Distributed Trust

How Data-Driven Applications, AI and Blockchain is Impacting Service Oriented Computing

Dr. Liming Zhu

Research Director, CSIRO's Data61

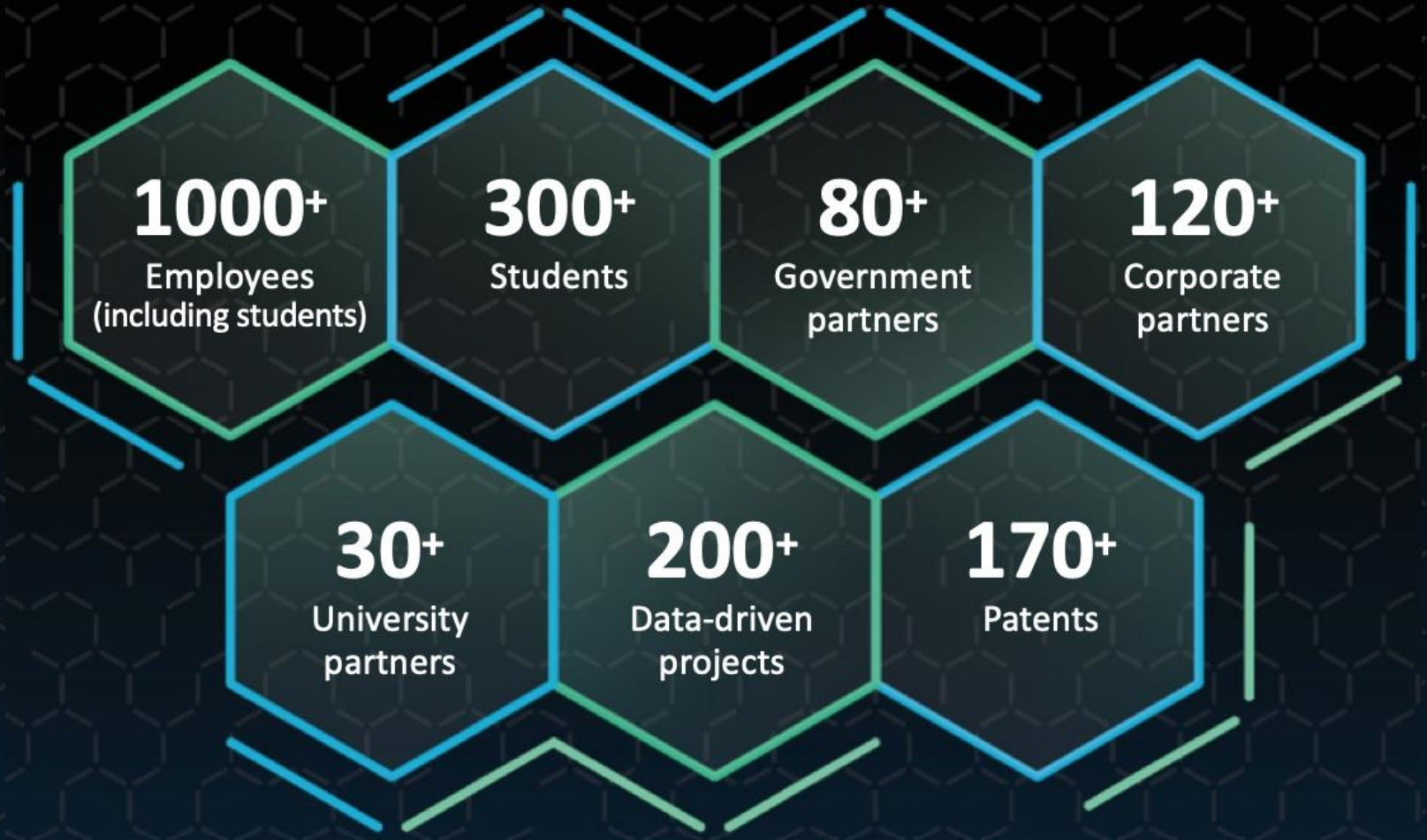
Conjoint Professor, UNSW

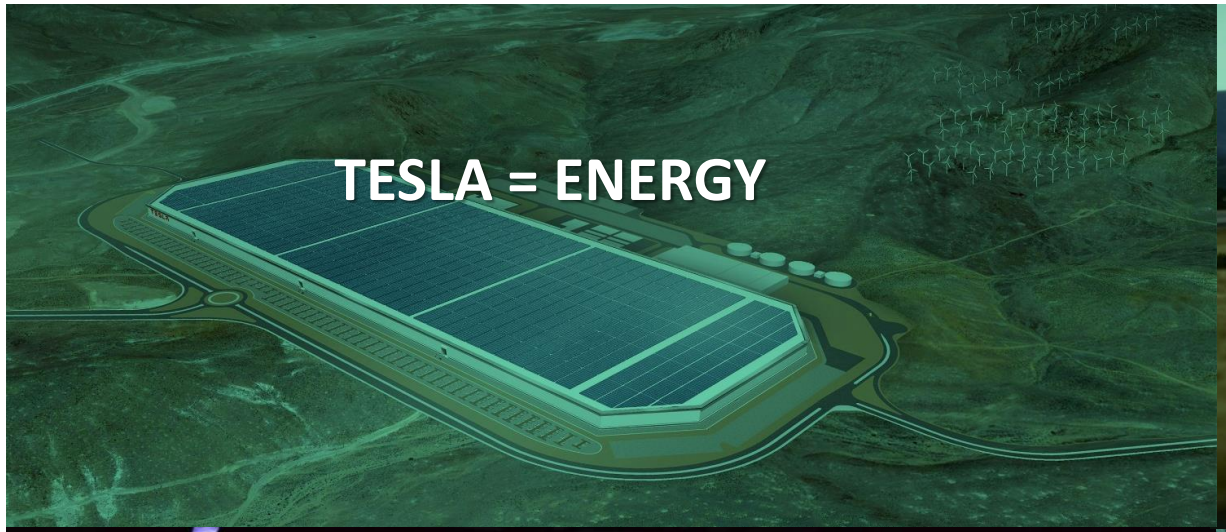
Chair, Blockchain and DLT Committee, Standards Australia

www.data61.csiro.au

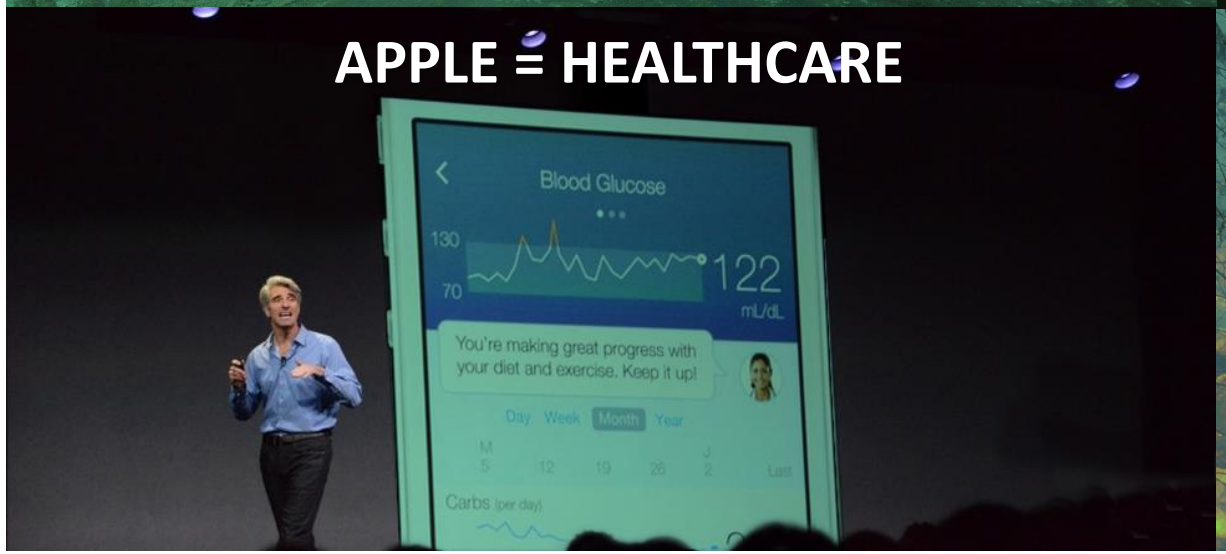
Data61: Australia's Digital Innovation Powerhouse

Creating Our Data-driven Future




TESLA = ENERGY


OTTO
UBER = TRANSPORTATION/LOGISTICS


APPLE = HEALTHCARE
Blood Glucose
130
70
122
mL/dL
You're making great progress with your diet and exercise. Keep it up!
Day Week Month Year
Carbs (per day)


NRMA = SAFETY
NRMA
FIREBLANKET

Trend: Products > (Data/Services) Platforms

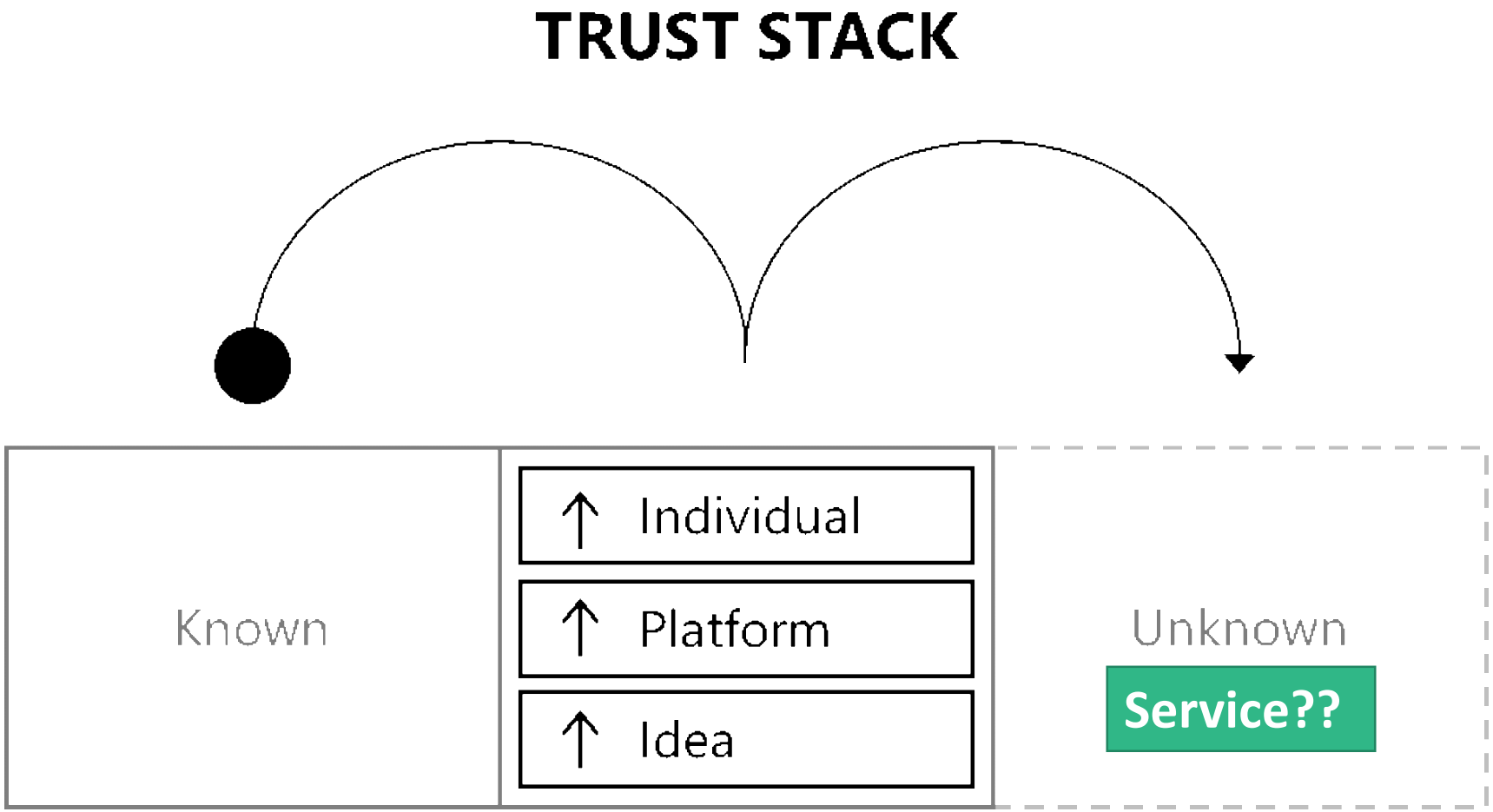
Every industry becomes data/service-driven

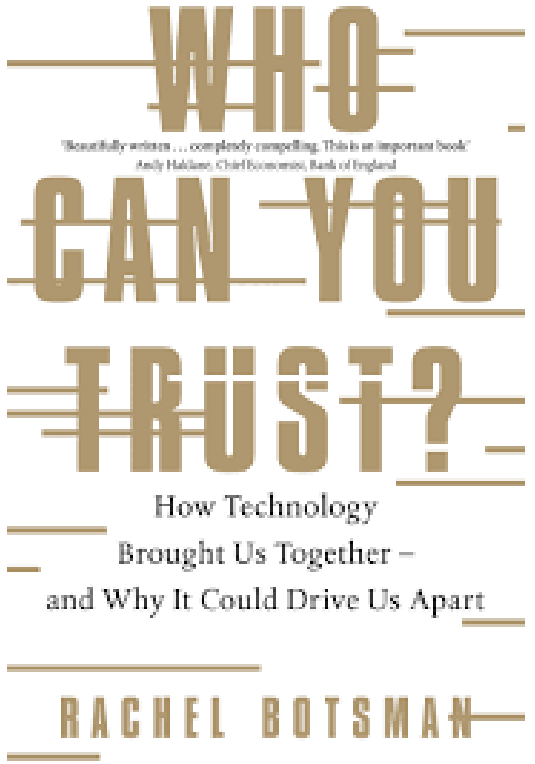
2018				2008			
Rank	Company	Founded	USbn	Rank	Company	Founded	USbn
1.	Apple	1976	890	1.	Exxon	1870	492
2.	Google	1998	768	2.	General Electric	1892	358
3.	Microsoft	1975	680	3.	Microsoft	1975	313
4.	Amazon	1994	592	4.	AT&T	1885	238
5.	Facebook	2004	545	5.	Proctor & Gamble	1837	226

Multi-sided Service Platforms & Platform Economics Become Ubiquitous

Trend: Trust Shifts to Distributed Trust

Local -> Institutional -> Distributed

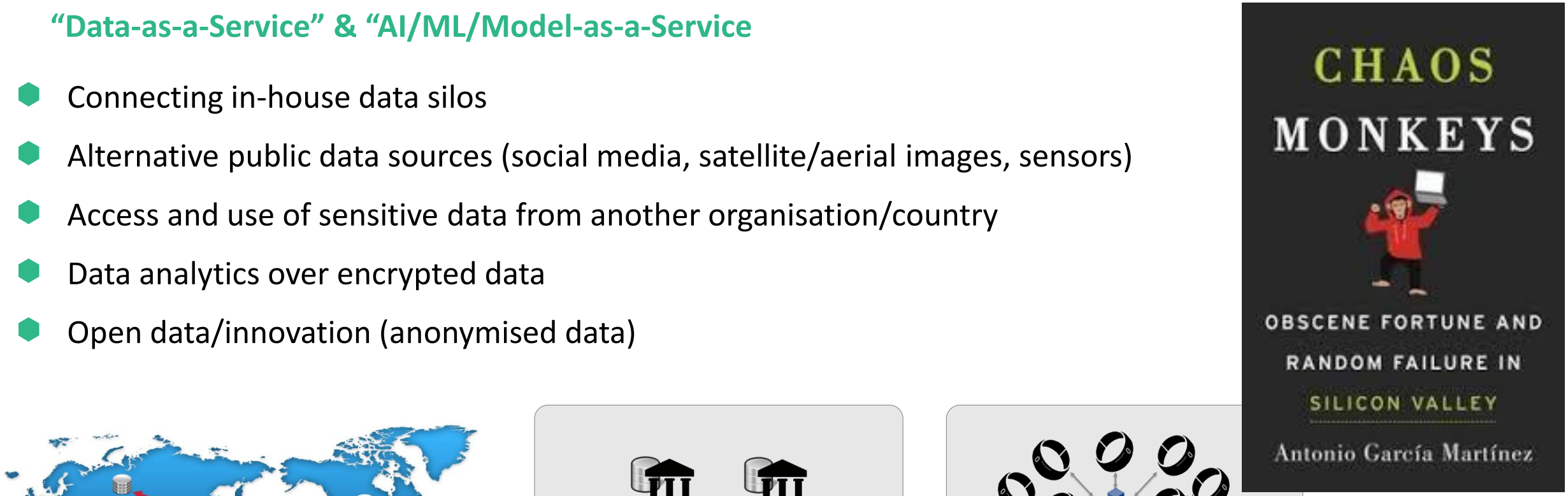


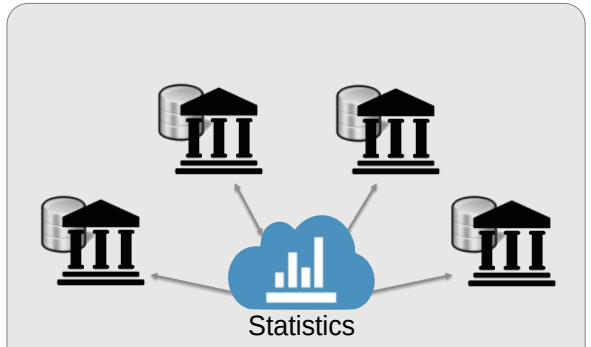


Trend: Value Arises from “Joint Analytics”/Services

“Data-as-a-Service” & “AI/ML/Model-as-a-Service

- Connecting in-house data silos
- Alternative public data sources (social media, satellite/aerial images, sensors)
- Access and use of sensitive data from another organisation/country
- Data analytics over encrypted data
- Open data/innovation (anonymised data)





Trend: Trust through Regulation?

Data Economy: Balancing Innovation & Regulation Burden

New Legislations

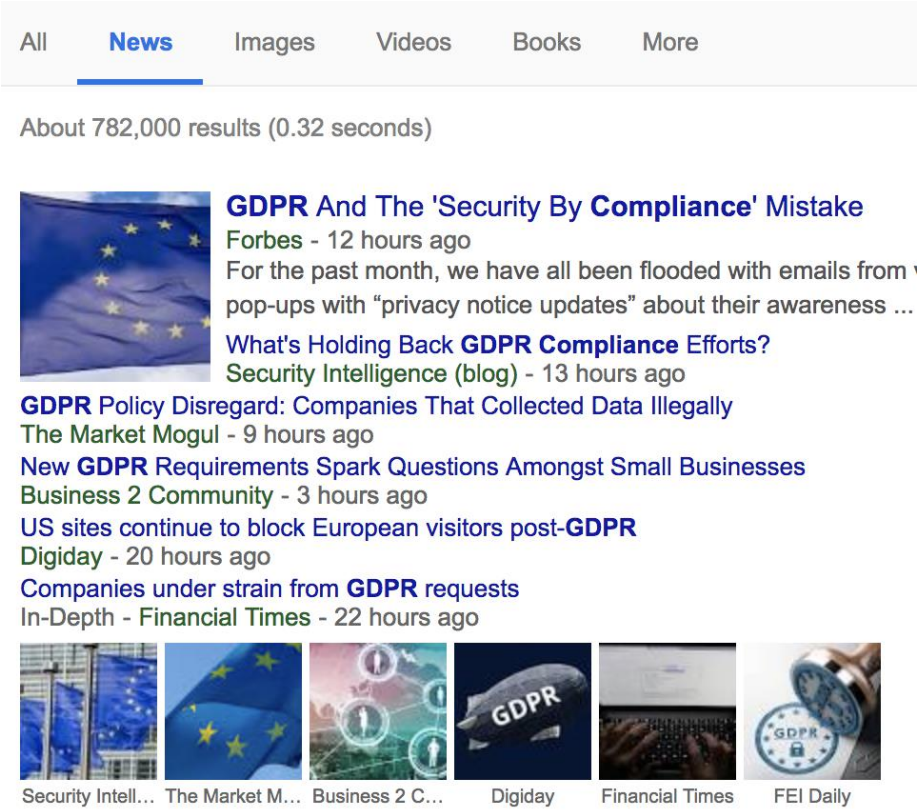
Trust behind **Data Services**?

- Privacy, security and “specific” purpose of use
- GDPR
- Australia
  - Data Breach Notification Scheme
  - Consumer Data Right (CDR): Open Banking, Energy, Telco

New Concerns

Ethical AI – Trust behind **AI Service?**

- Fairness, Accountability, Transparency
- Rights to explanation and redress



Budget 2018: National AI ethics framework on the way

Increased regulation signalled as part of $30m investment.

By Edward Pollitt on May 10 2018 02:20 PM

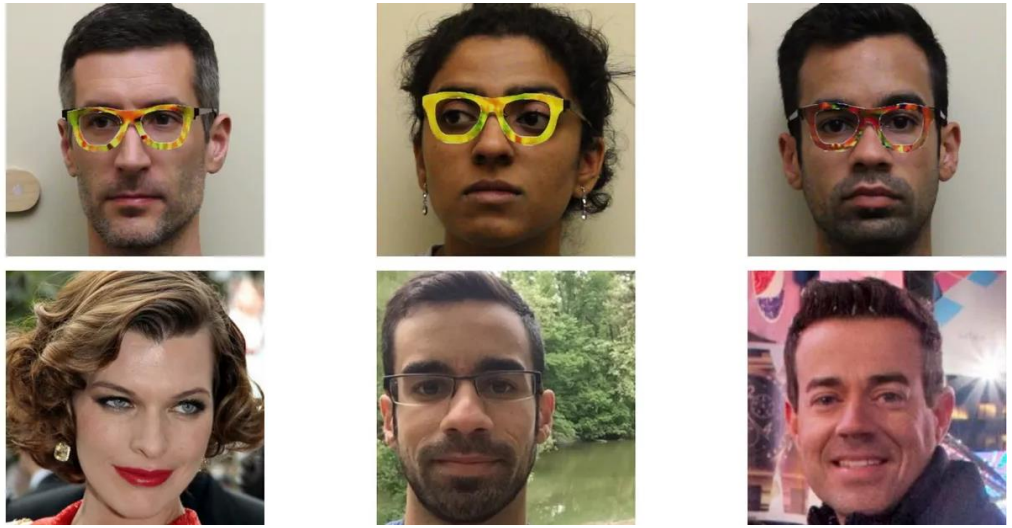
Trend: Challenges in Adversarial AI & Robust AI

Stealing model/data from AI-as-a-Service & Trust in AI-as-a-Service

Hype Cycle 2018: **AI-as-a-Service** on the rise

New threat vectors:

- **Trust in AI-as-a-Service:** counter examples
- **Stealing model/data from AI-as-a-Service:** Info leaking through models, model inversion

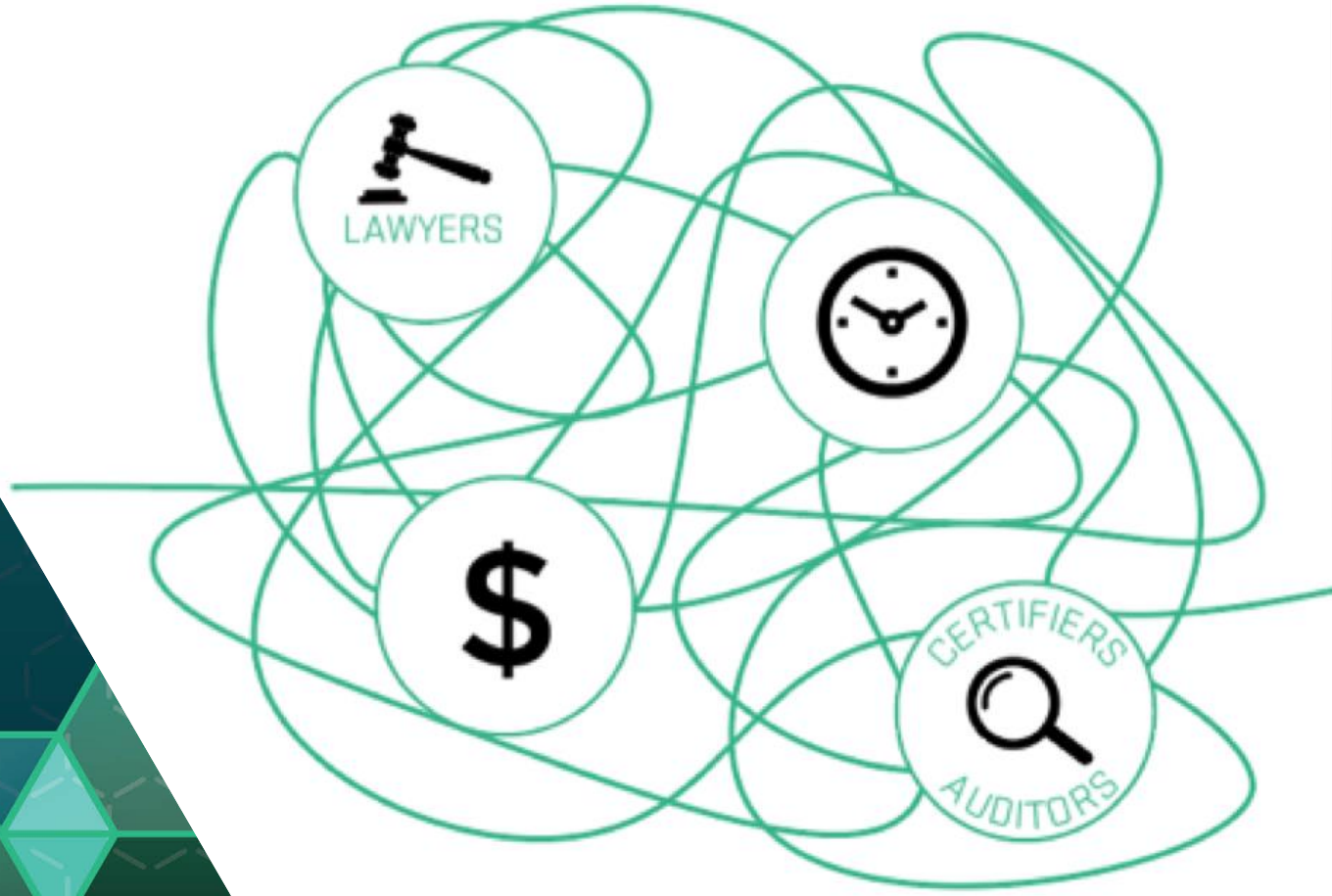


- Impersonating celebrities using printed glasses[1]

- Stop Sign ->
  Speed Limit Sign [2]

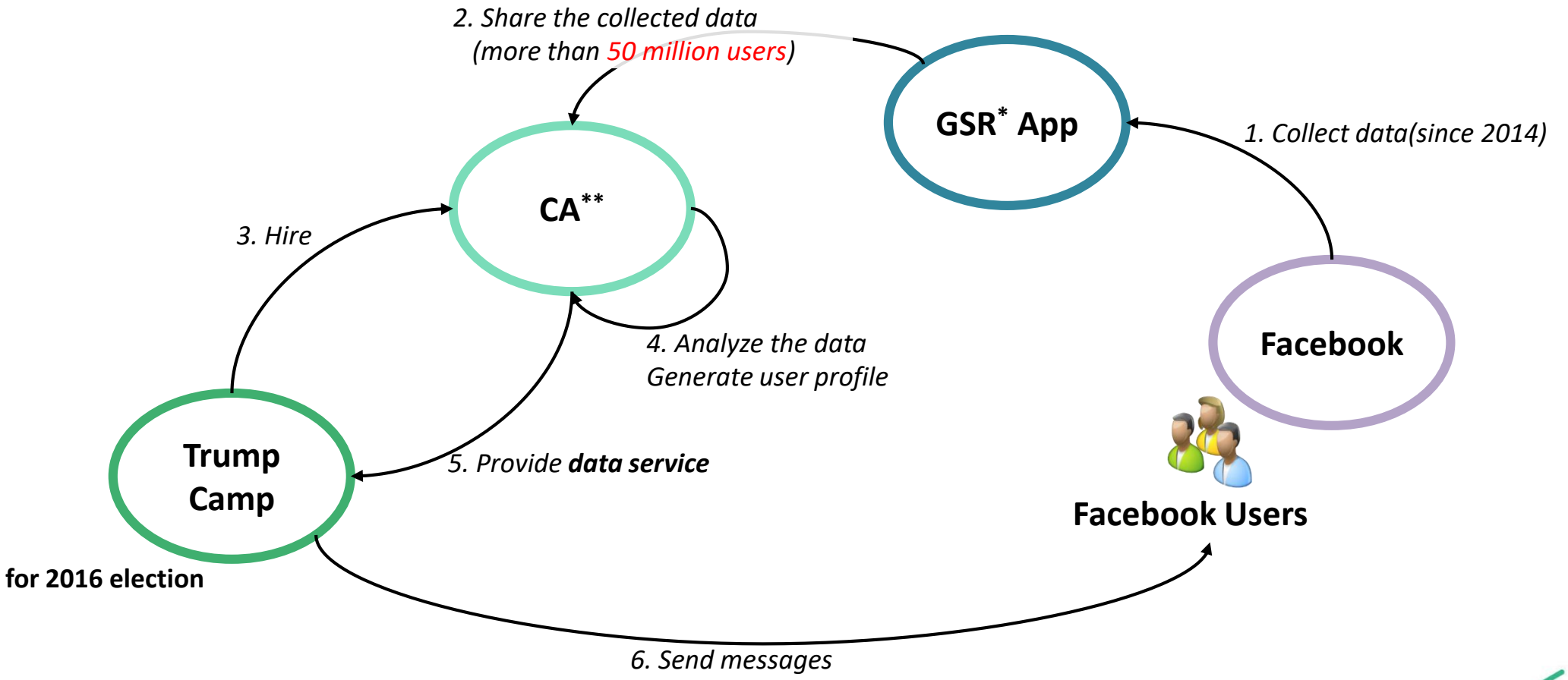
Trust is the New Currency

Service Governance & Regulation



Trust Can be Easily Breached

Facebook-Cambridge Analytica Data Scandal



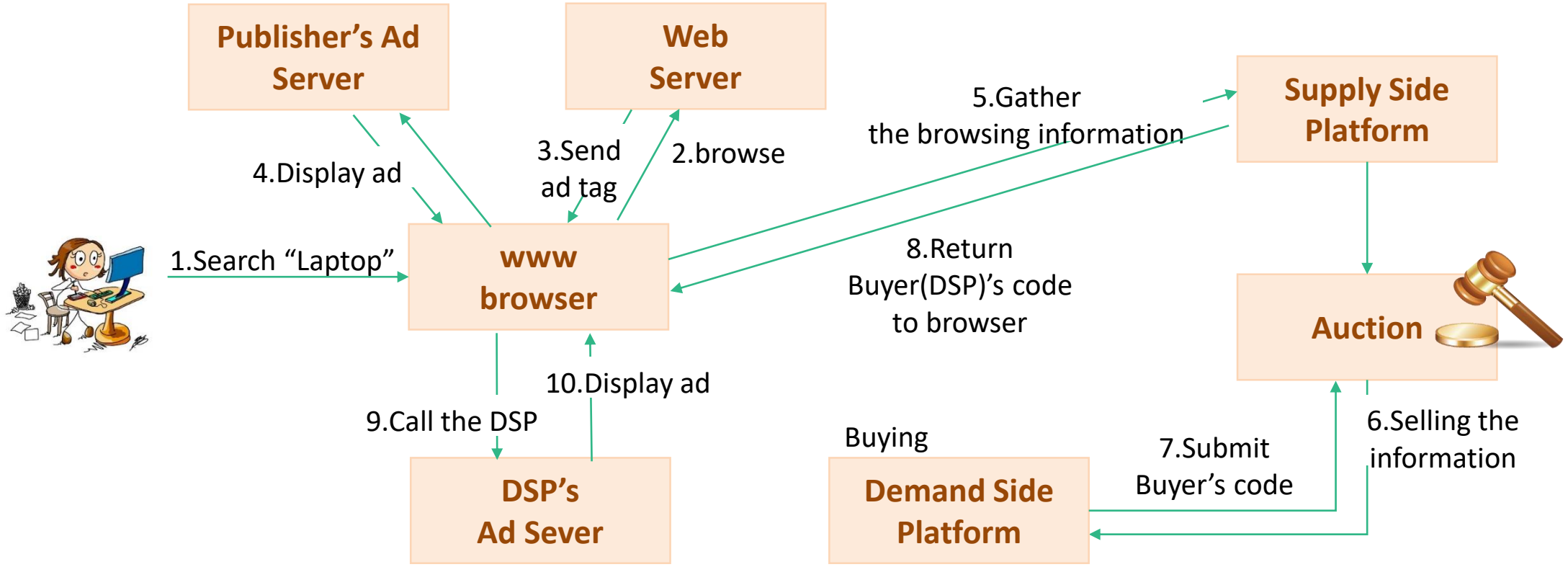
Consequence of Loss of Trust

- ✓ Facebook shares fell 13%.
- ✓ The reputation of Facebook is dropping.
  - * According to one survey, 1/3 respondent said “will leave Facebook”.

    MS (50%), Snap (46%), Uber (40%), Google (30%), Amazon (34%)
  - * Total participants are 2,600 IT employees.
- ✓ Facebook faces investigation (US, UK, Australia…).

Data Services & Flow in “Platform Ecosystem"

Targeted Advertising



Data Governance Needs in Platform Ecosystem

✓ Clarify data ownership/**rights** and data uses.

- Define clear roles, responsibility and data uses (for data owner/subject right protection).
- Extend and reinforce the roles and responsibilities for secondary use.

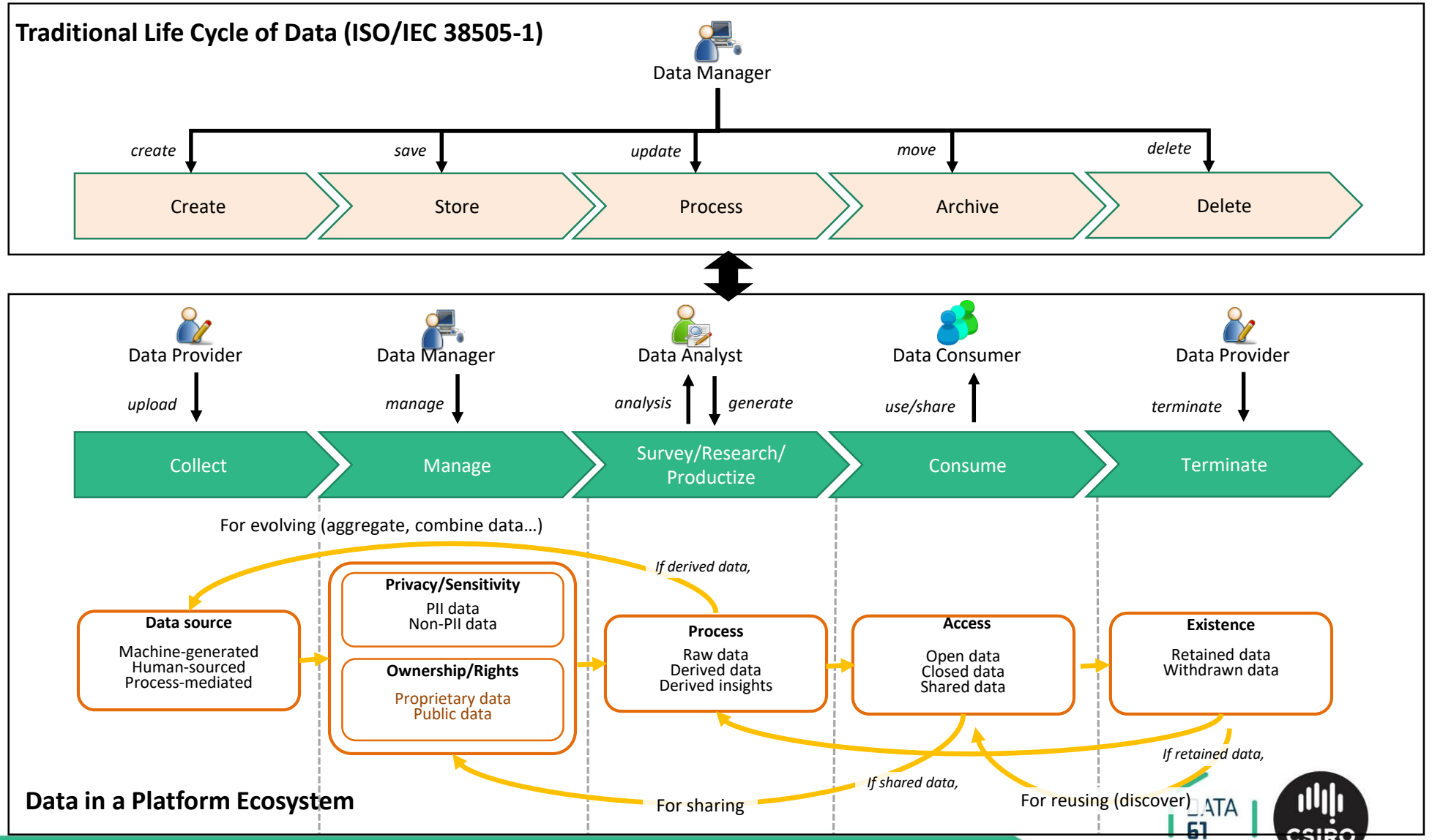
✓ Make data platform/services visible.

- Increase transparency of platforms/services and improve trust.
- Monitor, audit and trace all the activities taken place, and open to every stakeholders.

✓ Build and implement due processes.

- Include all the considerations of data governance, and make them executable.

New Data Governance Model for Platform Ecosystems

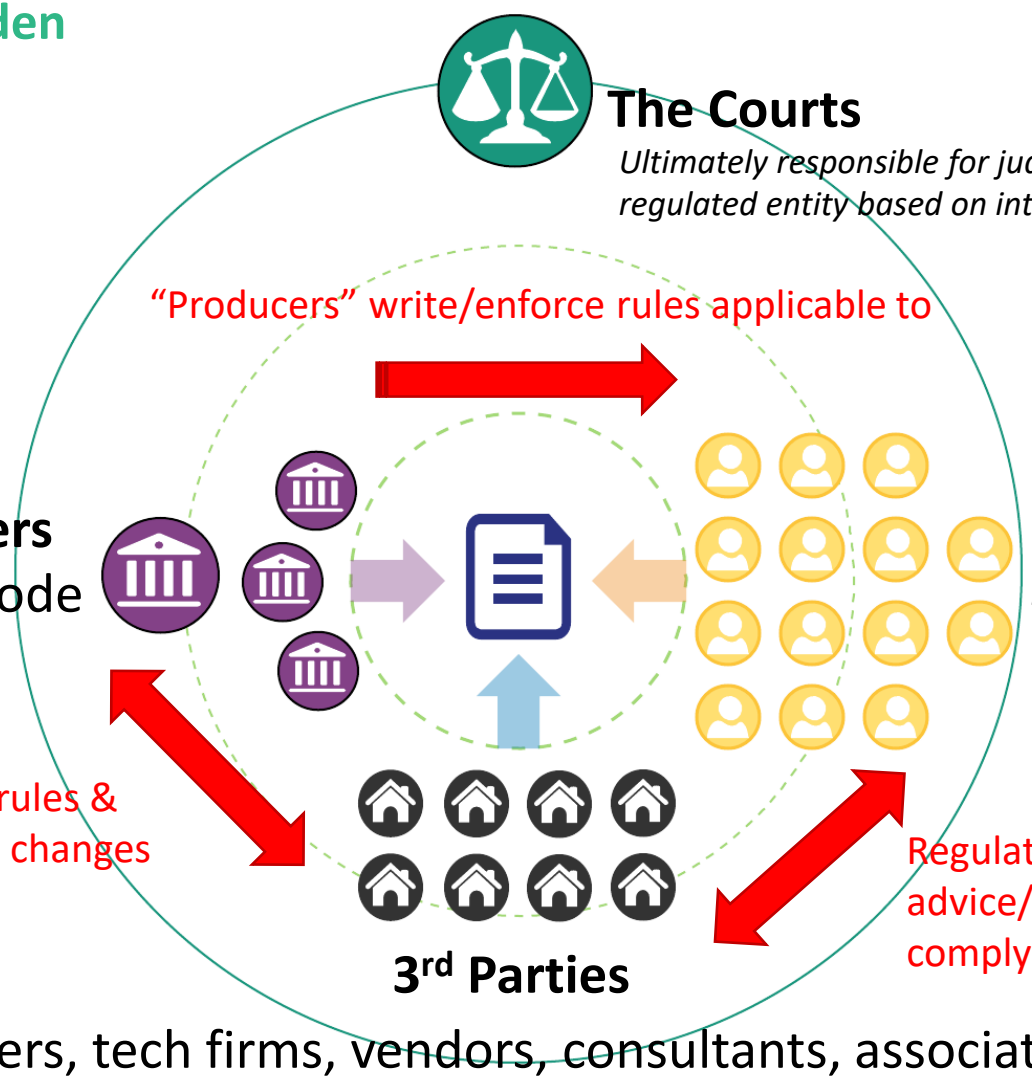


*S. Lee, Ross J., L. Zhu, "**A Contingency-Based Approach to Data Governance Design for Platform Ecosystem", PACIS 2018***

*S. Lee, R. Jeffery, L. Zhu, **"Data Governance Decisions for Platform Ecosystems", HICSS 2019***

What about Trust through Regulation?

Innovation vs. regulation burden



Legislation/Regulation-as-a-Services/Platform

Encode trust into digital rules once & benefit everyone through services

A service-oriented platform to help businesses …

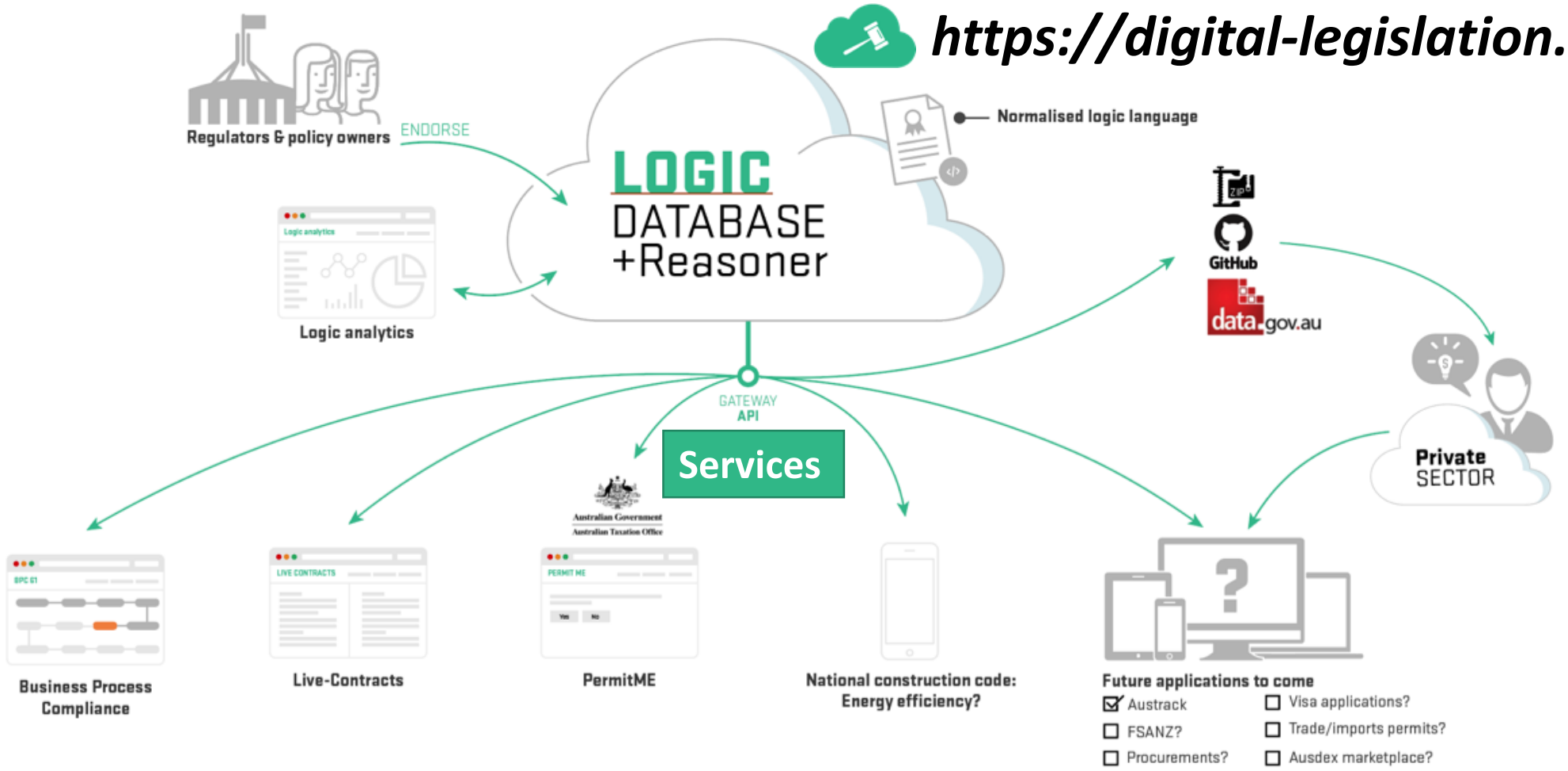
Link regulatory requirements to business process/services. Reduce manual effort and auditing costs

Untangle regulation and allow appropriate business outcomes from regulation changes

Assure compliance efficiently for business, auditors and regulators

Ensure consistent scalable experience

Digital Legislation as Services



Consumer Data Rights

How can consumer trust their data to organisations and their specific use

- **Australia's new legislation impacting consumer data and its services**
- Consumers can authorise 3rd parties to access their data
- Currently 3 designated sectors
  - Banking first (similarly to PSD II in Europe/Open Banking in UK)
  - Energy, Telecommunication to follow
  - Potentially applicable to all sectors which have consumer data

- Data61's Role
  - Setting Service/API standards + Security profiles standards
  - Advising on "directory" (related to service directory)

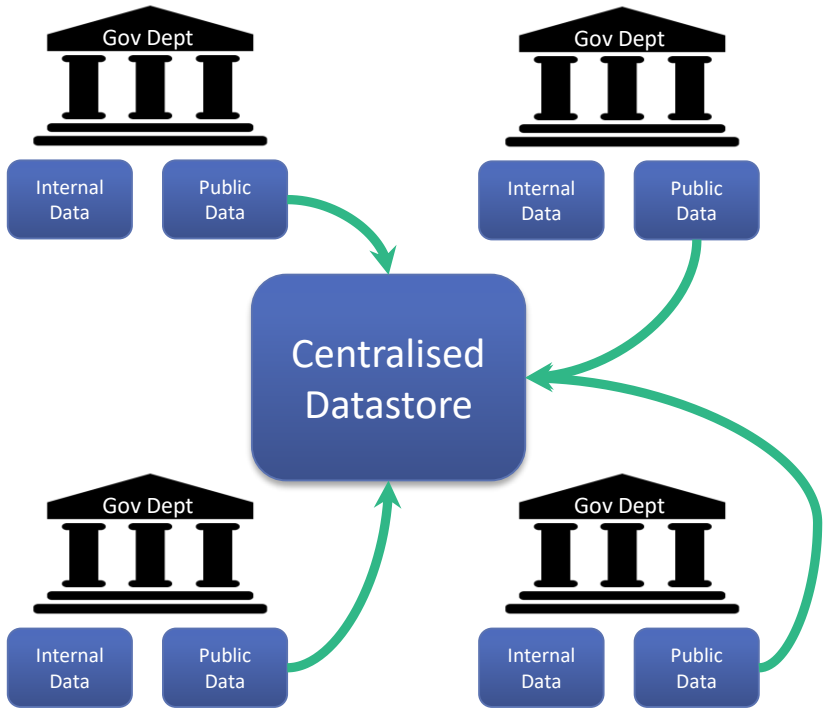
https://consumerdatastandards.org.au

Trust in Data Services

Distributed Trust through Federated Model

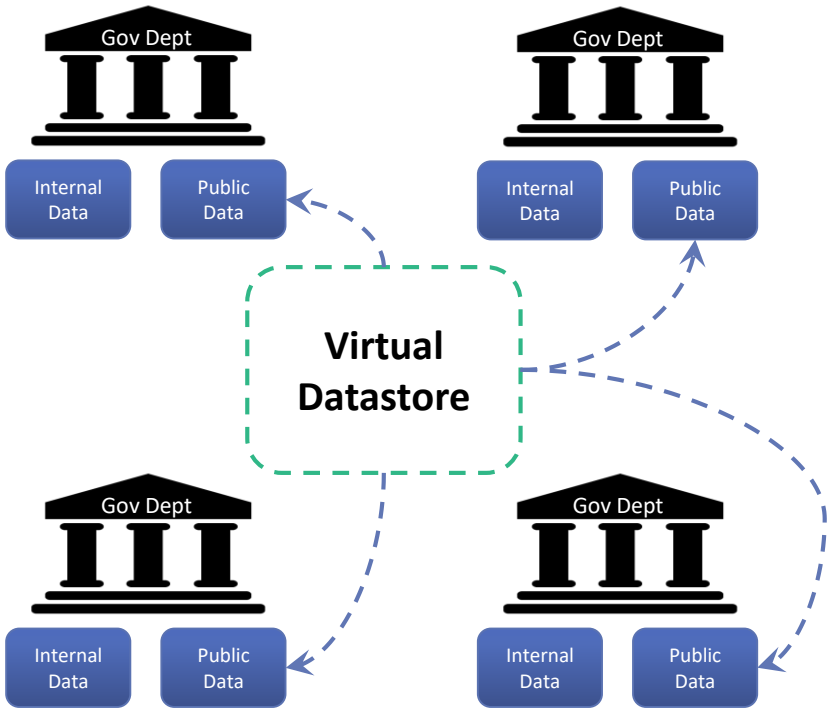
Joint analytics without moving data together

Centralised Model



Data is collected and analysed centrally

Federated Model

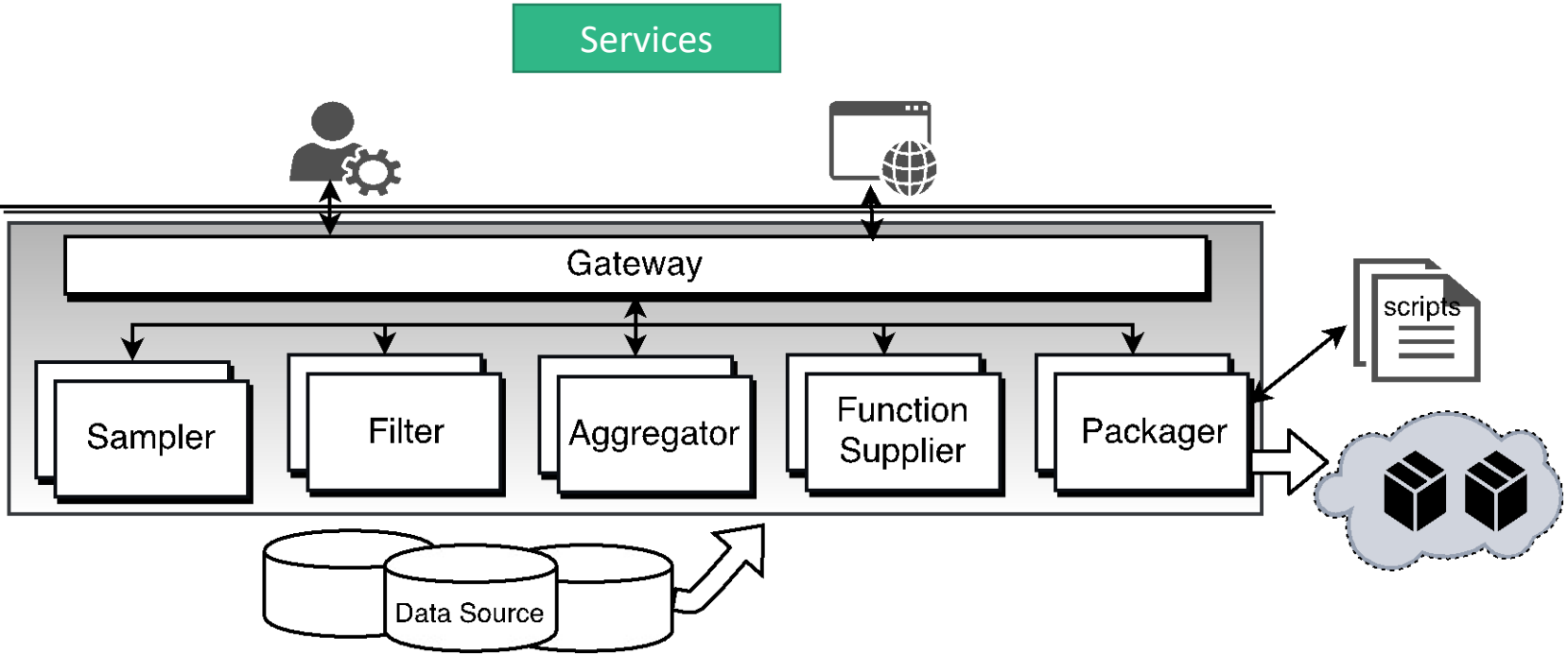


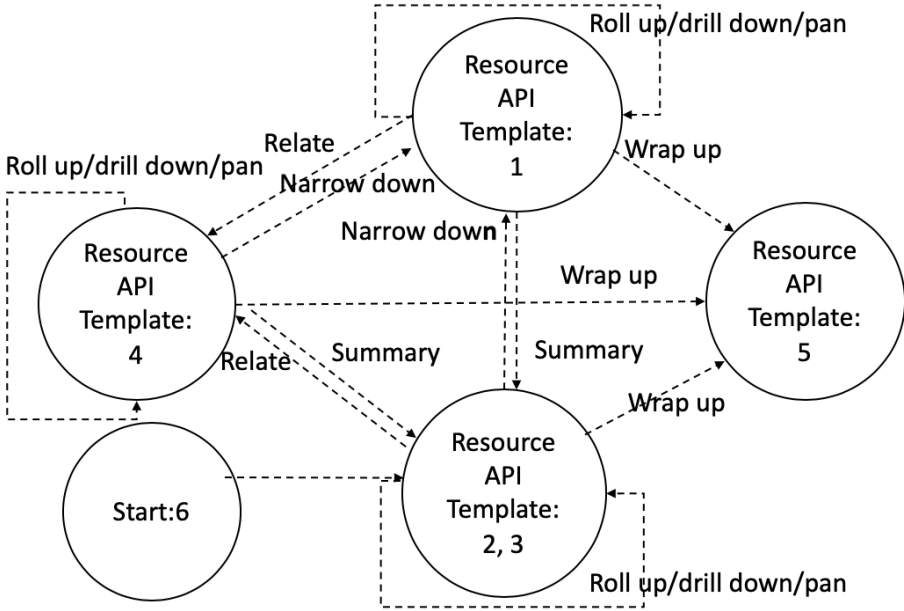
Data remains with its source department

(Data) Service API for “Analytics”

RESTful/HATEOS Services for Data Exploration/Analytics

Motivation: Data Service APIs were designed for retrieving info and perform transactions, *not* for Exploration & Analytics





Yun Zhang, L. Zhu, X. Xu… ***“Data Service API Design for Data Analytics”, SCC 2018***

Confidential Computing

Improving trust through “use without access” /“encryption-during-use”

Homomorphic Encryption

Encrypt (1) “+” Encrypt (2) = Encrypt (3)

New ways to secretly analyse data

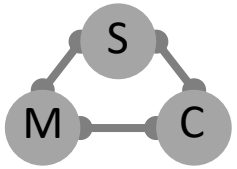
Federated Learning

Mary Smith – 10/4/1998 - 0403674324

00100101111010010100100100001011110101

Private linkage of identities

BLOCKCHAIN



Secure Multiparty Compute

Zero
Knowledge
Proofs

Many others!

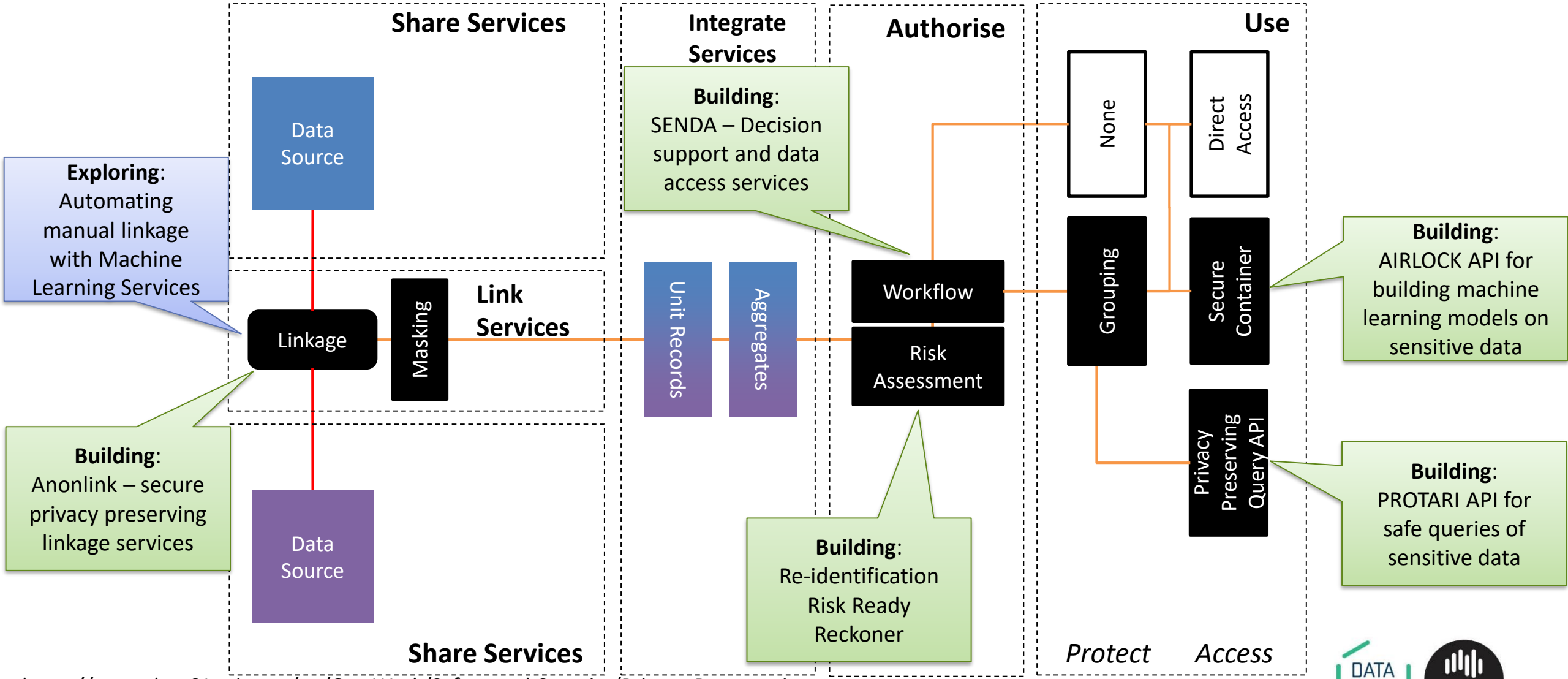
https://www.n1analytics.com

Data Service Provider's Constraints

Spectrum of Technologies to Solve the Problems

Desired output	Service Providers	Service Constraints	Solutions
Linked unit record data	Data custodian <-> Data custodian	Cannot share identifiers	Privacy Preserving Record Linkage
Linked unit record data	Data custodian <-> Data custodian	Cannot share unit record data	Confidential computing
Linked unit record data	Data custodian <-> Data custodian	Cannot store linked dataset	Virtual datasets
Query result over linked data	Data custodians <-> Data analysts	Cannot share unit record data	Confidential computing Secure & obfuscated API
Model trained over linked data	Data custodians <-> Data analysts	Cannot share unit record data	Confidential computing Data Airlock
Linked unit record data	Data custodians <-> Data analysts	Negligible re-identification risk	Privacy risk assessment & defense
Query result over linked data	Data custodians <-> Data analysts	Negligible re-identification risk	Privacy risk assessment & defense Secure & obfuscated API
Linked unit record data	Data custodians <-> Data analysts	Must retain control over data and environment	Secure containers

Enhancing Trust throughout Service API Life Cycle



https://www.data61.csiro.au/en/Our-Work/Safety-and-Security/Privacy-Preservation

Trust in ML/AI Services

Defend Trust by Protecting Models (not just data)

Against Adversarial Machine Learning/AI

New threat vectors

- Poisoning
- Adversarial examples
- Model inversion/Stealing through Services

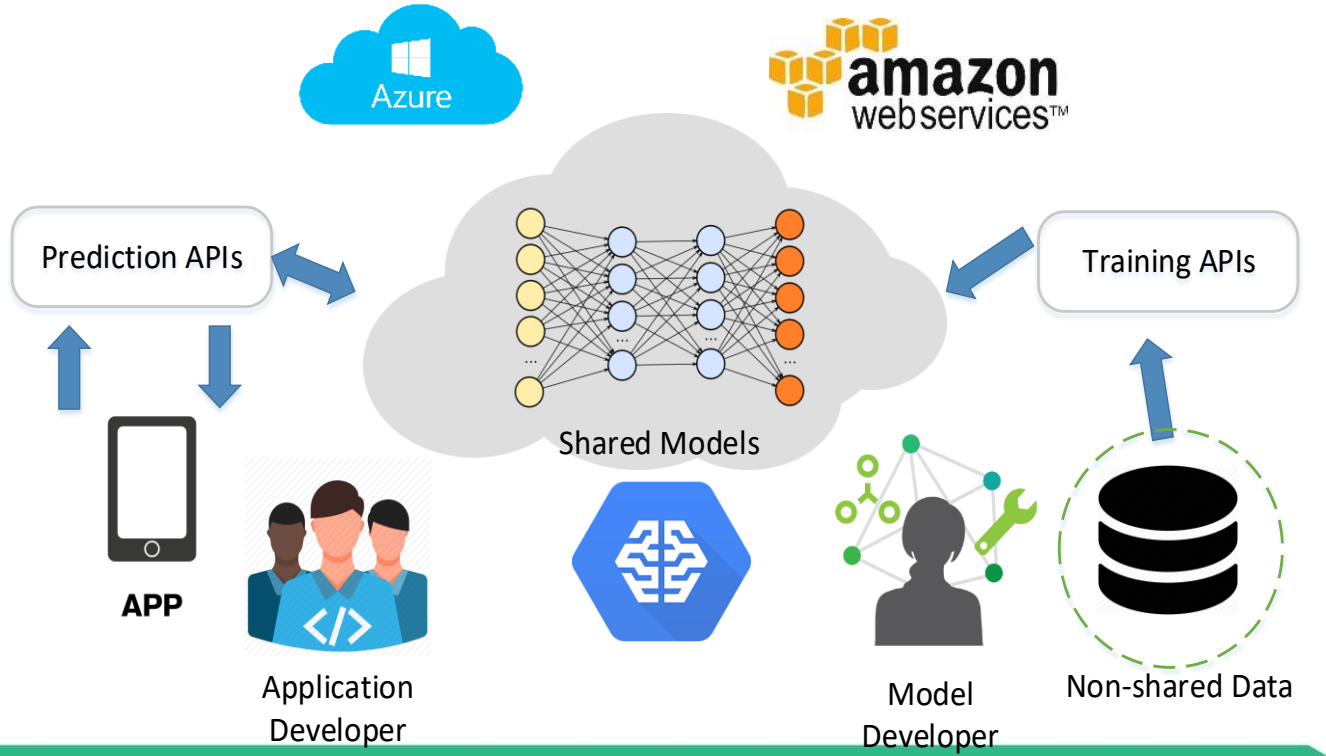
Solutions

- Game theory
- Data integrity and provenance
- Random projection & noise injection

Improve Trust through Sharing Models/Interpretation

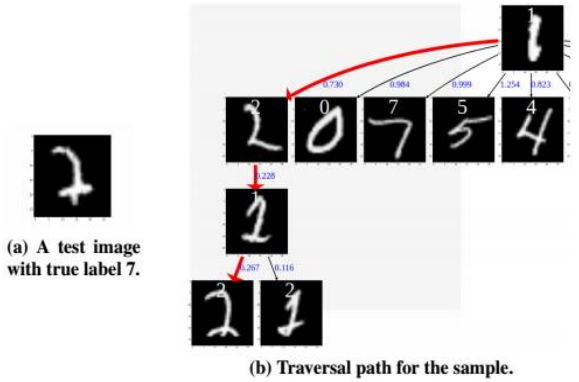
Sharing Deep Learning Models with Interpretation

- Models are shared in MLaaS (ML-as-a-Service) platforms
  - Deploy models as accessible query APIs
  - Models are *IP*

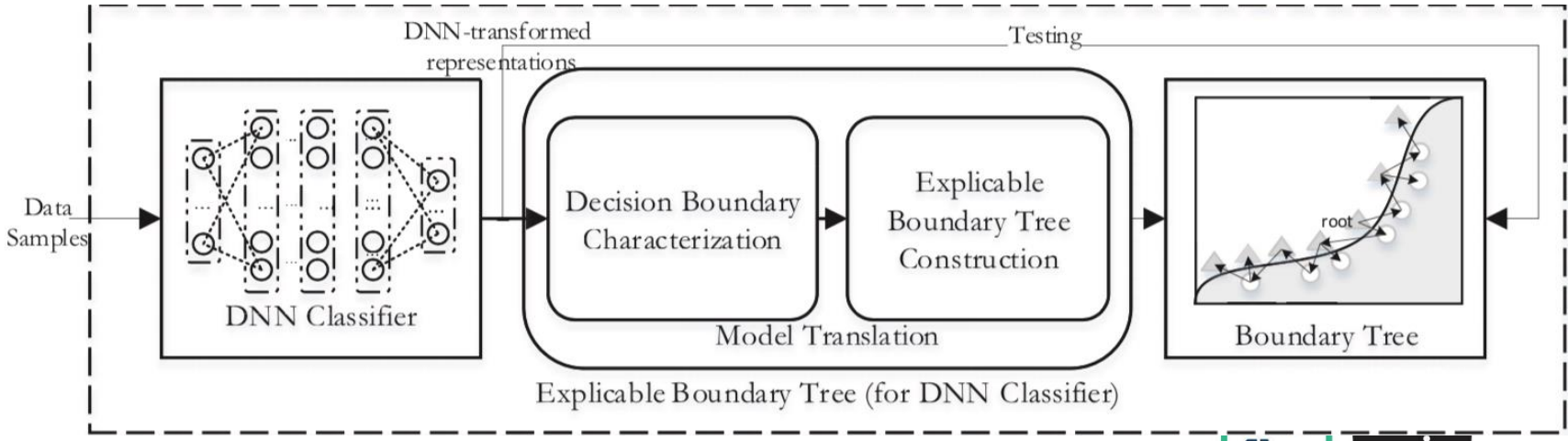


Sharing “Boundary Tree”

- DNN decision making remains a black-box
  - Too complicate to explain; training data dependency
  - There is no clear specification of a DNN model
- Providing interpretability to a model is helpful
  - Enable a user to get insight about model behaviours



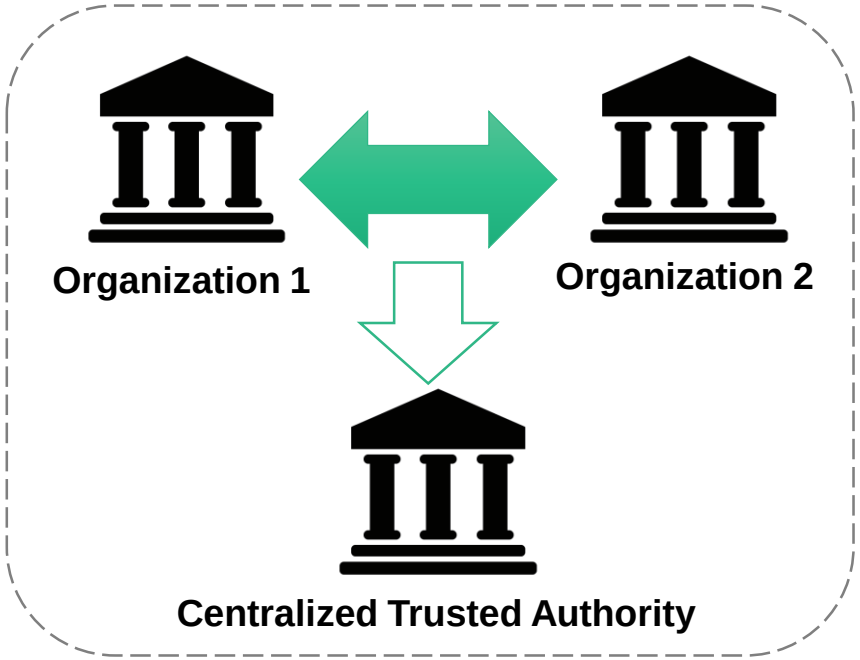
Huijun Wu, Chen Wang, Jie Yin, Kai Lu and Liming Zhu, ***"Sharing Deep Neural Network Models with Interpretation", WWW 2018***



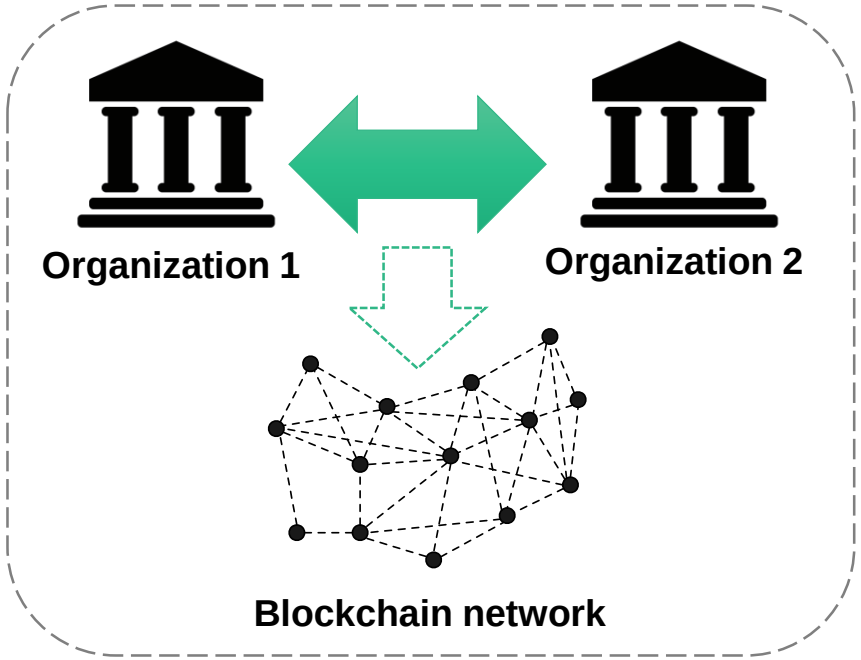
Trust in Service in a Trustless Environment

Blockchain

Shifts/Spreads Trust



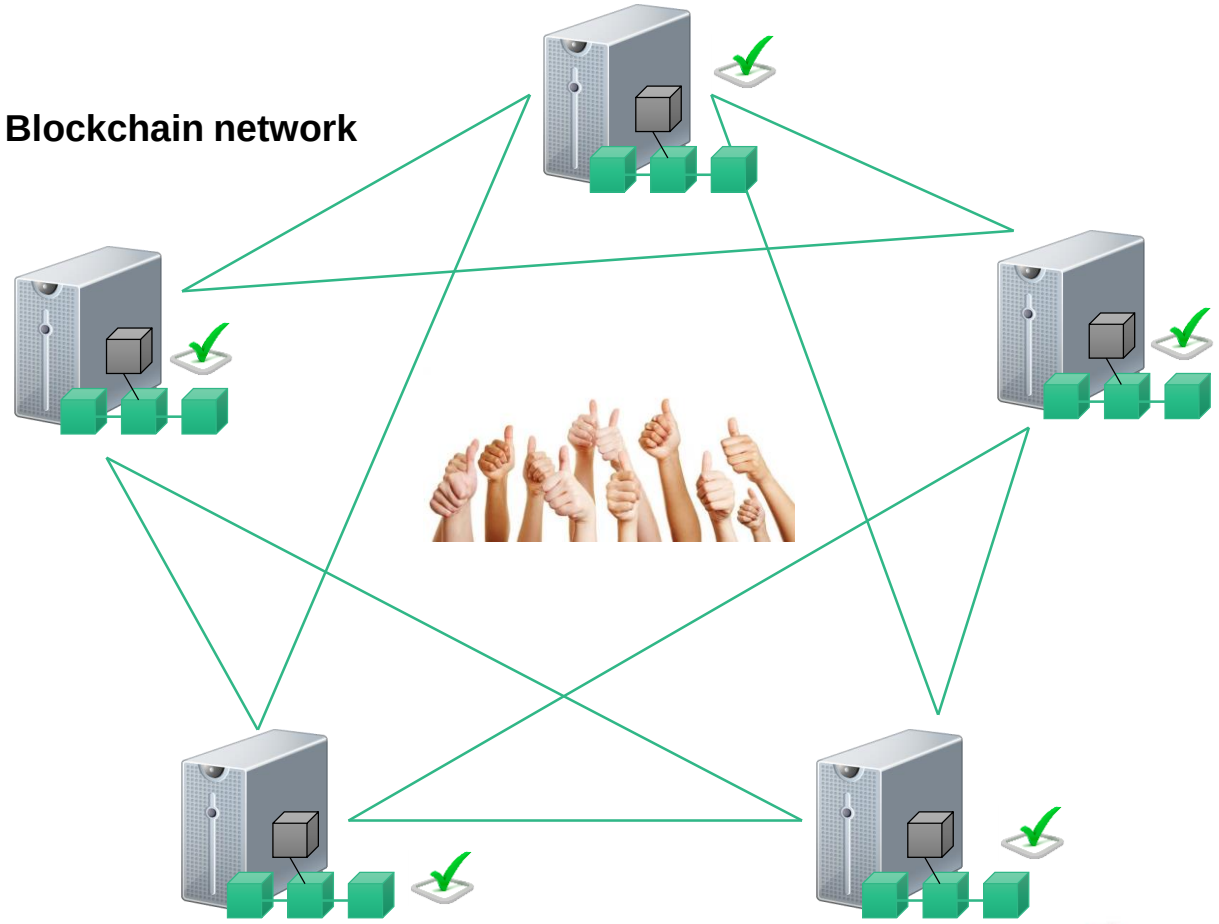
Traditional trusted environment



Blockchain trustless environment

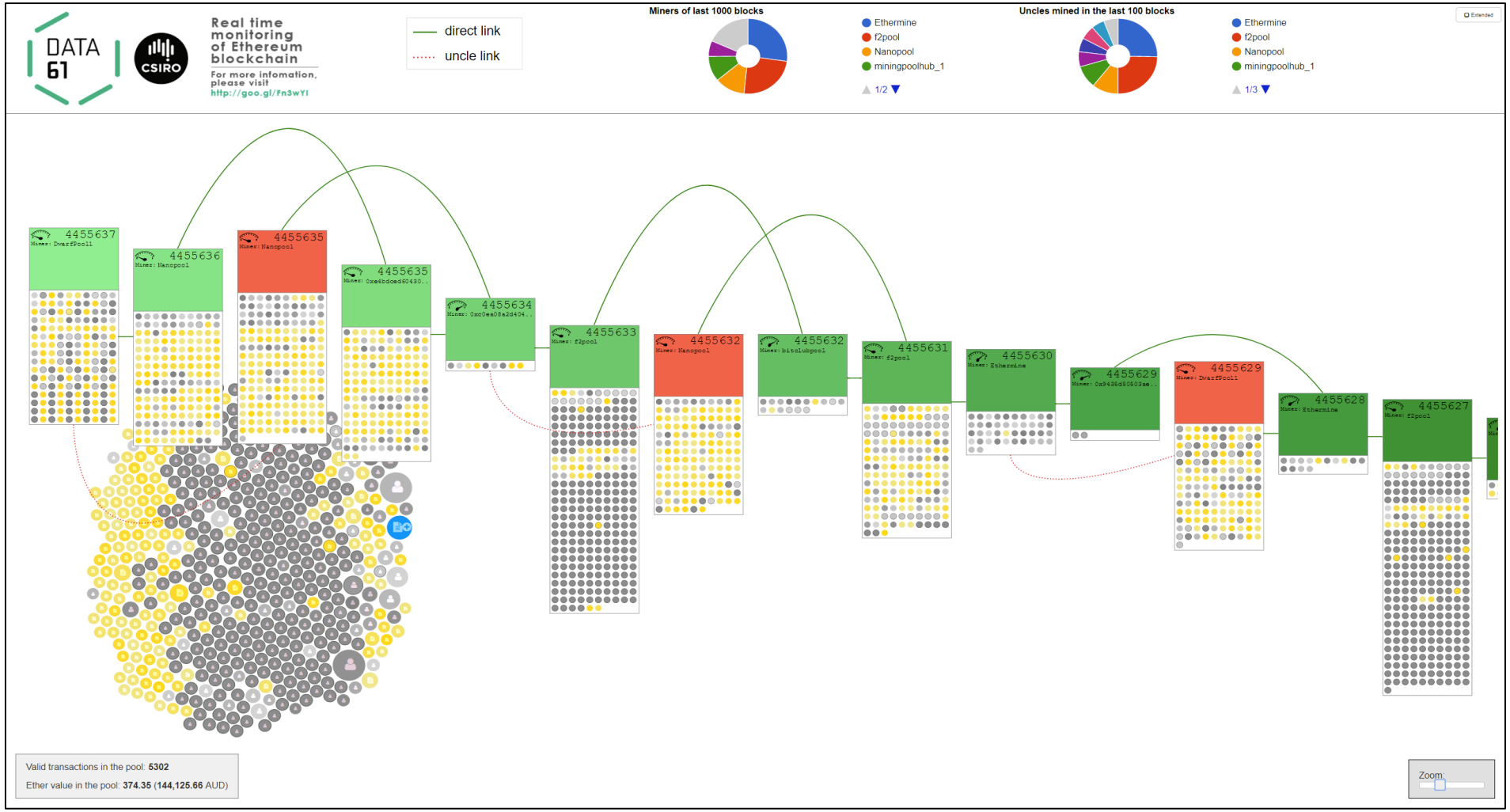
How?

- Immutable database
- Every node hosts a replica
- Transaction is verified by every node
- Combination of knowledge from
  - Distributed Systems,
  - Peer-to-Peer,
  - Cryptography
  - Incentive Systems
  - Game Theory



Visualization of a Blockchain http://ethviewer.live

Empirical Analysis of Transaction Data



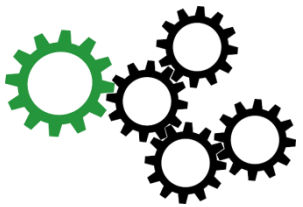
Blockchain 2nd gen – Smart Contracts

From Data to Programs/Services

- 1st gen Bitcoin transactions are financial transfers

- Blockchain ledger can also store/transact any kind of data

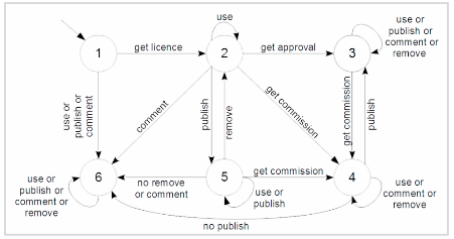
- Blockchain can deploy and execute programs: Smart Contracts
  - User-defined code, deployed on and executed by whole network
  - Can enact decisions on complex business conditions
  - Can hold and transfer assets, managed by the contract itself
  - Ethereum: pay per assembler-level instruction



Blockchain Research at Data61

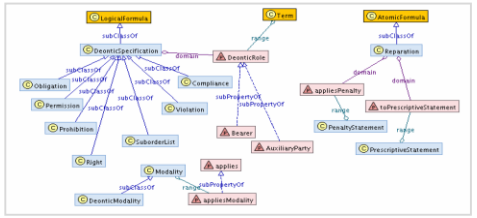
Improve Trust through Formal and Empirical Studies

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

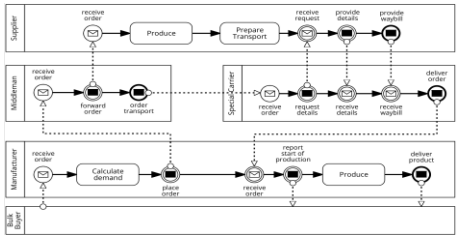


  - Empirical

- Defining & Using Smart Contracts
  - As legal contracts



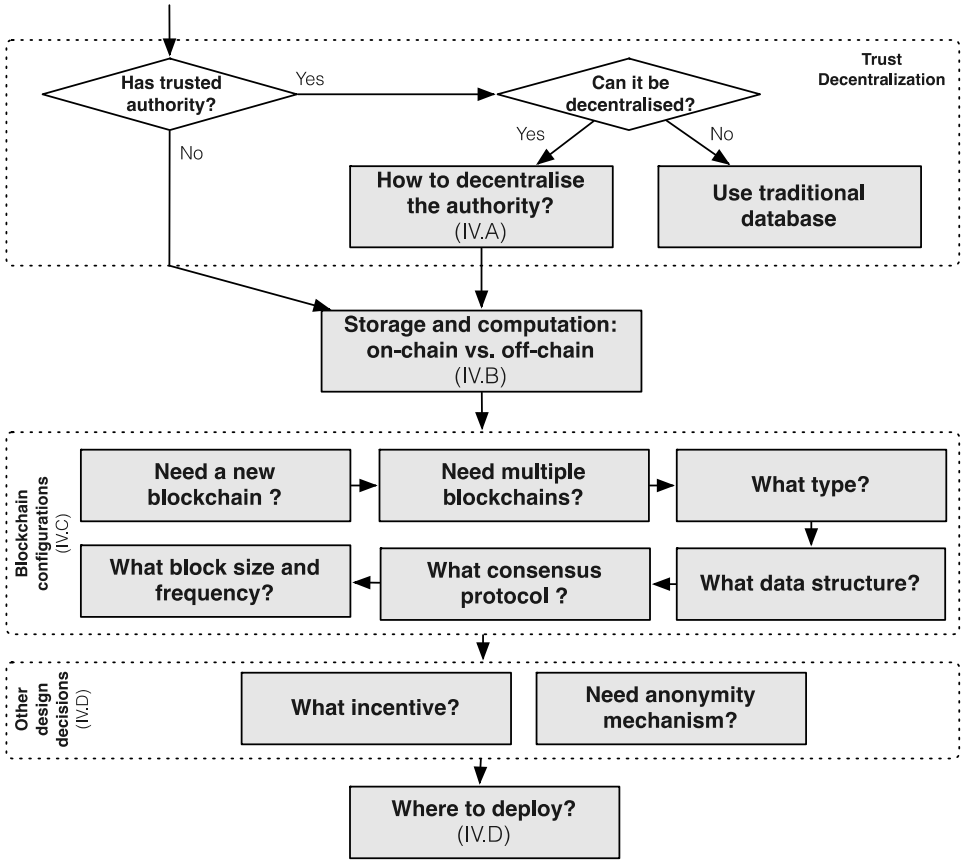
  - Cross-org business processes



Designing Trust with Blockchain (1/2)

Design process, quality analysis, design patterns and governance/risks

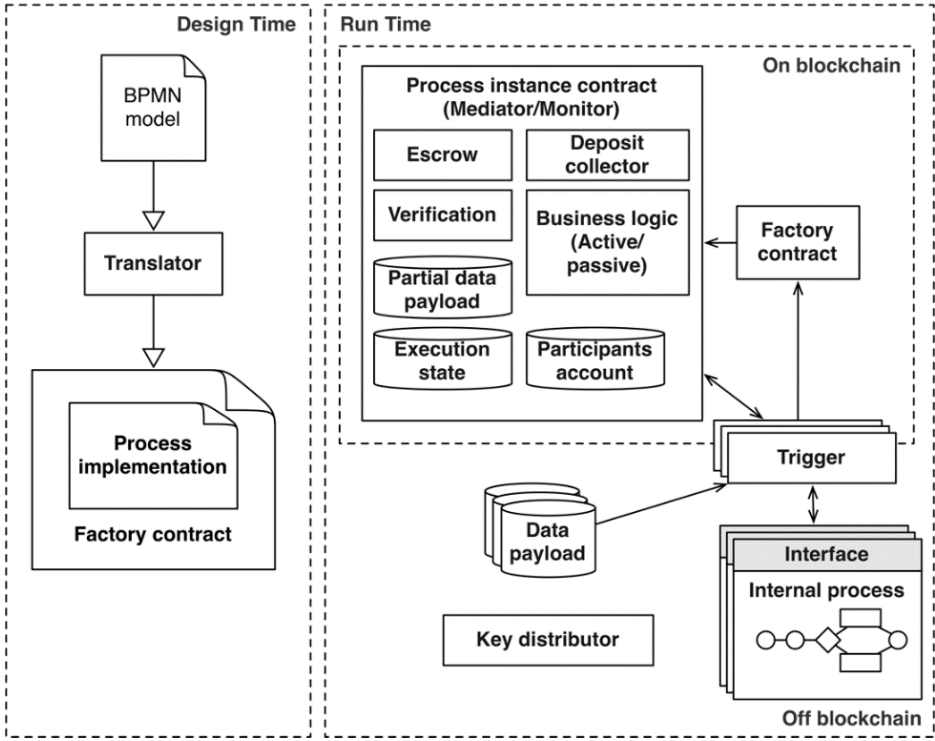
- **Design Process, including Suitability Analysis**
  - **A taxonomy of blockchain-based systems for architecture design,** X. Xu, I. Weber, M. Staples et al., ICSA2017.
  - **The blockchain as a software connector,** X. Xu, C. Pautasso, L. Zhu et al., WICSA2016.
- **Quality Analysis**
  - **Quantifying the cost of distrust: Comparing blockchain and cloud services for business process execution.** P. Rimba, A. B. Tran, I. Weber et al., Information Systems Frontiers, accepted August 2018 (previously SCAC 2017)
  - **Comparing blockchain and cloud services for business process execution,** P. Rimba, A. B. Tran, I. Weber et al., ICSA2017.
  - **Predicting latency of blockchain-based systems using architectural modelling and simulation**, R. Yasaweerasinghelage, M. Staples and I. Weber, ICSA2017.
- **Design Patterns**
  - **A pattern collection for blockchain-based applications.** X. Xu, C. Pautasso, L., Q. Lu, and I. Weber, EuroPLoP 2018
- **Integration with other systems**
  - **EthDrive: A Peer-to-Peer Data Storage with Provenance,** X. L. Yu, X. Xu, B. Liu, CAISE2017.
- **Governance and risk management**
  - **Risks and Opportunities for Systems Using Blockchain and Smart Contracts,** Treasury report



Designing Trust with Blockchain (2/2)

Cross-org focused, Process/Data/Assets/Artifact-based model-driven engineering

- **Business Process Execution**
  - **Untrusted business process monitoring and execution using blockchain**, I. Weber, X. Xu, R. Riveret et al., BPM 2016
  - **Optimized Execution of Business Processes on Blockchain**, L. García-Bañuelos, A. Ponomarev, M. Dumas, Ingo Weber, BPM 2017
  - **Caterpillar: A blockchain-based business process management system,** O. López-Pintado, L. García-Bañuelos, M. Dumas, and I. Weber, BPM 2017 Demo
  - **Runtime verification for business processes utilizing the Bitcoin blockchain,** C. Prybila, S. Schulte, C. Hochreiner, and I. Weber, Future Generation Computer Systems (FGCS), accepted August 2017
- **Data / Asset Modelling**
  - **Regerator: a Registry Generator for Blockchain,** A. B. Tran, X. Xu, I. Weber, CAISE 2017 Demo
- **Combined Asset & Process Modelling**
  - **Lorikeet: A Model-Driven Engineering Tool for Blockchain-Based Business Process Execution and Asset Management** A. B. Tran, Q. Lu, I. Weber, BPM 2018 Demo



Red Belly Blockchain (Data61 & USyd)

Achieving Trust at Scale

- New technology particularly for private / consortium blockchain
- ~660,000 tps with 300 nodes in one data center, and > 50,000 tps with globally distributed nodes

http://redbellyblockchain.io

Projects with Australian Treasury

DLT Foresight

What might plausibly happen, across society & economy?

Technical Risks & Opportunities

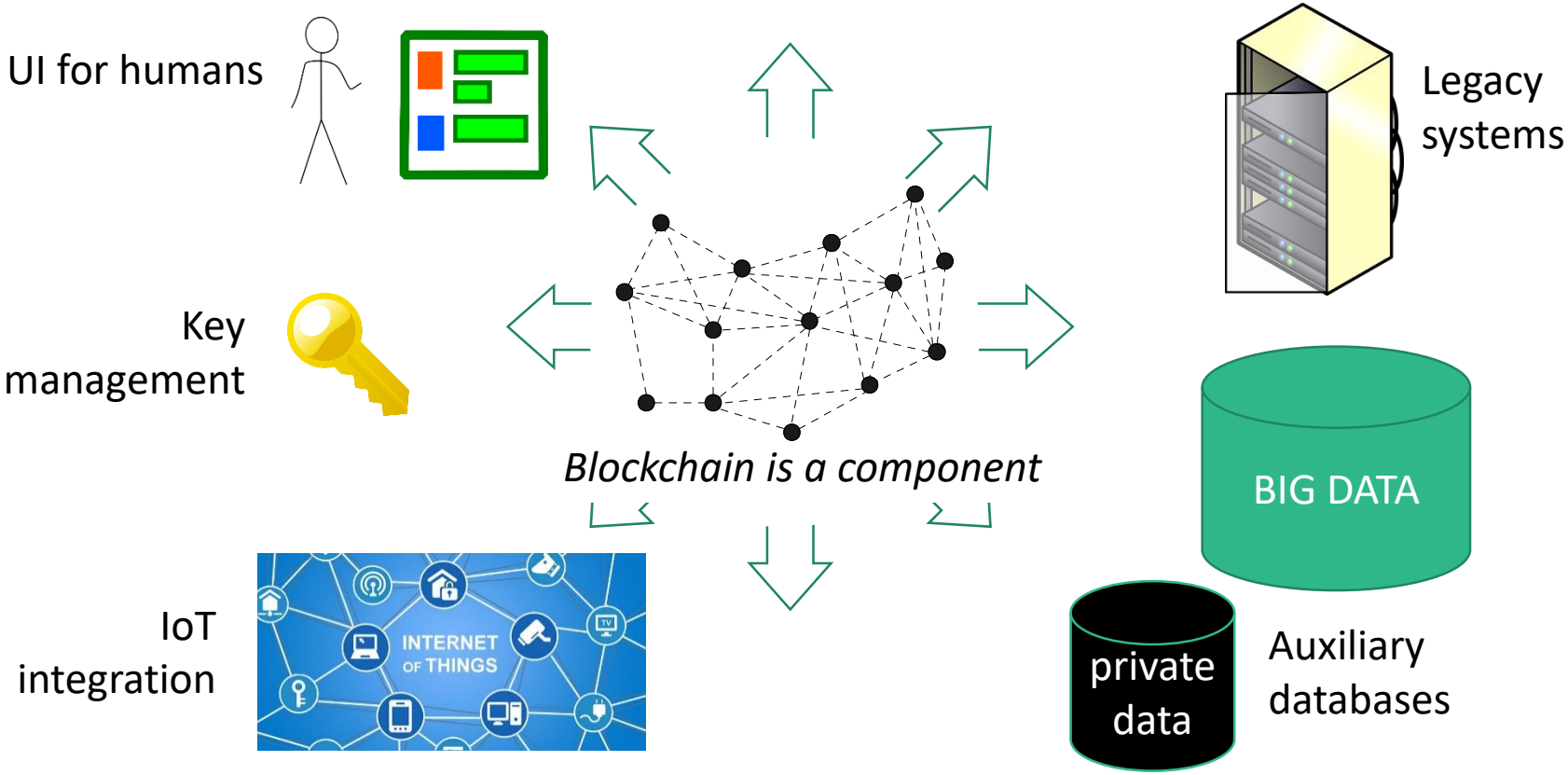
How do needs in various use cases fit blockchain's capabilities?

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

Service Composition using Smart Contract

Blockchains are Not Stand-Alone Systems

Service-oriented approach for connection

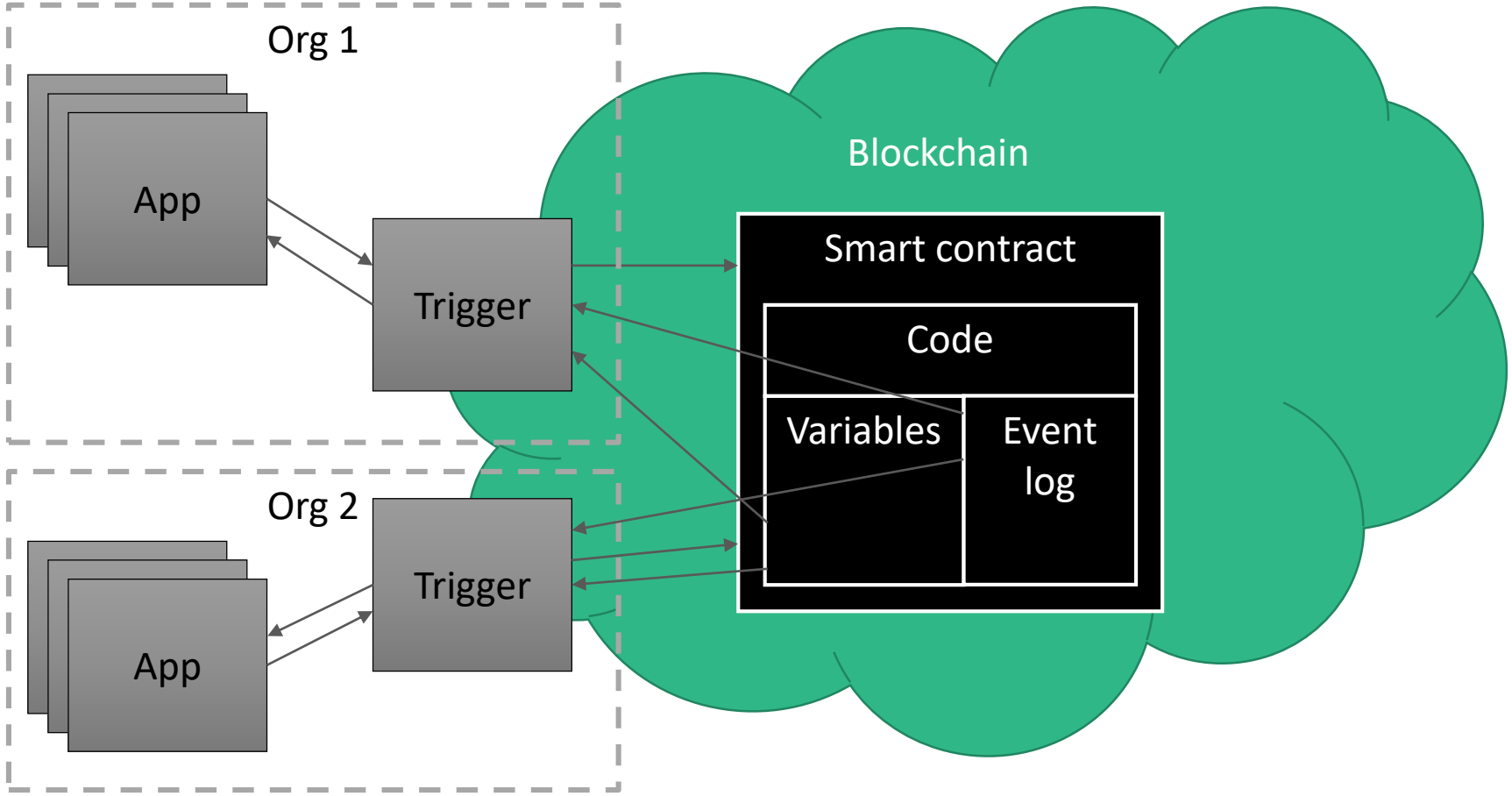


Blockchain is a closed-world system

- To interact with smart contracts on blockchain, need to:
  - Write: create and broadcast a blockchain transaction (BCTX) for each method call
  - Read: monitor smart contract variable values and/or event logs to see updates
- The outside world speaks Services
  - REST / SOAP-WSDL / JSON RPC
- How to bridge between the two worlds?
  - Recurring problem
  - Our solution: a *Trigger* component as bridge

Service-Oriented Approach to Integration

Trigger as bridge between blockchain and services



Motivation

- Integration of services across org: key driver of productivity gains
  - Means: service composition – orchestration and choreography

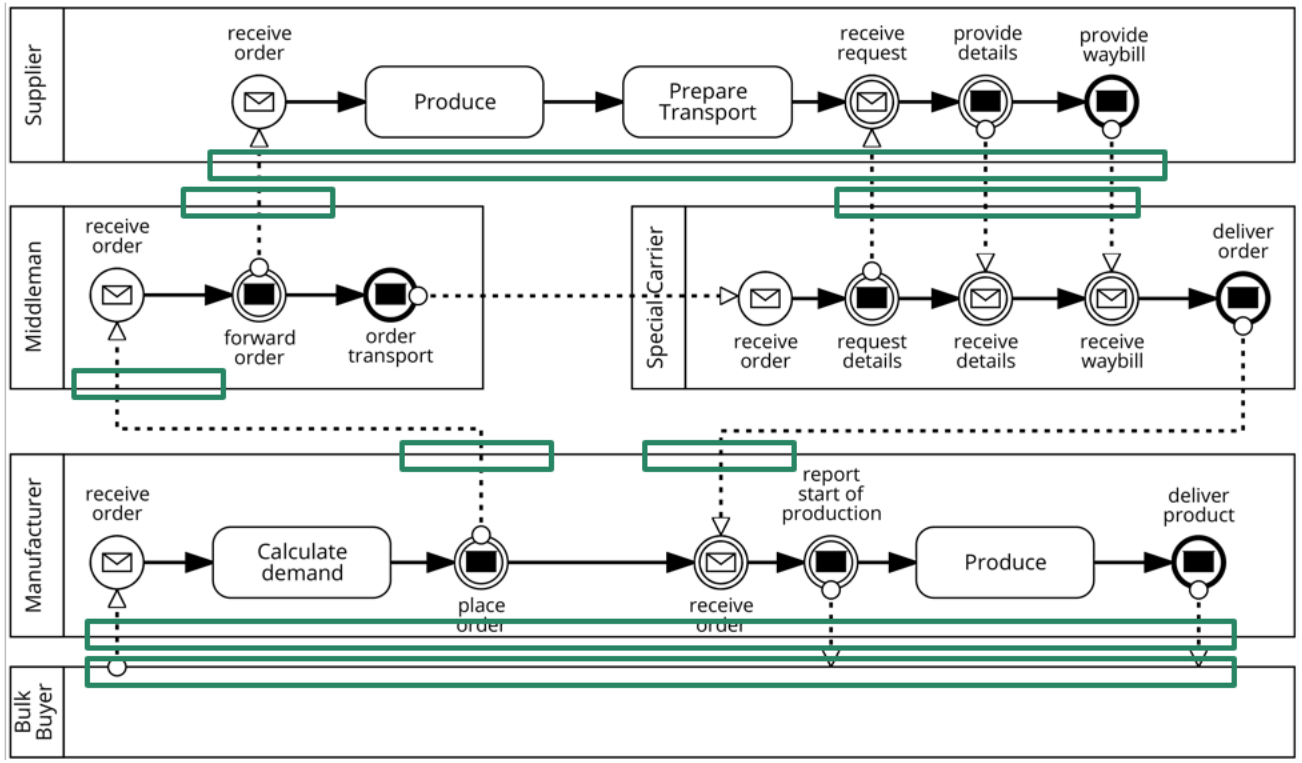
- Collaborative process execution
  - Doable when there is trust – supply chains can be tightly integrated

  - Problematic when involved organizations have a **lack of trust** in each other
    → if 3+ parties should collaborate, where to execute the process that ties them together?

  - Common situation in “coopetition”

Jan Mendling, Ingo Weber, Wil Van Der Aalst, ….Liming Zhu, ***“Blockchains for business process management – challenges and opportunities” ACM Transactions on Management Information Systems (TMIS), 2018.***

Motivation Example



Issues:

- Knowing the status, tracking correct execution
- Handling payments
- Resolving conflicts

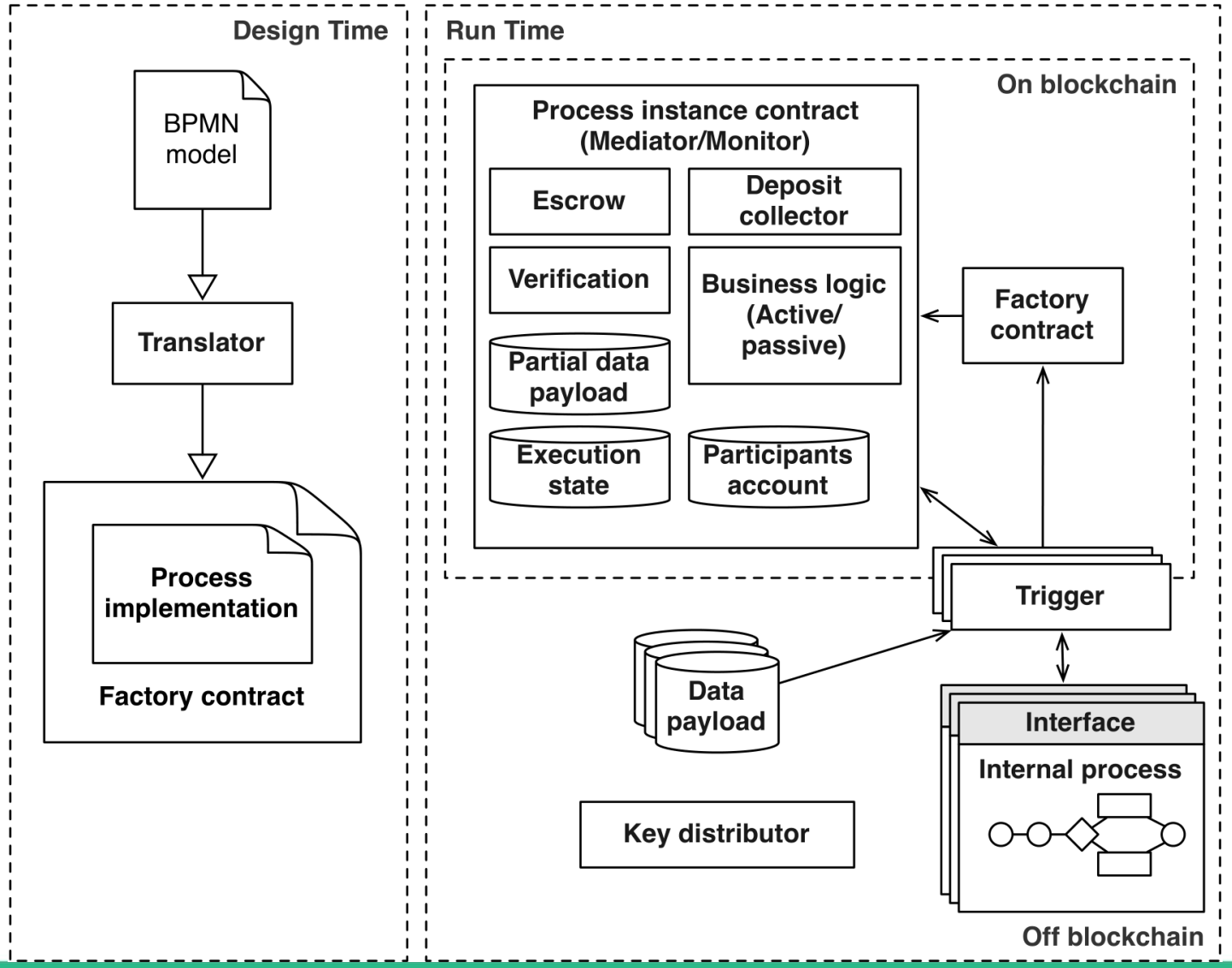
→ Trusted 3rd party ?
→ Blockchain!

Approach in a nutshell

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

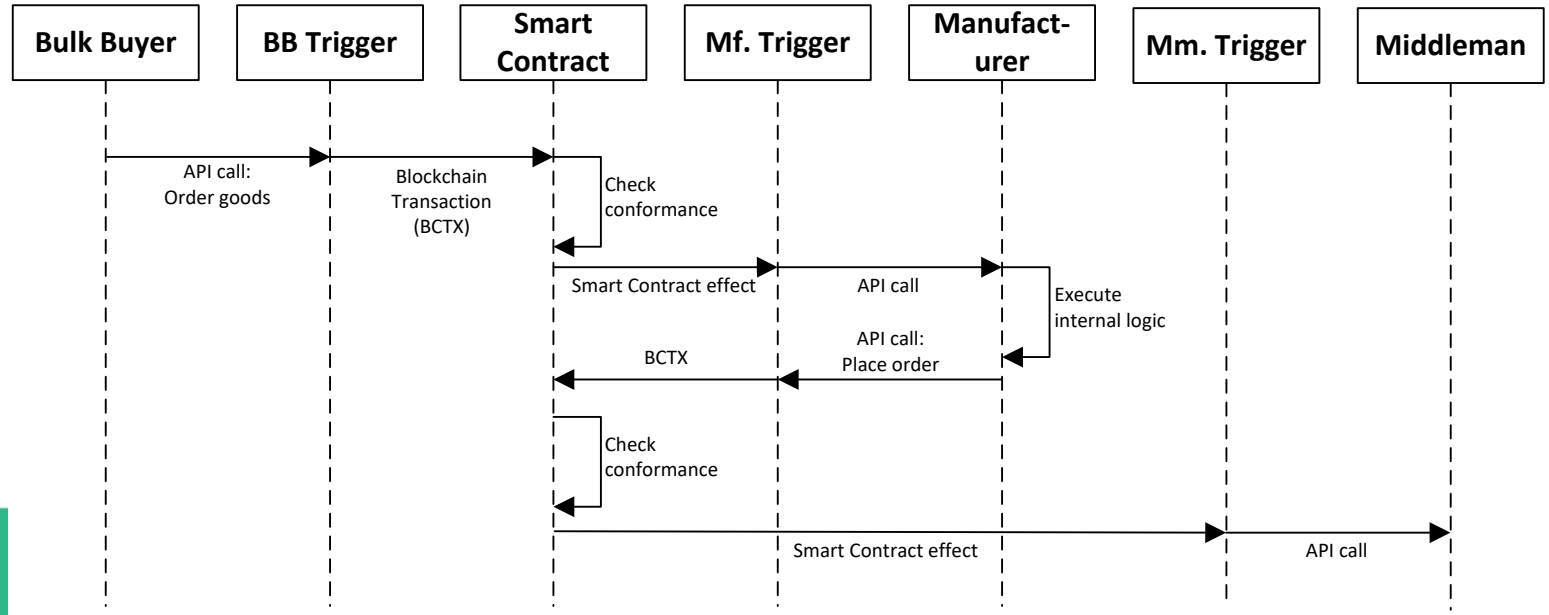
Ingo Weber, Sherry Xu, et. al, ***"Untrusted business process monitoring and execution using blockchain". BPM 2016***
O. López-Pintado, L. García-Bañuelos, M. Dumas, I. Weber. ***"Caterpillar: A blockchain-based business process management system" BPM 2017***
L. García-Bañuelos, A.Ponomarev, M. Dumas, I. Weber, ***"Optimized Execution of Business Processes on Blockchain". BPM 2017***

Architecture



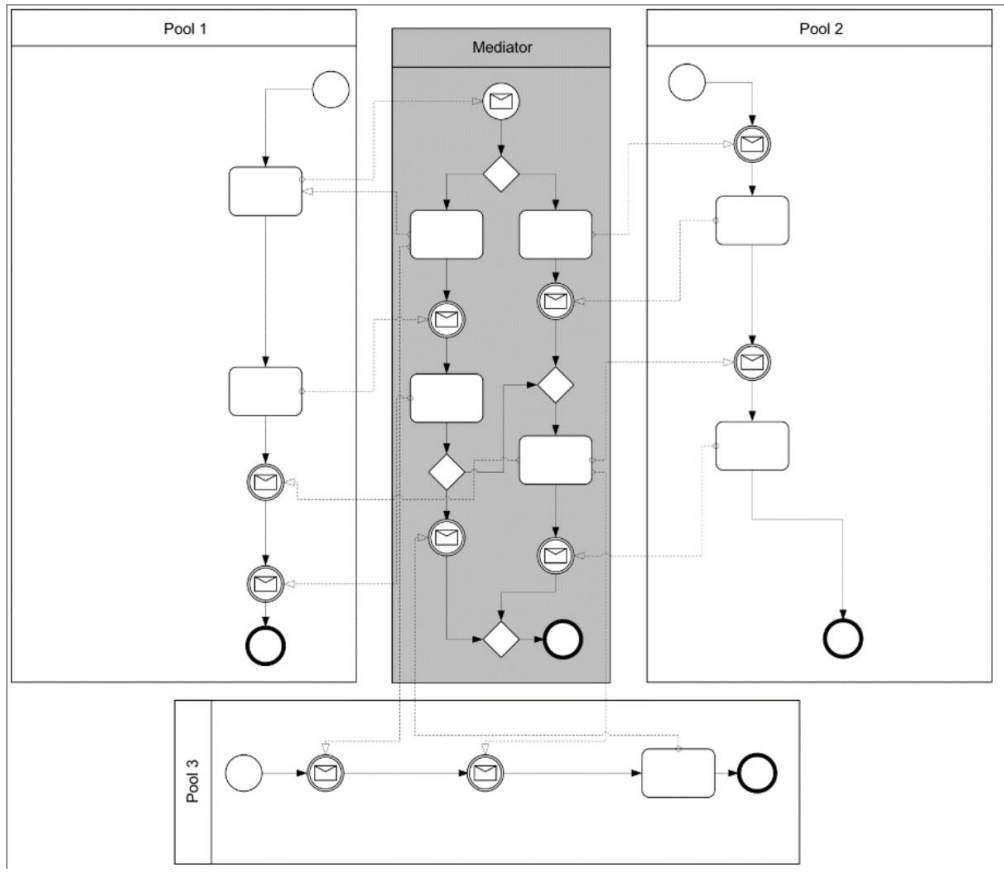
Runtime

- Instantiation:
  - New *instance contract* per process instance
  - Assign accounts to roles during initialization
  - Exchange keys and create secret key for the instance
- Messaging:
  - Instead of sending direct WS calls: send through triggers & smart contract
  - Instance contract handles:
    - Global monitoring
    - Conformance checking
    - Automated payments*
    - Data transformation*

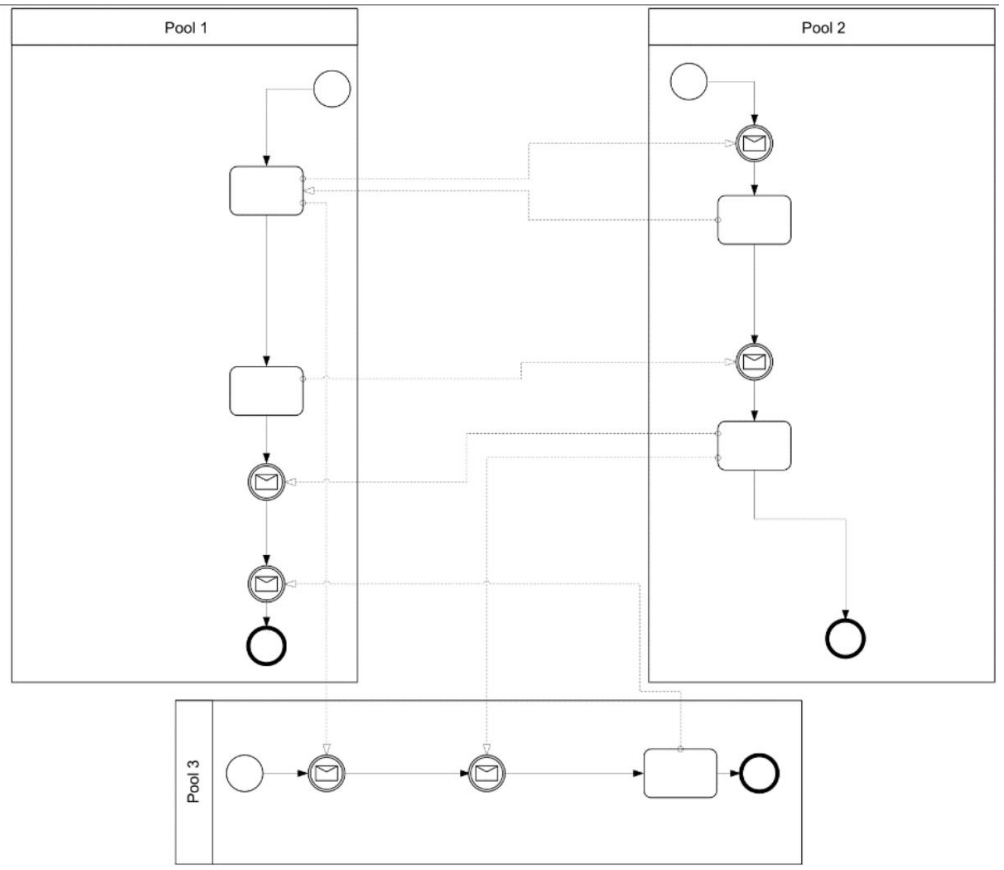


Service Composition Variants

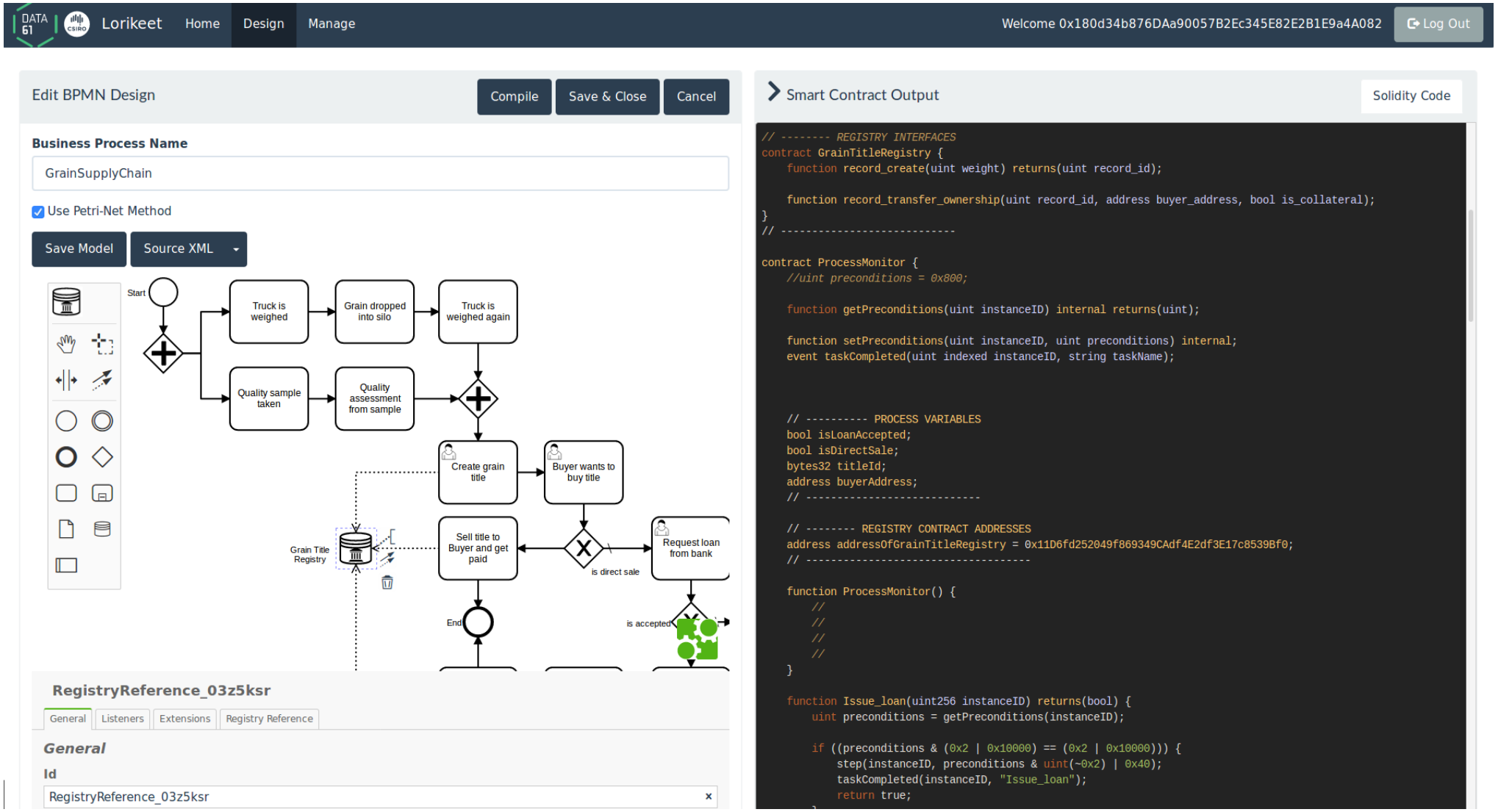
Mediator (orchestration)



Choreography → C-Monitor

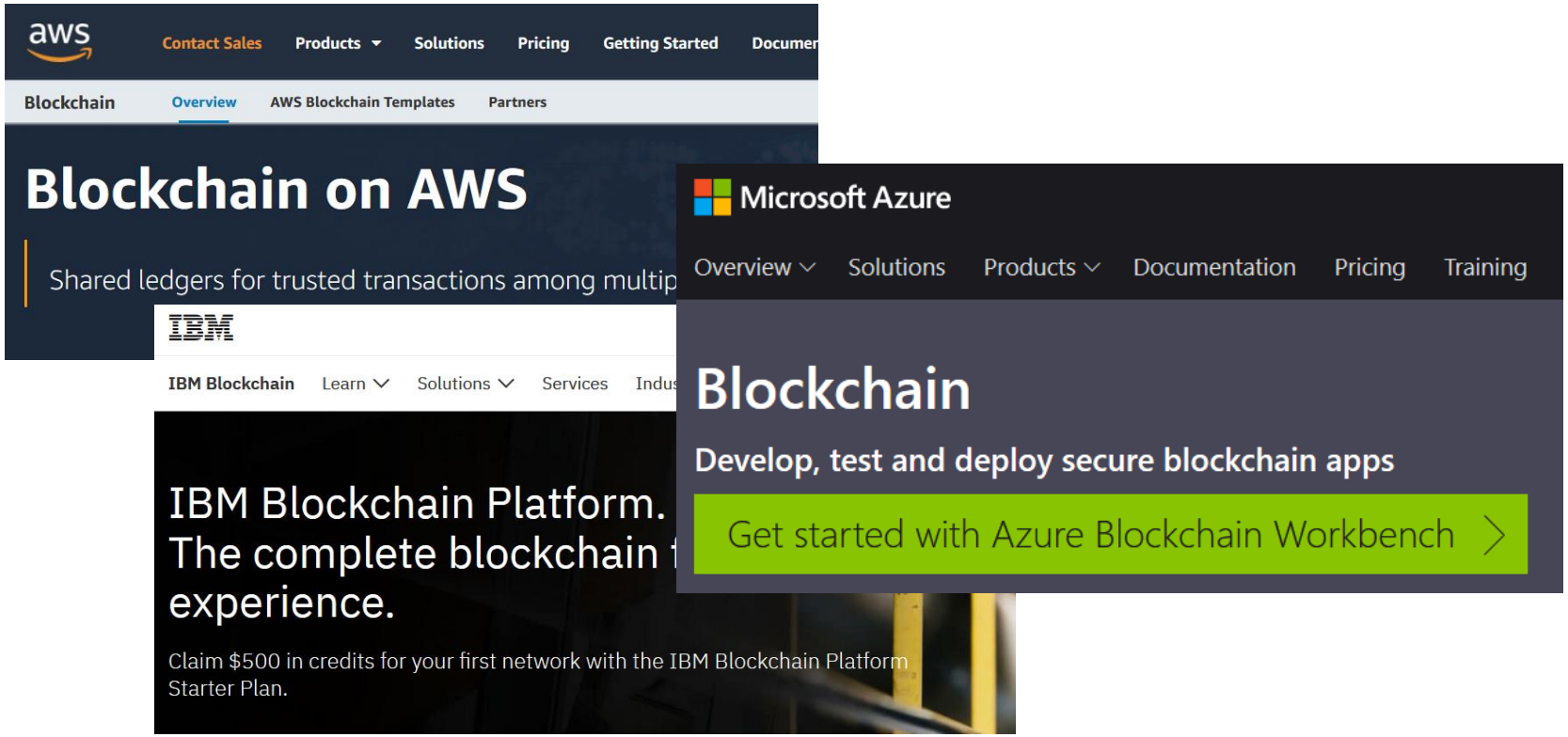


Tooling: Design and Runtime



Blockchain-as-a-Service

Commercial Offers

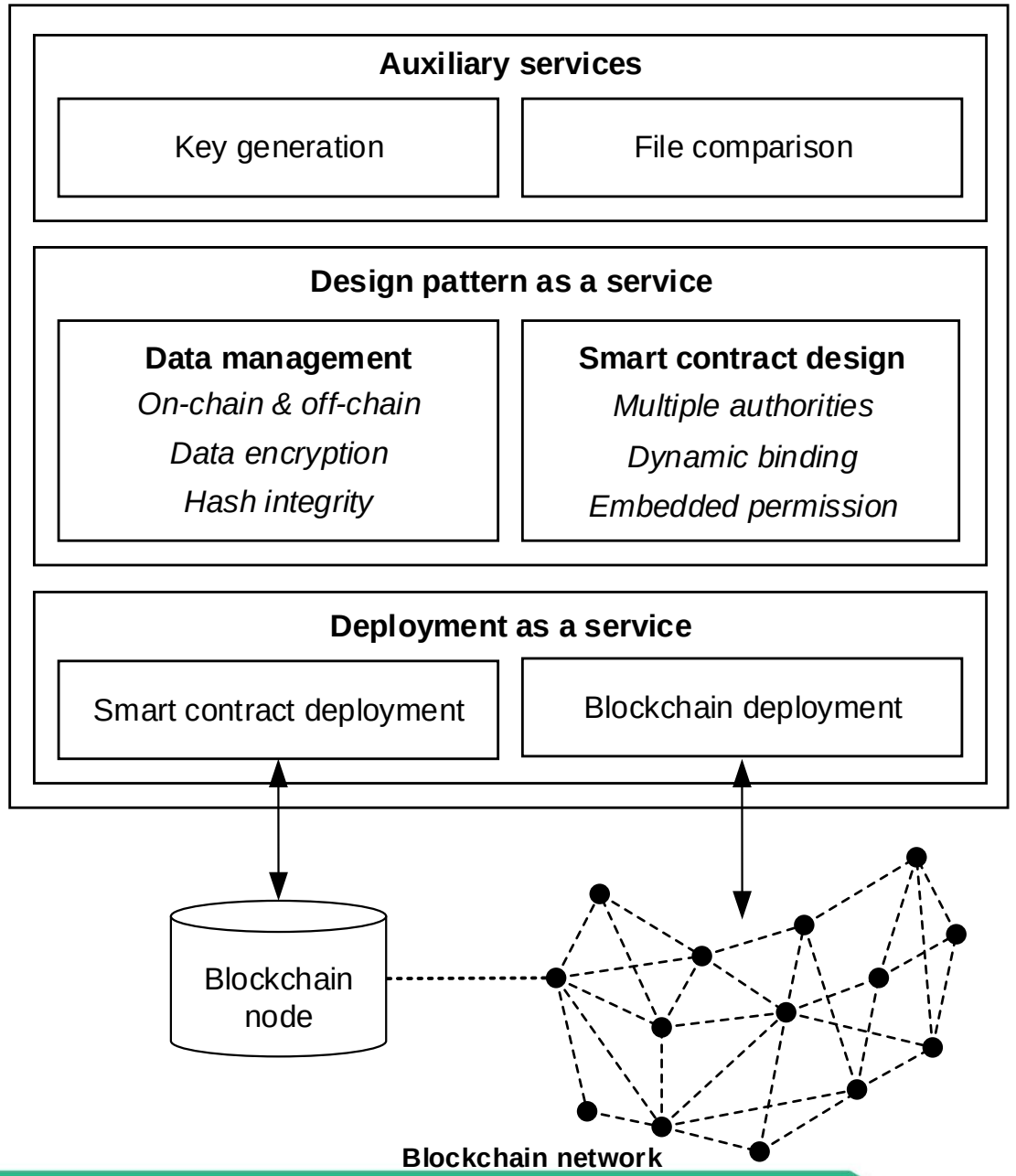


- But: some of these providers require that all nodes are using the same provider
  - Decentralization?

Early-stage Research: Unified Approach

- "uBaaS: A Unified Blockchain as a Service Platform"
  - Under review
- Deployment as a service
  - Includes a blockchain deployment service and a smart contract deployment service
  - Platform agnostic to avoid lock-in to specific cloud platforms
- Design patterns as a service
  - Common data management services and smart contract design services
  - Based on a design pattern to better leverage the unique properties of blockchain (i.e. immutability and data integrity, transparency) and address the limitations (i.e. privacy and scalability)

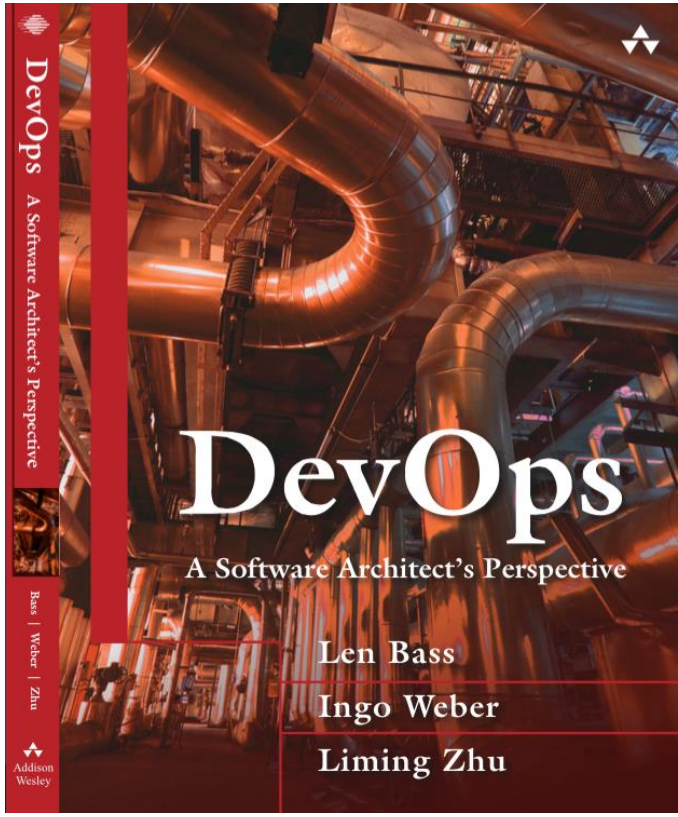
uBaaS architecture



Smart Contract as Microservices

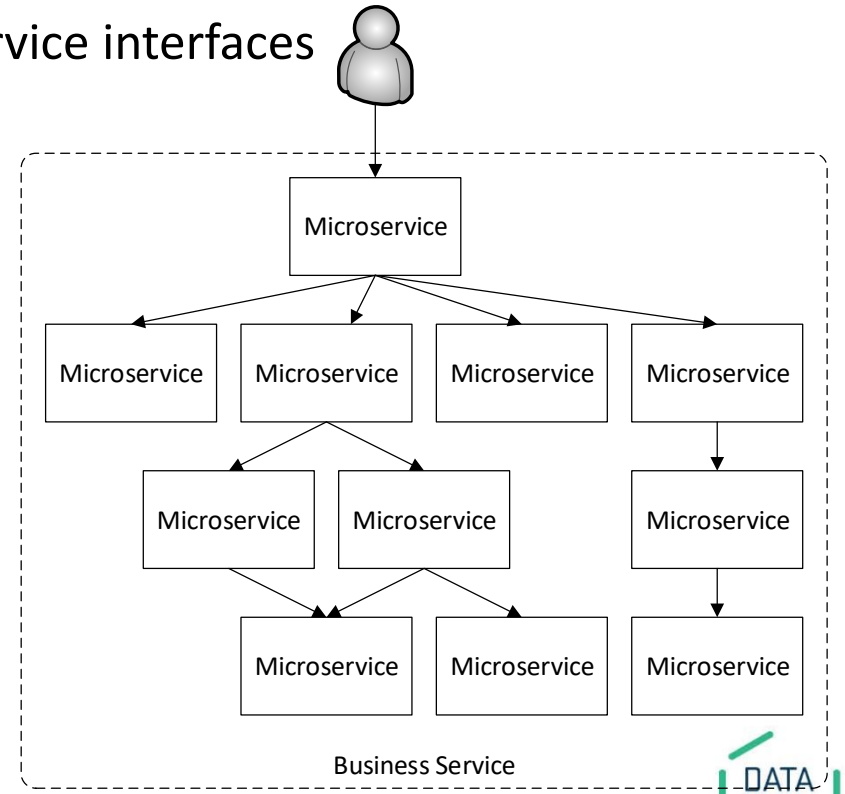
Microservice as DevOps Consequences

- Keep teams relatively small
  - Amazon's "two pizza rule": no team should be larger than can be fed with two pizzas
  - Advantages: make decisions quickly, less coordination overhead, more coherent units
- Team size becomes a major driver of the overall architecture:
  - Small teams develop small services → Microservices
  - Coordination overhead is minimized by channeling most interaction through service interfaces:
    - Team X provides service A, which is used by teams Y and Z
    - If changes are needed, they are communicated added to the interface.



Microservice Architecture

- Each user request is satisfied by some sequence of services
- Most services are not externally available
- Each service communicates with other services through service interfaces
- Service depth may be 70, e.g., LinkedIn



Smart Contracts as (Micro)Services?

- Analogy:
  - Smart contract code ≈ Java Class
  - Deployed smart contract ≈ Java Object, but with some properties
    - Defined interface
    - Standard way to invoke
    - Callable by anyone (who can send transactions to the blockchain)

    → Similar to Web service!

- Some design principles can apply

Service-Orientation Design Principles

× **Standardized Service Contract:** the public interfaces of a services make use of contract design standards.

✓ **Service Loose Coupling:** to impose low burdens on service consumers

× **Service Abstraction:** "to hide as much of the underlying details of a service as possible"

~ **Service Reusability:** services contain agnostic logic and "can be positioned as reusable enterprise resources"

✓ **Service Autonomy:** to provide consistent results, a service has to have strong control over its underlying environment

× **Service Statelessness:** services should be "designed to remain stateful only when required."

× **Service Discoverability:** "with communicative metadata by which they can be effectively discovered and interpreted."

~ **Service Composability:** "effective composition participants, regardless of the size and complexity of the composition."

× **Fundamental requirement – interoperability of services:** "...stating that services must be interoperable is just about as evident as stating that services must exist."

Based on SOA Principles of Service Design, Thomas Erl, Prentice Hall, 2007, http://serviceorientation.com/serviceorientation.
Summary: I. Weber, ***Semantic Methods for Execution-level Business Process Modeling. Springer LNBIP*** *Vol. 40, 2009.*

Microservice Principles

✓Small, focused functionality

✓ Split of responsibility

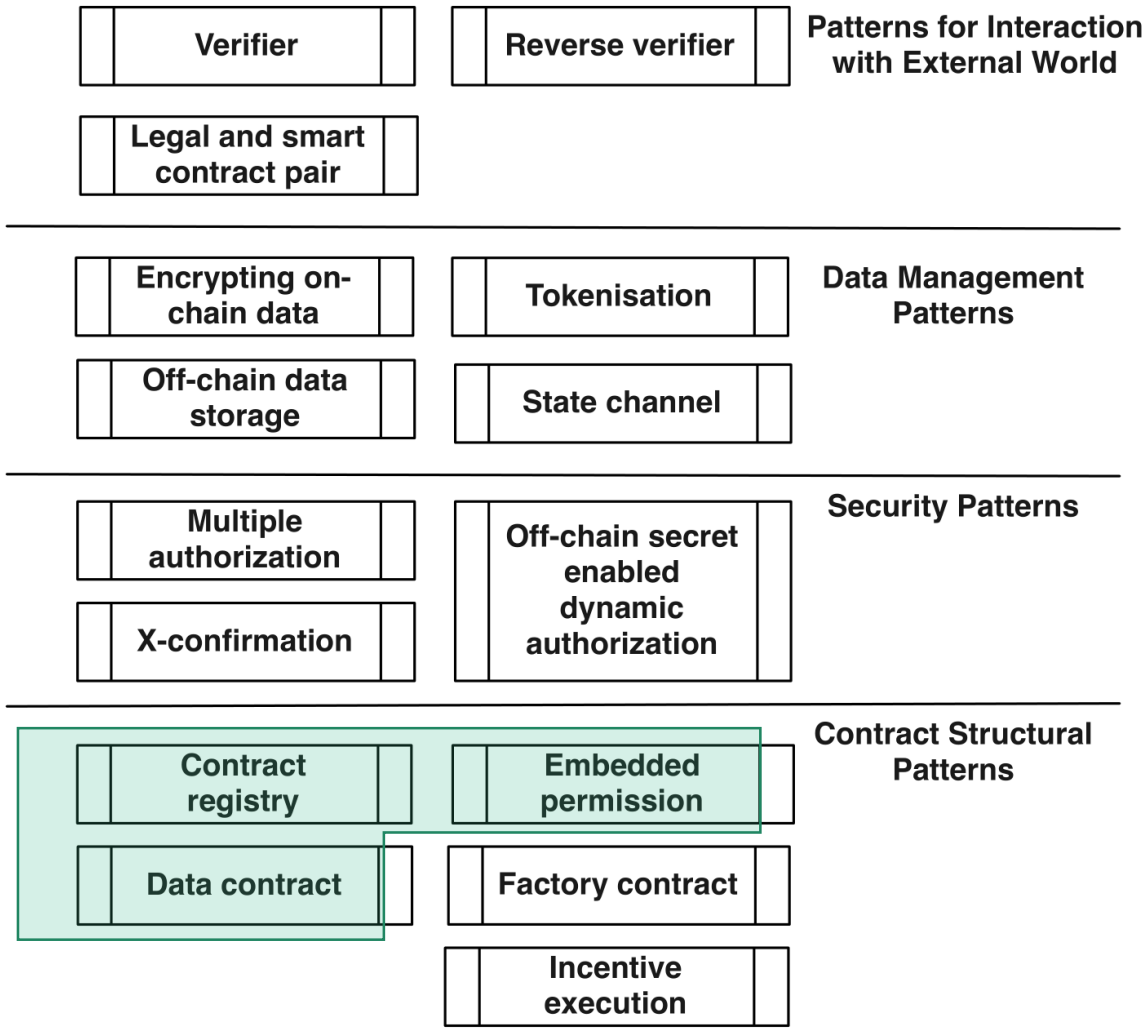
~ Full-stack & independently updatable without downtime

✘ Stateless

- While some design principles for Microservice Architectures apply, others do not
  - Updates *can* be independent
  - But reliance on the *inability* of anyone to update without consensus is one source of trust in a smart contract

(Service) Design Patterns for Blockchain

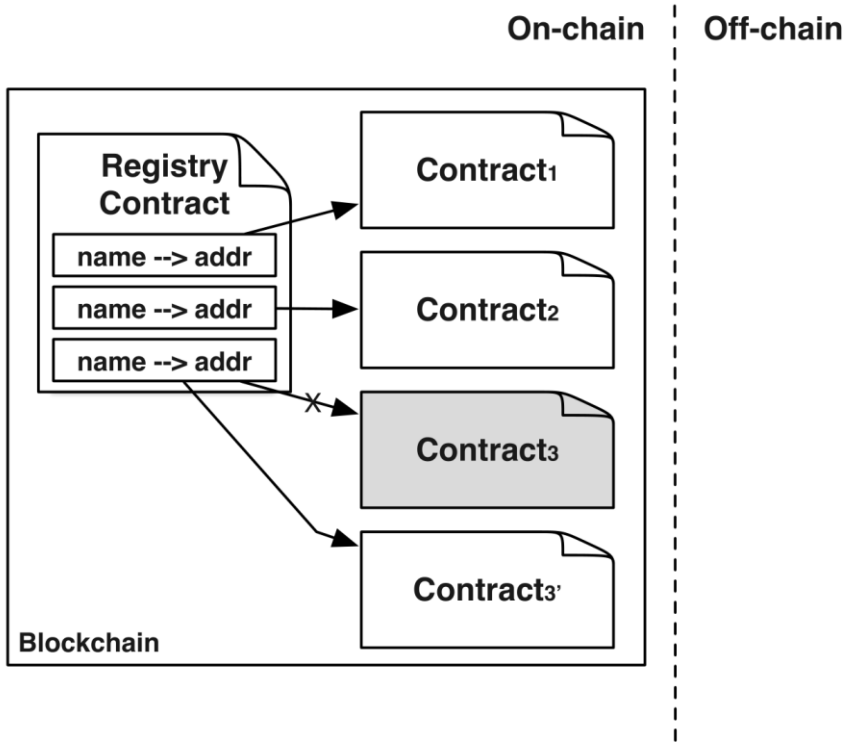
X. Xu, C. Pautasso, L. Zhu, Q. Lu , I. Weber,. et al. ***"A pattern collection for blockchain-based applications", EuroPLoP2018***



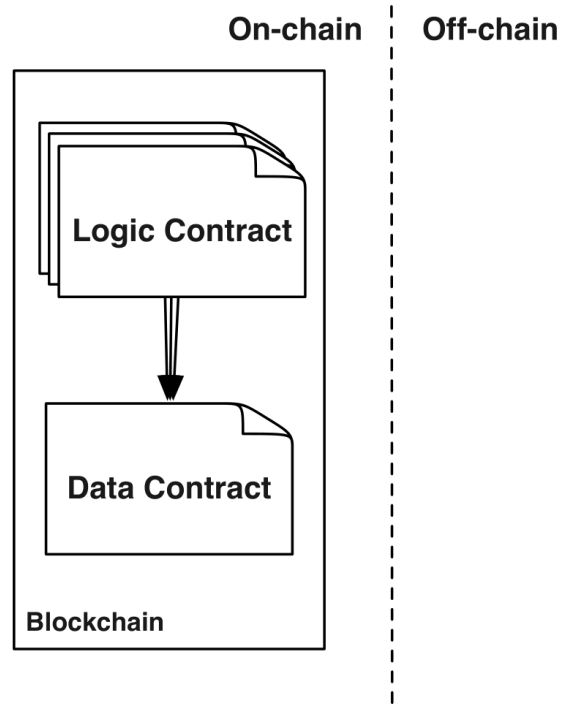
Some patterns of relevance here

Microservice Patterns for Smart Contracts

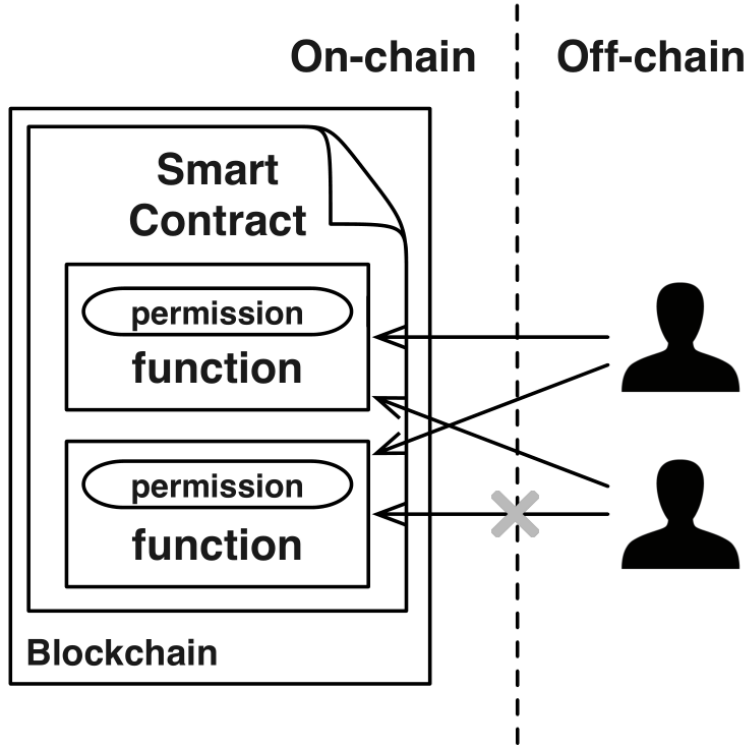
Pattern 11: Contract Registry



Pattern 12: Data Contract



Pattern 13: Embedded Permission



Summary: Connecting with Industry Impact

Trust in Data Services & ML/AI-as-a-Service

Data61 Challenge: Supply Chain Integrity

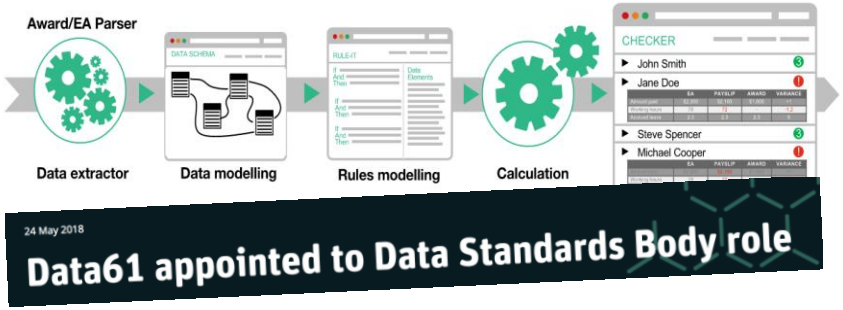
https://data61.csiro.au/en/Who-we-are/Our-programs/Challenge

Trust in the Space between Organisations

ANB: Smart Legal Contract as Services

https://www.australiannationalblockchain.com

Trust through Governance/Regulation



Consumer Data Rights & Regulation-as-Services

Building Distributed Trust Infrastructure

- behind Services &
- for the Space Between Services