
# Guest Lecture and Wrap-up

Mark Staples
Email: mark.staples@data61.csiro.au

www.data61.csiro.au

# Outline

- In-Class Time to Complete Course Survey: myExperience.unsw.edu.au
  - 5 mins, while lecturers briefly leave the room
- Guest Lecture – Bridie Ohlsson, Blockchain Lead @ AgriDigital
- Course Summary
- Other Topics
  - Standards and Interoperability
  - Governance, Regulation, and the Law
  - CryptoEconomics
  - Adoption, Disruptive Innovation, and New Business Models
  - Where to Next? – Possible Futures for Blockchain

Shape our Future

# Tell us about your experience and shape the future of education at UNSW.

**Click the myExperience link in Moodle**

**or login to myExperience.unsw.edu.au**

(use z1234567@ad.unsw.edu.au to login)

The survey is confidential, your identity will never be released

Survey results are not released to teaching staff until after your results are published

UNSW SYDNEY

# Guest Lecture: Bridie Ohlsson

- Blockchain Project Lead for AgriDigital
  - *Driving the blockchain innovation and implementation at AgriDigital, Bridie works across industry sectors to build communication channels and relationships with government, agribusinesses and the tech and startup ecosystem. Bridie manages AgriDigital's blockchain pilots, sitting at the juncture of industry and technology to develop transformational software solutions and solve embedded challenges across global supply chains.*
  - Contributing chapter “Blockchain Technology in the Trade and Finance of Agriculture Supply Chains” in the book *Architecture for Blockchain Applications*

# Course Summary

CSIRO DATA 61

# Course Summary 1/3

1. What is blockchain? Why does it matter?
   - Definitions
   - Role in Software Architecture
   - How it's different
2. Existing blockchain platforms
   - Example use cases
   - Cryptography basics
   - Nakamoto Consensus, PoW
   - Bitcoin
     - Script, UTXO
   - Ethereum
     - Solidity, Gas
   - Smart contract capabilities
3. Blockchain in Software Architecture
   - Taxonomy
   - Blockchain properties & limitations
   - Software Architecture basics
4. NFPs and Design Tradeoffs
   - What is Software Architecture
   - Non-Functional Properties
   - Views/Viewpoints; ATAM
   - UTXO vs Accounts
   - Arbitrary data storage

# Course Summary 2/3

5. Design Process & Cost
   - Is blockchain suitable?
   - Trade-offs between NFPs
   - Cost drivers for storage & computation
6. Performance
   - Load × Resources → Performance × Utilisation
   - Latency vs Throughput
   - Latency for Transaction Inclusion
   - Options for Latency & Throughput in Blockchain-Based Applications
7. Dependability & Security
   - Faults, Errors, Failures
   - Trust, Trustworthiness, Assurance
   - Functionality
     - Specification? Code is Law?
   - Integrity, Confidentiality, Privacy, Non-Repudiation, Accountability, Authenticity
   - Reliability, Availability

# Course Summary 3/3

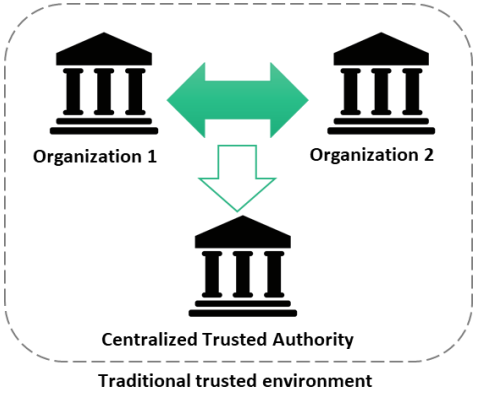
8. Design Patterns
   - Interaction with external world
   - Data management
   - Security
   - Contract structure
   - Deployment

9. Model-Driven Engineering
   - Models for Data & Tokens
   - Models for Business Process
   - Generation & Execution
   - Workflow on Blockchain

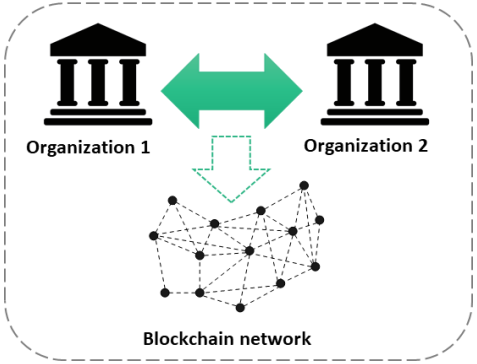
10. Summary
    - Some Key Points
    - Disruptive Potential

# What is blockchain? Why does it matter?

- You should know what these things are:
  - Blocks, Ledgers, Transactions
  - Nodes, Miners
  - Cryptocurrencies, Tokens, Digital Assets
  - Private vs Public Blockchains, vs. DLT
  - Smart Contracts
  - Oracles
  - dapps/ Decentralised Applications
  - Blockchain-Based Applications

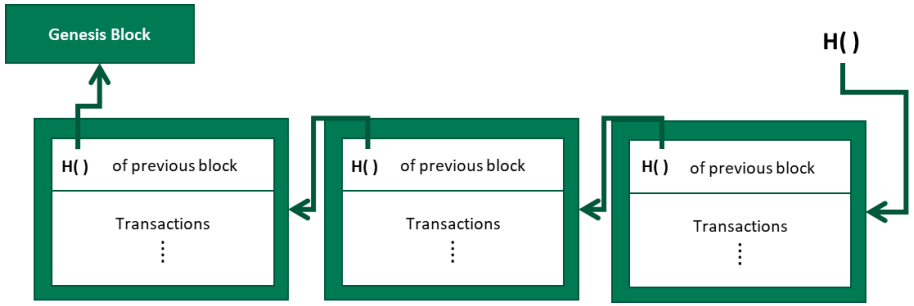


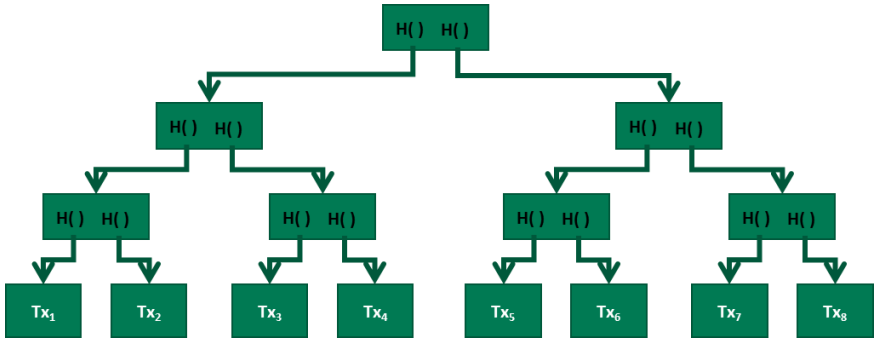
Traditional trusted environment

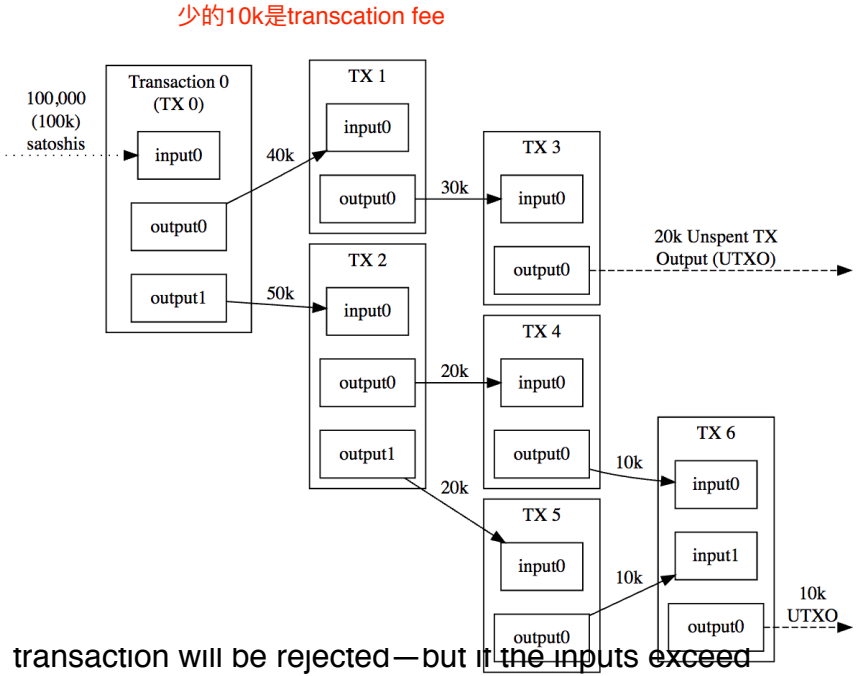


Blockchain trustless environment

# Bitcoin:
# Blocks-in-Ledger vs. Transactions-in-Block vs. UTXO

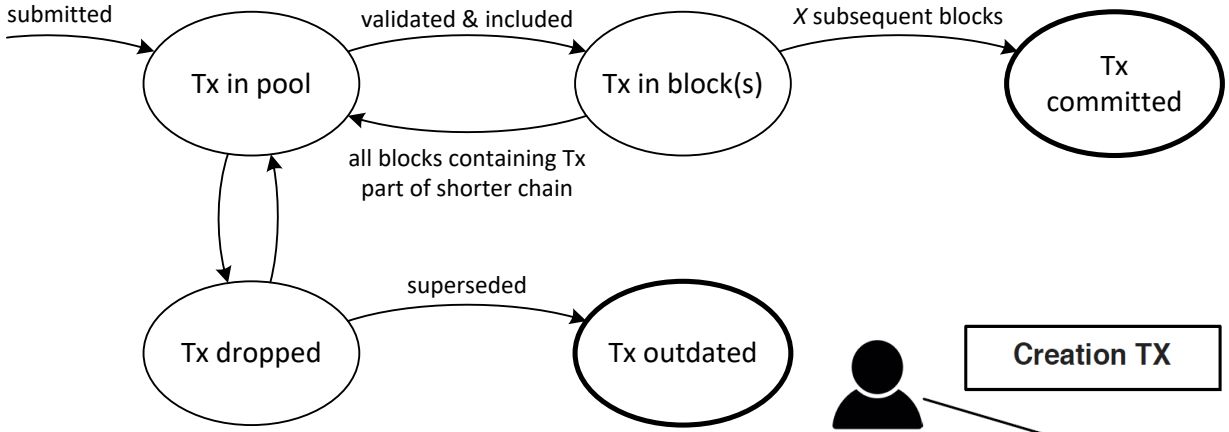


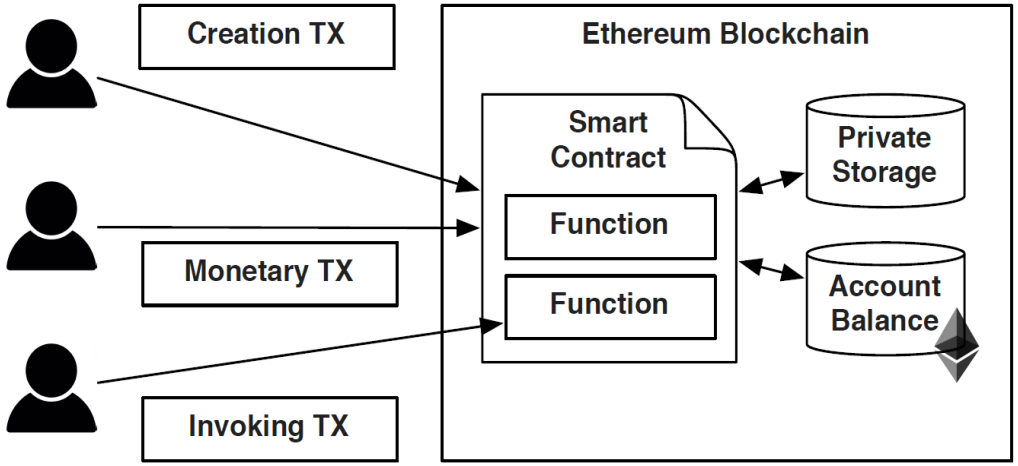




transaction will be rejected—but if the inputs exceed the value of the outputs, any difference in value may be claimed as a transaction fee by the Bitcoin miner who creates the block containing that transaction. For example, in the illustration above, each transaction spends 10,000 satoshis fewer than it rece[illegible]m its combined inputs, effectively paying a 10,00[illegible]toshi transaction fee

# Ethereum Transactions & Smart Contracts

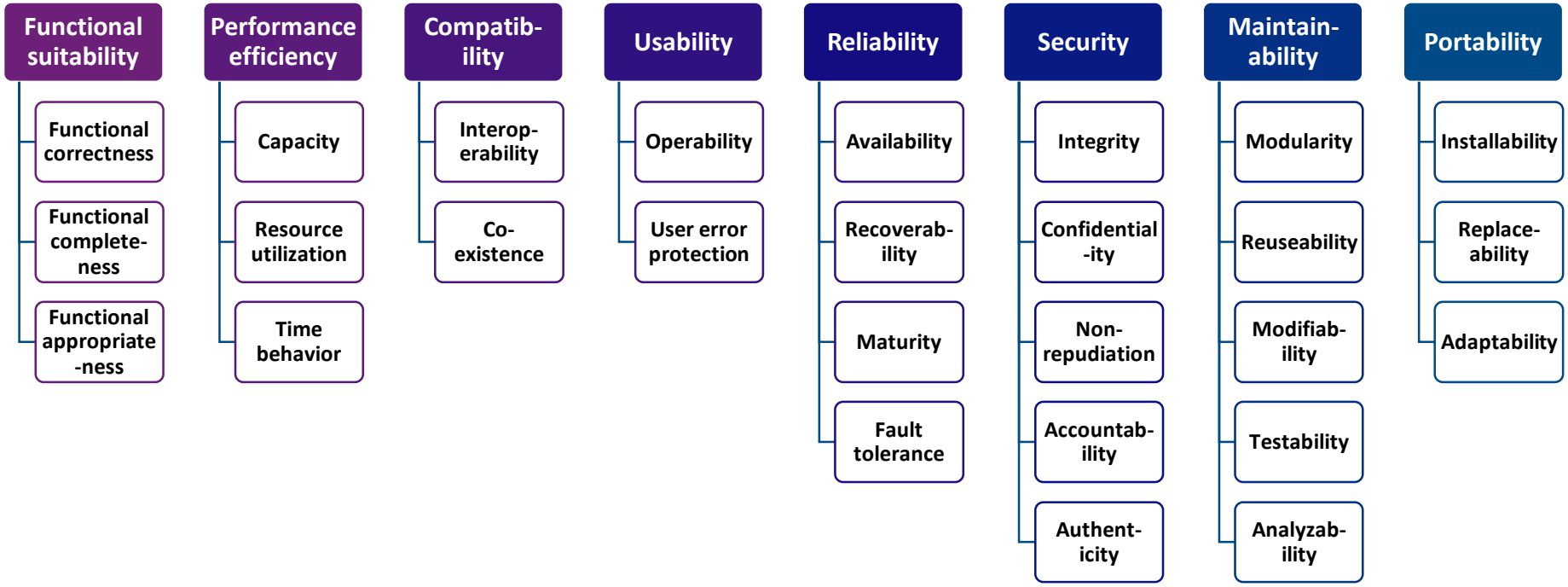


# Blockchain Taxonomy Dimensions

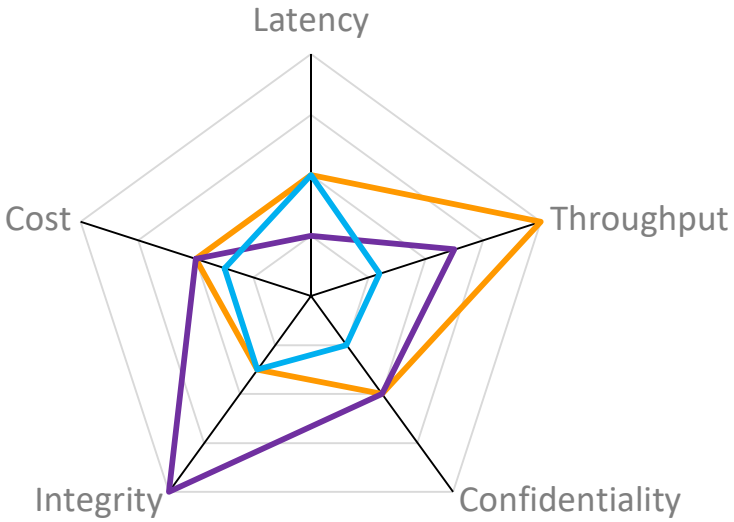
- (De)centralization
- Deployment
  - Public vs. Private
- Ledger Structure
  - List vs. DAGs vs. Networks of Ledgers
- Consensus Protocol
- Block Configuration and Data Structure
- Auxiliary Blockchain
  - Merged mining, sidechains, entangled, sharding,
- Anonymity
- Incentive

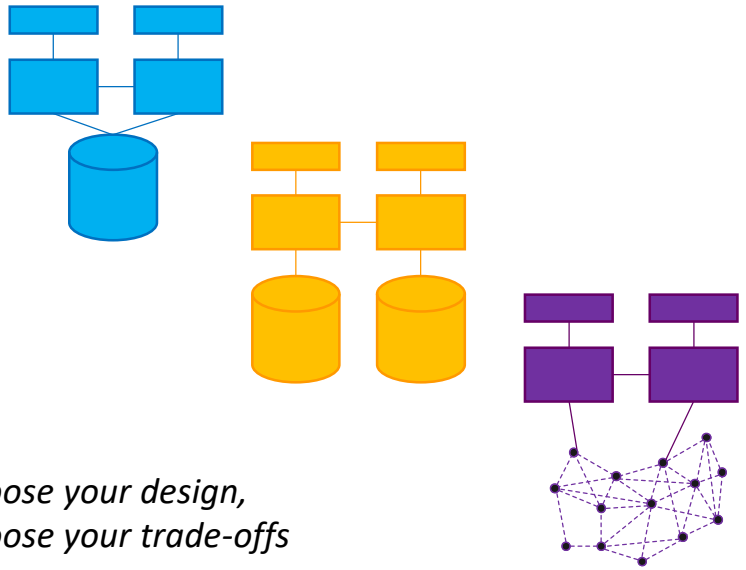
# ISO/IEC 25010:2011 Quality Model



# Software Architecture

Non-Functional Properties arise from Architectural Design Choices

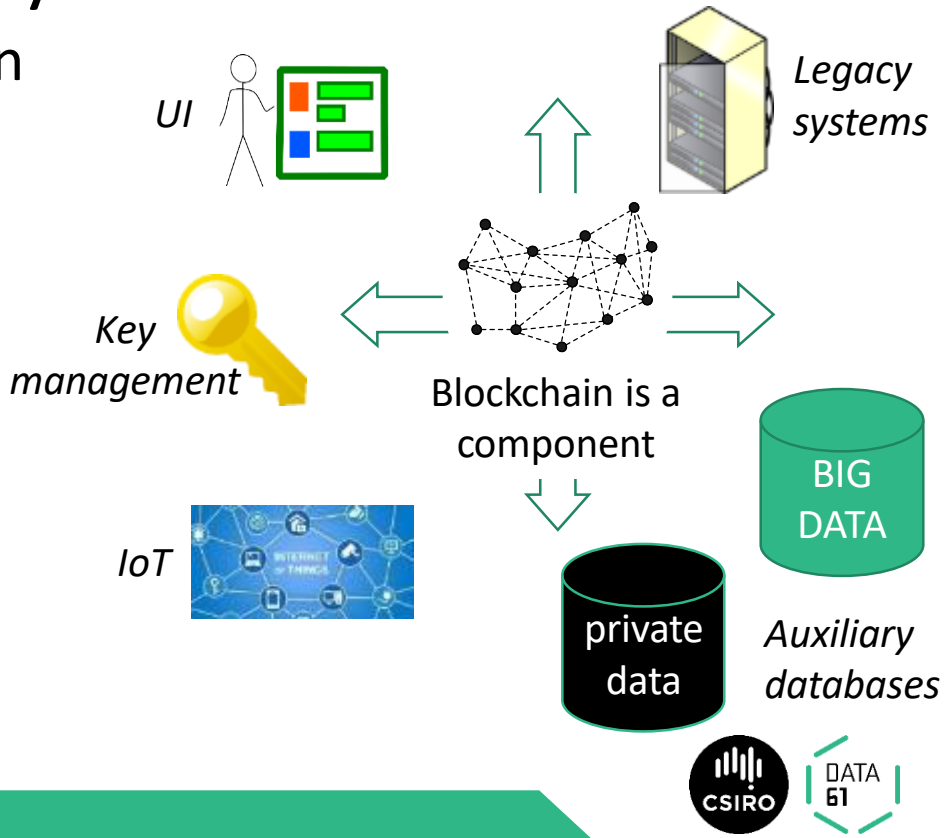


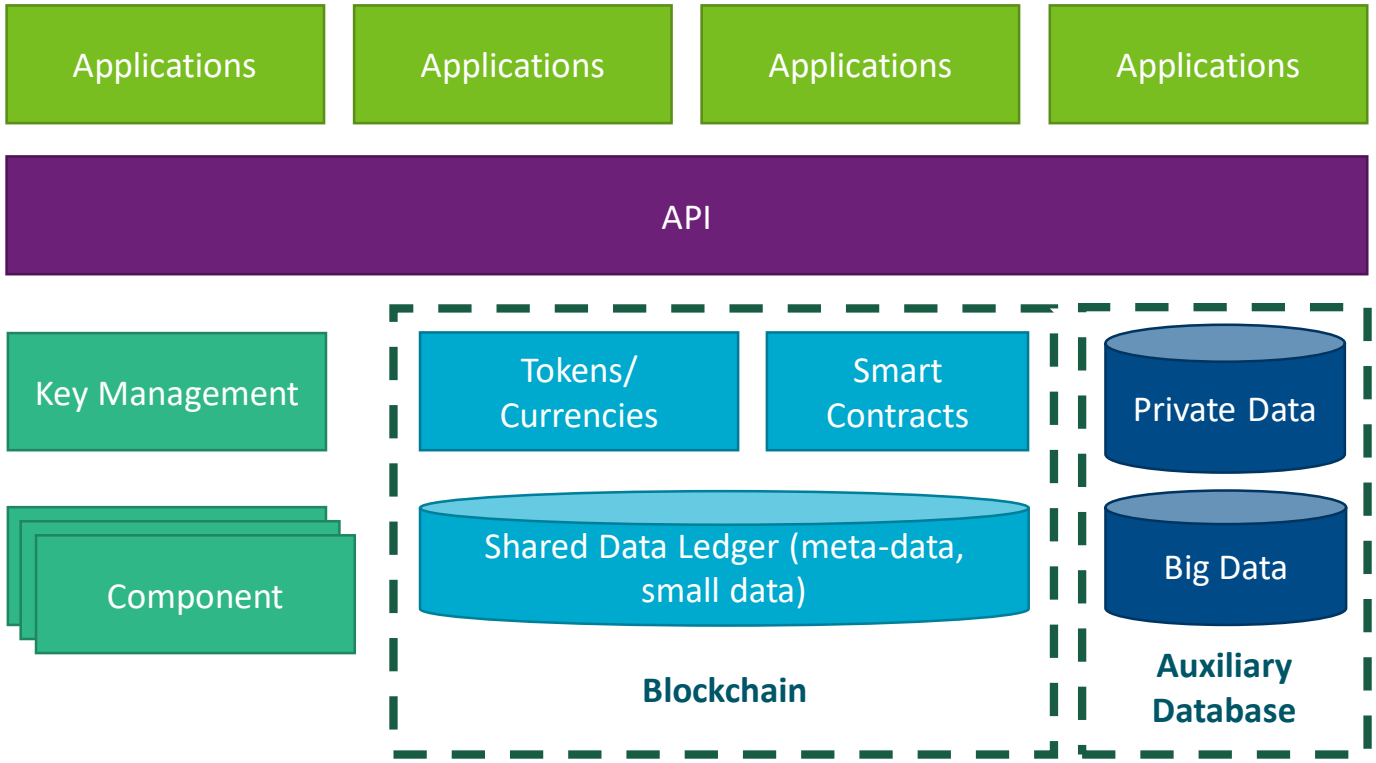
*Choose your design,*
*Choose your trade-offs*

# Not Stand-Alone; Design to Resolve Tradeoffs

- Non-Functional Property Trade-offs
  - (+) Integrity, Non-repudiation
  - (-) Confidentiality, Privacy
  - (-) Modifiability
  - (-) Throughput/ Scalability/ Big Data
  - (+ read/ - write) Availability/ Latency



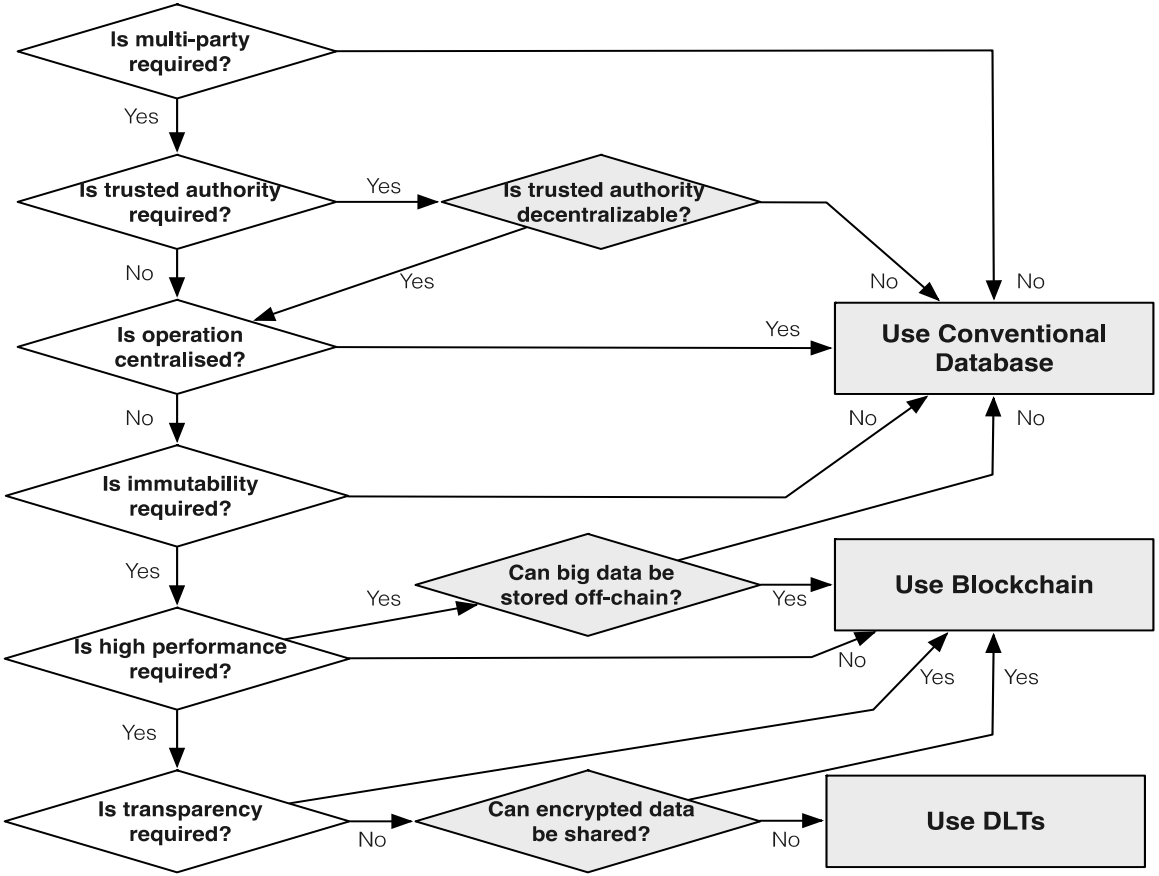
# Blockchain in Larger Software System



# Blockchain As An Architectural Element

- Storage
  - Shared append-only store of transaction data
- Computation
  - Smart contracts as little programs
- Communication
- Asset Management and Control

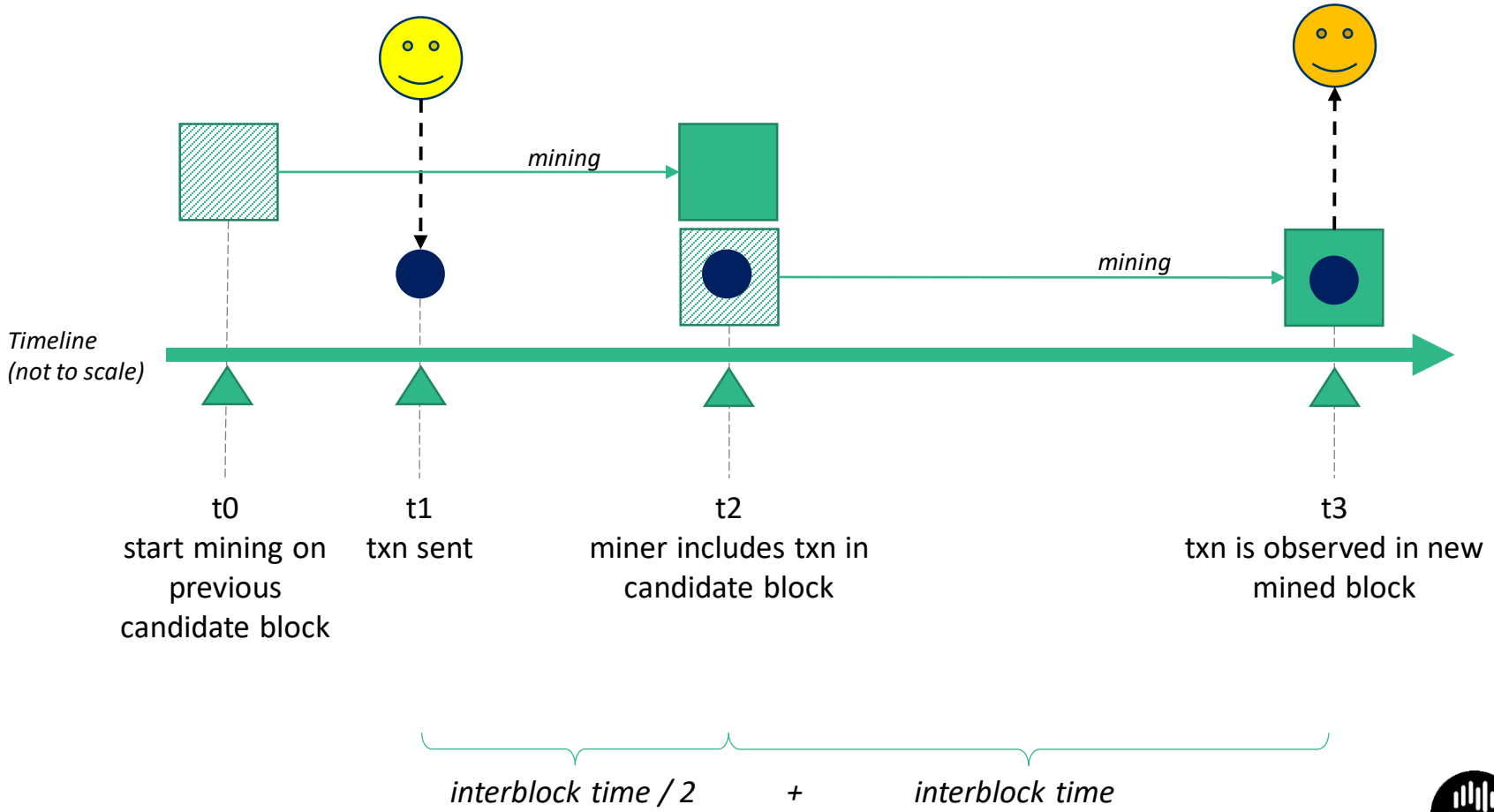
# Should You Use a Blockchain?



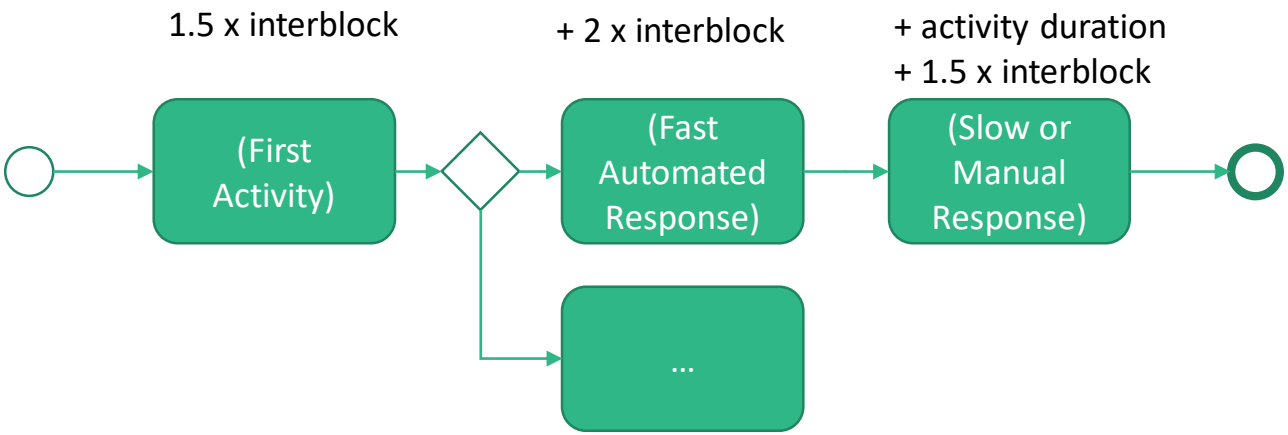
# Cost

- Cost of basic compute and storage on public blockchain have different cost structure than conventional cloud
  - One-off costs vs ongoing costs
  - But, public blockchain overall is orders of magnitude more expensive
- For public blockchain, cost model should include exchange rate
  - Highly volatility cryptocurrency exchange rates
- Cost tradeoffs with other non-functional properties
  - Maintainability and Scalability
  - Price of avoiding centralize control of data and computation

# Transaction Inclusion Time in Public Blockchain…



# Latency of Process Execution on Public Blockchain




Here, the design does not use confirmation blocks.

Quick & dirty!
Assumes averages (median or mean)!

- Transactions arrive in the middle of an interblock time window
- Uniform average interblock time

The first activity will arrive in “the middle” of some interblock time window, so takes 1.5 x interblock times.

If activity response is very fast (in the same interblock time window), it will still be too late to include in the next block, so will take 2 x interblock times.

If the activity response is very slow (longer than the interblock time window), it will arrive in “the middle” of some future interblock time window

# In Public Blockchains, Also Wait for Confirmation Blocks!?

- Nakamoto consensus can create “uncle” blocks
  - Blockchains using Proof-of-Work, Proof-of-Stake, …
  - Transaction you saw in a block turns out not to have been officially included in the blockchain
  - Only probabilistic, long-run, transaction inclusion

- Can increase confidence your transaction is really included, by waiting for subsequent “confirmation blocks”
  - For Bitcoin, often people say wait 6 blocks
  - For Ethereum, often people say wait 12 blocks
    - (Lower interblock time can increase likelihood of uncles)

- Waiting for confirmation blocks increases latency, and increases latency variability

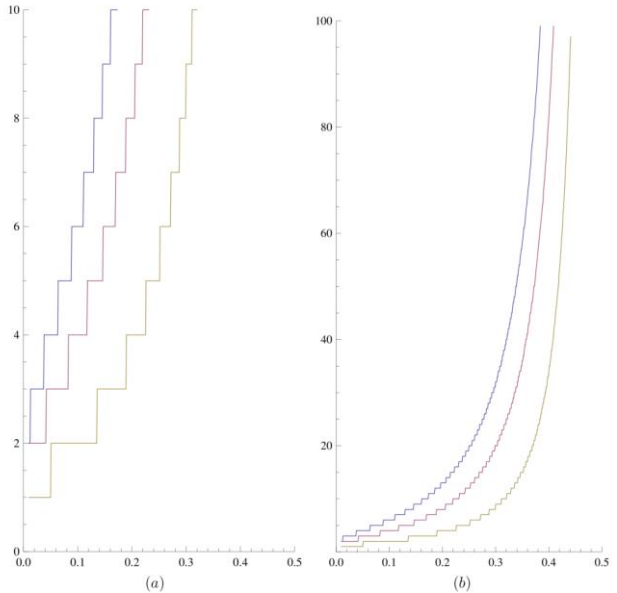



Figure 5: The number of confirmations required to keep the probability of success low, as a function of the attacker's hashrate, for various values of the target probability: 10% (yellow), 1% (purple) and 0.1% (blue). The graph is shown in two different vertical scales.

“Analysis of hashrate-based double spending”
https://arxiv.org/abs/1402.2009

# Blockchains as Dependable Systems

- Trust – accepted dependence
- Trustworthy – justified Trust
- Need Evidence to justify Engineering Assurances

- Validation
  - Sort-of-but-not-quite: “am I solving the right problem?”
  - Does my requirements/specification/system/design address/solve the user problem?
- Verification
  - Sort-of-but-not-quite: “am I solving the problem right?”
  - Does my system/design meet the requirements specification?

- Can provide evidence based on architectural analysis, early in the lifecycle
- Blockchains are increasingly relied on; need evidence they are trustworthy

# ISO/IEC 25010:2011 Dependability & Security

- Functional Suitability
  - Correctness
  - Completeness
  - Appropriateness
    - Code is Not Law
    - How to specify smart contracts?
- Security
  - Integrity & Clark-Wilson policy model
    - Blockchain “anomaly” in Nakamoto consensus: possible transaction reordering
  - Confidentiality
    - And its difference to Privacy
  - Non-Repudiation
  - Accountability
    - Avoid or support? KYC/AML-CTF
  - Authenticity
- Reliability
  - Availability: readiness for service
    - Affected by latency variability
  - Recoverability
    - Aborting transactions
  - “Maturity”
    - Reliability: continuous service
  - Fault Tolerance
    - Use of smart contracts for redundant oracles

# Design Patterns 1/2

## Interaction with External World

### Centralized Oracle
- Introducing external state into the blockchain environment through a centralized oracle

### Decentralized Oracles
- Introducing external state into the blockchain environment through decentralized oracles

### Voting
- A method for a group of blockchain users to make a collective decision

### Reverse Oracle
- A method for a group of blockchain users to make a collective decision

### Legal and smart contract pair
- A bidirectional binding between a legal agreement and the corresponding smart contract that codifies the legal agreement

## Data Management

### Encrypting On-chain Data
- Ensuring confidentiality of the data stored on blockchain by encrypting it

### Tokenisation
- Using tokens on blockchain to represent transferable digital or physical assets or services

### Off-chain Data Storage
- Using hashing to ensure the integrity of arbitrarily large datasets which may not fit directly on the blockchain

### State Channel
- Transactions that are too small in value or that require much shorter latency, are performed off-chain with periodic recording of net transaction settlements on-chain

## Security

### Multiple Authorization
- Transactions are required to be authorized by a subset of the pre-defined addresses

### Dynamic authorization
- Using a hash created off-chain to dynamically bind authority for a transaction

### X-Confirmation
- Waiting for enough number of blocks as confirmation to ensure that a transaction added into blockchain is immutable with high probability

### Security Deposit
- A deposit from a user, which will be paid back to the user for her honesty or given to others to compensate them for the dishonesty of the user

# Design Patterns 2/2

## Structural Patterns of Contract

### Contract Registry

- The address, and the version of the smart contract is stored in a contract registry

### Embedded Permission

- Embedded permission control is used to restrict access to the invocation of the functions defined in the smart contracts

### Data Contract

- Storing data in a separate smart contract

### Factory Contract

- An on-chain template contract used as a factory that generates contract instances from the template

### Incentive Execution

- A reward to the caller of a contract function for invocation

## Deployment

### dapp

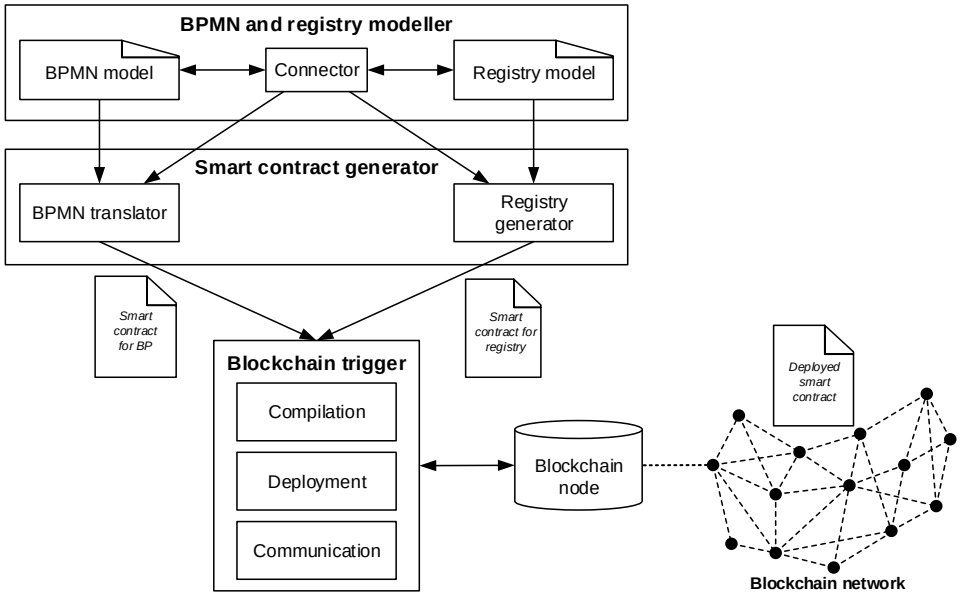
- Blockchain-based application hosted on P2P network, with a website that allows users to interact with smart contracts

### Semi-dapp

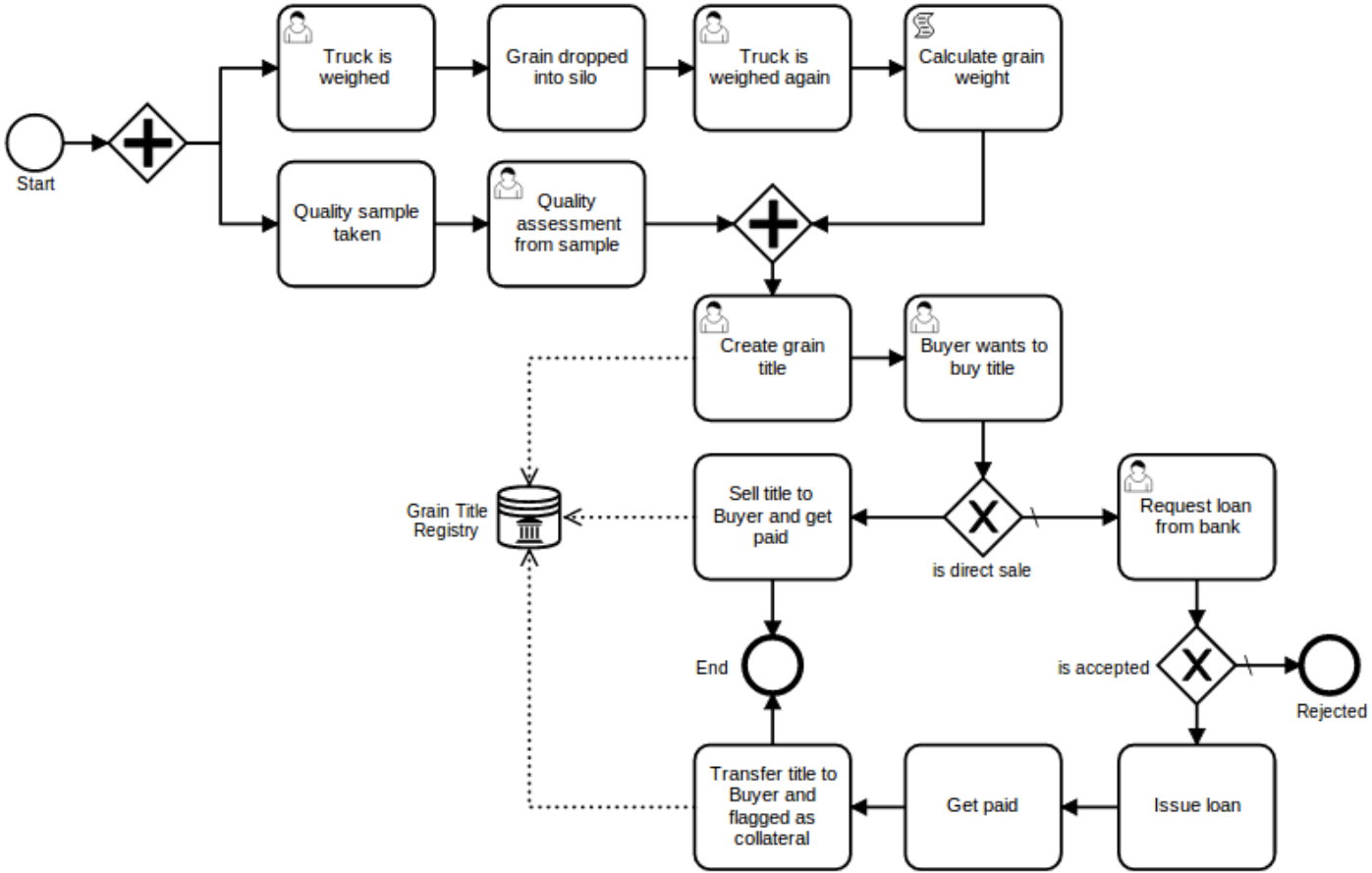
- Blockchain-based application with a website that can be browsed using a conventional web browser without any dapp plugin

# Model-Driven Blockchain-Based Applications

- Code generation for best practices and well-tested building blocks
- Improved productivity, especially for novices
- Models are often easier to understand than code
- Can be independent of specific blockchain technologies or platforms
  - Avoid lock-in to specific blockchain technologies



# Combining process and data/token models



# Other Topics

CSIRO DATA 61

# Other Topics

- Standards
- Interoperability
- Governance
- Regulation, and the Law
- CryptoEconomics
- Adoption, Disruption, and Business Models

# Standards

- ISO/TC 307 Blockchain and distributed ledger technologies
  - https://www.iso.org/committee/6266604.html
  - Led by Standards Australia
- IEEE
  - https://blockchain.ieee.org/standards
- ITU
  - Focus Group on DLT
    - https://www.itu.int/en/ITU-T/focusgroups/dlt/Pages/default.aspx
  - SG17/Q14 Security aspects for DLT
    - https://www.itu.int/en/ITU-T/studygroups/2017-2020/17/Pages/q14.aspx

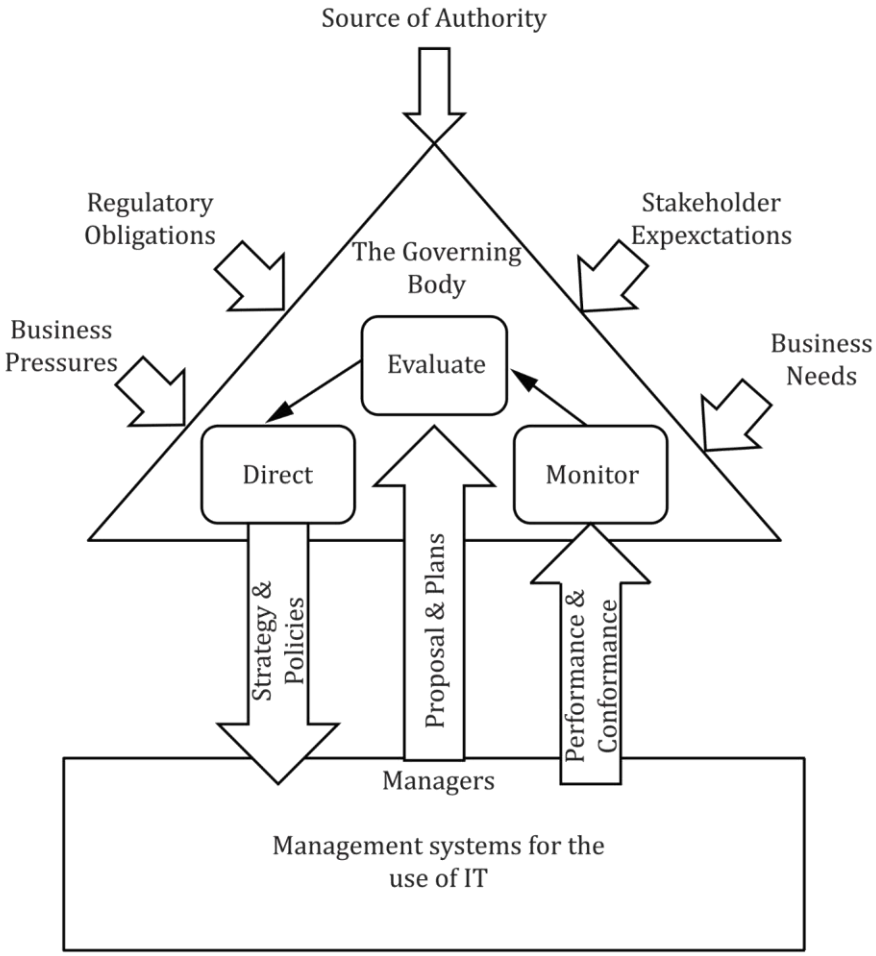
- Many standards in development
- Still very early days, but not a good reason to not trial/use blockchain

# Interoperability

- There won't be one blockchain to rule them all!
  - Different use cases have different requirements; need different infrastructure
  - Currently there is a proliferation of many blockchain technologies & blockchains
- How do we integrate different blockchains?
  - Different consensus mechanisms
  - Different digital asset representations
  - Different governance
- A major motivator for international standards, but we don't yet know how!

# Blockchain Governance

- How to define, control, manage, and evolve blockchain systems?
- Traditional (e.g. ISO 38500) governance assumes a single "Source of Authority"
  - Not always the case for blockchain/DLT systems
- Need new approach for blockchain and DLT
  - ISO TC307 WG5 – Governance
  - also a research challenge!



# Regulation, and the Law

- Regulation supports safe, humane, and efficient society & economy
- But, regulation has risks, costs and inefficiencies of its own!

- What risks does blockchain pose for economy and society?
- What controls (if any) should there be on blockchain technologies?

- How to define laws for blockchain?
  - What is a blockchain “token” in legal terms?
  - Can a blockchain be a legally-recognised register of title in assets?
  - Can a smart contract really be a legal contract, or not?
  - Can we use/adapt existing law, or do we need new laws?

# What Are Australian Regulators Doing?

- Active interest, good understanding, early guidance & regulation, e.g. …
- ATO treatment: cryptocurrency as assets
  - https://www.ato.gov.au/General/Gen/Tax-treatment-of-crypto-currencies-in-Australia---specifically-bitcoin/
- AUSTRAC treatment: AML/CTF & KYC for cryptocurrency exchanges
  - http://www.austrac.gov.au/digital-currency-exchange-providers
- ASIC guidance on DLTs and ICOs
  - https://asic.gov.au/regulatory-resources/digital-transformation/evaluating-distributed-ledger-technology/
  - https://asic.gov.au/regulatory-resources/digital-transformation/initial-coin-offerings/
- RBA remarks on eAUD, CBDCs
  - https://www.rba.gov.au/speeches/2017/sp-gov-2017-12-13.html
  - https://www.rba.gov.au/speeches/2018/sp-so-2018-06-26.html

# CryptoEconomics

- Cryptocurrency – Is it money? How should it be valued?
- Why does blockchain create economic (in)efficiencies?
- How should decentralised marketplaces work?
- How are they different to normal digital marketplaces?
- What would the consequences be of “programmable money”?
- What are the systemic financial infrastructure risks for blockchain?
  - e.g. Key management by individuals introduces a new kinds of risks

# Adoption, Disruption, and Business Models

- Why has adoption been so slow?  OR?  Why so fast?!
- 2019 is seeing emergence of production blockchains
  - TradeLens, FoodTrust, Marco Polo, ASX, …

- What are the main barriers to adoption?
- How to create new businesses or industries on blockchain?

- Data61/Treasury 2017 report
  - https://www.data61.csiro.au/~/media/D61/Files/Blockchain-reports/Blockchain-Scenarios-PDF.pdf
- Data61/ACS 2019 report
  - https://www.acs.org.au/content/dam/acs/acs-publications/ACS-Data61-Blockchain-2030-Report.pdf

# Summary

# Now you can…

- explain the principles of blockchain and which roles it can play in an application architecture
- decide the suitability of blockchains and how to design applications on them
- make functional and non-functional trade-offs for blockchain-based applications
- build small applications on blockchain


# THANK YOU

www.data61.csiro.au