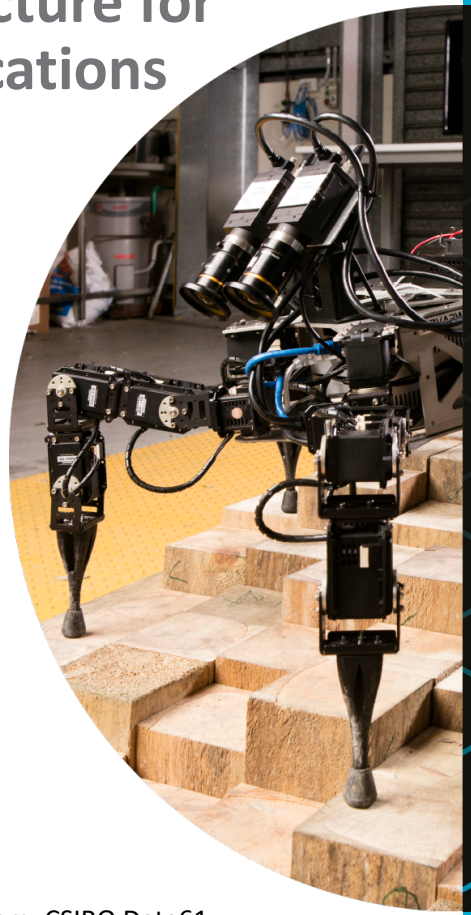
COMP6452
Software Architecture for Blockchain Applications



Summary & Wrap-up

Xiwei (Sherry) Xu

| Senior Research Scientist @ AAP team, CSIRO Data61
| Xiwei.Xu@data61.csiro.au

Outline

- Course Summary
- Other Topics
  - Standards
  - Interoperability
  - Governance, Regulation, and the Law

Course Summary

Course Summary 1/3

1. What is blockchain? Why does it matter?
   - Blockchain basics
   - Blockchain applications
   - Problem definition

2. Existing blockchain platforms
   - Cryptography basics
   - Smart contract basics
   - Bitcoin
     - Script, UTXO
   - Ethereum
     - Solidity, Gas
   - Hyperledger Fabric
     - Channel

3. Blockchain in Software Architecture
   - Software Architecture basics
     - What is Software Architecture
     - Non-Functional Properties
     - NFPs and Design Tradeoffs
     - Views/Viewpoints
   - Blockchain properties & limitations
   - Blockchain role in a larger software system

4. Taxonomy
   - Varieties of blockchain
   - Blockchain configuration

Course Summary 2/3

5. Design Process
   - Is blockchain suitable?
   - Trade-offs between NFPs
     - ATAM
   - Cost drivers for storage & computation
6. Blockchain Patterns
   - Interaction with external world
   - Data management
   - Security
   - Contract structure
   - Deployment
   - More smart contract patterns
   - Blockchain migration patterns
8. Performance
   - Latency vs throughput
   - Latency for transaction inclusion
9. Security & Reliability
   - Faults, Errors, Failures
   - Trust, Trustworthiness, Assurance
   - Functional Suitability
   - Integrity, Confidentiality, Privacy, Non-Repudiation, Accountability, Authenticity
   - Reliability, Availability

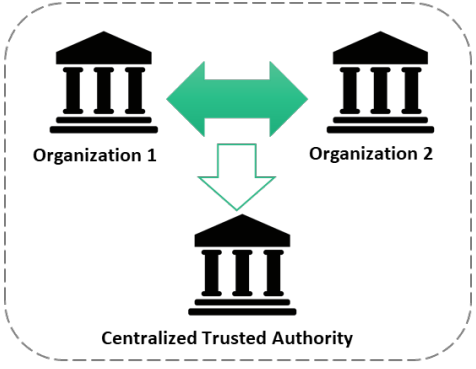
Course Summary 3/3

10. Smart Contract Testing
    - Types of testing
    - Known smart contract vulnerabilities
    - Tools and techniques
11. Model-Driven Engineering
    - Models for Data & Tokens
    - Models for Business Process
    - Generation & Execution
12. Data Management
    - Data Storage Architecture
    - Data Privacy
    - Data Quality
13. Cost
    - Data storage cost
    - Smart contract deployment cost
    - Smart contract execution cost

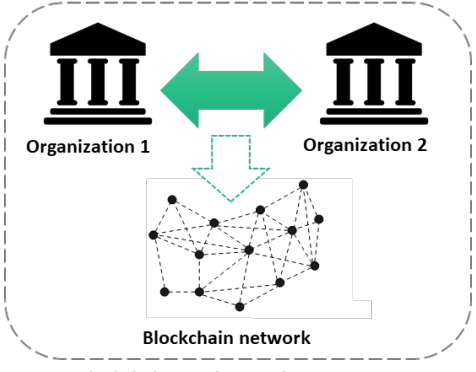
What is blockchain? Why it matters?

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

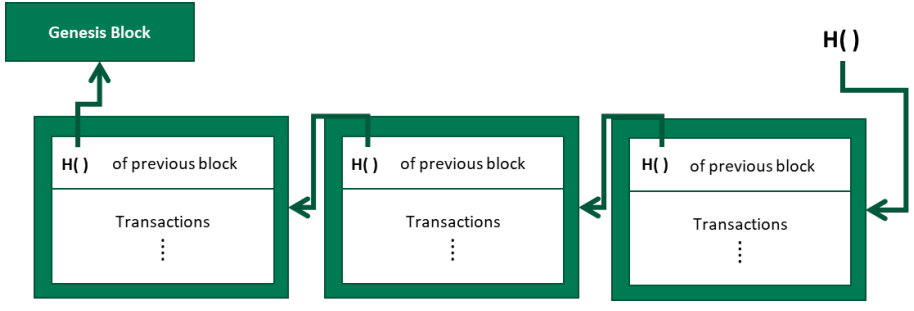
Properties of Blockchain

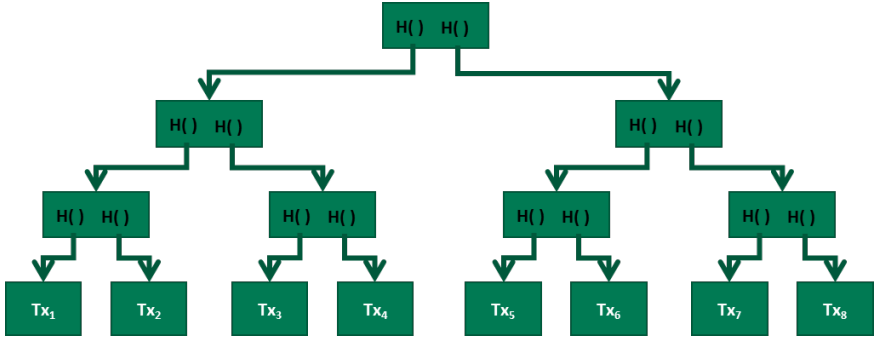
- Blockchain cannot meet requirements for all usage scenarios
  - E.g. those that require real-time processing
- Fundamental Properties
  - Immutability *from committed transaction*
  - Integrity *from cryptographic tool*
  - Transparency *from public access*
  - Equal rights *from consensus*
- Limitation
  - Data privacy
  - Scalability
    - Size of the data on blockchain
    - Transaction processing rate
    - Latency of data transmission

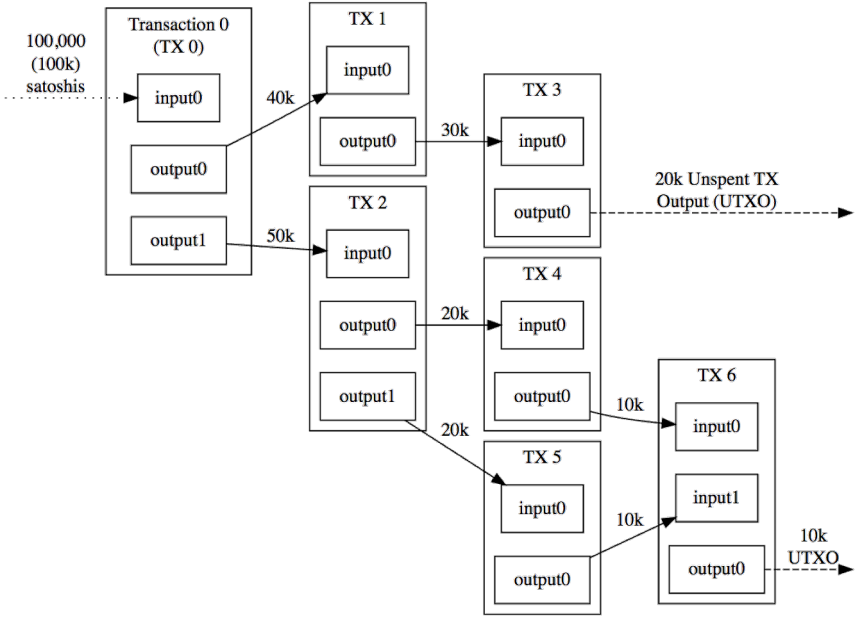
Blockchain platforms

- Bitcoin
- Ethereum
- Hyperledger fabric

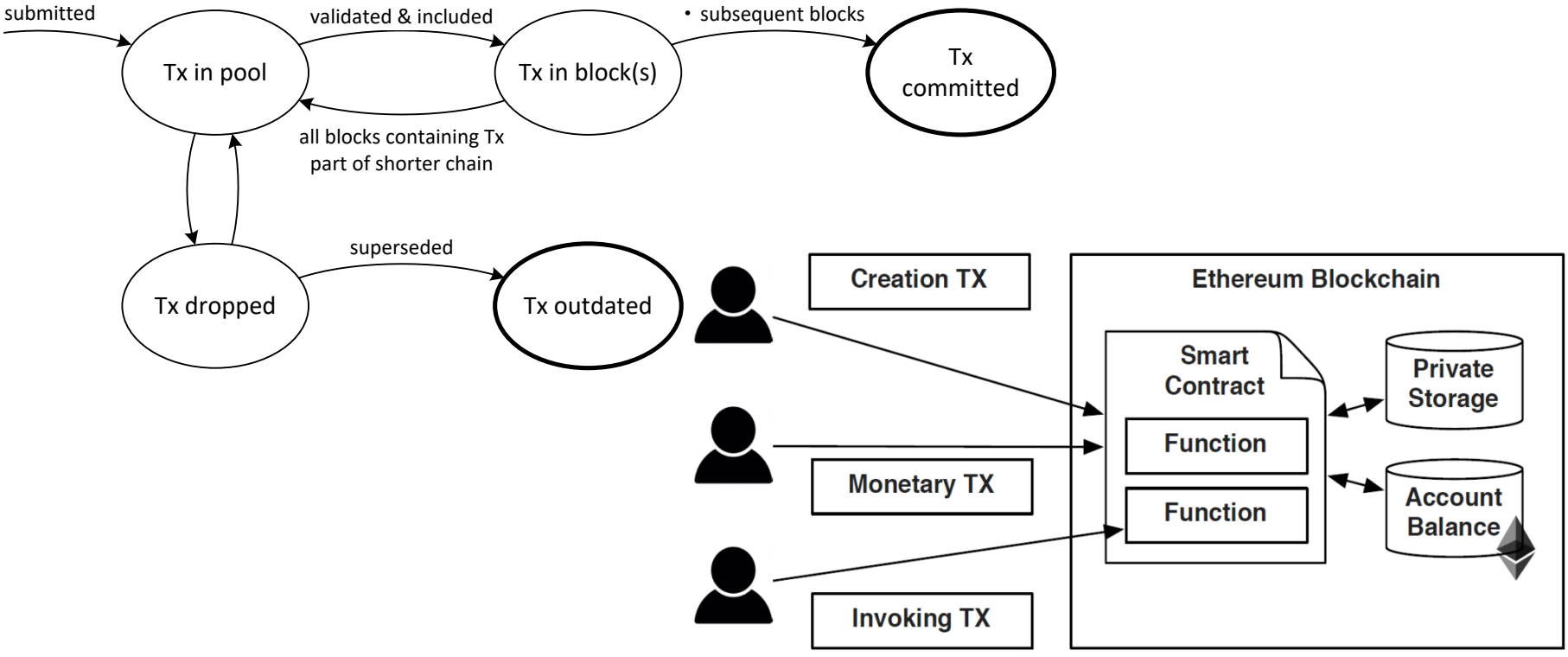
Bitcoin:
Blocks-in-Ledger vs. Transactions-in-Block vs. UTXO



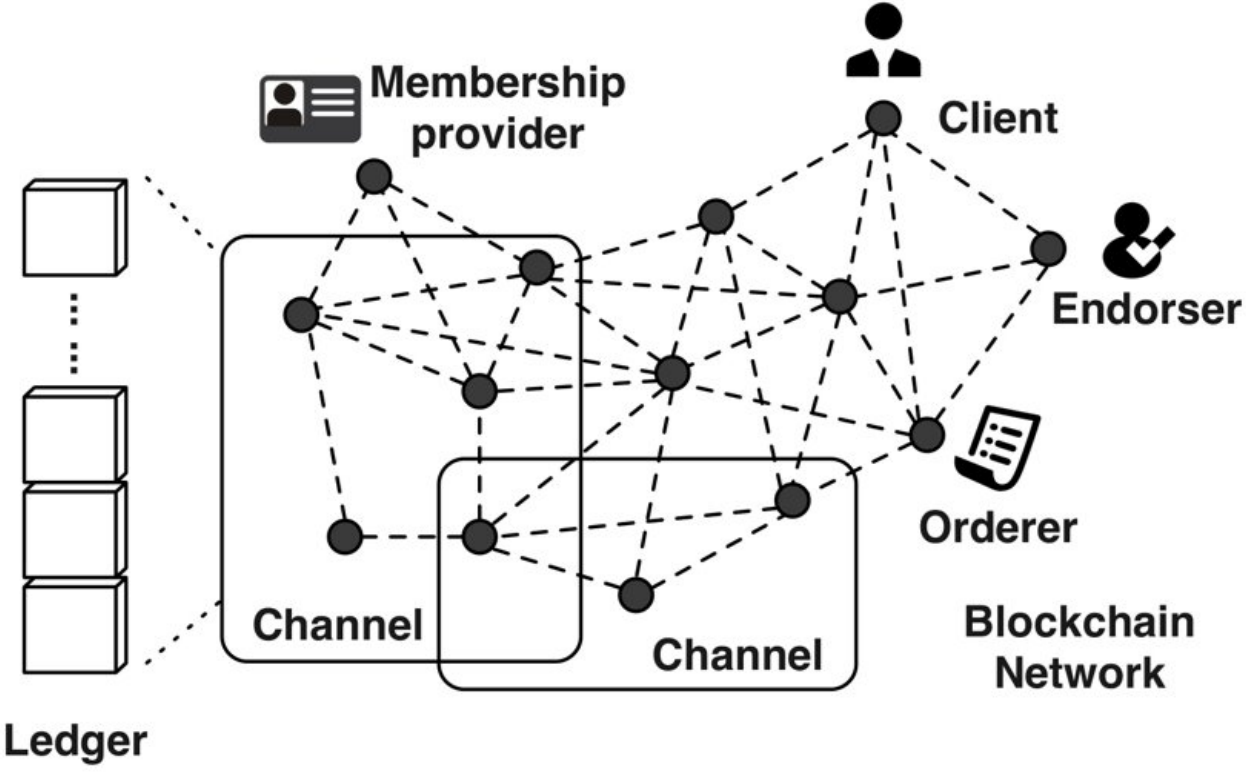




Ethereum Transactions & Smart Contracts

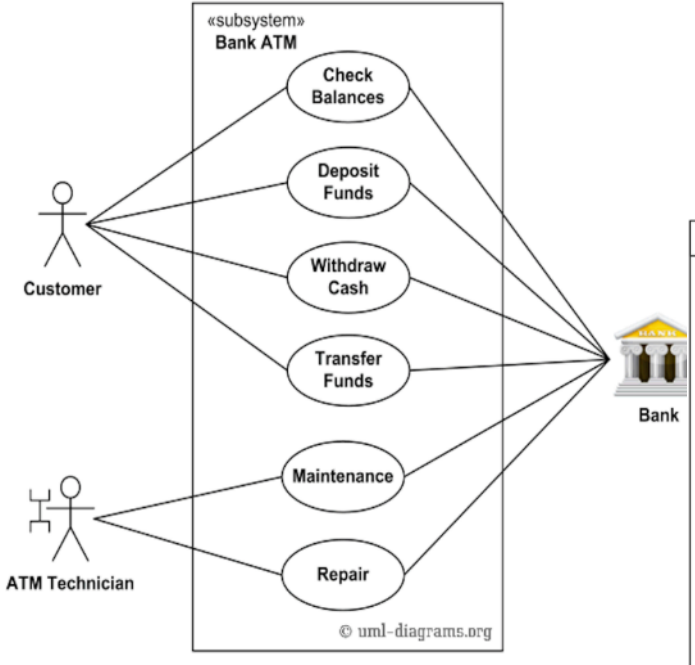


Hyperledger Fabric



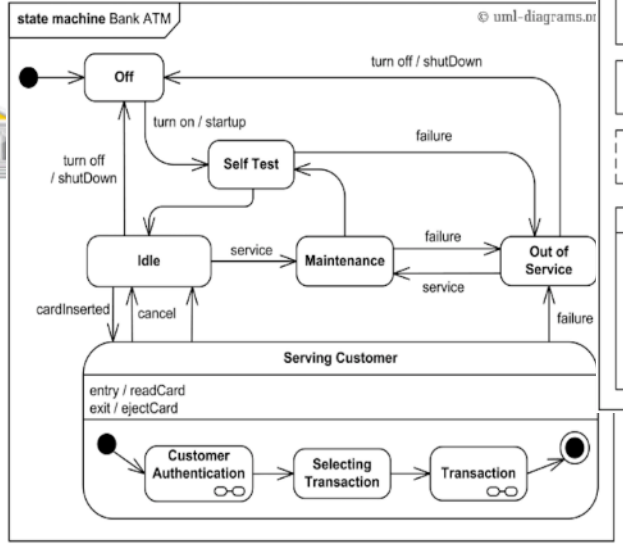
Blockchain in Software Architecture

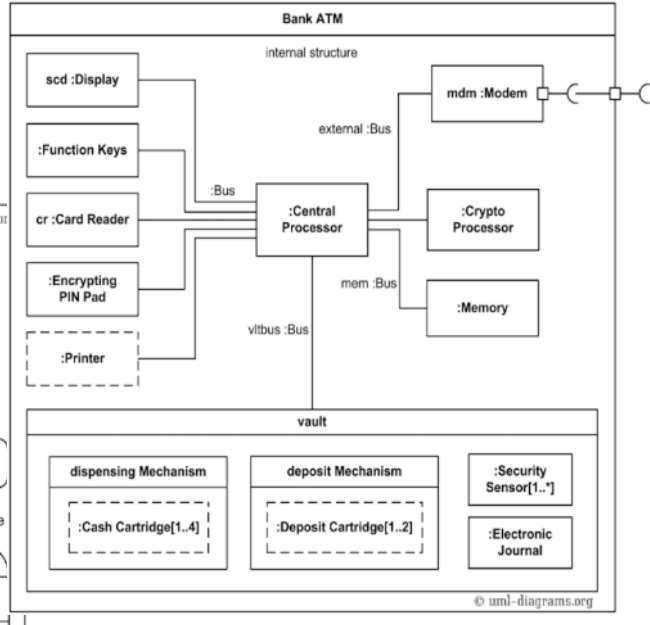
UML as Viewpoints and Views



UML Use Case

UML State chart

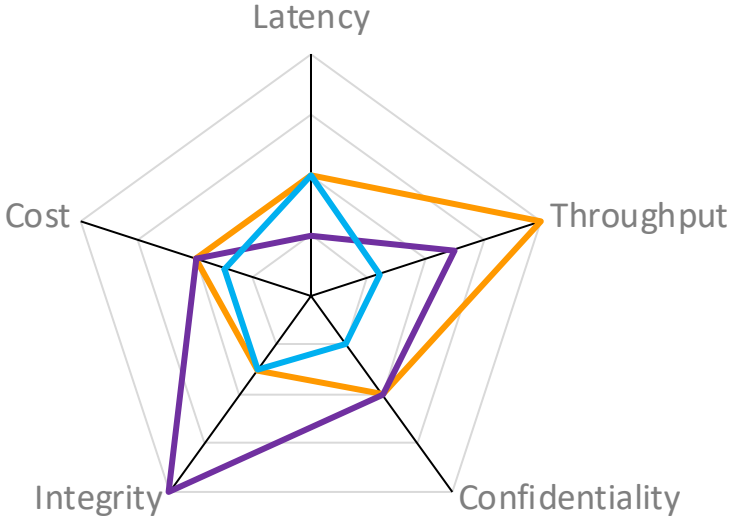


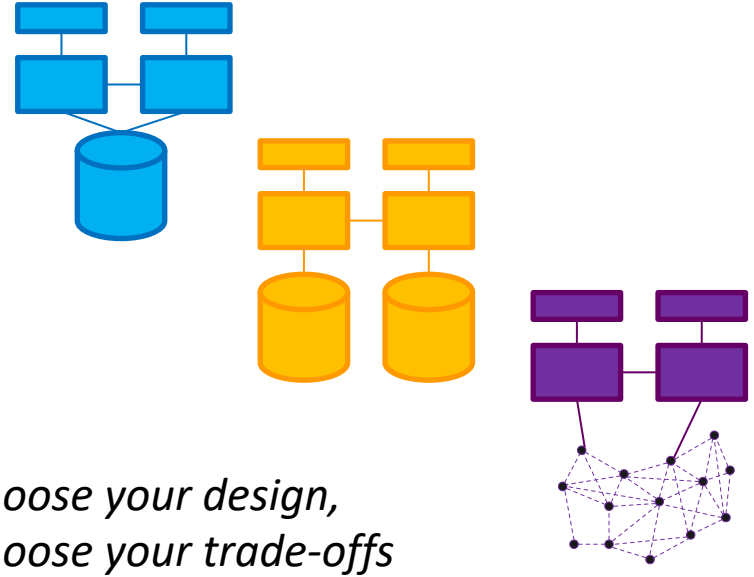


UML Composite Structure

Software Architecture

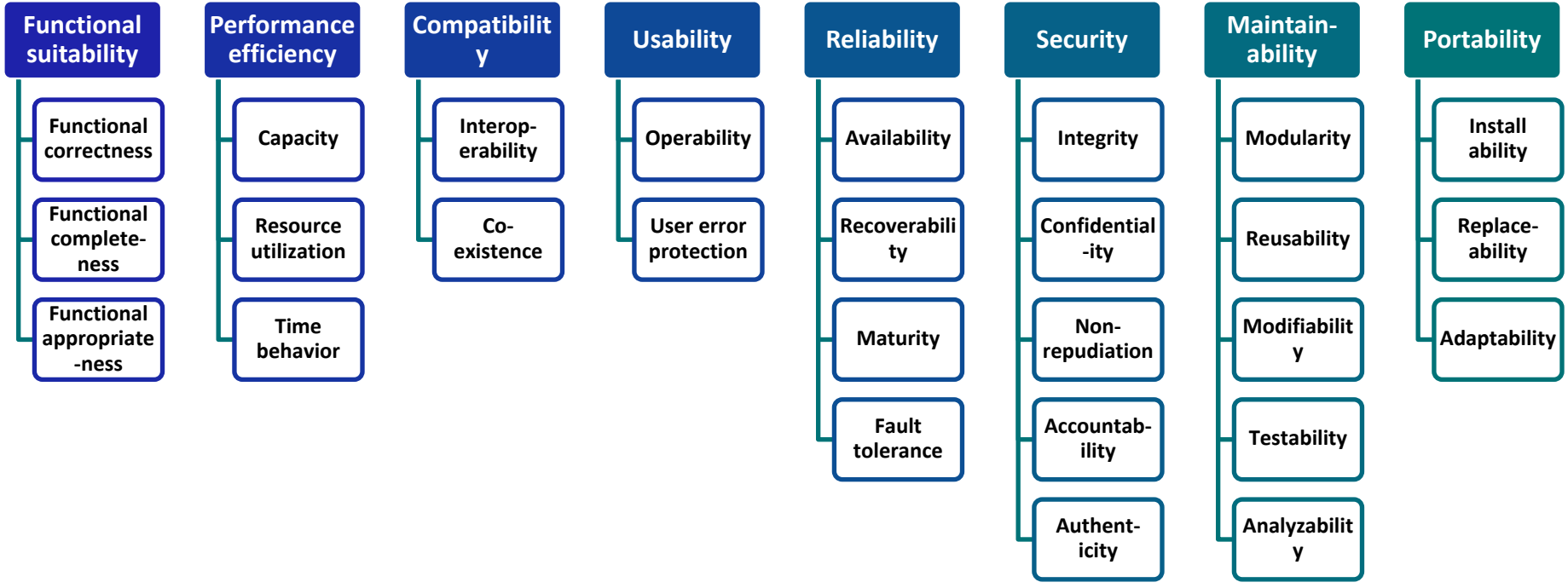
Non-Functional Properties arise from Architectural Design Choices





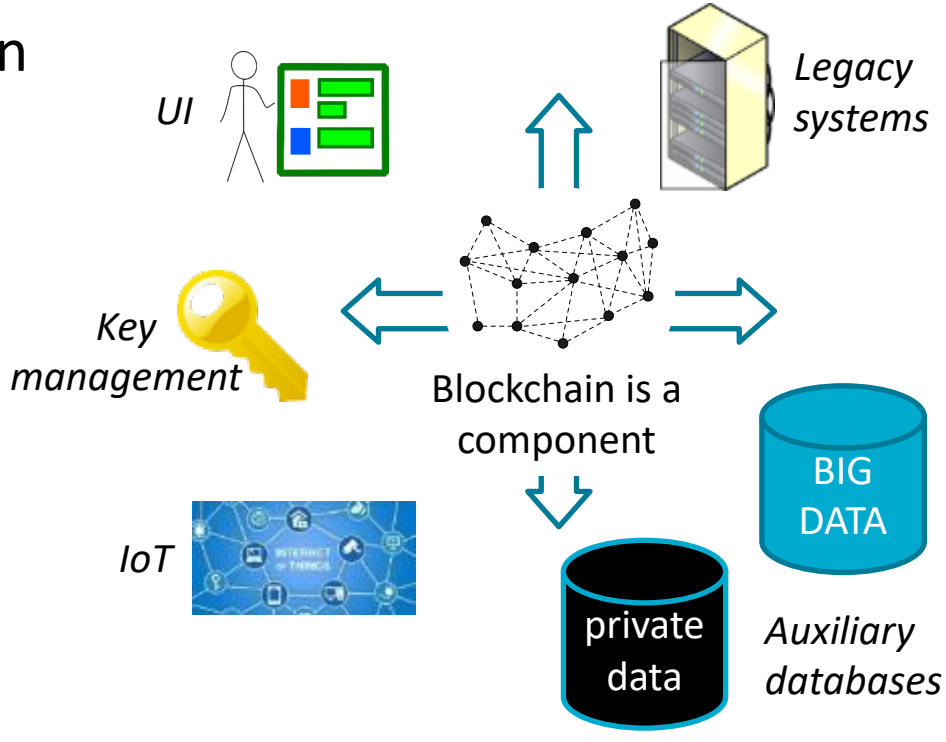
Choose your design,
Choose your trade-offs

ISO/IEC 25010:2011 Quality Model

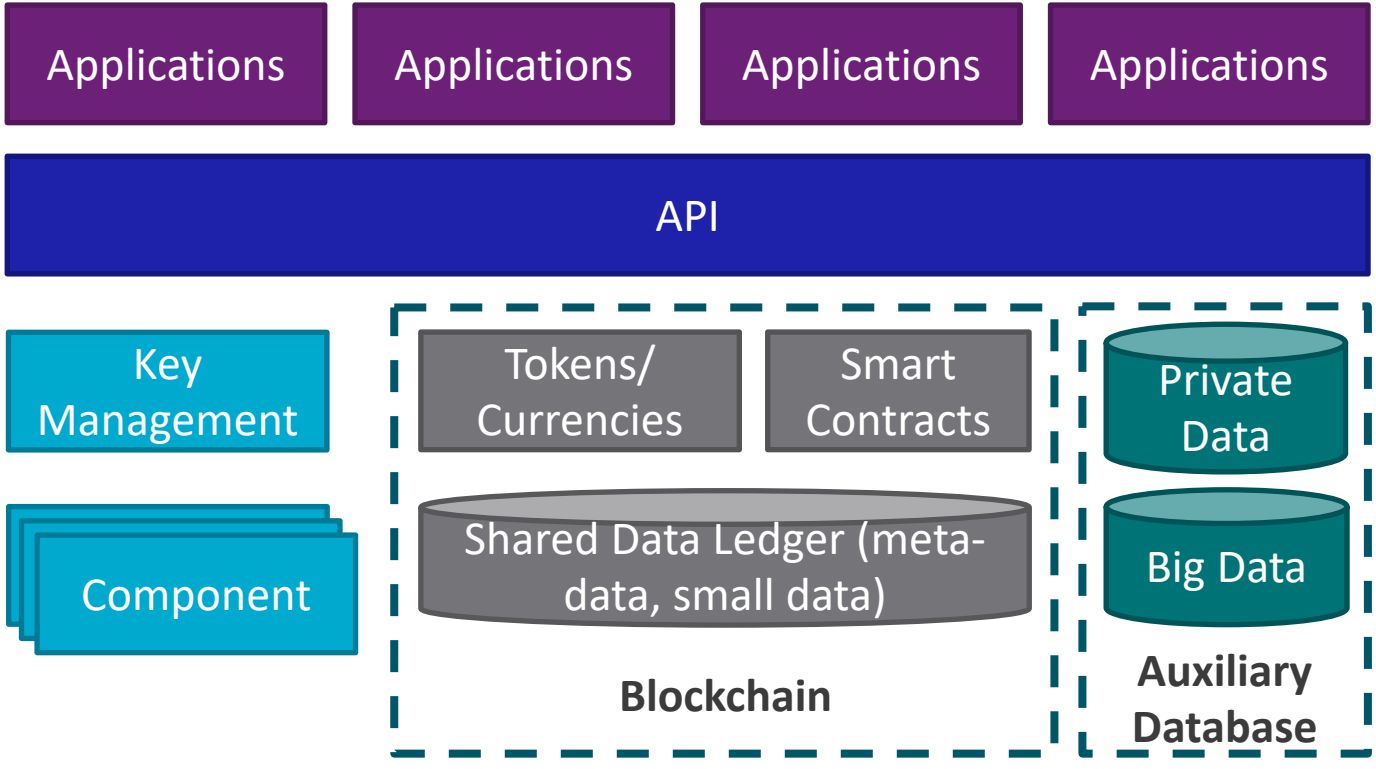


Not Stand-Alone; Design to Resolve Tradeoffs

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

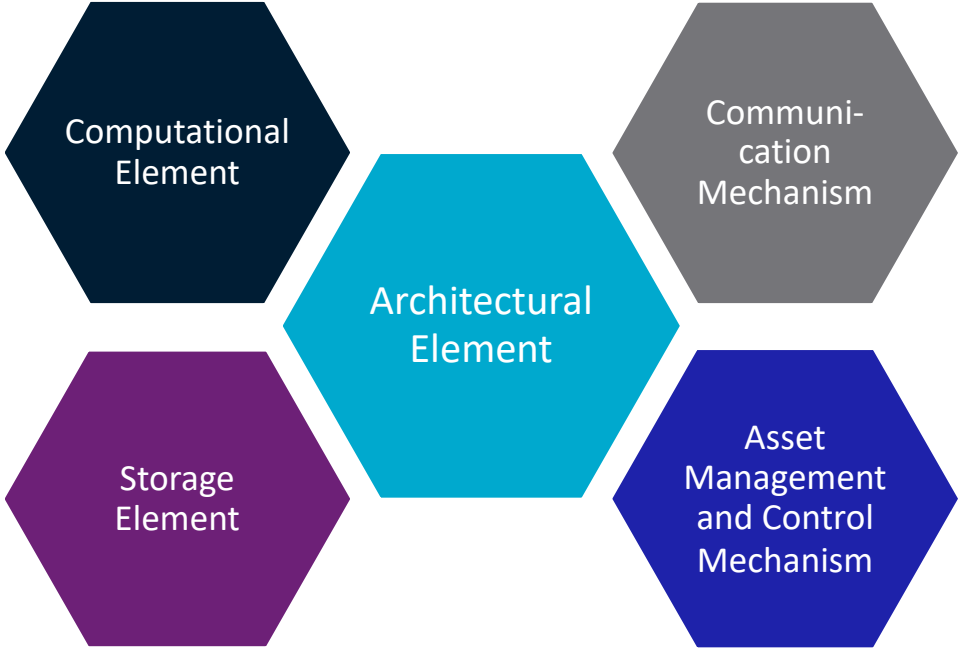


Blockchain in Larger Software System



Blockchain's Functions in Application

- Blockchain as…



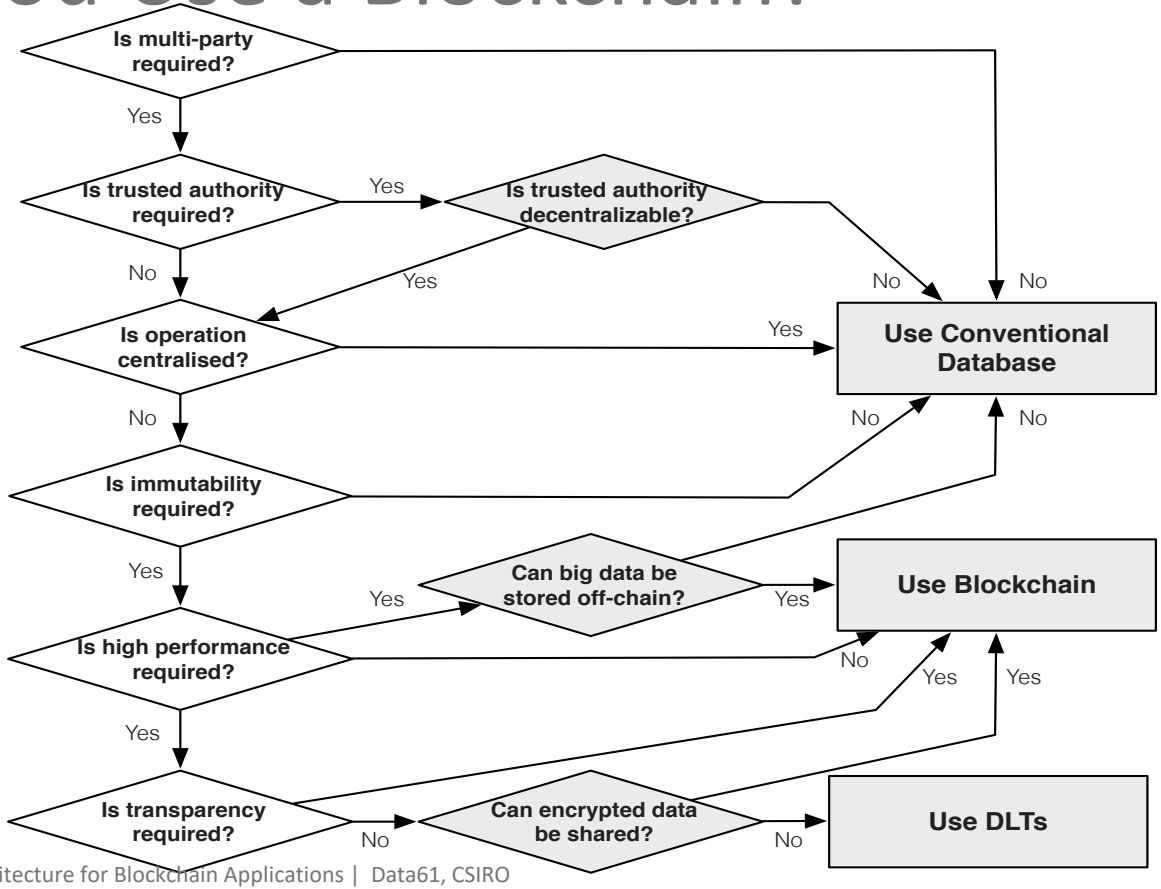
Blockchain Taxonomy

Blockchain Taxonomy

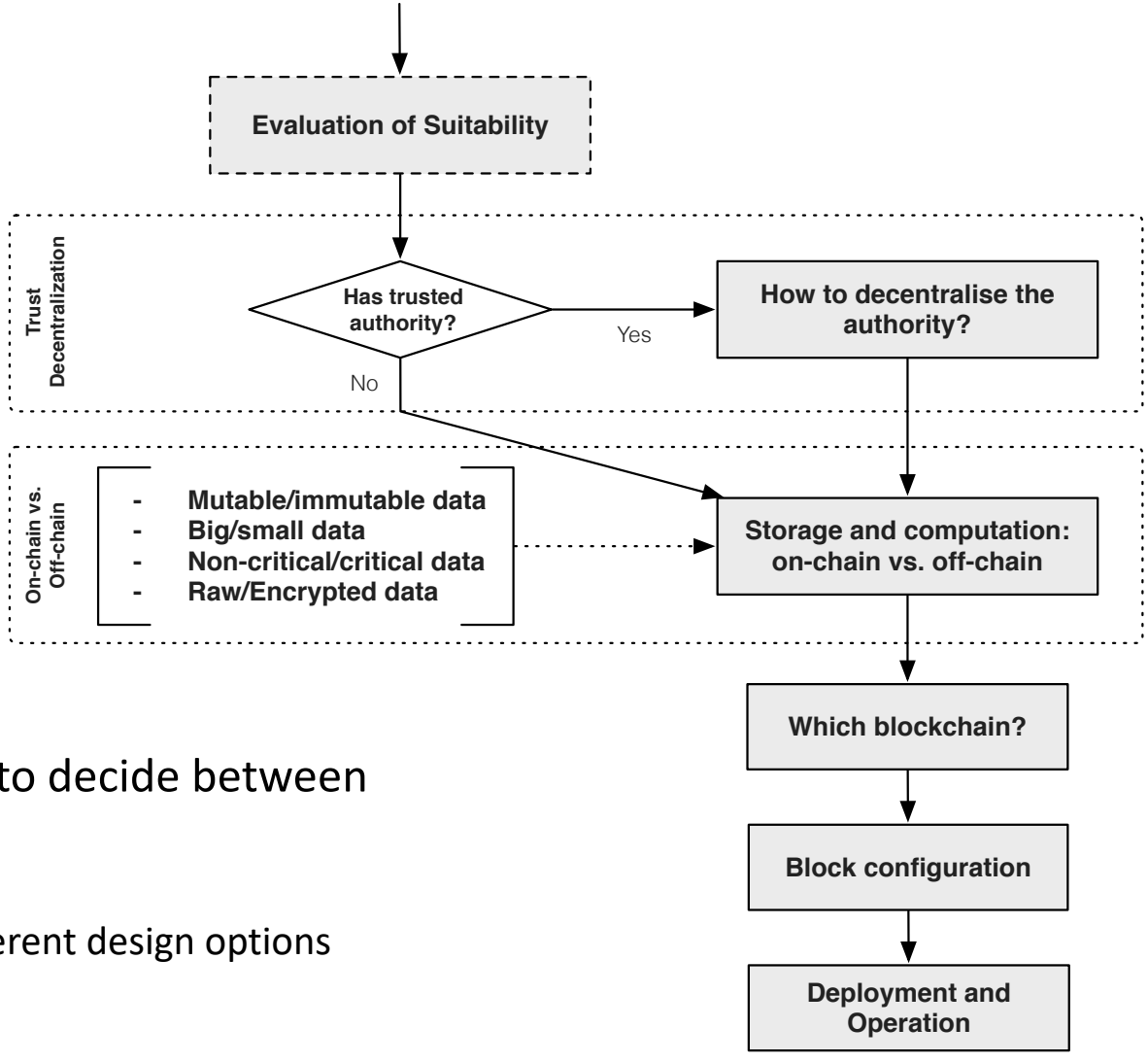
- (De)centralization
- Deployment
  - Public vs. Private
- Ledger Structure
  - List vs. DAGs vs. Networks of Ledgers
- Consensus Protocol
  - Proof-of-Work, Proof-of-Stake, PBFT
- Block Configuration and Data Structure
- Auxiliary Blockchain
  - Merged mining, sidechains, entangled, sharding etc.
-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

Design Process

Should You Use a Blockchain?



Design Process



- Every step is a procedure to decide between alternative options
  - Taxonomy
  - Systematic comparison of different design options

Patterns

Blockchain Application 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
- Reverse oracle of a system relies on smart contract to validate the requested data and check required status

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

Blockchain Application 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

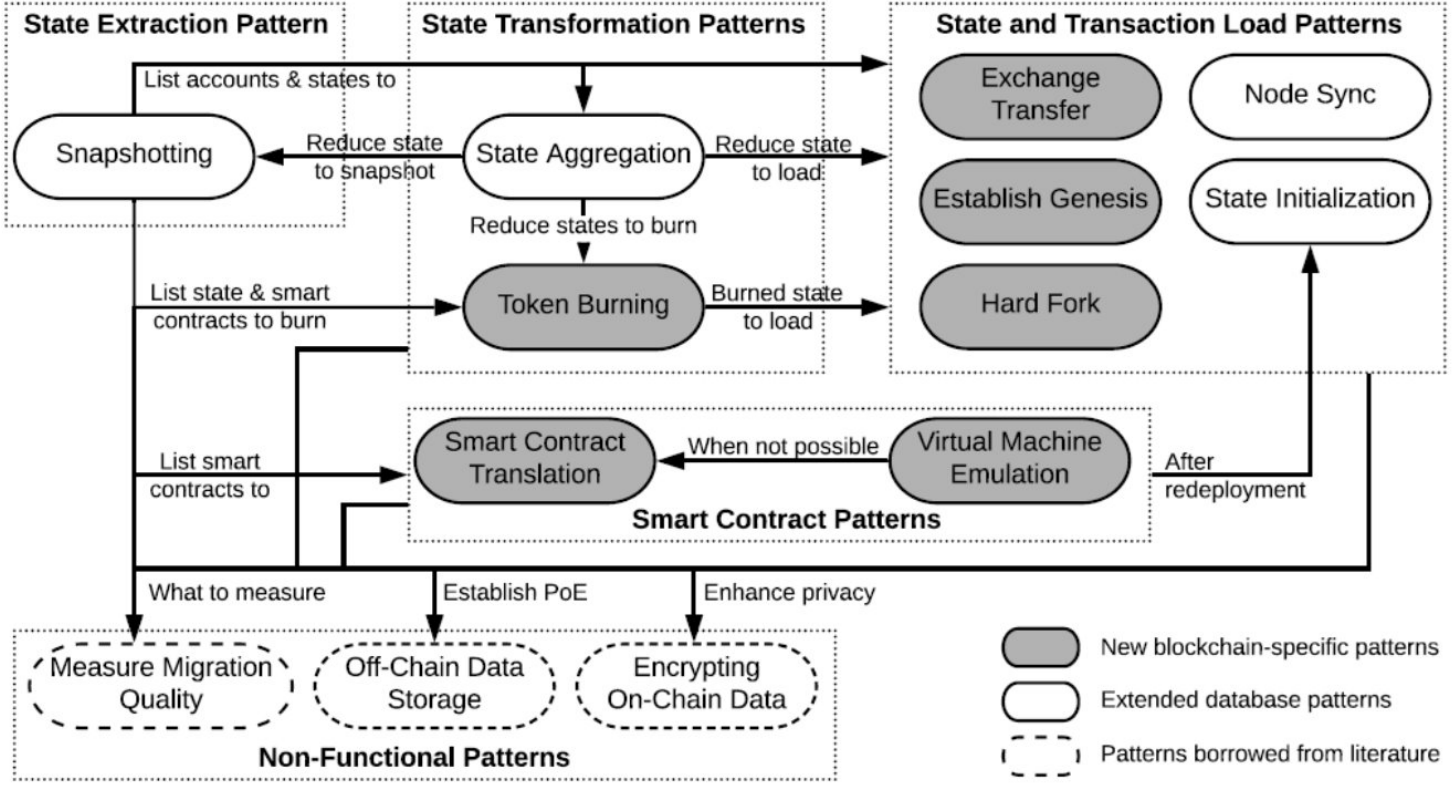
- Blockchain-based application with a website that can be browsed using a conventional web browser without any dapp plugin

More Smart Contract Patterns

BC Pattern Name	SC Pattern Name
Dynamic Authorization	Commit and Reveal
Centralized/Decentralized Oracle	Oracle (DataProvider)
Data Contract	Data Segregation
Contract Registry (within a contract)	Satellite
Contract Registry	Contract Register

Category	Pattern	Example Contract
Action and Control	Pull Payment	Cryptopunks
	State Machine	DutchAuction
	Commit and Reveal	ENS Registrar
	Oracle (Data Provider)	Etheroll
Authorization	Ownership	Ethereum Lottery
	Accesss Restriction	Etheroll
Lifecycle	Mortal	GTA Token
	Automatic Deprecation	Polkadot
Maintenance	Data Segregation	SAN Token
	Satellite	LATP Token
	Contract Register	Tether Token
	Contract Relay	Numeraire
Security[3]	Checks-Effects-Interaction	CryptoKitties
	Emergency Stop	Augur/REP
	Speed Bump	TheDAO
	Rate Limit	etherep
	Mutex	Ventana Token
	Balance Limit	CATToken

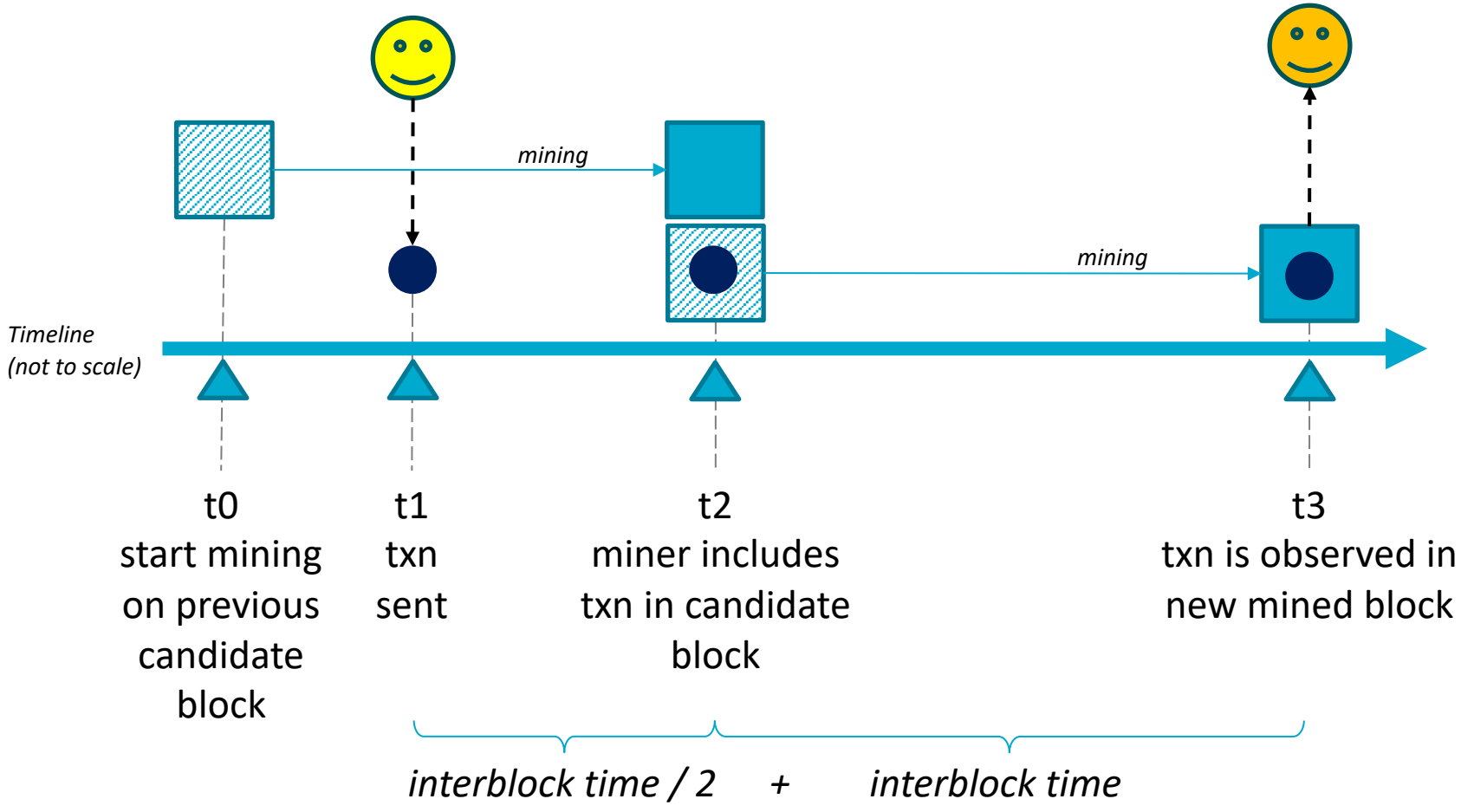
Blockchain Migration Patterns



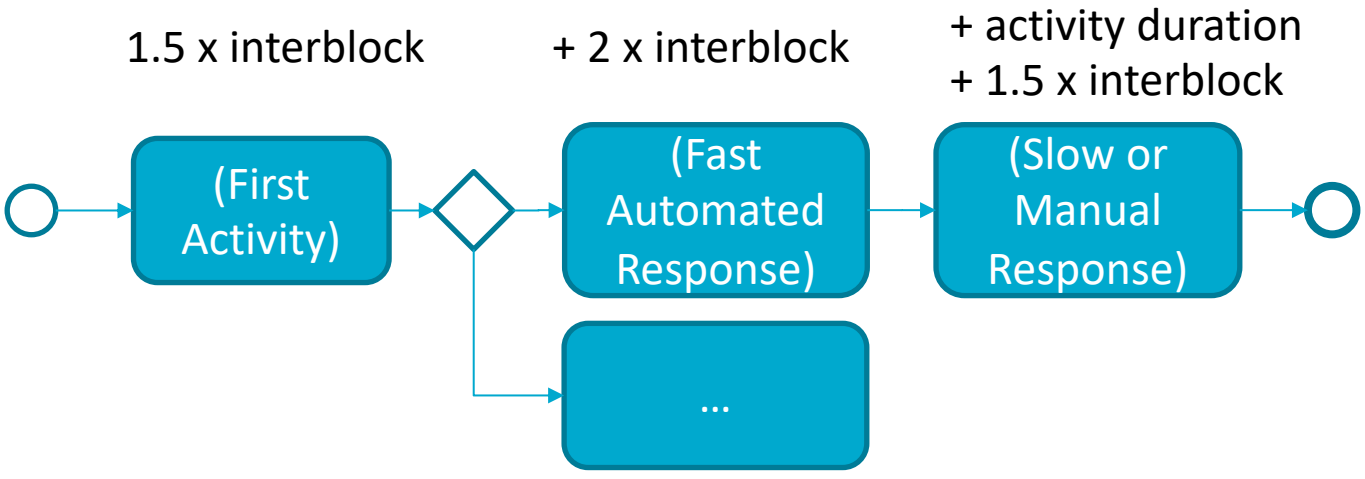
Non-functional Properties and Cost

- Performance
- Security & Reliability
- Cost

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)!

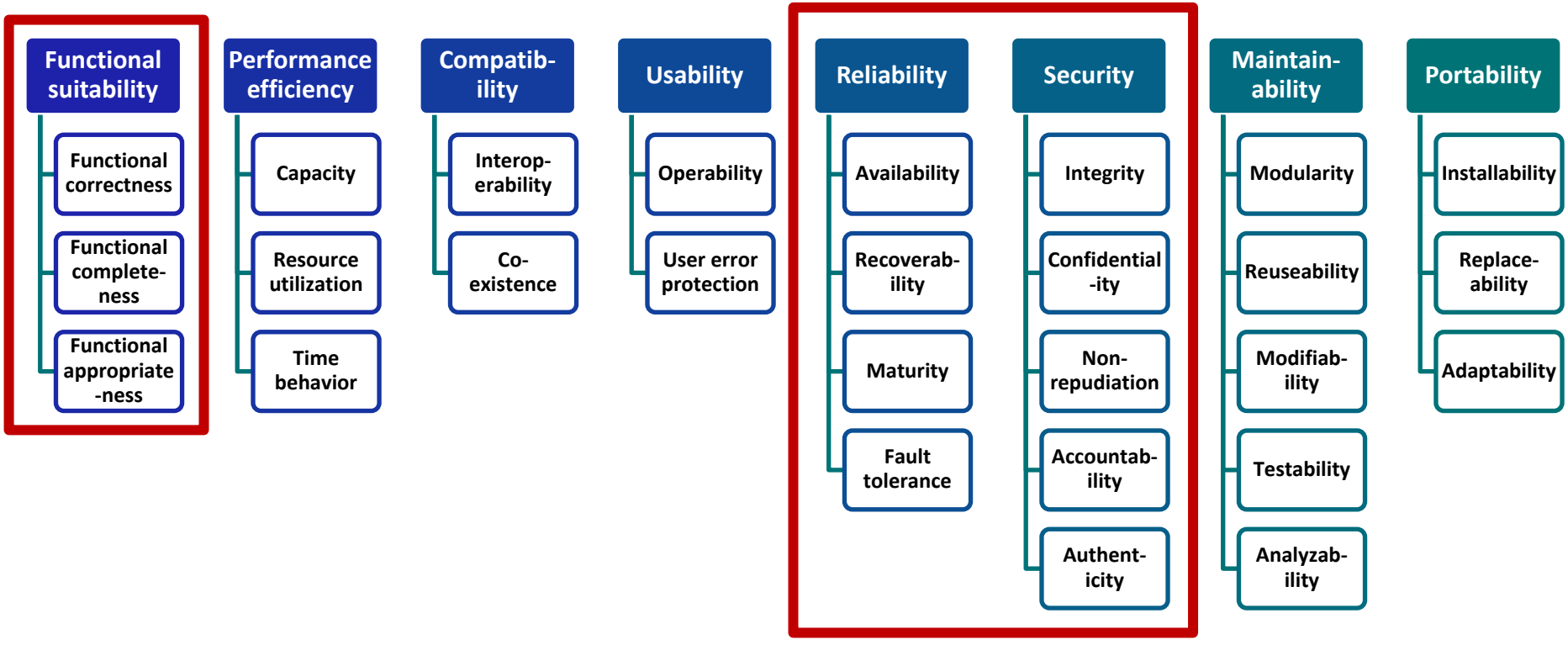
- Transactions arrive in the middle of an interblock time window
- Uniform average interblock time

- 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

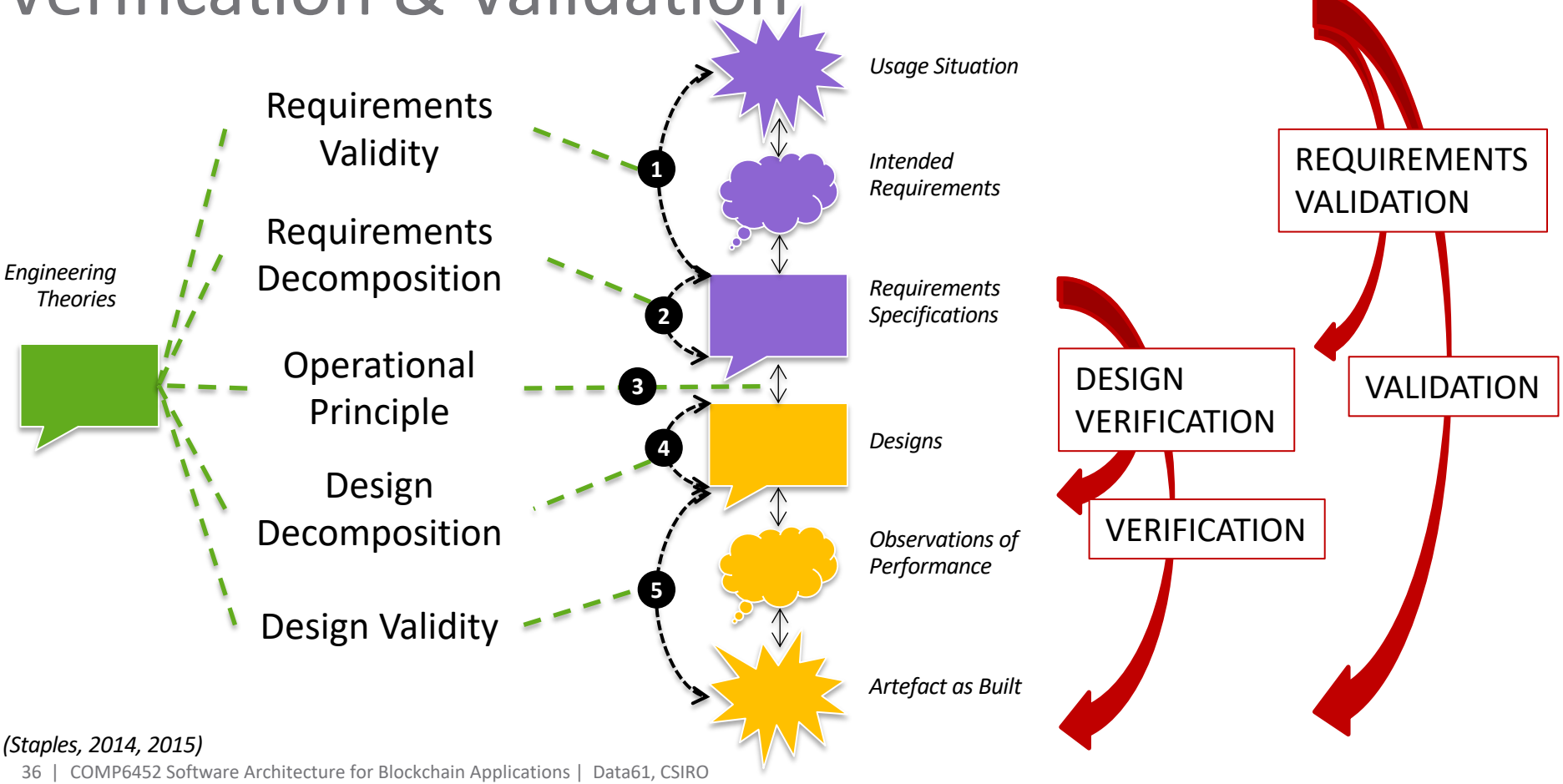
In Public Blockchains, Also Wait for Confirmation Blocks!?

- 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

ISO/IEC 25010:2011 Quality Model



Verification & Validation



(Staples, 2014, 2015)

Security & Dependability

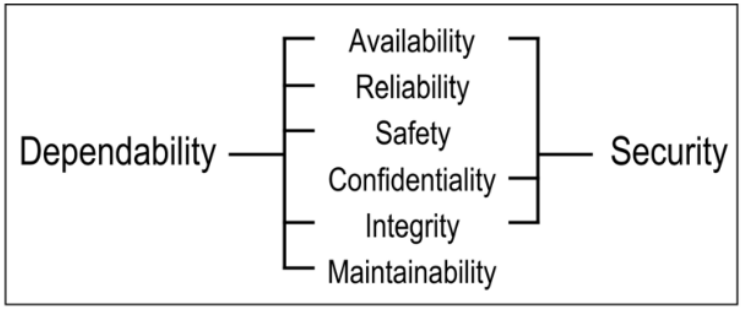


Fig. 1. Dependability and security attributes.

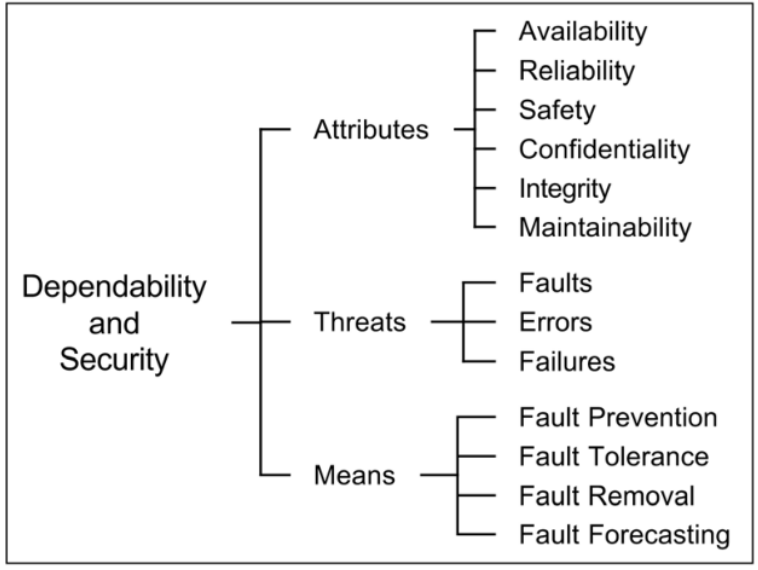


Fig. 2. The dependability and security tree.

Source: “Basic concepts and taxonomy of dependable and secure computing”
https://www.nasa.gov/pdf/636745main_day_3-algirdas_avizienis.pdf

ISO/IEC 25010:2011 Dependability & Security

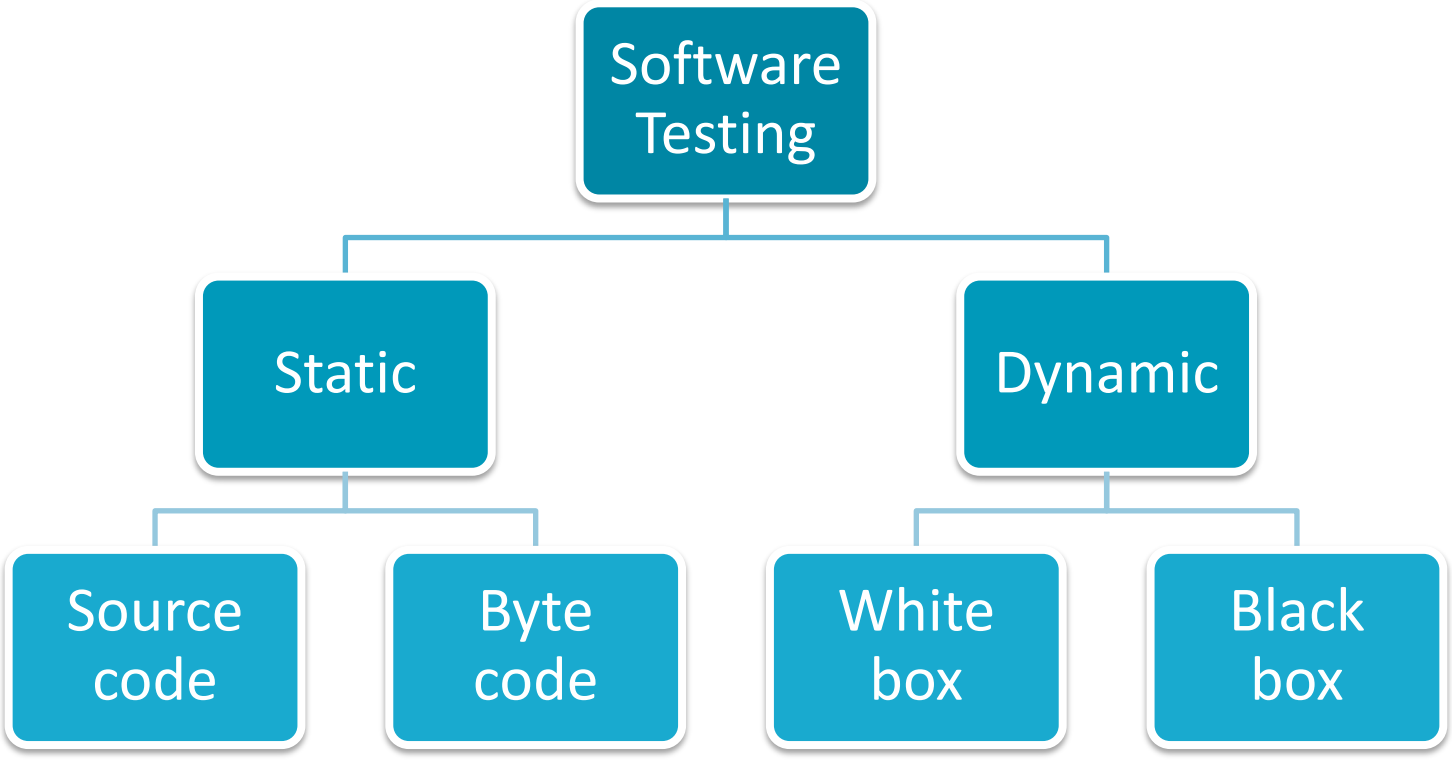
- Security
  - Integrity & Clark-Wilson policy model
    - Blockchain “anomaly” in Nakamoto consensus: possible transaction reordering
  - Confidentiality
    - And its difference to Privacy
  - Non-Repudiation
  - Accountability
    - Avoid or support? KYC/AML-CTF
  - Authenticity
- Reliability
  - Availability: readiness for service
    - Affected by latency variability
  - Recoverability
    - Aborting transactions
  - “Maturity”
    - Reliability: continuous service
  - Fault Tolerance
    - Use of smart contracts for redundant oracles

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

Smart Contract Testing

Types of Software Testing



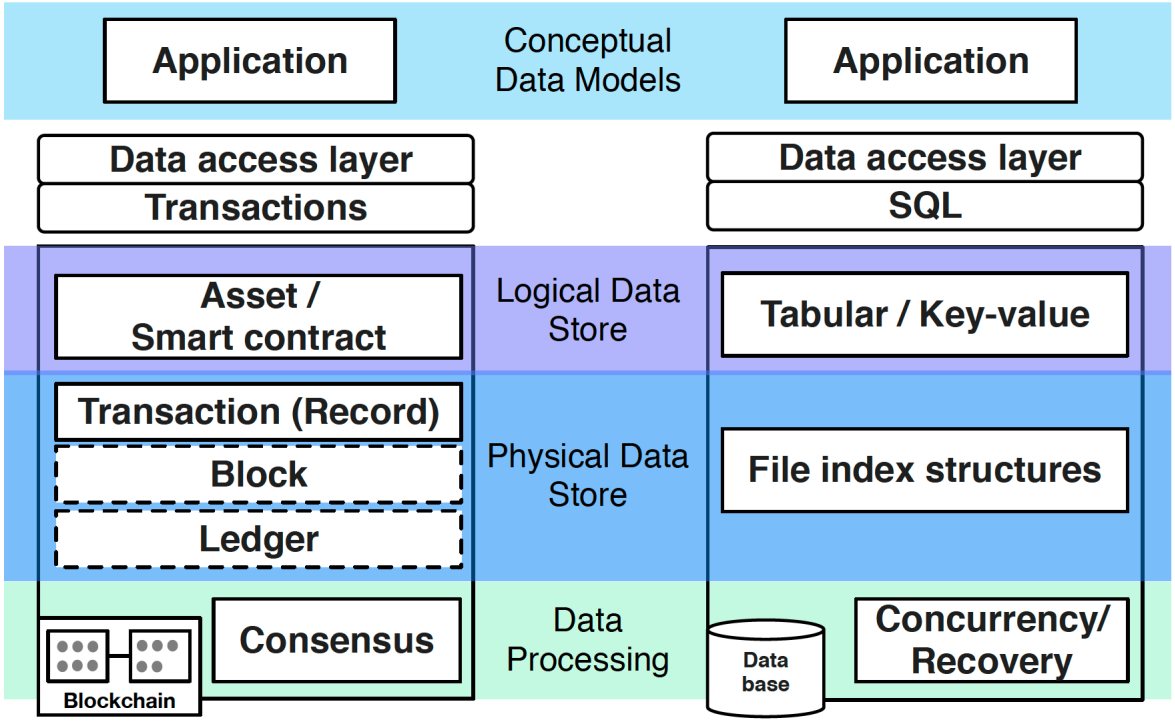
Known Issues/Vulnerabilities in SCs

- Race conditions
  - Re-entrancy
  - Cross-function race conditions
  - Deadlocks
- Denial of Service (DoS)
  - Unexpected throw
  - Gas limit
  - SC calls & block
- Arithmetic overflow/underflow
- TX order dependence (front running)
- Timestamp & block no dependence
  - Random no
- Access control
  - Ability to call `selfdestruct()`
- Bad error handling
- Language-specific behaviour
  - In solidity SC owner is set at time of initialization
  - Depreciated functions
  - Short address attack in EVM
  - Call stack depth

Data Management

- Blockchain Storage Architecture
- Data Privacy
- Data Quality

Blockchain architecture as a data store



Data privacy

- **Access** (and timeliness of the access) – the right to access their personal data and such access should be given without undue delay.
- **Rectification** – right to correct inaccurate data concerning him or her
- **Restriction of usage** – right to consent to use
- **Portability of the personal data** – right to receive the personal data in a machine-readable format for portability with other service
- **Right to be forgotten** – right to remove personal data concerning him/her without undue delay.

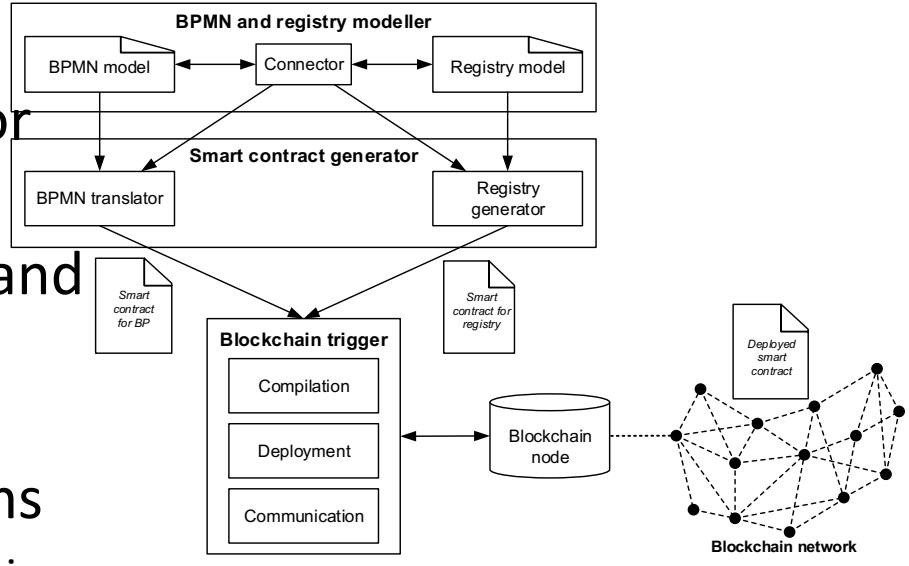
Data quality

- **Consistency** – data is free from contradiction
- **Traceability** – is there audit trail of access/changes to data
- **Availability** – is the data readily accessible
- **Compliance** – is the data compliant with standards or regulations in force
- **Confidentiality** – is the data only accessible to authorised users
- **Credibility** – is the data regarded as true and believable by users

Model Driven Engineering

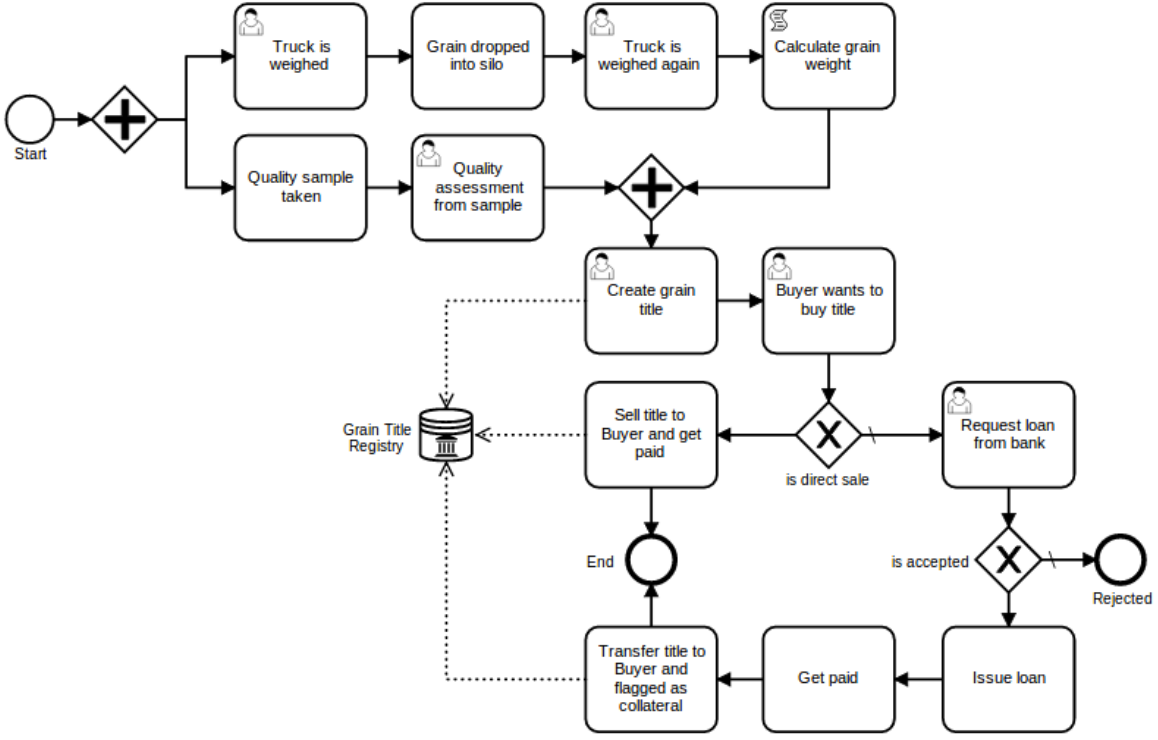
Model-Driven Blockchain-Based Applications

- 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



Data 61 Tool: Lorikeet

Combining process and data/token models



Other Topics

- Standards
- Interoperability
- Regulation, and the Law

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

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!

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?

Learning Outcomes

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

- **Xiwei Xu** | Senior Research Scientist
- Architecture & Analytics Platforms (AAP) team
  - **T** +61 2 9490 5664
  - **E** xiwei.xu@data61.csiro.au
  - **W** www.data61.csiro.au/