



# COMP6452

## Software Architecture for Blockchain Applications: Introduction

**Ingo Weber** | Principal Research Scientist & Team Leader  
Architecture & Analytics Platforms (AAP) team  
[ingo.weber@data61.csiro.au](mailto:ingo.weber@data61.csiro.au)

Conj. Assoc. Professor, UNSW Australia | Adj. Assoc. Professor, Swinburne University

[www.data61.csiro.au](http://www.data61.csiro.au)

## Agenda:

- **Part 1: Course Summary**
  - Lecturers and Tutor
  - Learning Outcomes, Course Outline, Assessments
- **Part 2: Topic Overview**
  - What is Blockchain, and Why Does it Matter?
  - Blockchain-based Applications
  - Blockchain Functionality
  - Blockchain Non-functional Properties
  - Blockchain Architecture Design
- **Part 3: Impact**
  - Use Cases
  - Disruptive Potential of Blockchains

# Agenda:

- Part 1: Course Summary

- Lecturers and Tutor
- Learning Outcomes, Course Outline, Assessments

- Part 2: Topic Overview

- What is Blockchain, and Why Does it Matter?
- Blockchain-based Applications
- Blockchain Functionality
- Blockchain Non-functional Properties
- Blockchain Architecture Design

- Part 3: Impact

- Use Cases
- Disruptive Potential of Blockchains



# Who are we?



# Lecturers and Tutor



- Dr Ingo Weber:

- Principal Research Scientist & Team Leader @ Data61, CSIRO; Conjoint Assoc. Prof. @ CSE, UNSW; Adjunct Assoc. Prof. @ Swinburne University.
- PhD from University of Karlsruhe (TH); MSc from the University of Massachusetts, Amherst, USA.



- Dr Xiwei (Sherry) Xu:

- Senior Research Scientist @ Data61, CSIRO & Conjoint Lecturer @ CSE, UNSW
- PhD from UNSW
- Working on blockchain since 2015



- Dr Mark Staples:

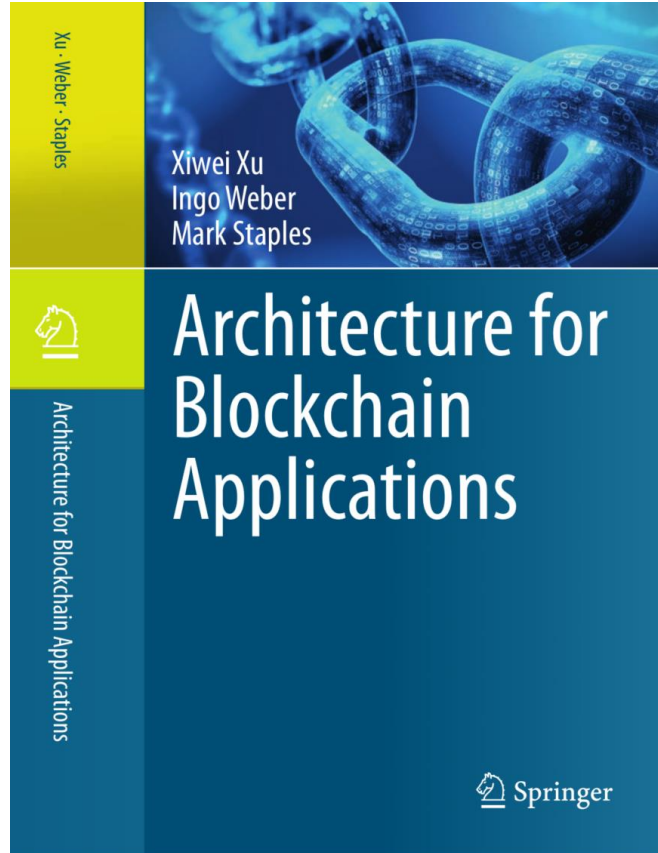
- Senior Principal Research Scientist & Team Leader @ Data61, CSIRO; Conjoint Assoc. Prof. @ CSE, UNSW
- PhD from University of Cambridge; BSc and BInfTech (Hons) from University of Queensland
- Leading member of ISO standardization for Blockchain (ISO/TC 307).



- Sin Kuang Lo:

- PhD student @ Data61, CSIRO and UNSW
- Tutor

# Book: Architecture for Blockchain Applications

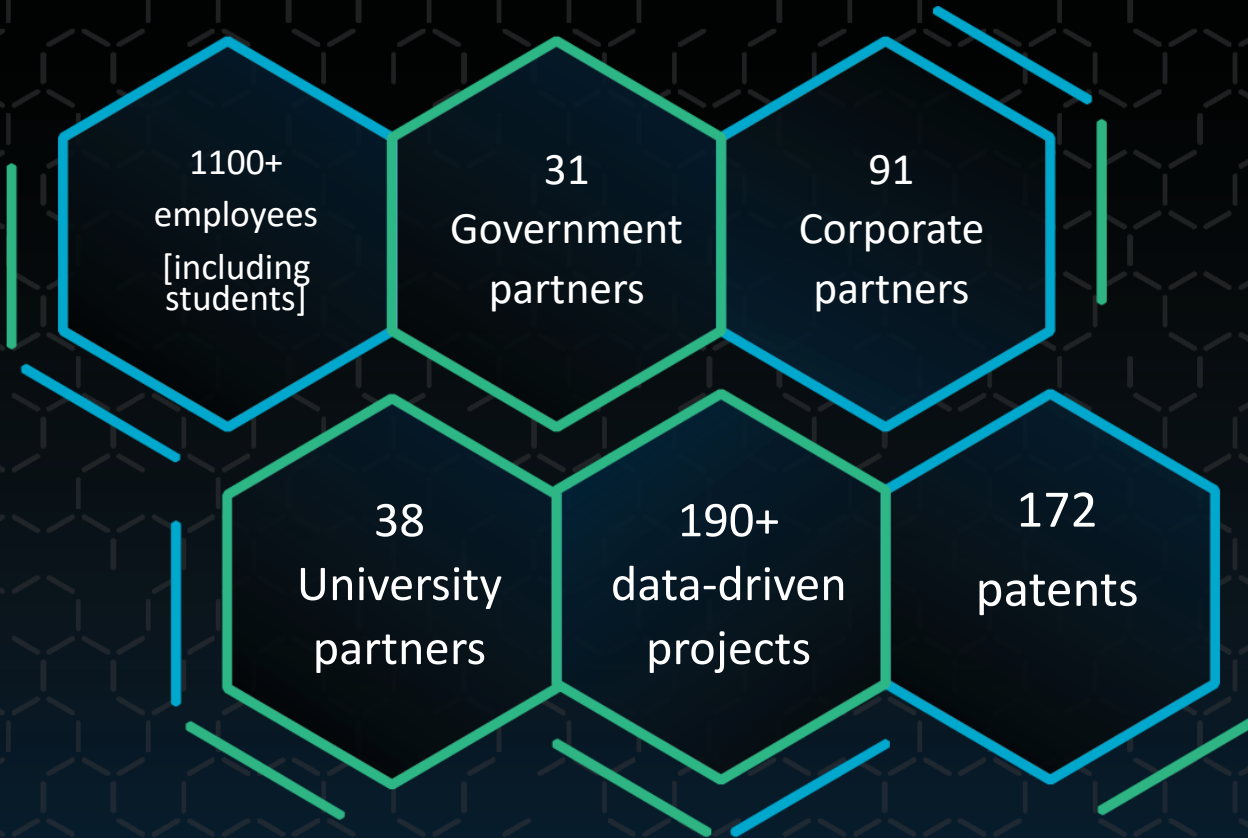


*Xiwei Xu, Ingo Weber, Mark Staples.*  
*Architecture for Blockchain Applications.*  
*Springer, 2019.*

# CSIRO: Australia's National Science Agency

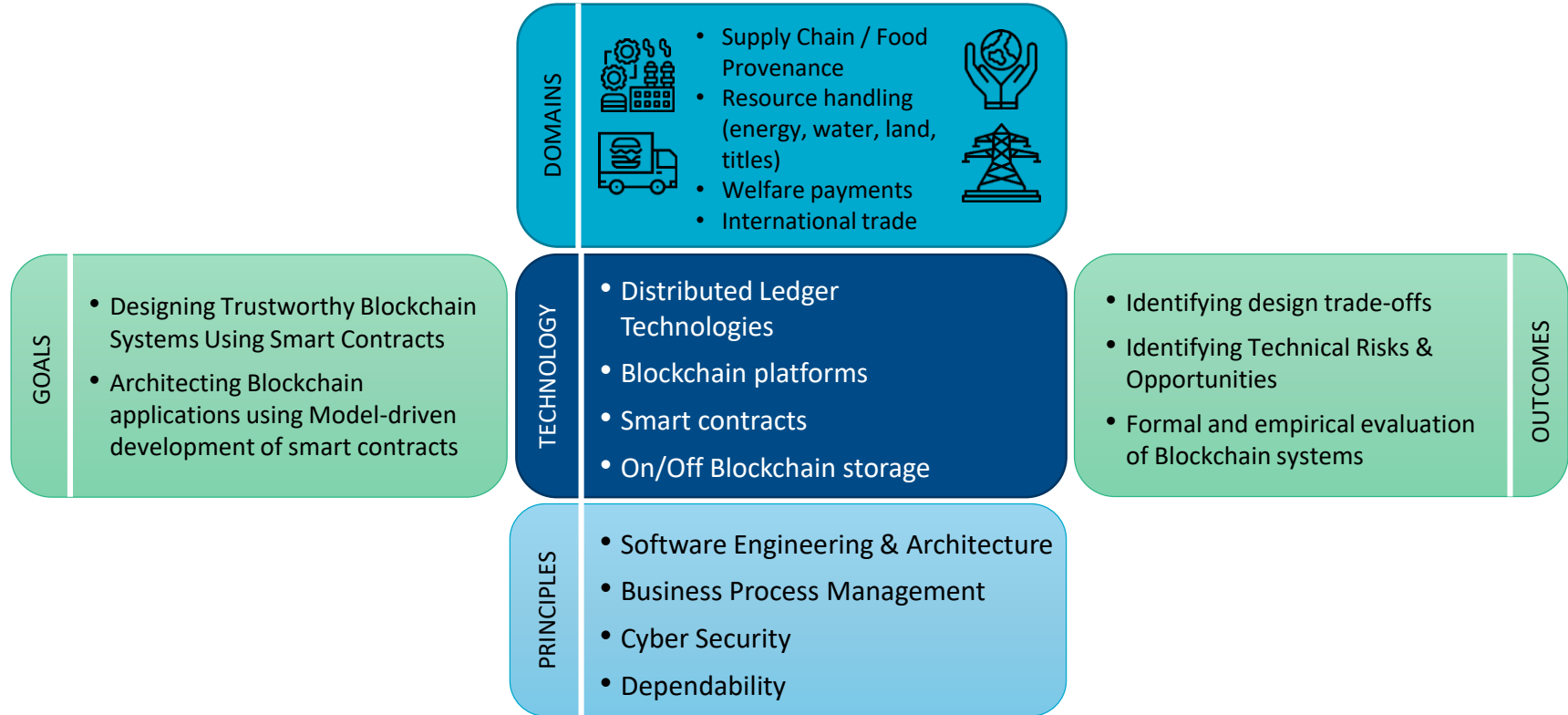


# Data61: Australia's Digital Innovation Powerhouse





# Blockchain Research at Data61 – Overview



# Course Structure

# Learning Outcomes

After the course, you will be able to:

- Explain the principles of blockchain and which roles it can play in an application architecture
- Decide on 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

# Course Outline (1)

| Week | Date   | Lecturer     | Lecture Topic                         | Relevant Book Chapters  | Notes  |
|------|--------|--------------|---------------------------------------|---|--|
| 1st  | 18 Feb | Ingo Weber   | Introduction                          | 1. Introduction<br>4. Example use cases   |  |
| 2nd  | 25 Feb | Ingo Weber   | Existing Blockchain Platforms         | 2. Existing Blockchain Platforms (1h on smart contract dev)   | Assignment 1 out (Monday before lecture)         |
| 3rd  | 4 Mar  | Sherry Xu    | Blockchain in Software Architecture 1 | 3. Varieties of blockchain<br>5. Blockchain in Software Architecture (including software architecture basics) 1/2 |  |
| 4th  | 11 Mar | Mark Staples | Blockchain in Software Architecture 2 | 5. Blockchain in Software Architecture (Non-functional properties and trade-offs) 2/2                             | Pitching session<br>Assignment 1 due (Wednesday) |
| 5th  | 18 Mar | Sherry Xu    | NFPs 1                                | 6. Design Process for Applications on Blockchain (DevOps of blockchain)<br>9. Cost                                |  |
| 6th  | 25 Mar | Mark Staples | NFPs 2                                | 10. Performance   | Mid-term Exam (1 hour)                           |

# Course Outline (2)

| Week | Date   | Lecturer                      | Lecture Topic                               | Relevant Book Chapters   | Notes   |
|------|--------|-------------------------------|---|--|---|
| 7th  | 1 Apr  | Mark Staples                  | NFPs 3                                      | 11. Dependability and Security   | Assignment 2 out (Monday before lecture)        |
| 8th  | 8 Apr  | Sherry Xu                     | Design Patterns for Blockchain Applications | 7. Blockchain Patterns (design pattern basics, design pattern language)  |   |
| 9th  | 15 Apr | Ingo Weber                    | Model-Driven Engineering                    | 8. Model-driven Engineering for Applications on Blockchains (model-driven basics, UML)                             | Assignment 2 due (After tutorial)               |
| 10th | 22 Apr | Easter Holiday                |   |  |   |
| 11th | 29 Apr | Guest Lecturer + Mark Staples | Guest Lecture and Summary                   | AgriDigital? Summary (including Epilogue content) Discuss disruptive potential, high-level opportunities and risks | Guest Lecture 2-hour and Summary of the Lecture |



# Assessments

| Assessment Title | Assessment Type  | Marks   |
|------------------|--|---------|
| Assignments (2)  | Assignment   | 2x 12.5 |
|                  | Two assignments where you will analyse and implement blockchain scenarios.   |         |
| Mid-term quiz    | Quiz Test  | 25      |
|                  | Mid-term quiz conducted in class on material covered up to that point.   |         |
| Final Exam       | Examination (central)  | 50      |
|                  | A written examination, testing all course content, with a focus on material from the second half of the course.<br>Students must obtain a <b>passing grade on the exam</b> in order to pass the course (i.e. < 50% of the exam points means failing the course). |         |

# Marking & Course Website

- To pass the course, your **overall mark must be 50 or higher**, and your **mark in the final exam must be 25 or higher**. The overall final mark will be the sum of your marks for each component if you pass the course.
- Marking will be done according to this formula:

if (final  $\geq$  25)

then total = ass1 + ass2 + mid\_exam + final\_exam;

else

total = final\_exam \* 2;

- Course website:

<https://webcms3.cse.unsw.edu.au/COMP6452/19T1/>

## Agenda:

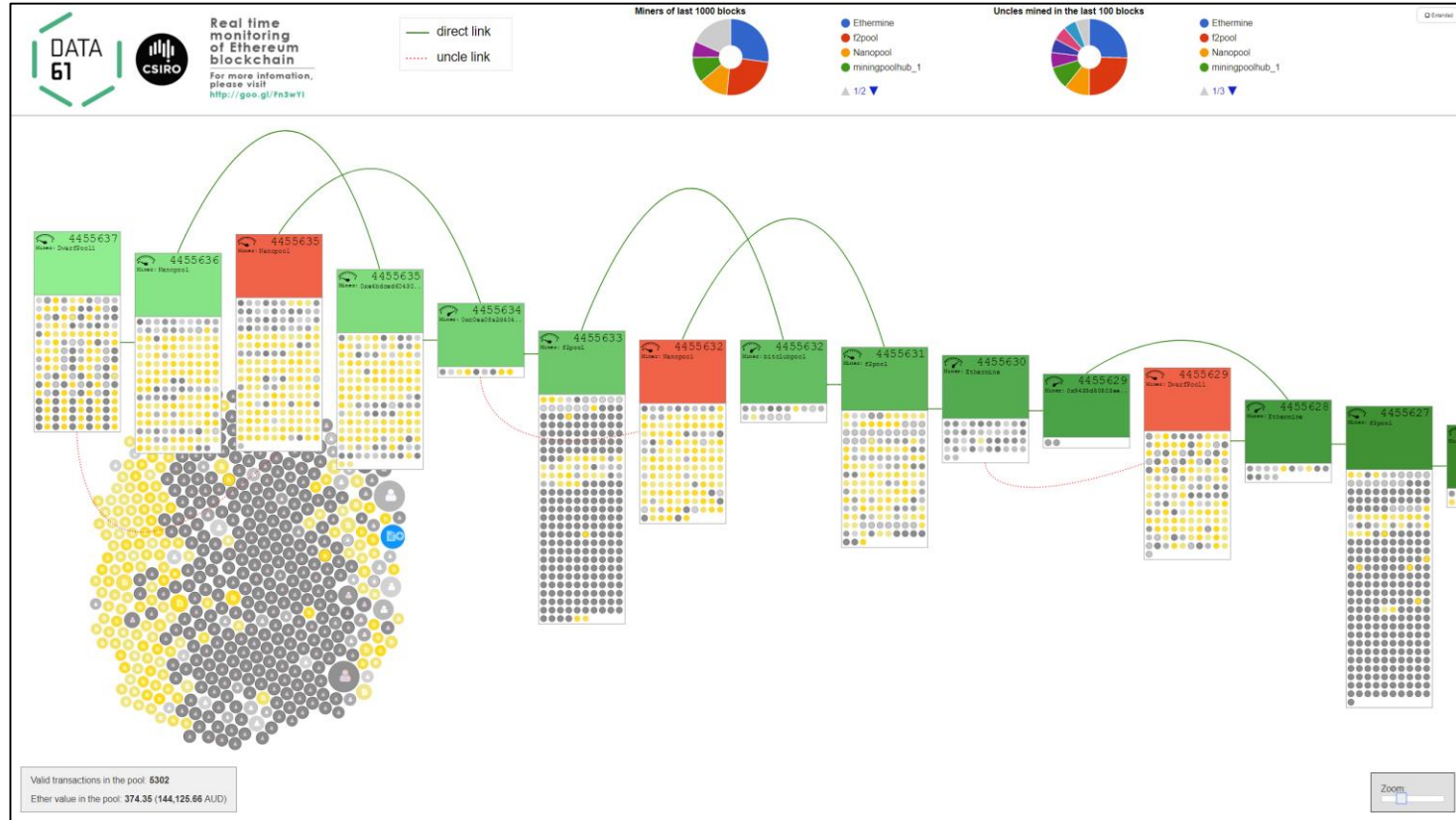
- **Part 1: Course Summary**
  - Lecturers and Tutor
  - Learning Outcomes, Course Outline, Assessments
- **Part 2: Topic Overview**
  - What is Blockchain, and Why Does it Matter?
  - Blockchain-based Applications
  - Blockchain Functionality
  - Blockchain Non-functional Properties
  - Blockchain Architecture Design
- **Part 3: Impact**
  - Use Cases
  - Disruptive Potential of Blockchains

# What is Blockchain, and Why Does it Matter?



# What is a Blockchain?

Visualization of a Blockchain: <http://ethviewer.live>





# What is the Beef about Blockchain?

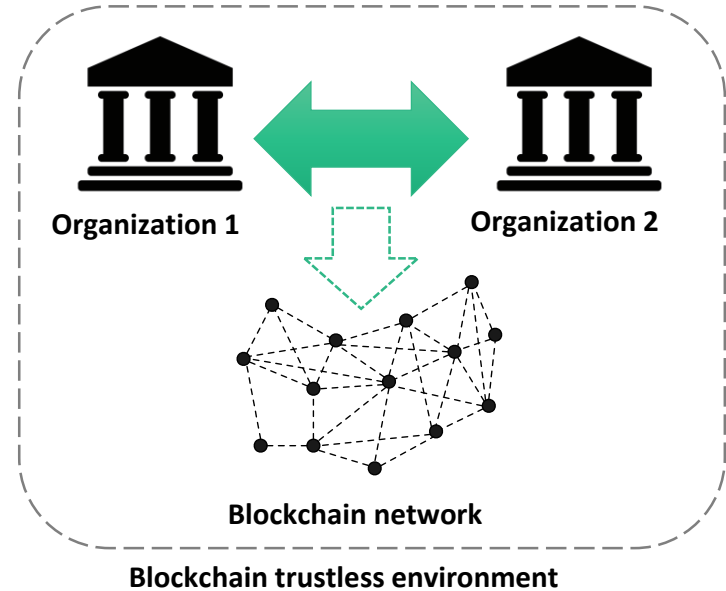
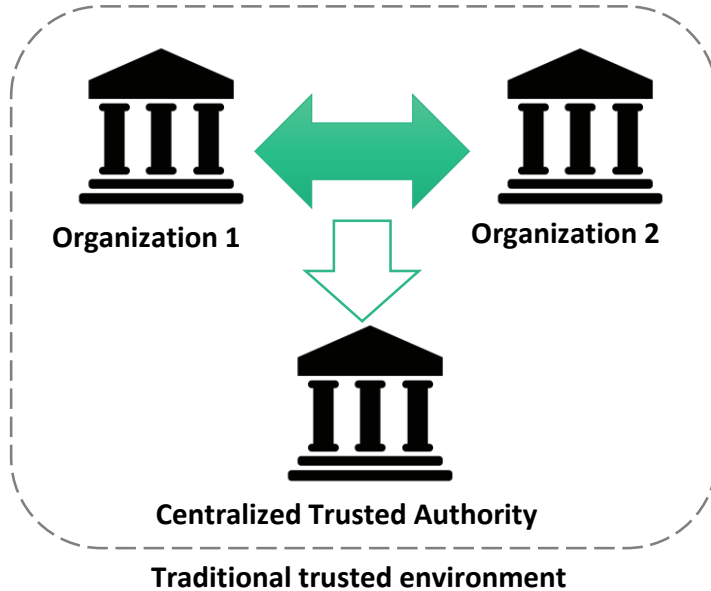


Applications coming soon:

- NSW electronic driver licences
- ASX system for settlement of trades

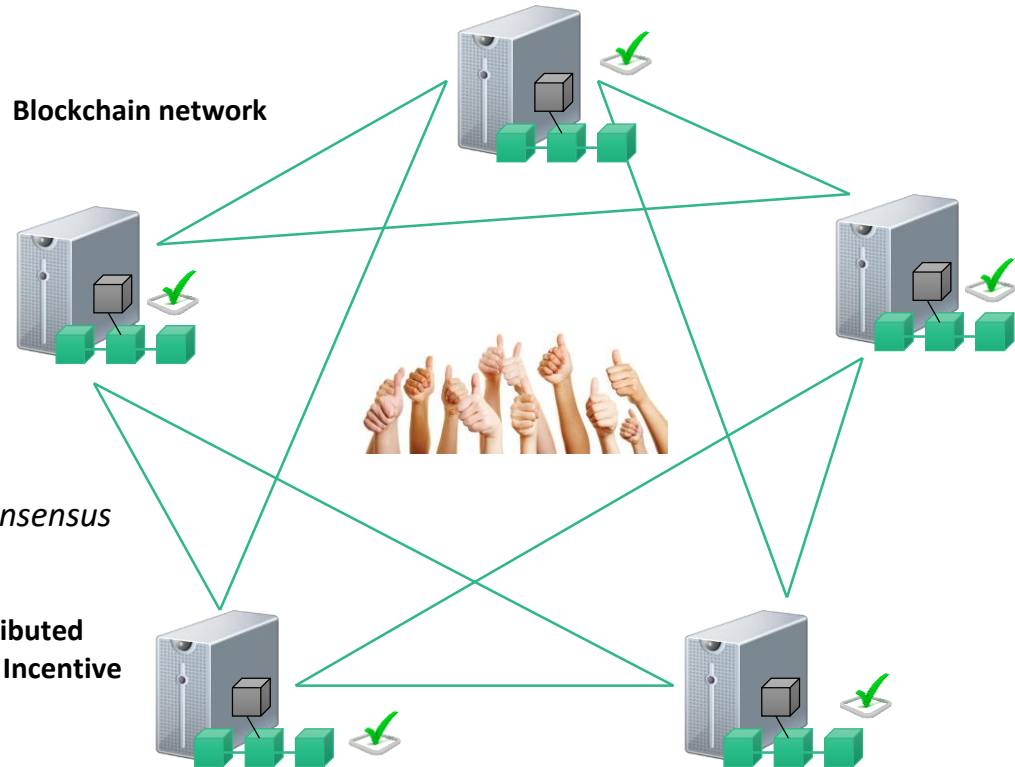
Video & reports: <https://www.data61.csiro.au/blockchain>

# Blockchain – replacing centralized trusted authority

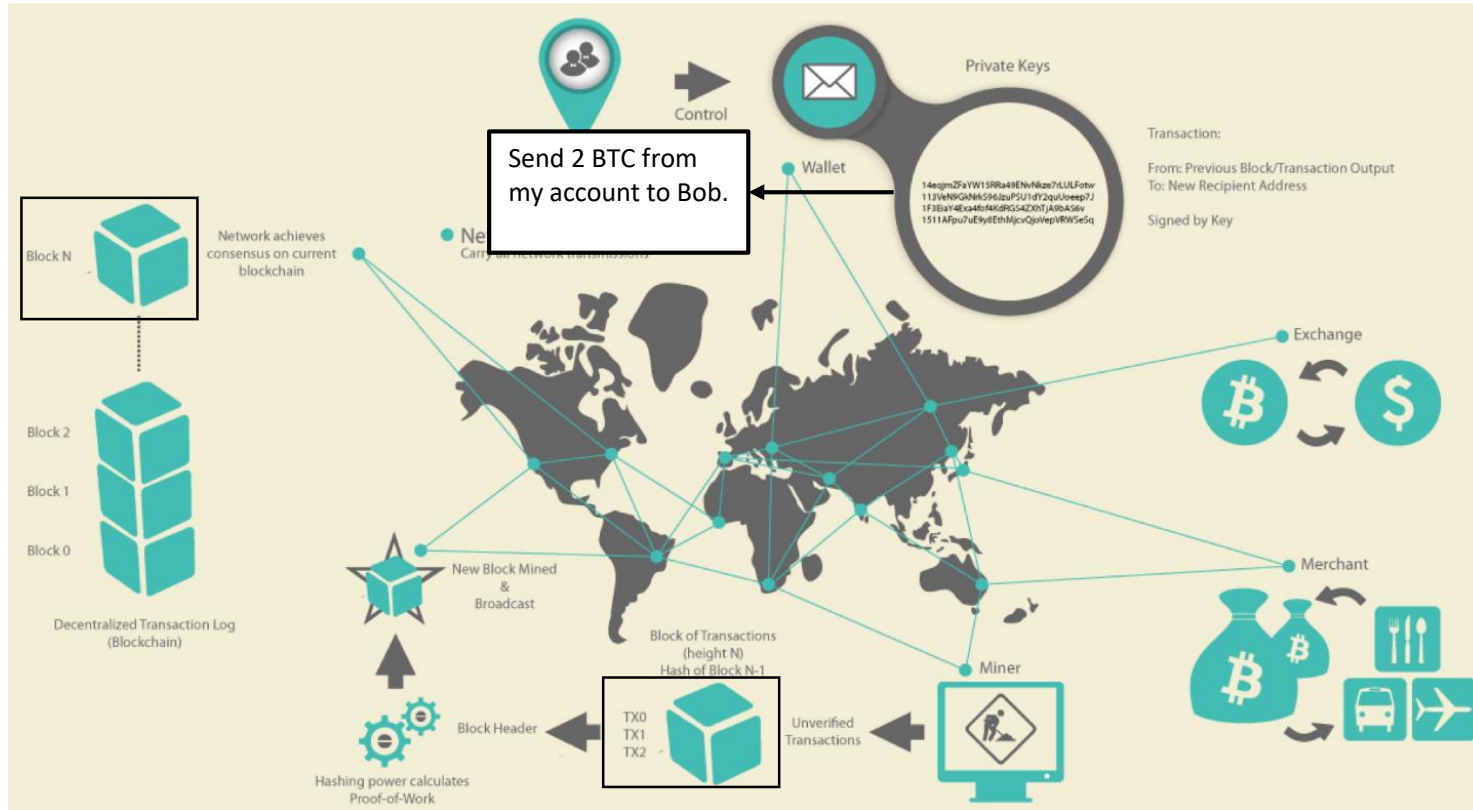


# How?

- **Immutable data base**
  - Public ledger
- **Every node hosts a replica**
  - Distributed consensus
    - *No central owner of consensus*
- **Transaction is verified by every node**
- **Combination of knowledge from Distributed Systems, Peer-to-Peer, Cryptography, Incentive Systems and Game Theory**



# Blockchain 1<sup>st</sup> gen — Cryptocurrency



Users:

- create transactions,
- sign them, and
- announce them to network

Miners:

- receive transactions
- include them in a new block,
- (try to) append the new block to the data structure

When a transaction is part of the data structure, it has taken place (though it's a bit more complicated – more later).

# Blockchain 2<sup>nd</sup> gen – Smart Contracts

- 1<sup>st</sup> gen blockchains: transactions are financial transfers
- Now Blockchain ledger can do that, and store/transact any kind of data
- Blockchain can deploy and execute programs: Smart Contracts
  - User-defined code, deployed on and executed by whole network
  - Can enact decisions on complex business conditions
  - Can hold and transfer assets, managed by the contract itself
  - Ethereum: pay per assembler-level instruction





# So what?

- Well, blockchains are exciting because they can be used as a new foundation for re-imagining systems:
  - a neutral infrastructure for processing transactions and executing programs
  - potentially interesting for innovation at **all touch-points** between organizations or individuals
  - **blockchain applications have the potential to disrupt the fabric of society, industry, and government**
- Blockchains can also be used as a technology platform to handle hard issues of data replication and system state synchronization with high integrity.

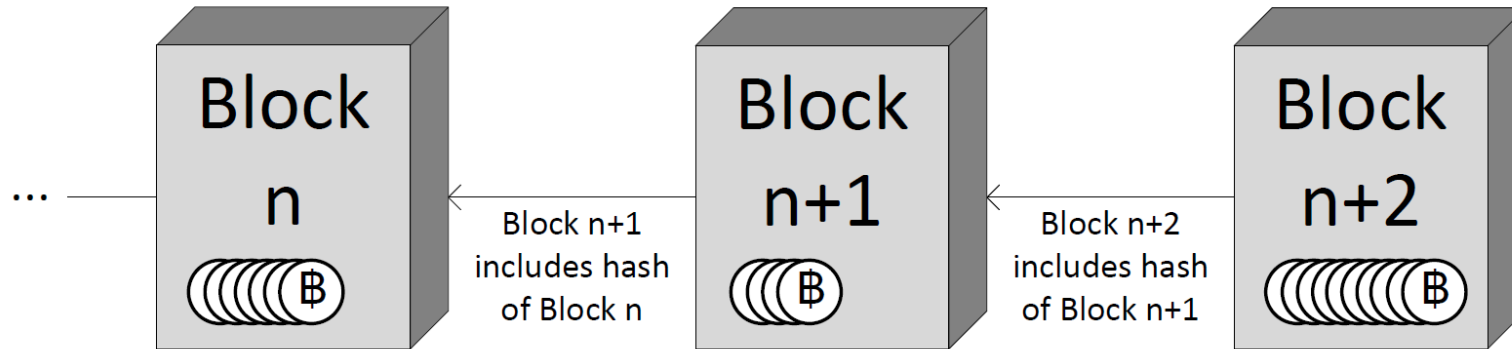
# Defining Blockchain (1)

- **Distributed Ledger**

- An “append-only” transaction store distributed across machines
- A new transaction might reverse a previous transaction, but both remain part of the ledger

- **Blockchain**

- A distributed ledger structured into a linked list of blocks.
- Each block contains an ordered set of transactions
- Use cryptographic hashes to secure the link from a block to its predecessor.



# Defining Blockchain (2)

- A **Blockchain System** consists of
  - A blockchain network of nodes
  - A blockchain data structure
    - For the ledger replicated across the blockchain network
    - Full nodes hold a full replica of the ledger
  - A network protocol
    - Defines rights, responsibilities, and means of communication, verification, validation, and consensus across the nodes in the blockchain network
    - Includes ensuring authorisation and authentication of new transactions, mechanisms for appending new blocks, incentive mechanisms

# Defining Blockchain (3)

- A **Public Blockchain** is a blockchain system with the following characteristics:
  - Has an open network
    - Nodes can join and leave without requiring permission from anyone
  - All full nodes can verify new transactions and blocks
  - Incentive mechanism to ensure the correct operation
    - Valid transactions are processed and included in the ledger and invalid transactions are rejected
- A **Blockchain Platform** is the technology needed to operate a blockchain
  - Blockchain client software for processing nodes
  - The local data store
  - Alternative clients to access the blockchain network

# Decentralised Applications and Smart Contracts

- **Smart contracts**

- Programs deployed as data and executed in transactions on the blockchain
- Blockchain can be a computational platform (more than a simple distributed database)
- Code is deterministic and immutable once deployed
- Can invoke other smart contracts
- Can hold and transfer digital assets

- **Decentralized applications or dapps**

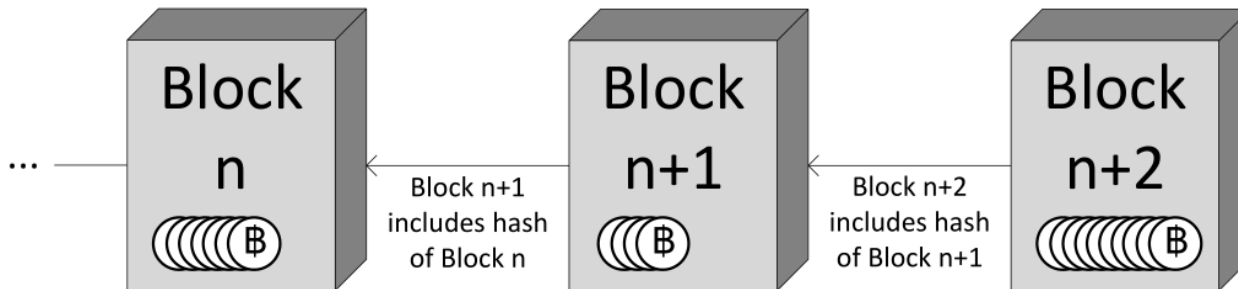
- Main functionality is implemented through smart contracts
- Backend is executed in a decentralized environment
- Frontend can be hosted as a web site on a centralized server
  - Interact with its backend through an API
- Could use decentralized data storage such as IPFS
- “State of the dapps” is a directory recorded on blockchain: <https://www.stateofthedapps.com/>



# Blockchain defined (1/4)

## Verbatim from the Book

- **Definition 1 (Distributed Ledger).** A Distributed Ledger is an *append-only store of transactions* which is distributed across many machines.
- **Definition 2 (Blockchain (Concept)).** A Blockchain is a *distributed ledger* that is structured into a *linked list of blocks*. Each block contains an ordered set of transactions. Typical solutions use cryptographic hashes to secure the link from a block to its predecessor.



# Blockchain defined (2/4)

## Verbatim from the Book

- **Definition 3 (Blockchain System).** A Blockchain System consists of:
  - a *blockchain network* of machines, also called *nodes*;
  - a *blockchain data structure*, for the ledger that is replicated across the blockchain network. Nodes that hold a full replica of this ledger are referred to as *full nodes*;
  - a network *protocol* that defines rights, responsibilities, and means of communication, verification, validation, and consensus across the nodes in the network. This includes ensuring *authorization and authentication* of new transactions, mechanisms for appending new blocks, incentive mechanisms (if needed), and similar aspects.

# Blockchain defined (3/4)

## Verbatim from the Book

- **Definition 4 (Public Blockchain).** A Public Blockchain is a *blockchain system* that has the following characteristics:
  - it has an *open network* where nodes can join and leave as they please without requiring permission from anyone;
  - all full nodes in the network *can verify each new piece of data* added to the data structure, including blocks, transactions, and effects of transactions; and
  - its protocol includes an *incentive mechanism* that aims to ensure the correct operation of the blockchain system including that valid transactions are processed and included in the ledger, and that invalid transactions are rejected.

# Blockchain defined (4/4)

## Verbatim from the Book

- **Definition 5 (Blockchain Platform).** A blockchain platform is the *technology needed to operate a blockchain*. This **comprises** the blockchain client software for processing nodes, the local data store for nodes, and any alternative clients to access the blockchain network.
- **Definition 6 (Smart Contract).** Smart contracts are *programs* deployed as data in the blockchain ledger, and executed in transactions on the blockchain. Smart contracts can *hold and transfer digital assets* managed by the blockchain, **and can invoke other smart contracts** stored on the blockchain. Smart contract code is **deterministic and immutable** once deployed.
- **Definition 7 (dapp).** A **decentralized application** or dapp is a software system that is designed to provide its main functionality through smart contracts.

# Cryptocurrencies and Tokens

- **Cryptocurrencies**

- 'Baked in' to the core platform of public blockchains -base currency of blockchains
- Symbiotic relationship
  - Blockchain keeps track of the ownership of portions of that currency, e.g. Alice owned 2Ether, transferred 1 Ether to Bob, offered 0.01Ether to miner
  - Cryptocurrency enables the incentive mechanism for blockchain operations

- **Digital tokens**

- Created and exchanged using smart contracts
- Represent assets
  - Fungible asset: individual units are interchangeable, e.g. company share, gold
  - Non-fungible asset: represents a unique asset, e.g. cryptokitties, car title

- **Not all applications are the same:**

- Transferring coins / tokens vs. tracking movement of physical goods
- Core difference: where is the default version of the truth, on or off-chain?

# Fungible and Non-fungible Tokens

- Fungible tokens: interchangeable

- E.g., \$2 coin, \$10 note
- Main concern: how many?
- Ethereum: ERC20 standard

- [https://theethereum.wiki/w/index.php/ERC20\\_Token\\_Standard](https://theethereum.wiki/w/index.php/ERC20_Token_Standard)

- Example: OmiseGO (OMG).

“The OmiseGO blockchain comprises a decentralized exchange, liquidity provider mechanism, clearinghouse messaging network, and asset-backed blockchain gateway. ... It uses the mechanism of a protocol token to create a proof-of-stake blockchain to enable enforcement of market activity amongst participants. Owning OMG tokens buys the right to validate this blockchain, within its consensus rules.”

- Non-fungible tokens

- E.g., houses, cars, patents
- Main concern: which ones?
- Ethereum: ERC721

- <https://github.com/ethereum/EIPs/issues/721>

- Example: cryptokitties

- <https://www.cryptokitties.co/>



- -fungible, individual, and their appearance depends on the individual features



# Blockchain Applications



# Blockchain-based Application

- A **blockchain-based application** (or just **blockchain application**) makes significant use of blockchain
  - dapps are an example, but the concept is far broader
  - significant portions of such applications can be based on traditional systems.
- Globally, many financial services companies, enterprises, startups, and governments are exploring suitable applications
- Areas include supply chain, electronic health records, voting, energy supply, ownership management, and protecting critical civil infrastructure
- By now, most if not all industry sectors have explored blockchain use
- The course (and the book) are about what you need to know to design and build blockchain-based applications

# Application Areas – Enterprises and Industry

- **Supply chain:** store key events to ensure goods provenance and logistic visibility
- **IoT:** device access control and software/configuration updates
- **Utility resources and services:** monitoring and payment of usage
- **Digital rights and IP management:** A trusted media asset registry to store hashes, meta-data or other identifier on blockchain and manage access and right information
- **Data management:** A metadata layer for decentralized data sharing and analytics - to discover and integrate large datasets and data analytics services
- **Proof of existence:** store a timestamped record of a hash of the document
- **Inter-divisional accounting:** A shared distributed ledger of inter-divisional accounts
- **Corporate affairs:** Board and shareholder voting and registration



# Application Areas – Financial Services

- **Digital currency**
  - New form of money transferred without 3rd parties
  - Programmable money: attach policies to specific parcels of currency
- **(International) payments**
  - Via digital currency with local exchanges between the digital currency and FIAT currencies
  - Pseudonymous: but usually have regulatory requirements to have identity, e.g. Anti-Money Laundering (AML)
- **Reconciliation for correspondent banking**
  - Banks can create a single shared ledger of accounts maintained in real-time
- **Securities settlement**
  - The exchanged assets are represented by tokens using smart contracts
  - Payment are made using tokens or native cryptocurrency
- **Markets**
  - Provides a platform for making and accepting offers to trade assets or services
  - Record the status of the offers
- **Trade finance**
  - Evidence trade-related documents
  - Automate payments

# Application Areas – Government Services

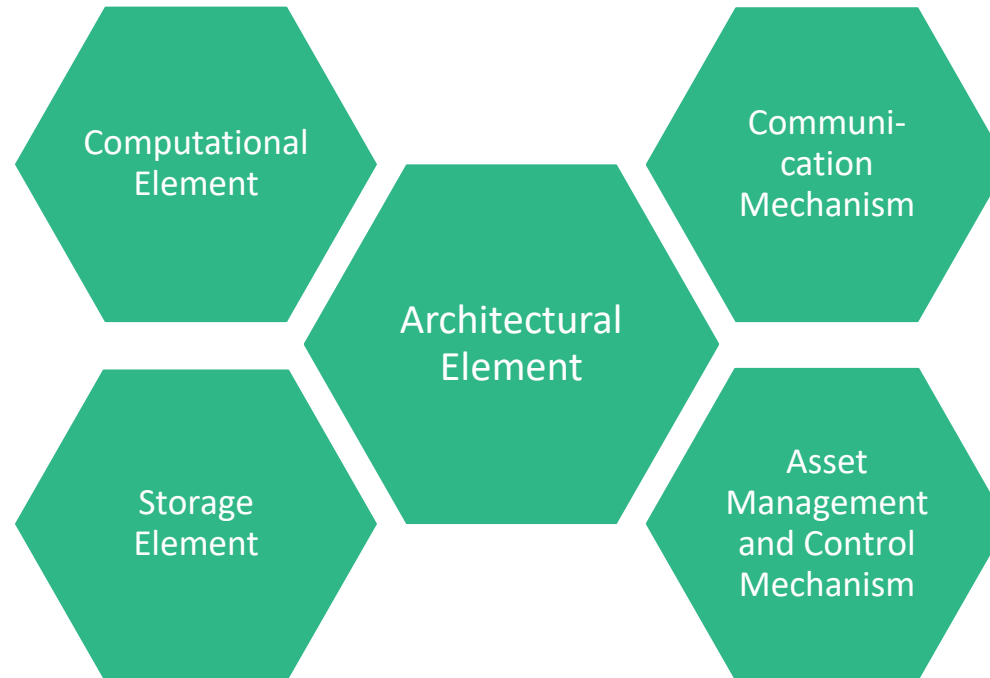
- **Registries and identity**
  - Including identities of persons, companies or devices; licensing; qualification; and certification
- **Grants and social security**
  - Automate the process coordination for application, decision making, and payment distribution
  - Allow conditional payments through programmable money – e.g. NDIS – where the money checks the conditions for spending it when attempting to do so
- **Resource quota management**
  - Government granted quotas, allocations, rights to physical resources could be awarded and tracked through tokens
  - E.g. water access licenses can provide rights to take a certain volume of water from specific sources during specific time frames
- **Taxation**
  - Automate tax collection using smart contract

# Blockchain Functionality



# Functions Blockchain can provide in a dapp

- Blockchain as...

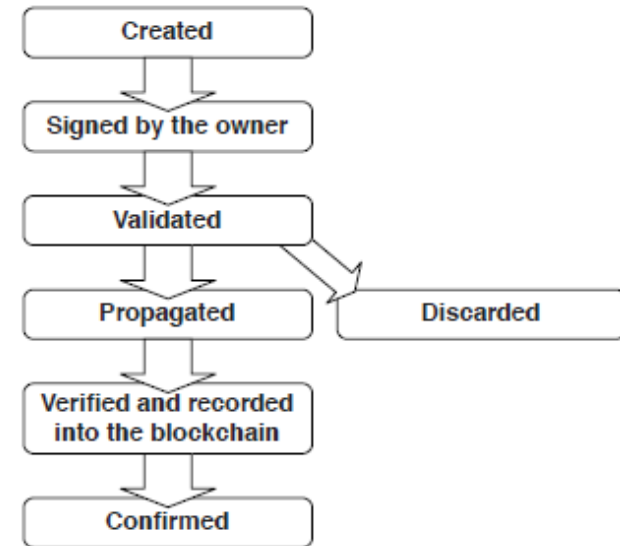




# Blockchain Functionality

## Blockchain as Storage Element

- Append only data store – prevent tampering of data
  - An ordered list of blocks, where each block contains a small (possibly empty) list of transactions.
  - Each block in is “chained” back to the previous block, by containing a hash of the previous block
- Transactions
  - Information is recorded within the transactions
  - Update blockchain states
  - Store code, variables, and results of function call results
  - Public key and digital signatures are normally used to identify accounts and to ensure integrity and authorisation of transactions



# Blockchain Functionality

## Blockchain as a Computational Infrastructure

- Smart contracts: programs, typically small in size, on the blockchain (e.g. escrow)
- Architectural decision
  - Which parts of data and computation should be placed on-chain or kept off-chain since the amount computation power, data storage space and control of read accesses on a blockchain can be limited
  - A common practice: store hashed data, meta-data and small-sized public data on-chain and keep large and private data off-chain
- There are existing platforms (e.g. IPFS) that can be used for providing a data layer on top of blockchain
- Oracles: interacting with the external world

# Blockchain Functionality

## Blockchain as a Communication Mechanism

- Transactions that are announced get broadcast across the network
- Committing a transaction to the data structure may be required before regarding the transaction as final
  - ...or the receiver can already act based on the announcement

## Blockchain as an Asset Management and Control Mechanism

- Cryptocurrency and tokens are virtual assets
- Typically, only the owner can control them
- More complex models can be implemented, e.g.:
  - Multi-signature or threshold voting (more than one account controls an asset)
  - Escrow: smart contract code controls an asset
- Very customizable through smart contracts

# Blockchain Non-functional Properties



# Blockchain Non-functional Properties

- **Immutability**
  - If data is contained in a committed transaction, it will eventually become immutable
- **Non-repudiation**
  - The immutable chain of cryptographically-signed historical transactions provides non-repudiation of the stored data
- **Integrity**: cryptographic tools support data integrity
- **Transparency**: public access (on permission-less blockchains)
- **Equal rights**
  - Allows every participant the same ability to access and manipulate the blockchain
  - **Trust**: Achieved from the interactions between nodes within the network

# Blockchain Non-functional Properties

## Limitations

- Data privacy
  - No privileged users
  - Tradeoff between data privacy and transparency
- Scalability
  - Size of the data: due to the global replication of all data across the network
  - Transaction throughput: 3-20 transactions per second (tps)
    - VISA network handles 1700 tps on average
  - Latency of data transmission
    - Write latency caused by propagation
    - Number of transactions included in each block (1MB for bitcoin)
    - Latency between submission and confirmation caused by consensus protocol
- All addressed through various research and development
  - But there is no one-size-fits-all solution (yet?)

# Decentralization

## Deployment

| Deployment Option               | Impact                 |                 |             |             |
|---------------------------------|------------------------|-----------------|-------------|-------------|
|                                 | Fundamental properties | Cost efficiency | Performance | Flexibility |
| Public blockchain               | ⊕⊕⊕                    | ⊕               | ⊕           | ⊕           |
| Consortium/community blockchain | ⊕⊕                     | ⊕⊕              | ⊕⊕          | ⊕⊕          |
| Private blockchain              | ⊕                      | ⊕⊕⊕             | ⊕⊕⊕         | ⊕⊕⊕         |



# Decentralization

## Permissions

- A blockchain may be permissioned: having one or more authorities act as a gate for participation
  - Permission to join the network, to initiate transactions, to mine, to create an asset etc.
  - Suitable in regulated industries. E.g. banks are required to establish real-world identity of transacting parties
  - Depends on the size of the network
  - Tradeoffs between permissioned and permission-less blockchain
    - Transaction processing rate, cost, flexibility, reversibility, etc.
  - Permission management mechanism may become a single point failure

# Anonymity

- Zero-knowledge proofs
  - Maintain a secure ledger
  - Private payment without disclosing the parties or amounts involved
  - E.g. Zcash encrypts the payment information in the transactions and uses a cryptographic method to allow verifying the encrypted transaction
- Mixing services
  - Groups several transactions together
    - Payment contains multiple input addresses and multiple output addresses
  - Examples: Monero, CoinJoin, Blindcoin, CoinSwap

# Blockchain Architecture Design



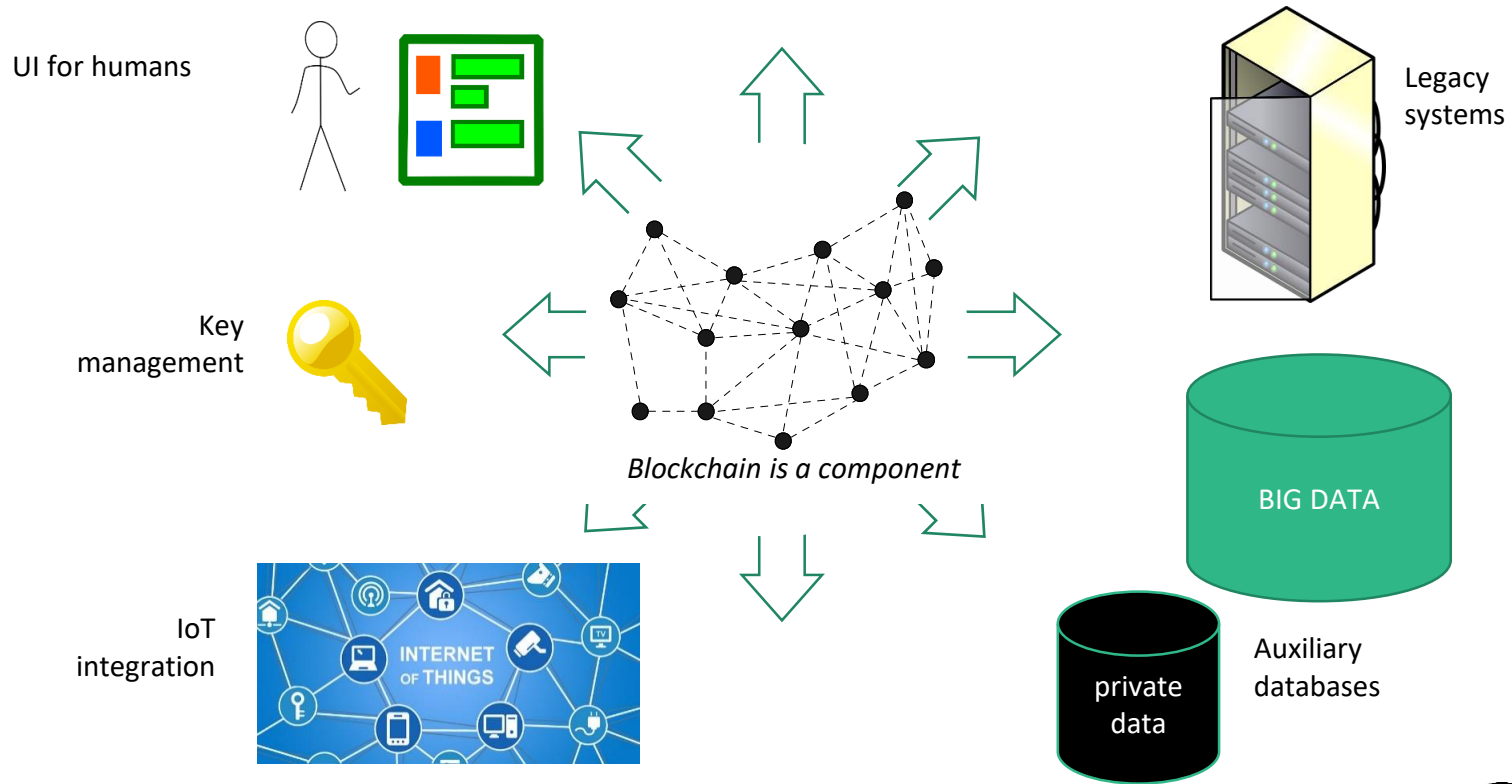
# Overview

- 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?

# Relevant Topics in the Course

- Architecting applications on Blockchain:
  - Taxonomy and design process
  - Blockchain as an element in an architecture
    - Functional and non-functional properties of blockchain
  - Blockchain Patterns
  - Cost: how much will using blockchain cost?
  - Latency: simulation under changes
  - Security and Dependability
- Model-driven development of smart contracts
  - Business process execution
  - Model-based generation of registries and UIs

# Blockchains are Not Stand-Alone Systems

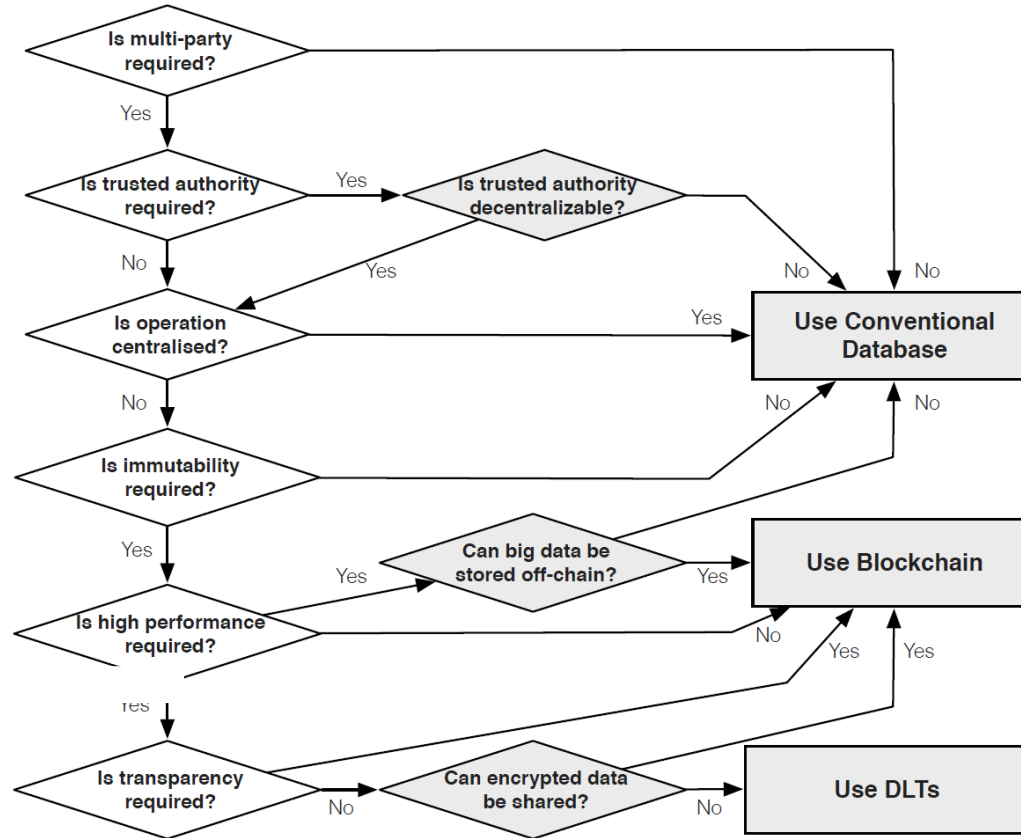


# Non-Functional Trade-Offs

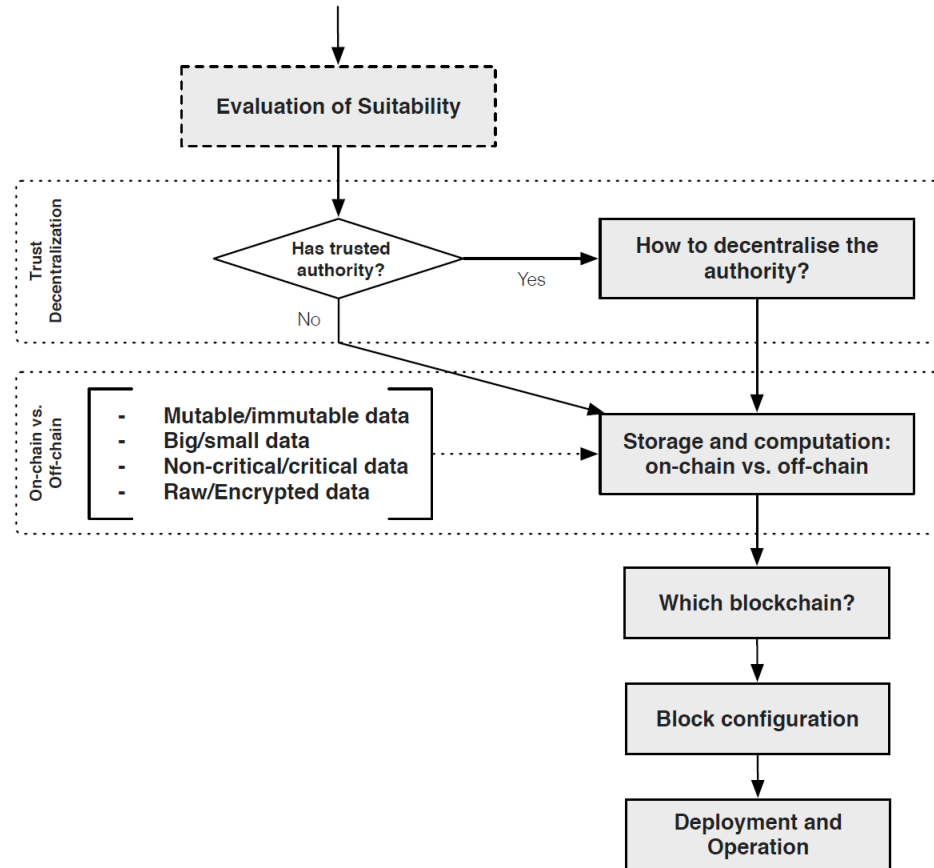
- Compared to conventional database & script engines, blockchains have:
    - (-) Confidentiality, Privacy
    - (+) Integrity, Non-repudiation
    - (+ read/ - write) Availability
    - (-) Modifiability
    - (-) Throughput / Scalability / Big Data
    - (+ read/ - write) Latency
- } Security: combination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



# Evaluation of Suitability



# Design Process for Blockchain-Based Systems



# 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   | Fundamental properties | Impact          |             | #Failure points                |
|--|--|------------------------|-----------------|-------------|--------------------------------|
|  |  |                        | Cost efficiency | Performance |                                |
| Fully Centralised                                      | Services with a single provider ( <i>e.g.</i> , governments, courts)   | ⊕                      | ⊕⊕⊕             | ⊕⊕⊕         | 1                              |
|  | Services with alternative providers ( <i>e.g.</i> , banking, online payments, cloud services)  |                        |                 |             |                                |
| Partially Centralised & Partially Decentralised        | Permissioned blockchain with permissions for fine-grained operations on the transaction level ( <i>e.g.</i> , 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) |
| Also consider skills available for a specific platform |  |                        |                 |             |                                |
|  |  | 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)          |

# Taxonomy

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

| Design Decision |           | Option                                     | Fundamental properties | Impact          |             |             |
|-----------------|-----------|--|------------------------|-----------------|-------------|-------------|
|                 |           |  |                        | 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                          | Fundamental properties                       | Impact          |             |             |
|------------------------|---------------------------------|--|-----------------|-------------|-------------|
|                        |                                 |  | 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         |
|                        |                                 | Increase block size / Decrease mining time   | ⊕               | ⊕⊕⊕         | n/a         |
| New blockchain         | Security-wise                   | Merged mining                                | ⊕⊕⊕             | ⊕           | ⊕           |
|                        |                                 | Hook popular blockchain at transaction level | ⊕⊕              | ⊕⊕          | ⊕⊕⊕         |
|                        |                                 | Proof-of-burn                                | ⊕               | ⊕⊕⊕         | ⊕⊕          |
|                        | Scalability-wise                | Side-chains                                  | ⊕⊕⊕             | ⊕           | ⊕           |
|                        |                                 | Multiple private blockchains                 | ⊕               | ⊕⊕⊕         | ⊕⊕⊕         |



# Agenda:

- **Part 1: Course Summary**

- Lecturers and Tutor
- Learning Outcomes, Course Outline, Assessments

- **Part 2: Topic Overview**

- What is Blockchain, and Why Does it Matter?
- Blockchain-based Applications
- Blockchain Functionality
- Blockchain Non-functional Properties
- Blockchain Architecture Design

- **Part 3: Impact**

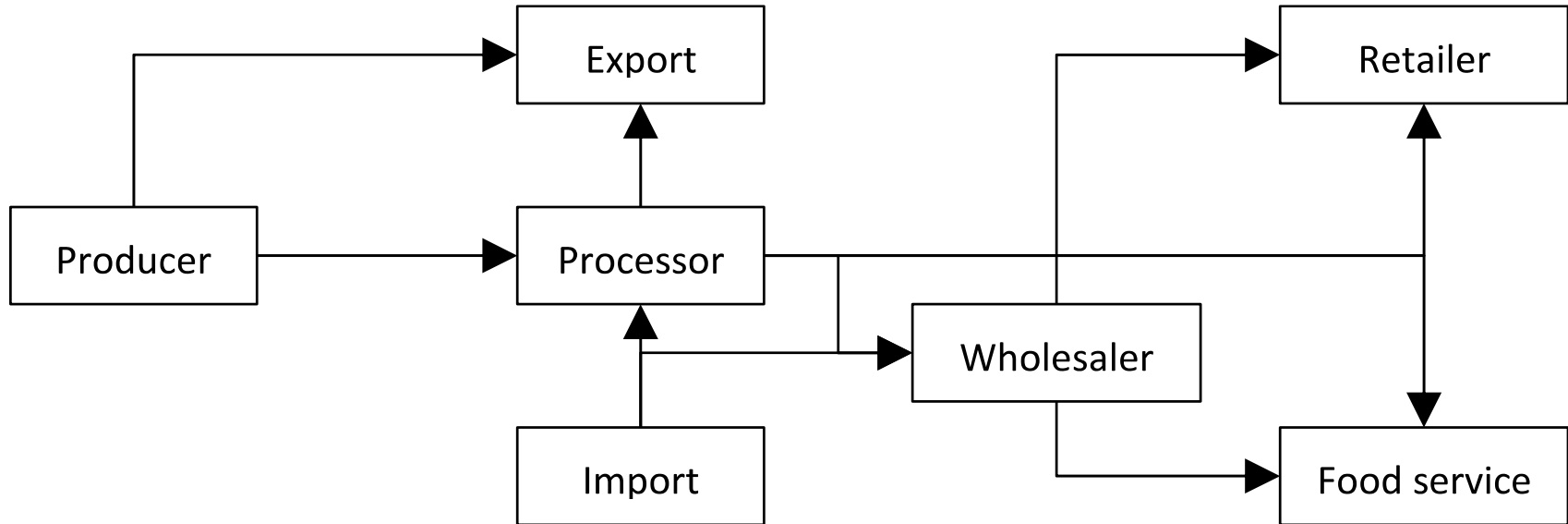
- Use Cases
- Disruptive Potential of Blockchains

# Use Cases



# Use Case: Supply Chains

## Sample Agricultural Supply Chain Network





# Using Blockchain for Supply Chains

- Why?

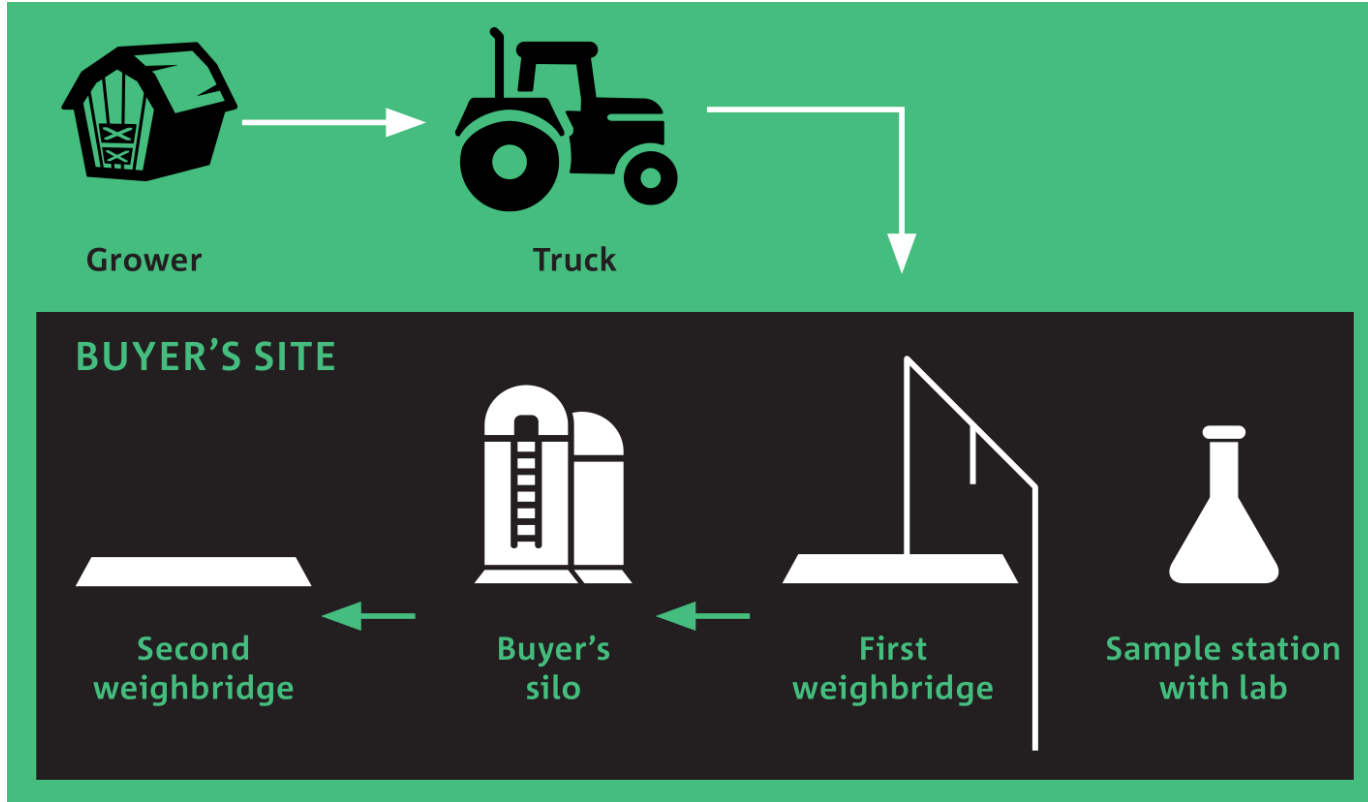
- Irrefutable, tamper-proof data store
  - Prevent or detect counterfeiting
- Smart contracts can check integrity and authorization / authentication
- Can solve other problems:
  - Counter-party risks
  - Lack of trust, e.g., in coopetition
  - Supply chain transparency
- ...

- How?

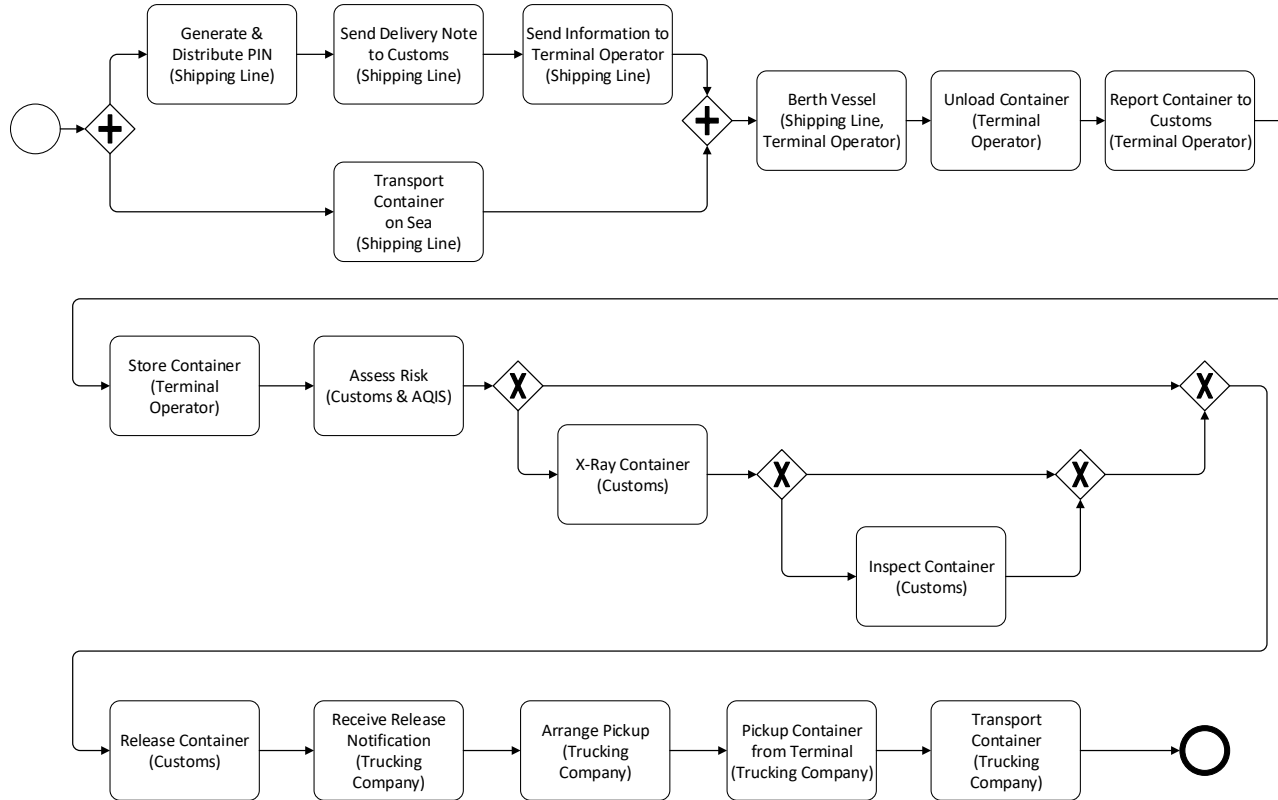
- Record supply chain events on blockchain, e.g.:
  - GS1 EPCIS events
  - Other tag scans
  - Phytosanitary certificates
- Check that event sequences are correct, e.g. through
  1. Process conformance
  2. Business rules adherence
  - Can be on-chain or off-chain
  - Regulatory compliance

# Example: AgriDigital's first pilot

see <https://www.data61.csiro.au/blockchain> / Chapter 12 in the book



# Example: Sea Import to Australia



# Some Benefits of using Blockchain in Supply Chains

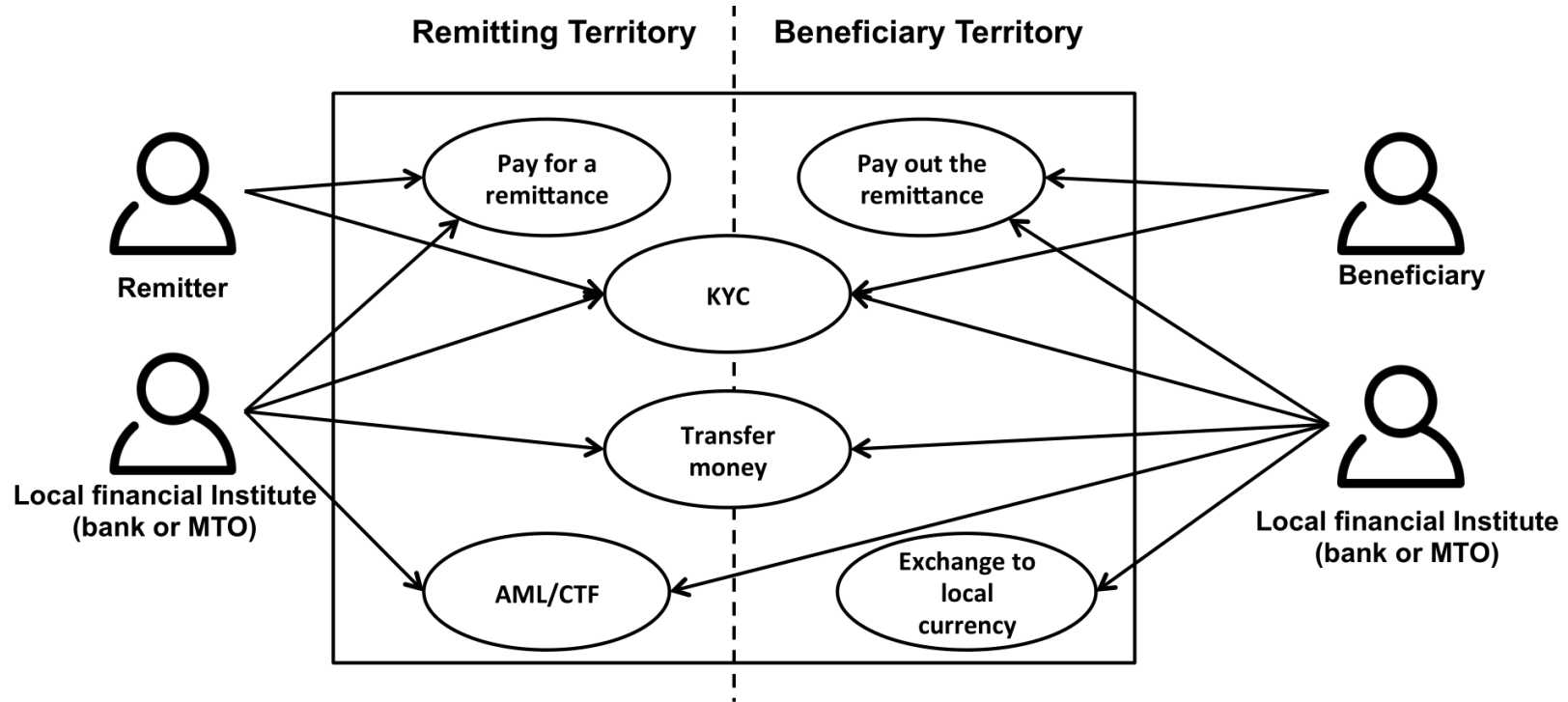
- Electronic titles to supply chain goods
  - Ensure ownership, right to sell, etc are handled correctly
  - Reduce financial risk – e.g.: if a buyer goes bankrupt before paying for the goods, the seller still owns them
- Establish identity and authenticity for:
  - Requester
  - Other relevant supply chain participants
- Check financial record / trustworthiness
- Ensure correctness of specific supply chain documents
  - E.g., invoice, purchase order, ...

# Use Case: International Money Transfers

- Many workers in Australia regularly send money back to their families overseas
  - Up to 10% of GDP in some developing countries (and even 27% in Tonga and 20% in Samoa)
  - High remittance costs have serious effects in these countries
- Remittance costs in Pacific Island countries are among the highest in the world
  - For example, to send \$200 from Australia to Vanuatu costs \$33.20 and \$28.60 to Samoa
- Issues:
  - Many parties involved, sometimes little transparency
  - Difficulties of satisfying AML/CTF (Anti-Money Laundering/Counter-Terrorism Financing) regulation, especially where the receiving party may not have a bank account.
  - High latency, with transaction times up to 5days.

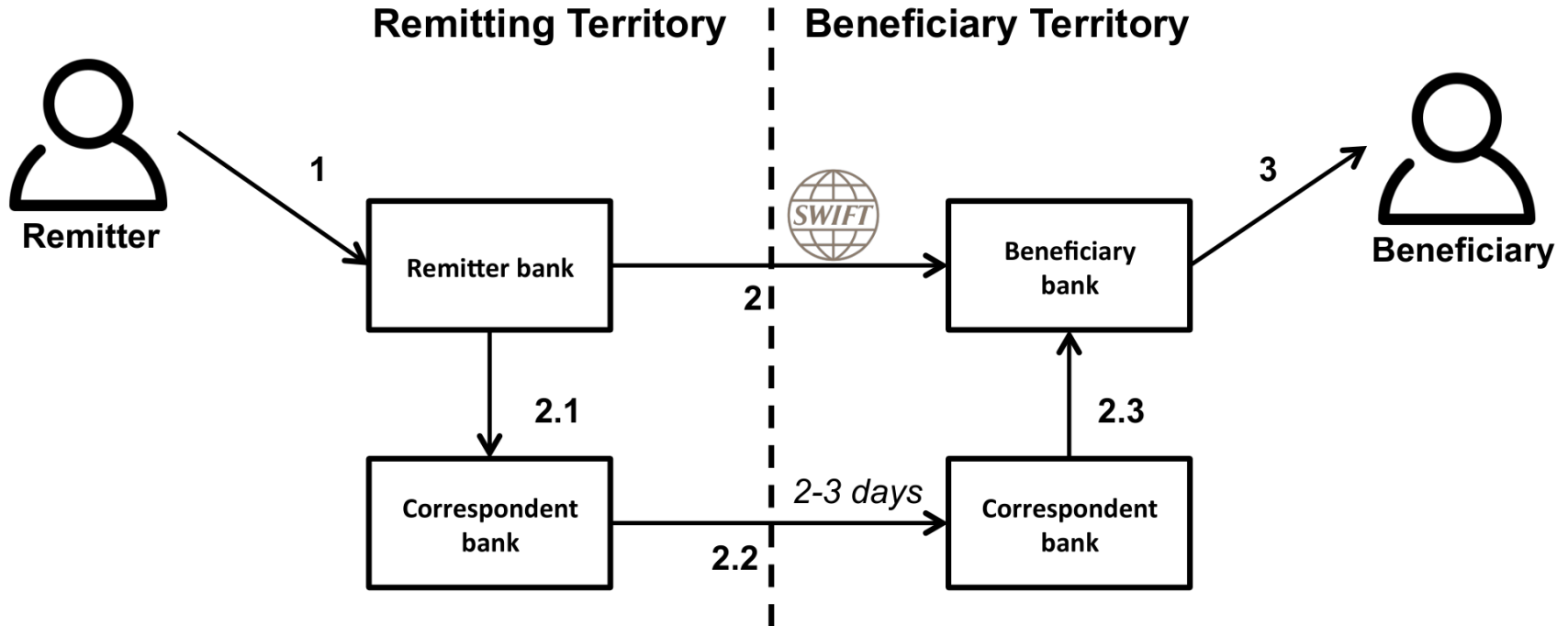
# Use Case: International Money Transfers

## Stakeholders and Functions



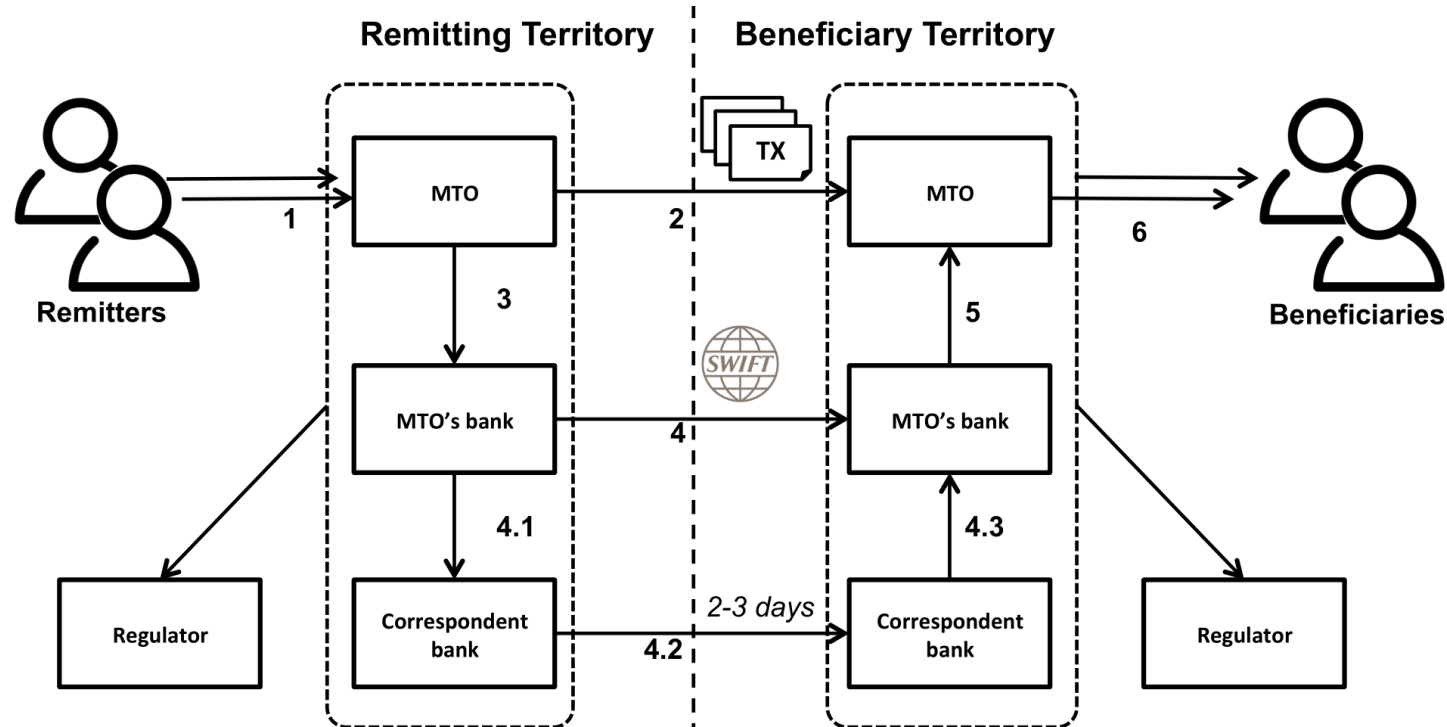
# Use Case: International Money Transfers

## Remittance through banks



# Use Case: International Money Transfers

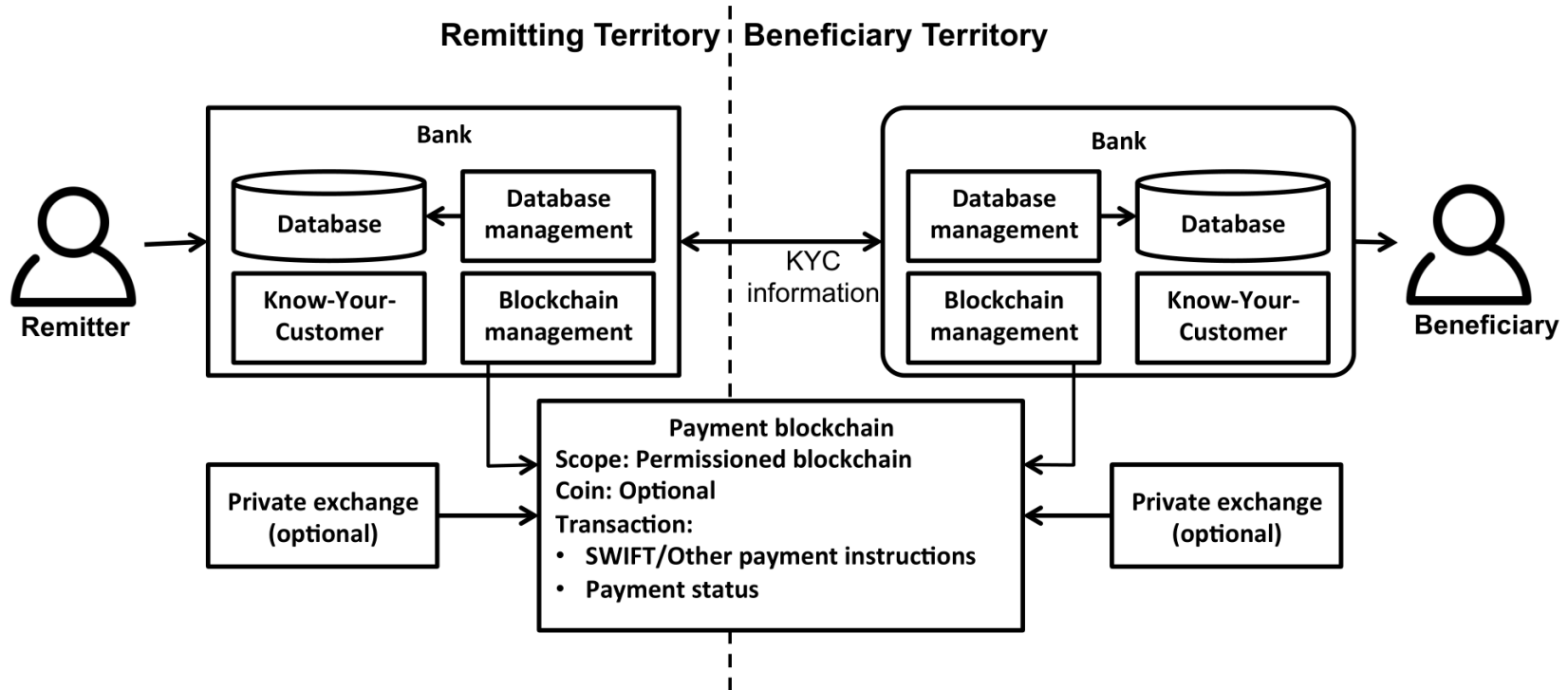
## Remittance through a Money Transfer Operator (MTO)





# Use Case: International Money Transfers

## Remittance through Blockchain



Alternatively, use existing cryptocurrencies like Bitcoin.

# Intl. Money Transfers Non-functional Properties

- Transaction Latency:
  - From days (conventional) to hours or minutes (with blockchain)
- Cost:
  - Depends on the fees charged by various parties – but more parties are involved in the conventional designs
- Transparency:
  - Greater in the blockchain setting; foreign exchange rates might still be unfavourable for the customers
- Barriers to Entry
  - Conventional design requires participants to have banking / financial services licenses, and business relationships with correspondent banks
  - Public blockchains have low barriers to entry, but local regulation still applies to end-points within countries

# Disruptive Potential of Blockchains



# Disruptive potential? (1/3)

Based on Foreword by Len Bass, PhD, CMU

- Many articles call blockchain a disruptive technology
- Think about the disruption caused by the World Wide Web and the cloud:
  - WWW changed the lives of consumers, cloud changed the lives of the producers
- Blockchain's most direct impact will likely be in the backend
  - But there are exceptions – especially where there is no centralized authority or it cannot be trusted, or going around it pays off
  - Example: the Zaatari refugee camp in Jordan that 'runs' on blockchain

# Disruptive potential? (2/3)

Based on Foreword by Len Bass, PhD, CMU

- Example: the Zaatari refugee camp in Jordan that 'runs' on blockchain
  - See <https://www.technologyreview.com/s/610806/inside-the-jordan-refugee-camp-that-runs-on-blockchain/>
- Iris scan allows refugees to spend money from their blockchain accounts
- Outcomes:
  - Saves approx. 98% of fees - similar to half of the international money transfer use case
  - Might serve as identity solution, in absence of government IDs



# Disruptive potential? (3/3)

Based on Foreword by Len Bass, PhD, CMU

- Disruptive use cases?
  - Supply chain: increase efficiency, decrease latency and fees
  - Proof of identity:
    - Most disruptive if respective government authorities are dysfunctional or uncooperative
    - NSW virtual driver licence said to use blockchain (but details unknown as yet)
- Advantages of using blockchain can be substantial, but not necessarily disruptive to the consumer
- However, Amara's law says:

*“We tend to overestimate the effect of a technology in the short run and underestimate the effect in the long run.”*

## Summary of today:

- Part 1: Course Summary
  - Lecturers and Tutor
  - Learning Outcomes, Course Outline, Assessments
- Part 2: Topic Overview
  - What is Blockchain, and Why Does it Matter?
  - Blockchain-based Applications
  - Blockchain Functionality
  - Blockchain Non-functional Properties
  - Blockchain Architecture Design
- Part 3: Impact
  - Use Cases
  - Disruptive Potential of Blockchains



# Course Outline – next two weeks

| Week | Date   | Lecturer   | Lecture Topic                         | Relevant Book Chapters  | Notes                                    |
|------|--------|------------|---------------------------------------|---|--|
| 1st  | 18 Feb | Ingo Weber | Introduction                          | 1. Introduction<br>4. Example use cases   |  |
| 2nd  | 25 Feb | Ingo Weber | Existing Blockchain Platforms         | 2. Existing Blockchain Platforms<br>(1h on smart contract dev)  | Assignment 1 out (Monday before lecture) |
| 3rd  | 4 Mar  | Sherry Xu  | Blockchain in Software Architecture 1 | 3. Varieties of blockchain<br>5. Blockchain in Software Architecture (including software architecture basics) 1/2 |  |

Note: **tutorials** run from week 2 to week 9.  
They are not mandatory, but helpful,  
especially for completing the assignments.





# End of Lecture / Consultation

**Ingo Weber** | Principal Research Scientist & Team Leader

Architecture & Analytics Platforms (AAP) team

[ingo.weber@data61.csiro.au](mailto:ingo.weber@data61.csiro.au)

Conj. Assoc. Professor, UNSW Australia | Adj. Assoc. Professor, Swinburne University

[www.data61.csiro.au](http://www.data61.csiro.au)