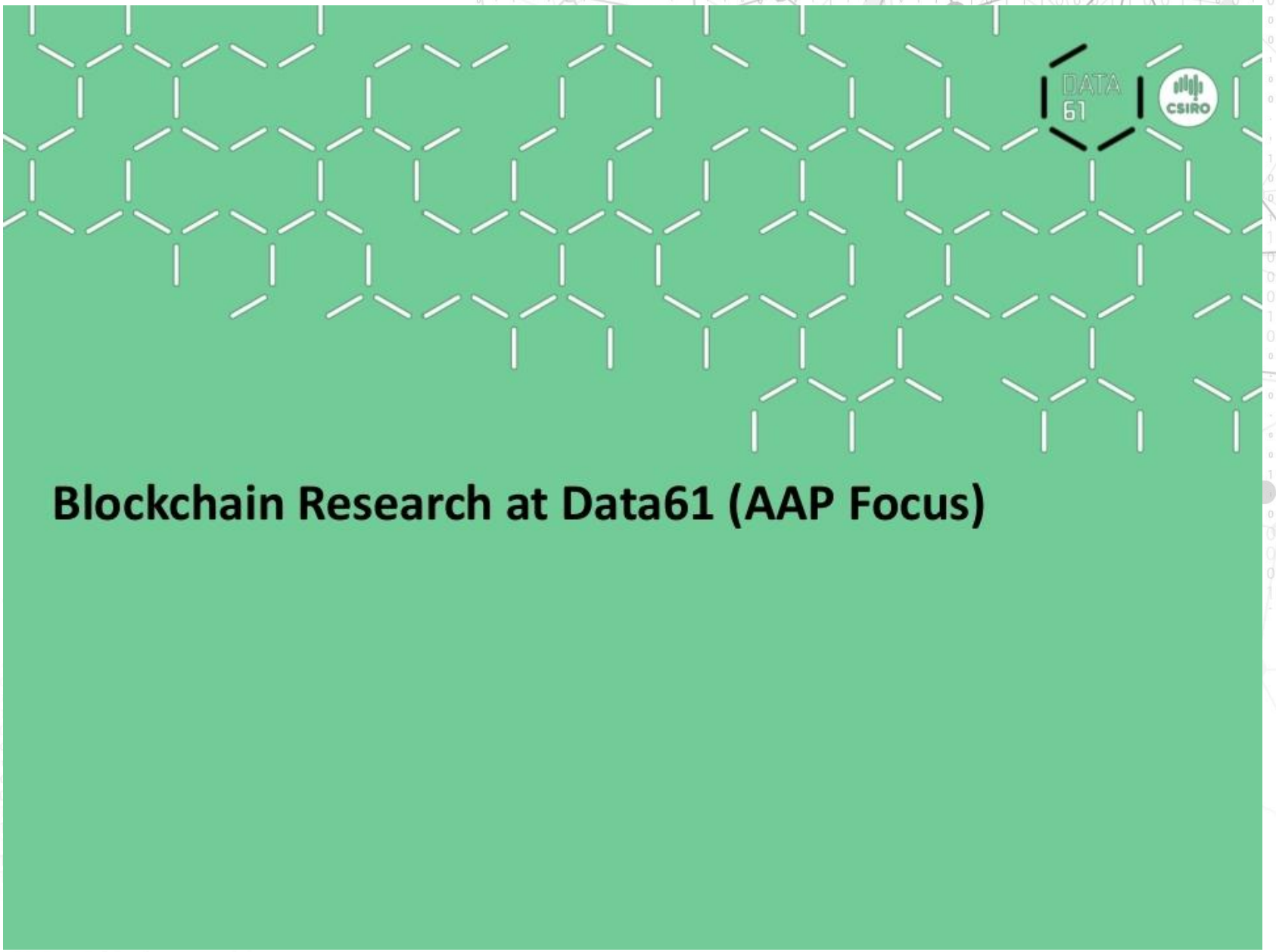
Key Note Speaker 1

"Blockchain projects in Australia"

Ph.D. Ingo Weber

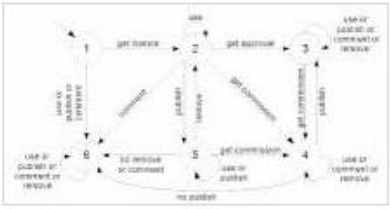
Principle Research Scientist


DATA
61
Blockchains and Smart Contracts:
Architecture Design and Model-
Driven Development
Ingo Weber | Principal Research Scientist & Team Leader
Architecture & Analytics Platforms (AAP) team
ingo.weber@data61.csiro.au
May 2017
www.csiro.au
CSIRO


DATA 61
CSIRO
Blockchain Research at Data61 (AAP Focus)

Blockchain Research at Data61

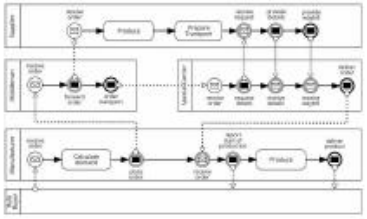
- Designing Systems with Blockchain
  - Design Trade-offs
  - Model-driven development
  - Governance and risk management
- Trustworthy Blockchain Systems
  - Formal



  - Empirical

- Defining and Using Smart Contracts
  - As Legal Contracts

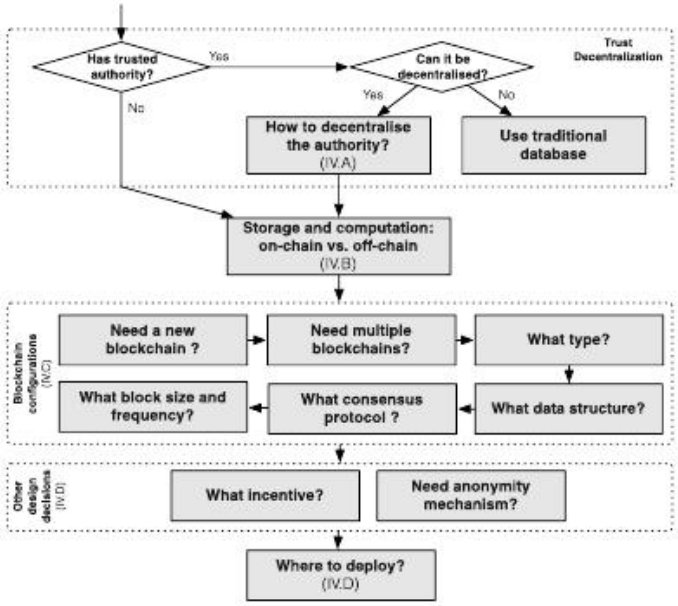
  - Business Process



3 | Blockchain | Data61, CSIRO

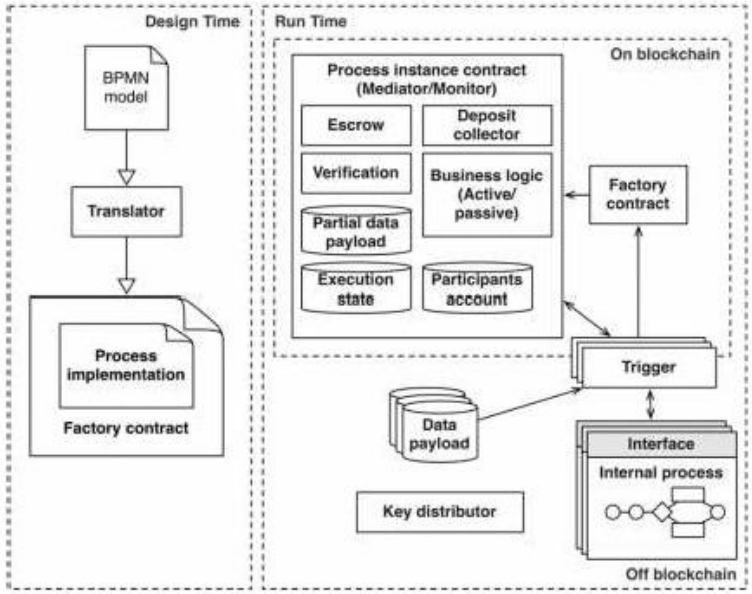
Designing Systems with Blockchain

- Design Process
  - **A taxonomy of blockchain-based systems for architecture design,** X. Xu, I. Weber, M. Staples et al., ICSA2017.
  - **The blockchain as a software connector,** X. Xu, C. Pautasso, L. Zhu et al., WICSA2016.
- Quality Analysis
  - **Comparing blockchain and cloud services for business process execution,** P. Rimba, A. B. Tran, I. Weber et al., ICSA2017.
  - **Predicting latency of blockchain-based systems using architectural modelling and simulation**, R. Yasaweerasinghelage, M. Staples and I. Weber, ICSA2017.
- Model-Driven
  - **Regerator: a Registry Generator for Blockchain, A. B. Tran, X. Xu, I. Weber, CAISE2017.**
  - **From business process models, see next slide**
- Integration with other systems
  - **EthDrive: A Peer-to-Peer Data Storage with Provenance, X. L. Yu, X. Xu, B. Liu, CAISE2017.**
- Governance and risk management
  - **Risks and Opportunities for Systems Using Blockchain and Smart Contracts,** Treasury report



Defining and Using Smart Contracts

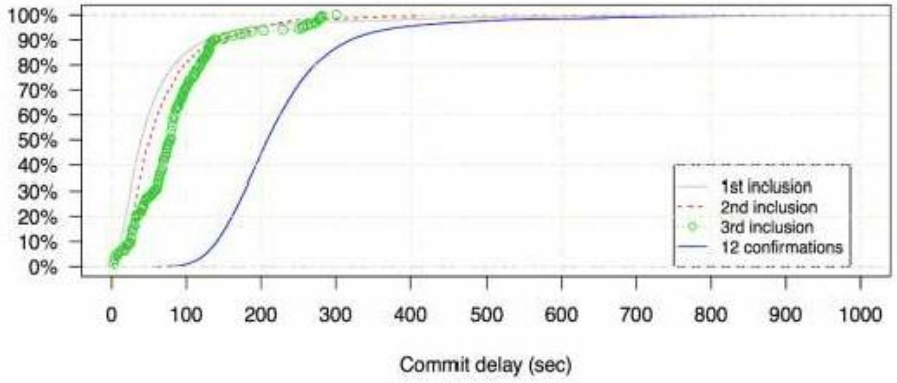
- Business Process
  - **Untrusted business process monitoring and execution using blockchain,**
    I. Weber, X. Xu, R. Riveret et al., BPM2016
  - **Optimized Execution of Business Processes on Blockchain,**
    L. García-Bañuelos, A. Ponomarev, M. Dumas, Ingo Weber, BPM2017
- As Legal Contracts
  - **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, *NCA*2016
  - **On the Danger of Private Blockchains,** V. Gramoli,*DCCL*2016
  - **(Leader/Randomization/Signature)-free Byzantine Consensus for Consortium Blockchains,** T. Crain, V. Gramoli, M. Larrea, M. Raynal, arXiv:1702.03068, 2016

- Empirical
  - **New kids on the block: an analysis of modern blockchains,** L. Anderson, R. Holz, A. Ponomarev et al., arXiv:1606:06530, 2016
  - ...



6 | Data61 : Preliminary Findings on Blockchain/DLT Projects

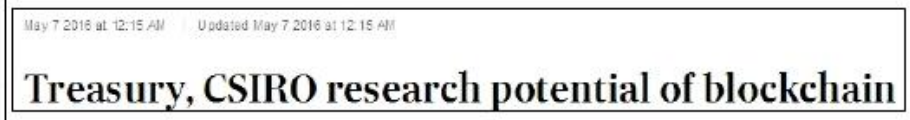
Projects with Australian Treasury

- Jul 2016 – Mar 2017
- Funded by National Innovation Science Agenda
- With assistance of The Treasury


Budget 2016: Data61's blockchain review welcomed by fintech leaders

- **DLT Foresight**
  - What might plausibly happen, across society & economy?
- **Blockchain Proof-of-Concepts**
  - What are technical risks & opportunities for use cases?


CSIRO's Data61 and Treasury join forces to examine the blockchain
Blockchain expected to change the way Australia's economy operates


May 7 2016 at 12:15 AM Updated May 7 2016 at 12:15 AM
Treasury, CSIRO research potential of blockchain


DATA 61
CSIRO
Architecting Applications on Blockchain
Based on [1,2,3]

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?

9 | Blockchain Smart Contracts: Use and Application in BPM | Ingo Weber

Our work – AAP team

- Architecting applications on Blockchain:
  - Taxonomy and design process [1]
  - "Cost of Distrust": how much more expensive is blockchain? [2]
  - Latency: simulation under changes [3]
- Model-driven development of smart contracts
  - Business process execution [4,5]
  - Model-based generation of registries and UIs: "Regerator" [6]

Taxonomy

Blockchain-related design decisions regarding (de)centralisation, with an indication of their relative impact on quality properties
Legend: ⊕: Less favourable, ⊕⊕: Neutral, ⊕⊕⊕: More favourable

Design Decision	Option	Impact			
		Fundamental properties	Cost efficiency	Performance	#Failure points
Fully Centralised	Services with a single provider (*e.g.*, governments, courts)	⊕	⊕⊕⊕	⊕⊕⊕	1
	Services with alternative providers (*e.g.*, banking, online payments, cloud services)				
Partially Centralised & Partially Decentralised	Permissioned blockchain with permissions for fine-grained operations on the transaction level (*e.g.*, permission to create assets)	⊕⊕	⊕⊕	⊕⊕	*
	Permissioned blockchain with permissioned miners (write), but permission-less normal nodes (read)				
Fully Decentralised	Permission-less blockchain	⊕⊕⊕	⊕	⊕	Majority (nodes, power, stake)
		Fundamental properties	**Cost efficiency**	**Performance**	**#Failure points**
Verifier	Single verifier trusted by the network (external verifier signs valid transactions; internal verifier uses previously-injected external state)	⊕⊕	⊕⊕	⊕⊕	1
	M-of-N verifier trusted by the network	⊕⊕⊕	⊕	⊕	M
	Ad hoc verifier trusted by the participants involved	⊕	⊕⊕⊕	⊕⊕	1 (per ad hoc choice)

11 | Blockchain Smart Contracts: Use and Application in BPM | Ingo Weber

Taxonomy

Blockchain-related design decisions regarding storage and computation, with an indication of their relative impact on quality properties

Design Decision		Option	Impact			
			Fundamental properties	Cost efficiency	Performance	Flexibility
Item data	On-chain	Embedded in transaction (Bitcoin)	⊕⊕⊕⊕	⊕	⊕	⊕⊕
		Embedded in transaction (Public Ethereum)		⊕⊕⊕⊕	⊕	⊕⊕⊕
		Smart contract variable (Public Ethereum)		⊕⊕	⊕⊕⊕	⊕
		Smart contract log event (Public Ethereum)		⊕⊕⊕	⊕⊕	⊕⊕
	Off-chain	Private / Third party cloud	⊕	~KB Negligible	⊕⊕⊕⊕	⊕⊕⊕⊕
		Peer-to-Peer system		⊕⊕⊕⊕	⊕⊕⊕	⊕⊕⊕
Item collection	On-chain	Smart contract	⊕⊕⊕⊕	⊕⊕⊕⊕ (public)	⊕⊕⊕⊕	⊕
		Separate chain		⊕ (public)	⊕	⊕⊕⊕⊕
Computation	On-chain	Transaction constraints	⊕⊕⊕⊕	⊕	⊕	⊕
		Smart contract				
	Off-chain	Private / Third party cloud	⊕	⊕⊕⊕⊕	⊕⊕⊕⊕	⊕⊕⊕⊕

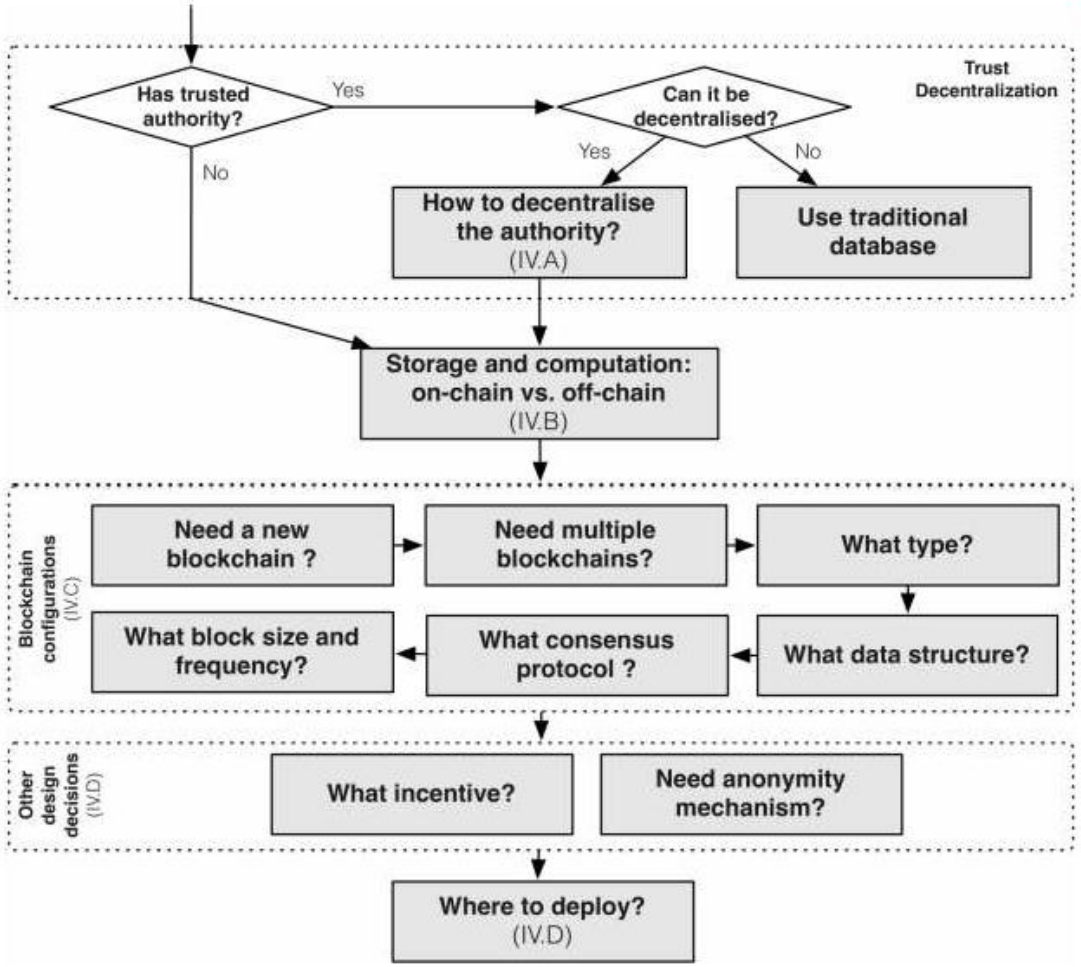
Taxonomy

Blockchain-related design decisions regarding blockchain configuration

Design Decision		Option	Impact			
			Fundamental properties	Cost efficiency	Performance	Flexibility
Blockchain scope		Public blockchain	⊕⊕⊕	⊕	⊕	⊕
		Consortium/community blockchain	⊕⊕	⊕⊕	⊕⊕	⊕⊕
		Private blockchain	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕
Data structure		Blockchain	⊕⊕⊕	⊕	⊕	⊕
		GHOST	⊕⊕	⊕⊕	⊕⊕	⊕
		BlockDAG	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕
		Segregated witness	⊕⊕⊕	⊕⊕	⊕	⊕
Consensus Protocol	Security-wise	Proof-of-work	⊕⊕⊕	⊕	⊕	⊕
		Proof-of-retrievability	⊕⊕⊕	⊕	⊕	⊕
		Proof-of-stake	⊕⊕	⊕⊕	⊕⊕	⊕⊕⊕
		BFT (Byzantine Fault Tolerance)	⊕	⊕⊕⊕	⊕⊕⊕	⊕
	Scalability-wise	Bitcoin-NG	⊕⊕⊕	⊕	⊕	⊕
		Off-chain transaction protocol	⊕	⊕⊕⊕	⊕⊕	⊕⊕⊕
		Mini-blockchain	⊕⊕	⊕⊕	⊕	⊕⊕
Protocol Configuration	Security-wise	X-block confirmation	⊕	⊕	⊕	⊕⊕⊕
		Checkpointing	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕	⊕
	Scalability-wise	Original block size and frequency	⊕⊕⊕	n/a	⊕	n/a
		Increase block size / Decrease mining time	⊕	n/a	⊕⊕⊕	n/a
New blockchain	Security-wise	Merged mining	⊕⊕⊕	⊕⊕	⊕	⊕
		Hook popular blockchain at transaction level	⊕⊕	⊕	⊕⊕	⊕⊕⊕
		Proof-of-burn	⊕	⊕	⊕⊕⊕	⊕⊕
	Scalability-wise	Side-chains	⊕⊕⊕	⊕	⊕	⊕
		Multiple private blockchains	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕

13 | Blockchain Smart Contracts: Use and Application in BPM | Ingo Weber

Proposed design process (using Taxonomy)



14 | Blockchain Smart Contracts: Use and Application in BPM | Ingo Weber

Cost of Distrust

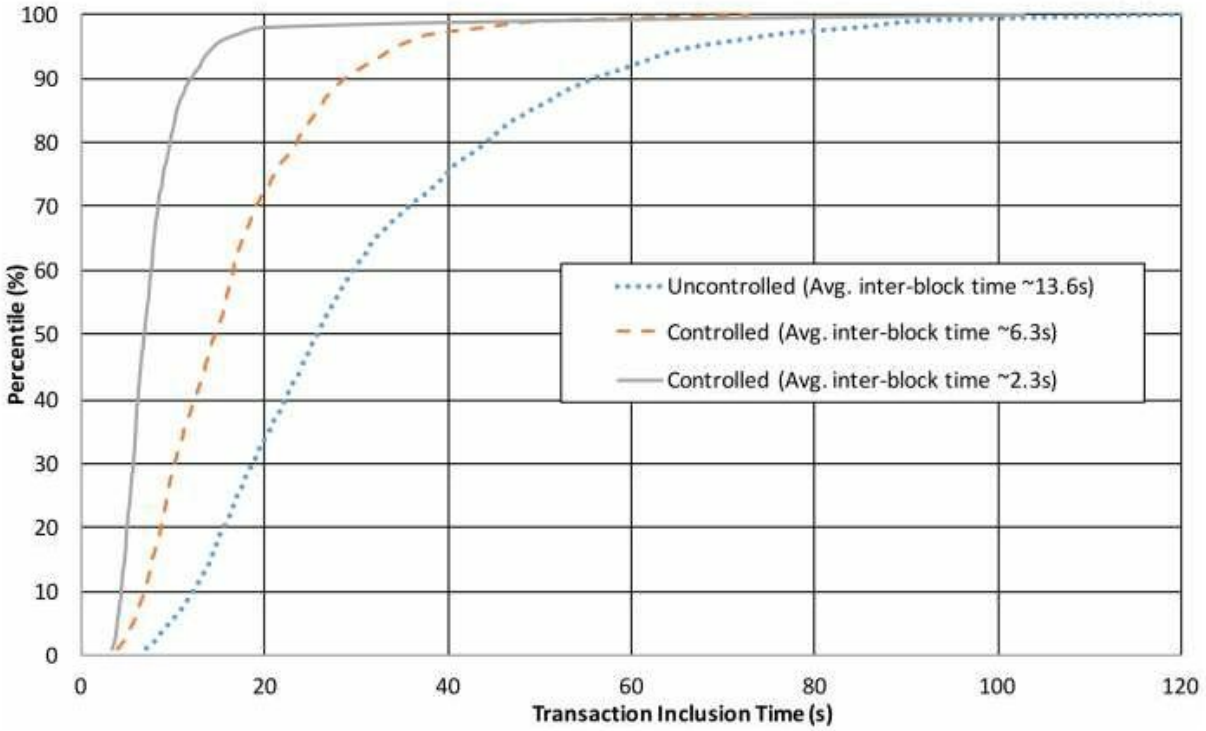
- RQ: How much more expensive is blockchain over Cloud services?
  - Lens: business process execution
  - AWS SWF vs. Ethereum public blockchain
    - In both cases: pay per instruction
  - Experiments on two use cases:
    - Incident management (literature)
      - 32 instances on public Ethereum vs. 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
  - ~US$ 0.35 per process instance on public blockchain
    - outweighed by cost of escrow (if needed) for about US$ 10 of value

15 | Blockchain Smart Contracts: Use and Application in BPM | Ingo Weber

Latency simulation

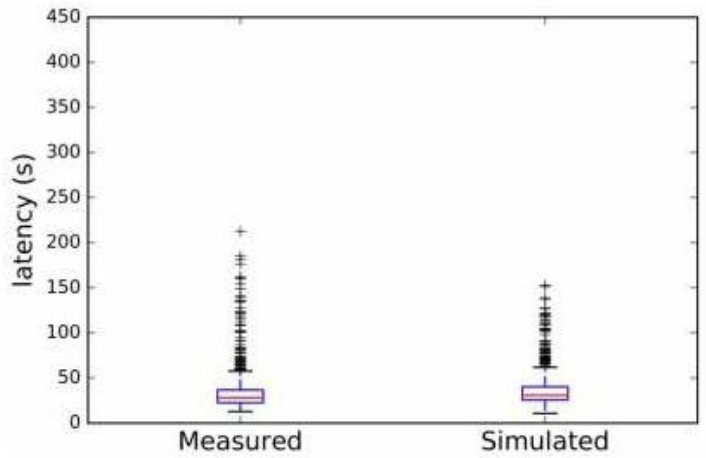
- Goal: predict latency for blockchain-based application before building it
  - Challenge specifically for latency: mean and variation
- Means: Architecture performance modeling
  - Paladio Component Models with individual latency distributions + connections + probability of branching
  - Allows changing 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

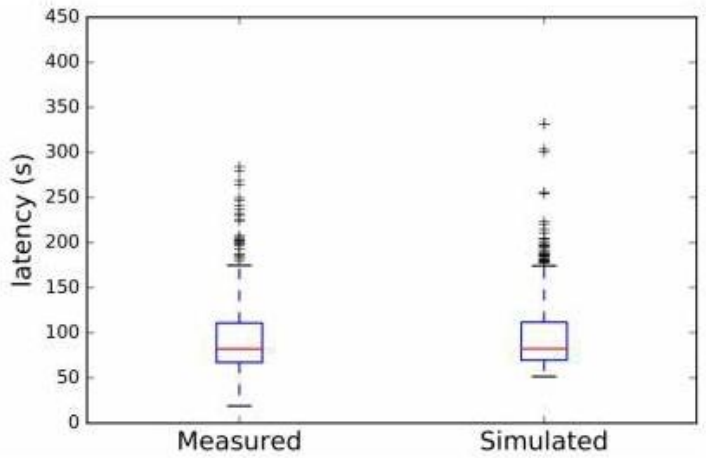


17 | Blockchain Smart Contracts: Use and Application in BPM | Ingo Weber

Latency: what if we change required number of confirmation blocks?

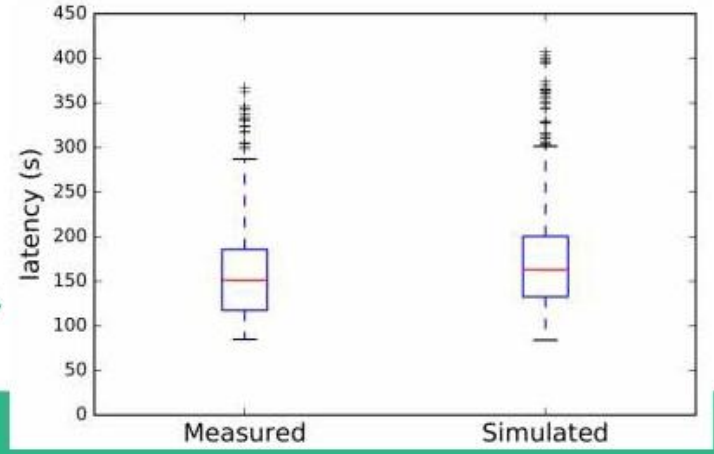


1 block



6 blocks

12 blocks



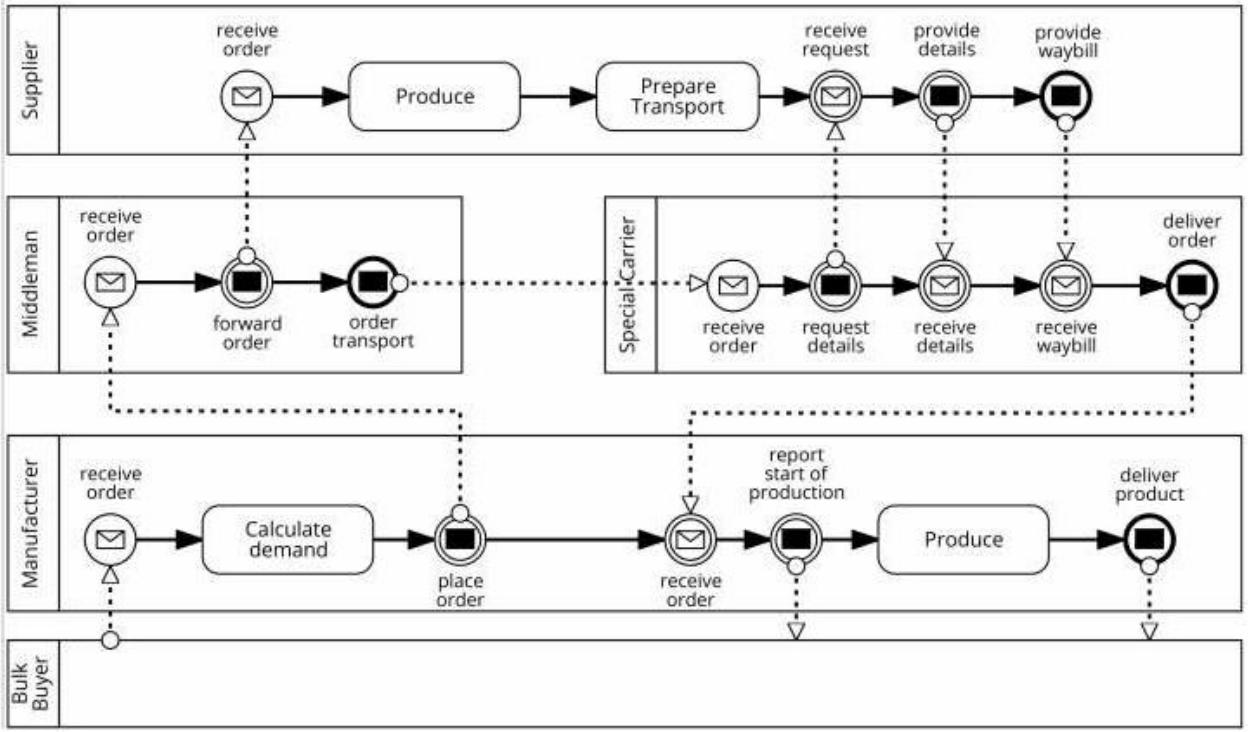
18 | Blockchain Smart Contracts: Use and Application in BPM | Ingo Weber


DATA 61
CSIRO
Using Smart Contracts for Business Process Monitoring and Execution
Based on [4,5]

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
- Handling payments
- Resolving conflicts

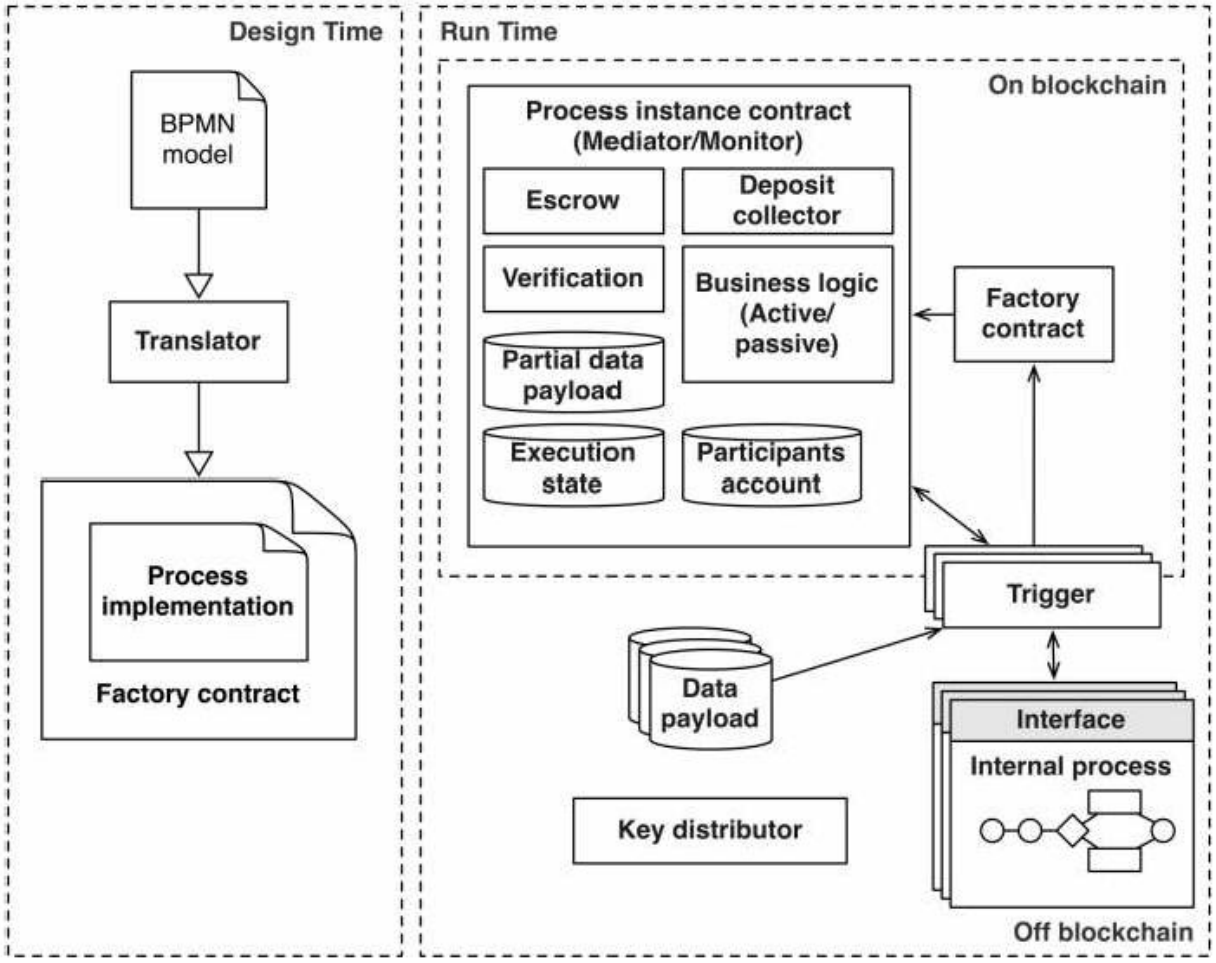
→ ~~Trusted 3rd party ?~~
→ Blockchain!

21 | Blockchain Smart Contracts: Use and Application in BPM | Ingo Weber

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



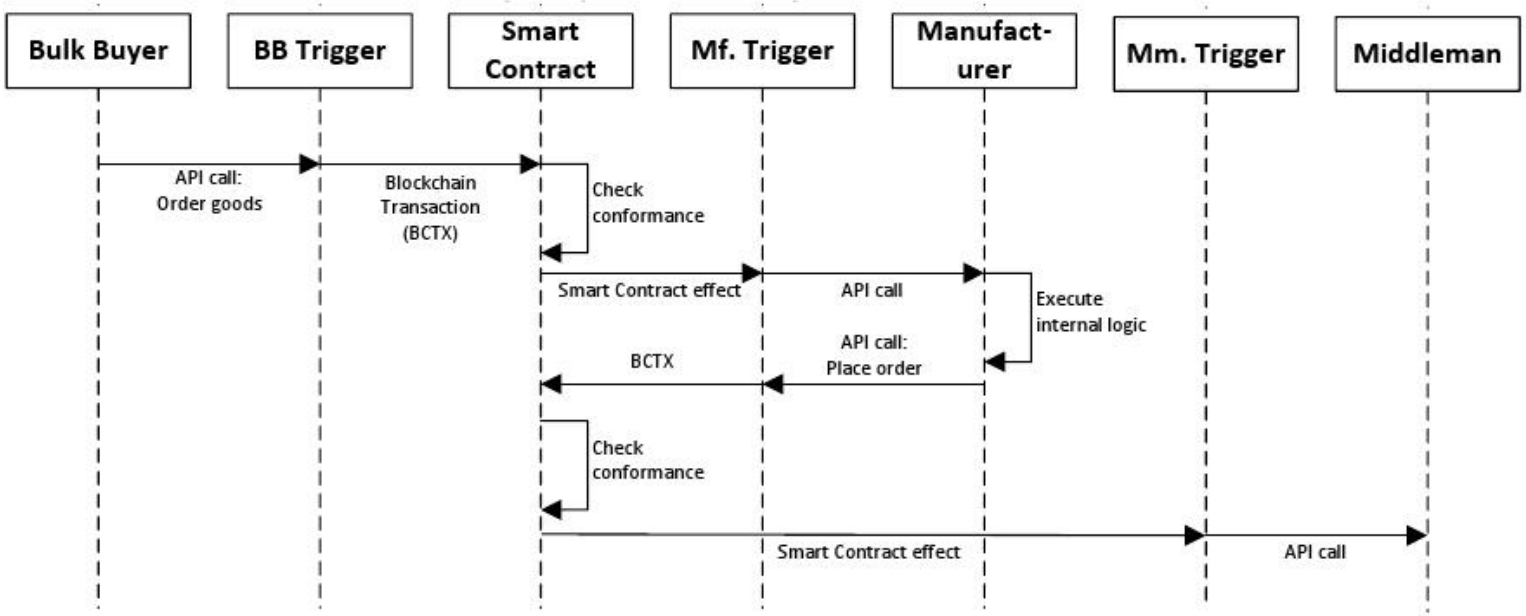
23 | Blockchain Smart Contracts: Use and Application in BPM | Ingo Weber

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

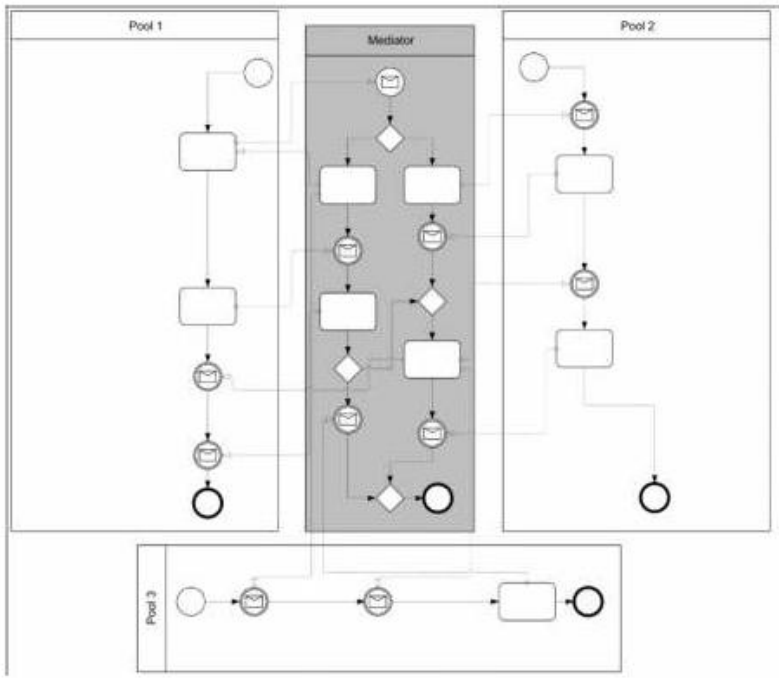
- Instantiation:
  - New *instance contract* per process instance



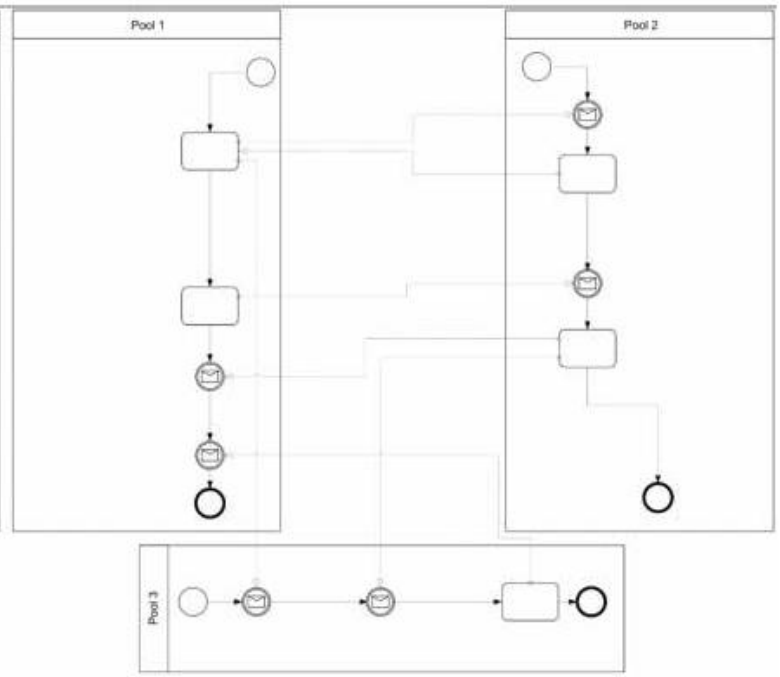
25 | Blockchain Smart Contracts: Use and Application in BPM | Ingo Weber

Collaborative processes: variants

Mediator (orchestration)



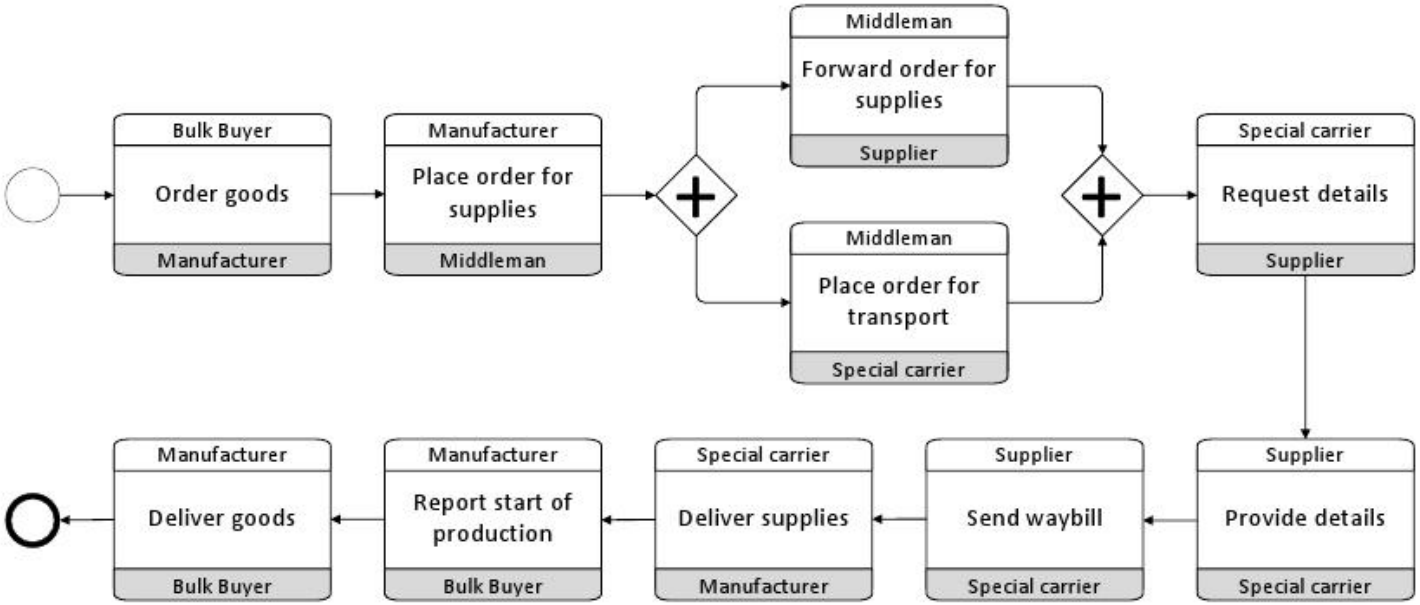
Choreography → C-Monitor



26 | Blockchain Smart Contracts: Use and Application in BPM | Ingo Weber

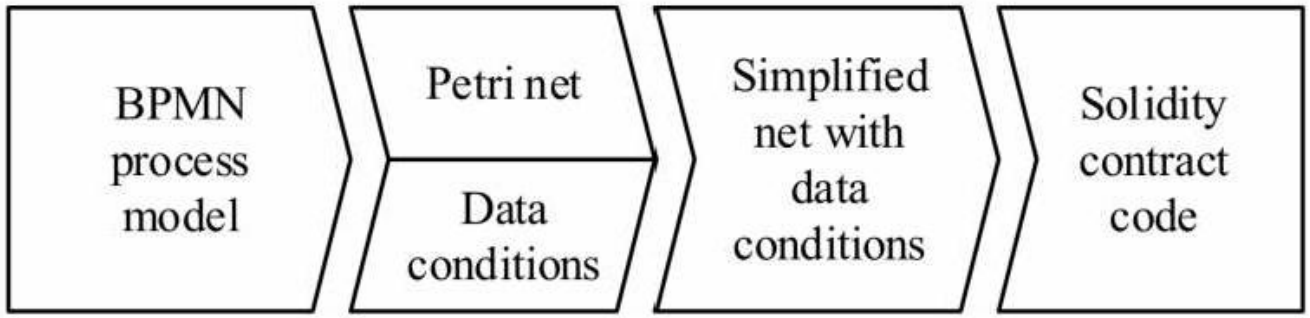
Translator

- Translate subset of BPMN elements to *Solidity*
  - BPMN Choreography diagrams or regular BPMN models with pools



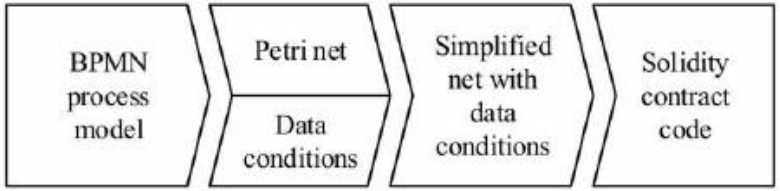
27 |

Translator

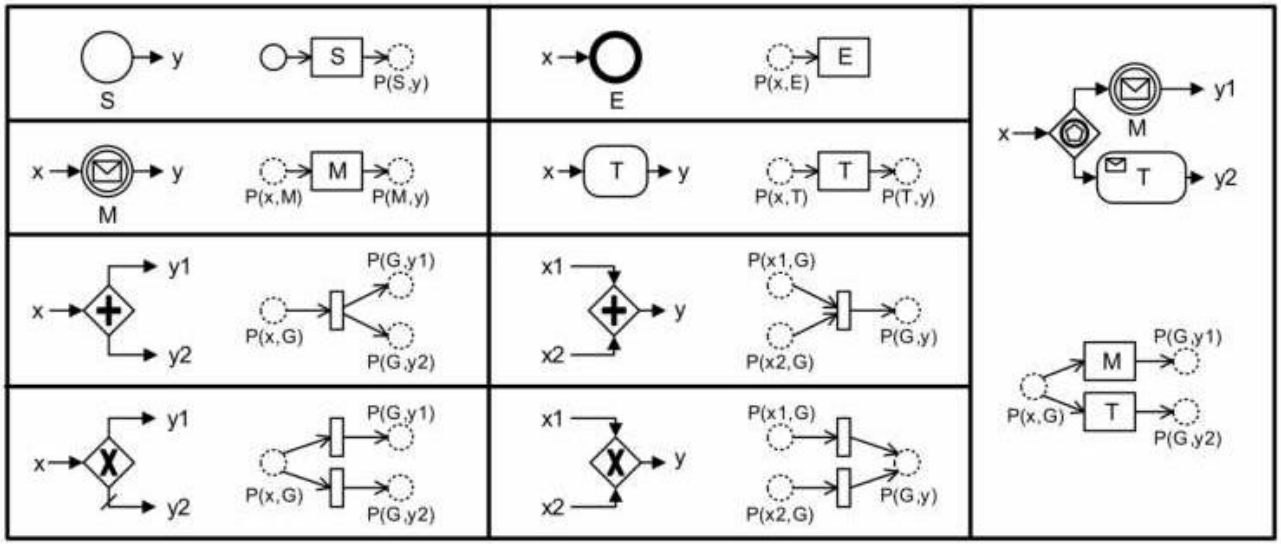


28 |

Translator

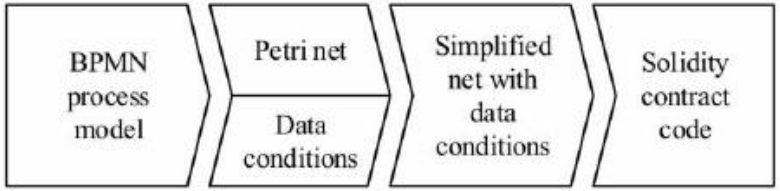


1.Translate BPMN control flow to WFnet (proven to be safe)



29 |

Translator

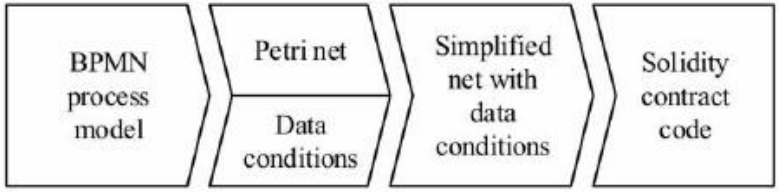


1.Translate BPMN control flow to WFnet (proven to be safe)

2.Capture dataflow and conditions

30 |

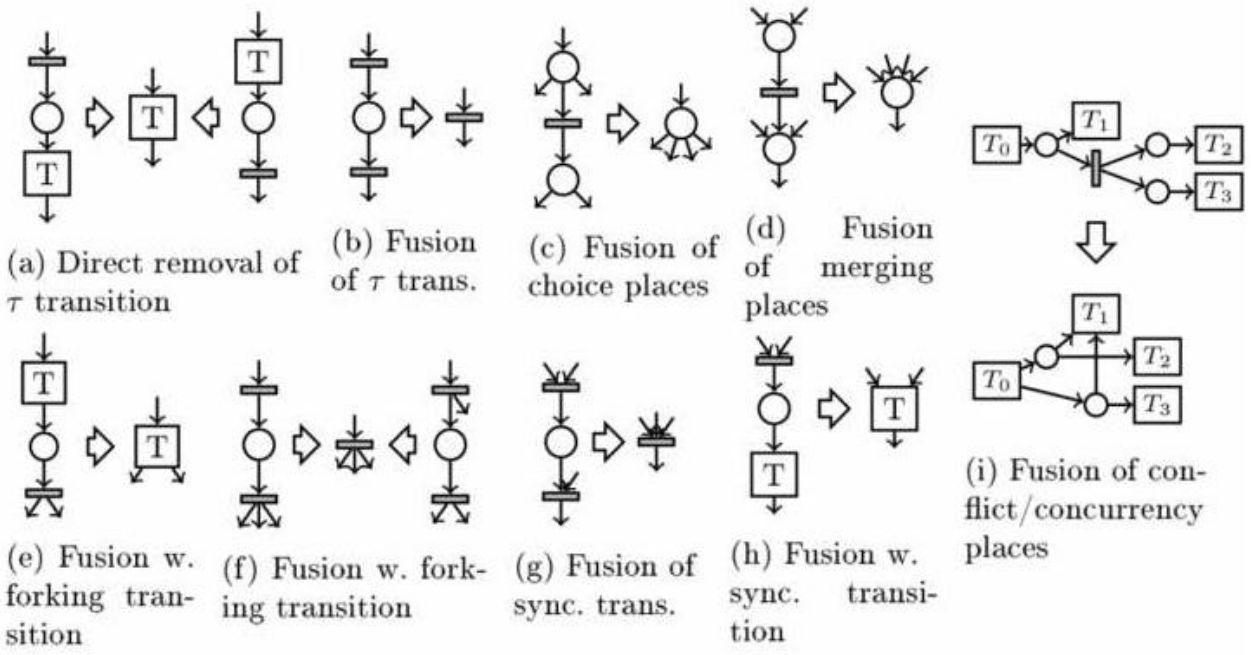
Translator



1.Translate BPMN control flow to WFnet (proven to be safe)

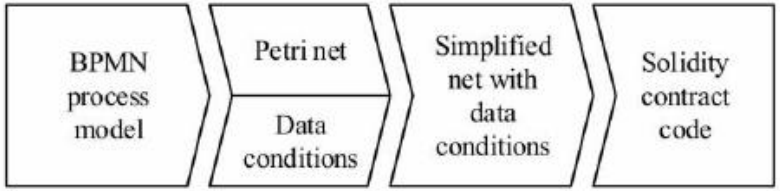
2.Capture dataflow and conditions

3.Reduce WFnet and annotate dataflow



31 |

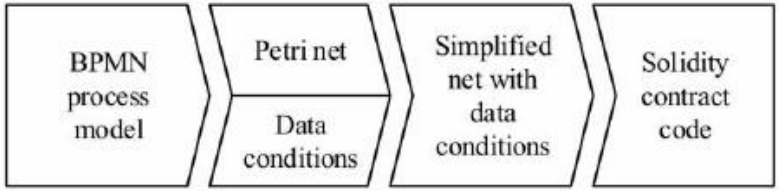
Translator



1. Translate BPMN control flow to WFnet (proven to be safe)
2. Capture dataflow and conditions
3. Reduce WFnet and annotate dataflow
4. Translate into Solidity code
   - Status of the process is kept in a bit vector
   - Updates are bit-wise operations

32 |

Translator



1. Translate BPMN control flow to WFnet (proven to be safe)
2. Capture
3. Reduce
4. Translat
   - Status
   - Update

Bit vector check: does pos 1 have value "1"?

```
1  contract BPMNContract {
2    uint marking = 1;
3    uint predicates = 0;
4
5    function CheckApplication( -input params- ) returns (bool) {
6      if (marking & 2 == 2) { // is there a token in place p1?
7        // Task B's script goes here, e.g. copy value of input params to contract variables
8        uint tmpPreds = 0;
9        if ( -eval P- ) tmpPreds |= 1; // is loan application complete?
10       if ( -eval Q- ) tmpPreds |= 2; // is the property pledged?
11       step(
12         marking & uint(~2) | 12,        // New marking
13         predicates & uint(~3) | tmpPreds // New evaluation for "predicates"
14       );
15       return true;
16     }
17     return false;
18   }
```

Bit vector update: set pos 1 to "0"

set pos 2 and 3 to "1"

33 |

Payments, escrow, data handling

- Payment / escrow using crypto-coins:
  - Instance contract has an account
  - Only the contract code governs what gets paid out, but anyone can pay in
  - Contract can base validity of transactions on associated payments
  - Anyone can see balance of the contract – enables trust:
    - All parties know when the money is there
    - The contract code specifies who gets paid how much, and when
- Data:
  - Status update data has to readable for the contract, most other data can be encrypted
    - Data used in conditions has to be readable
  - Sending data over blockchain is costly – can split on-chain vs. off-chain
    - On-chain: URI to the data + hash
    - Off-chain: reachable and addressable data store (IPFS, S3, ...)

Evaluation (1/3)

- 4 use case processes, 1 of them from industry with 5316 traces and 65K events
- Executed ~150K transactions on private and 256 TXs on public Ethereum blockchain

1. Correct execution (conformance checking): 100% correct classification
2. Cost:
   - Per process instance: cost on average of 0.0347 Ether (~$0.40) before optimization
   - After optimization:
     - About 25% reduction for industrial process (Opt-CF)
     - Up to 75% reduction if smart contract can be reused (Opt-Full)

Evaluation (1/3)

- 4 use case processes, 1 of them from industry with 5316 traces and 65K events
- Executed ~150K Ethereum blockc
1. Correct executio classification
2. Cost:
  - Per process instar optimization
  - After optimizatior
    - About 25% redu
    - Up to 75% redu

Process	Tested Traces	Variant	Avg. Cost		Savings (%)
			Instant.	Exec.	
Invoicing	5316	Default	1,089,000	33,619	–
		Opt-CF	807,123	26,093	-24.97
		Opt-Full	54,639	26,904	-75.46
Supply chain	62	Default	304,084	25,564	–
		Opt-CF	298,564	24,744	-2.48
		Opt-Full	54,248	25,409	-42.98
Incident mgmt.	124	Default	365,207	26,961	–
		Opt-CF	345,743	24,153	-7.04
		Opt-Full	54,499	25,711	-57.96
Insurance claim	279	Default	439,143	27,310	–
		Opt-CF	391,510	25,453	-8.59
		Opt-Full	54,395	26,169	-41.14

Evaluation (2/3)

3. Latency and throughput

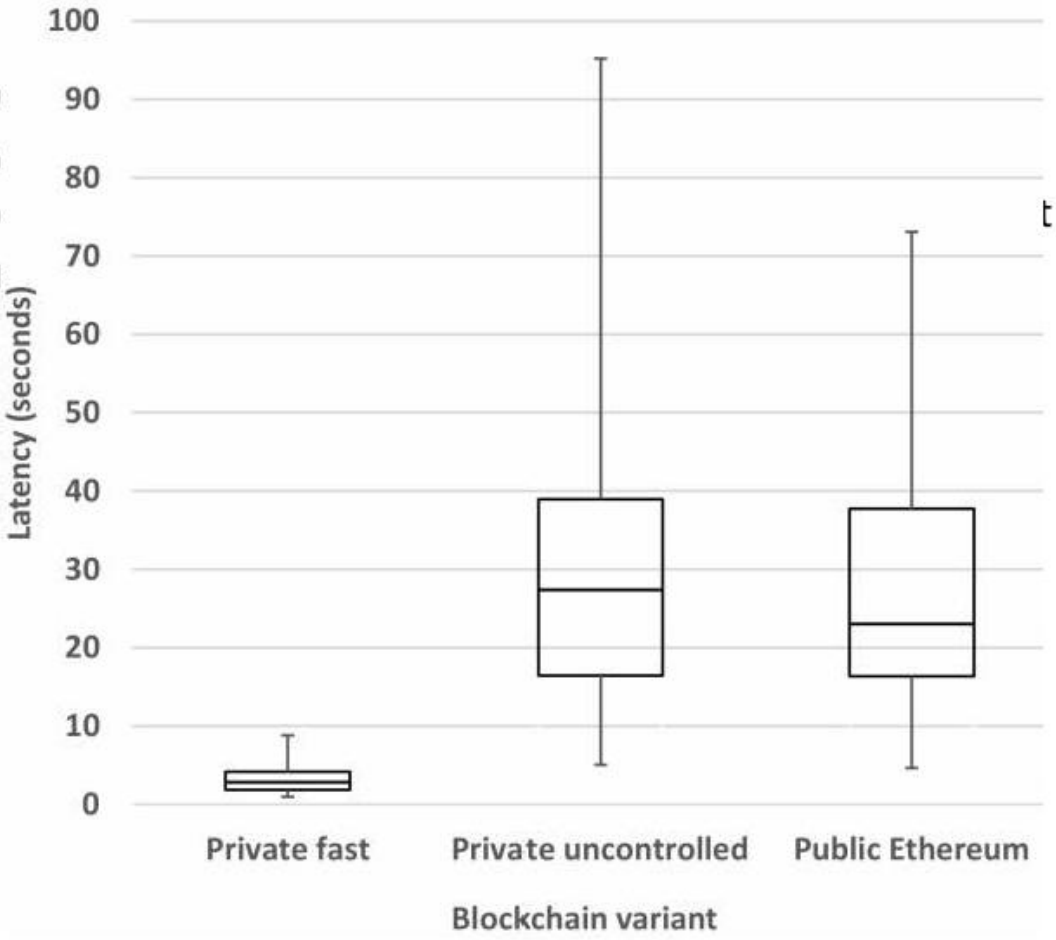
- Primary source of latency: mining time
  - Public Ethereum blockchain: median time between blocks set to 13-15s
  - Private blockchain: can control it

Evaluation (2/3)

3. Latency

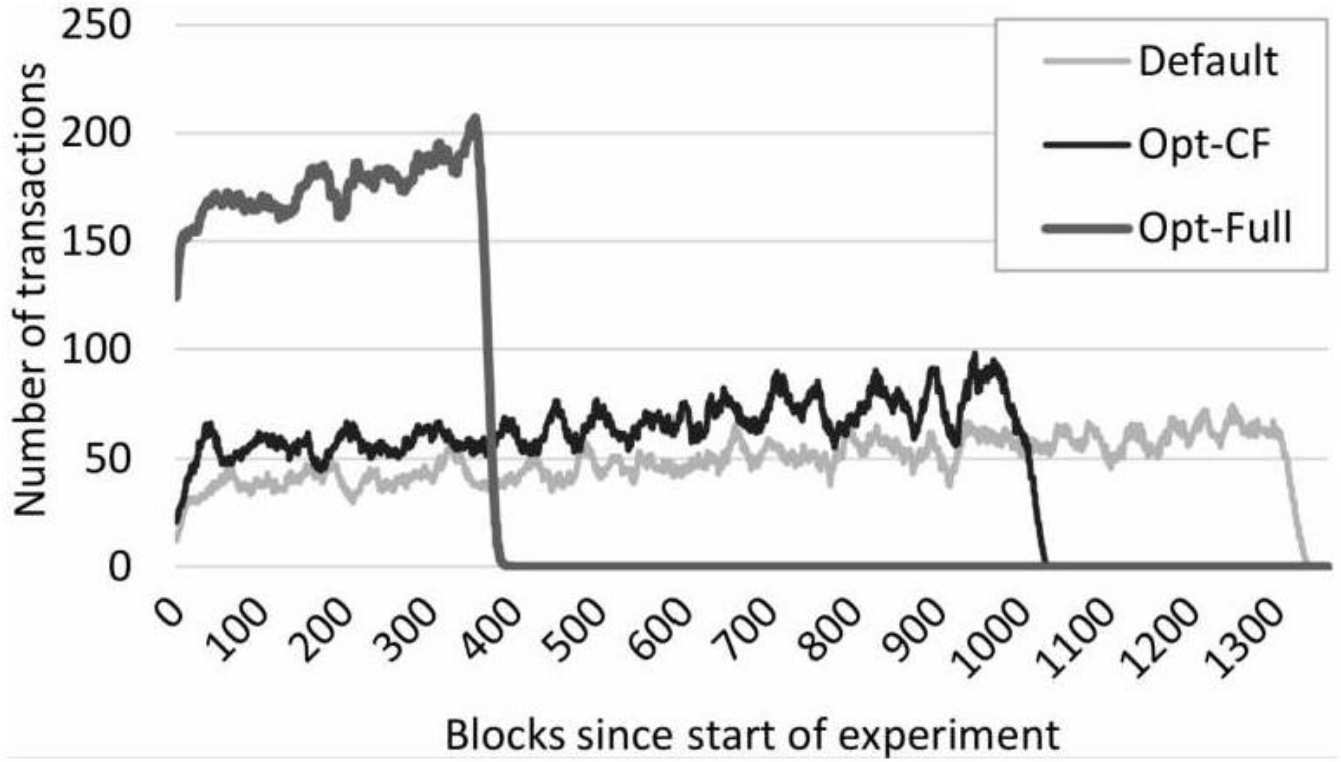
- Primary
  - Publi
  - Privat

t to 13-15s



Evaluation (3/3)

4. Throughput: optimized implementation can fill up each block
   - Limit: blockchain network-defined gas limit



- Unoptimized
- Control flow optimized
- Fully optimized

Summary

- Blockchain is a hot topic
  - Goal for AAP team: excellent science where there is genuine value
- Software design and model-driven development for applications using blockchain
  - Taxonomy and design process
  - Latency simulation
  - Cost comparison
  - Model-driven development: registries and business processes
- Using Blockchain for process monitoring and execution
  - Applicable in lack-of-trust settings for collaborative process execution
  - Compile process model into smart contract, with highly beneficial features


DATA
61
Blockchains and Smart Contracts:
Architecture Design and Model-
Driven Development
Ingo Weber | Principal Research Scientist & Team Leader
Architecture & Analytics Platforms (AAP) team
ingo.weber@data61.csiro.au
May 2017
www.csiro.au
CSIRO

References

1. Xiwei Xu, Ingo Weber, Mark Staples, Liming Zhu, Jan Bosch, Len Bass, Cesare Pautasso, and Paul Rimba. *A taxonomy of blockchain-based systems for architecture design.* In ICSA'17: IEEE International Conference on Software Architecture, Gothenburg, Sweden, April 2017.
2. Paul Rimba, An Binh Tran, Ingo Weber, Mark Staples, Alexander Ponomarev, and Xiwei Xu. *Comparing blockchain and cloud services for business process execution.* In ICSA'17: IEEE International Conference on Software Architecture, short paper, Gothenburg, Sweden, April 2017.
3. Rajitha Yasaweerasinghelage, Mark Staples, and Ingo Weber. *Predicting latency of blockchain-based systems using architectural modelling and simulation.* In ICSA'17: IEEE International Conference on Software Architecture, short paper, Gothenburg, Sweden, April 2017.
4. Ingo Weber, Sherry Xu, Regis Riveret, Guido Governatori, Alexander Ponomarev and Jan Mendling. *Untrusted business process monitoring and execution using blockchain.* In BPM'16: International Conference on Business Process Management, Rio de Janeiro, Brazil , September, 2016
5. Luciano García-Bañuelos, Alexander Ponomarev, Marlon Dumas, and Ingo Weber. *Optimized Execution of Business Processes on Blockchain.* In BPM'17: International Conference on Business Process Management, Barcelona, Spain, September 2017
6. An Binh Tran, Xiwei Xu, Ingo Weber, Mark Staples, and Paul Rimba. *Regerator: a registry generator for blockchain.* In CAiSE'17: International Conference on Advanced Information Systems Engineering, Forum Track (demo), June 2017.
7. Xiwei Xu, Cesare Pautasso, Liming Zhu, Vincent Gramoli, Alexander Ponomarev, An Binh Tran and Shiping Chen. *The blockchain as a software connector.* In: WICSA, Venice, Italy, April, 2016
8. Luke Anderson, Ralph Holz, Alexander Ponomarev, Paul Rimba, Ingo Weber. *New kids on the block: an analysis of modern blockchains.* http://arxiv.org/abs/1606.06530