GovCore

An open source government platform concept

Simon Weller

Think bigger …

Acknowledgements:
Daniela Mazzone
Colin Fairweather
Grantly Mailes

Background

This story has its genesis in a question … what would be really disruptive to local government?

… imagine if you could download everything you need to run a council - for free - from GitHub

Which begs the questions:

Why even try?

- What is the pain we're trying to solve?
- What is the prize if we do?

What would the new ecosystem, look like?

- Where are the market opportunities?

How would you build it?

- Technology components
- Architecture and data governance

And then beyond minimum viable product:

Solving endemic challenges in public sector systems:

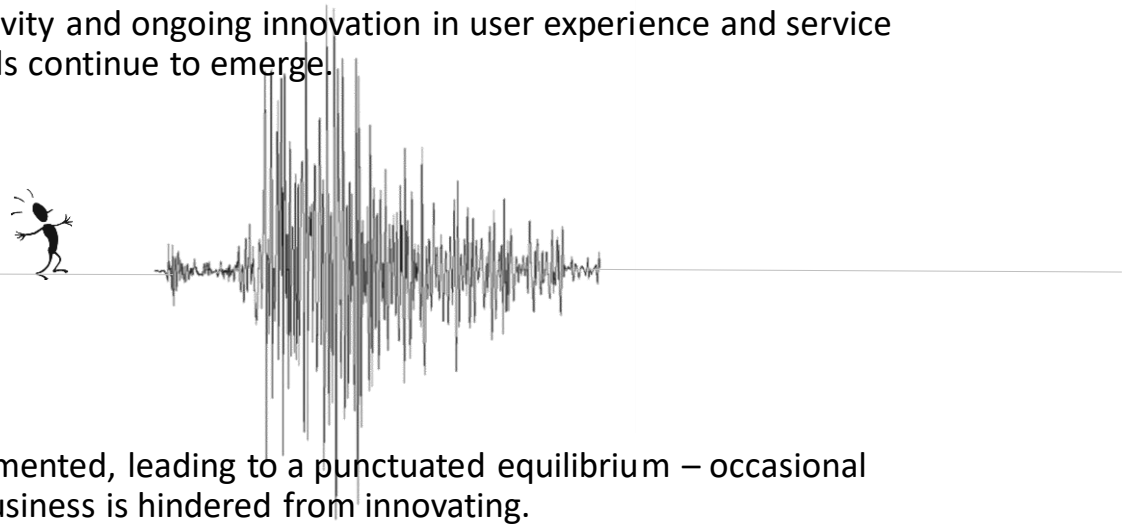
- Satisfying audit
- Supporting flexibility and continuous improvement
- Resilience and scale in deployment options
- Onboarding new functions and organisations (data migration)
- Managing attachments like images, documents and records
- Embedding spatial mapping
- Managing roles and responsibilities
- Adopting international and industry standards
- Satisfying records management obligations

Frontend / Backend

This architecture vision is virtually silent on design and technology of the frontend. This should not be taken to mean that user experience is less important. Nothing could be further from the truth.

The frontend is far and away the most important component to our customers, our staff, our sponsors and key stakeholders.

This architecture vision imagines a backend that maximises the potential for creativity and ongoing innovation in user experience and service design, acknowledging that design paradigms continually change and new channels continue to emerge.



Traditional platforms are difficult to implement and difficult to change once implemented, leading to a punctuated equilibrium – occasional cataclysmic change with long periods of suspended animation during which the business is hindered from innovating.

The vision for GovCore is a backend that supports ongoing low-cost changes to data definitions and business processes. This positions the business to innovate the user experience as well as data needs, supporting processes, freeing service designers to continually experiment, iterate and transform end-to-end service delivery.

INFLECTION

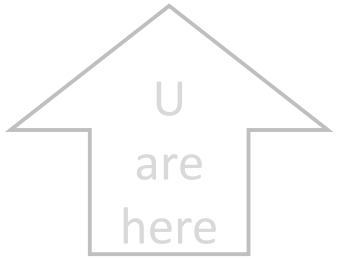
Current digital transformation approaches have reached the limits of speed and scale – a new approach is needed.

Arts & Crafts

- Design and Manufacture co-located
- Artisan workshops
- One-off and small batch

Modernity

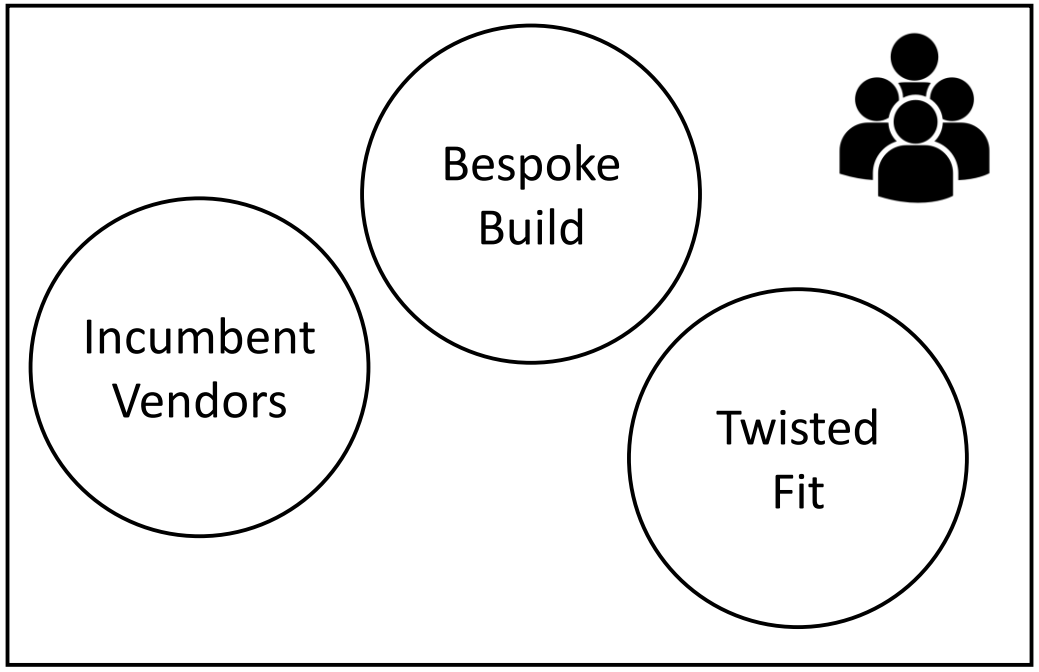
- Design separate from manufacture
- Production lines
- Manufacture at speed and scale



We are at the point of transition – we need to retool and rethink our delivery models

Why even try?

The pain



Our reality is three system options:

1. Incumbent vendors are tending to leverage existing platform investments rather than innovating
2. Bespoke built solutions are innovative but incur all the costs of establishing and running an internal development capability and risk of knowledge-loss and obsolescence
3. Old-school platform solutions (e.g. ERP or CRM) twisted to fit government's needs with vast upfront and ongoing cost of customisation and generally sub-optimal outcomes

- The prison: there are no options outside this reality – outside the box – so every agency is cyclically hopping from one option to another, generating massive duplication and redundancy across the sector
- Vast amounts of public resources spent migrating from one poor solution to another

Existing options

Incumbent Vendors

- Infor
- Civica
- Technology One

- Foundations laid pre-2000
- Most have not re-architected*
- No APIs (or very limited)
- No mobile customer interface
- Not cloud native (or capable)
- No flexibility to design new user experiences and services

Twisted fit

- Oracle
- SAP
- Salesforce

- Designed for other industries
- “Configured” for government (customised, in reality)
- Older examples have similar constraints to incumbents

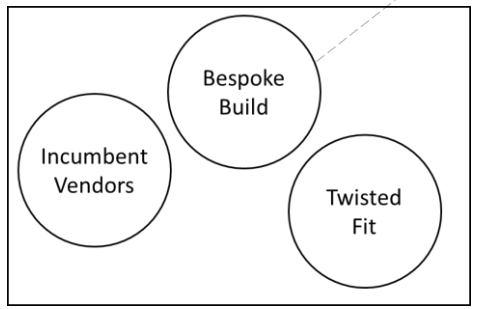
Bespoke build

- Handbills & Fundraising
- PRISM (historic)
- SASSI (historic)

- No economy of scale (each agency bares 100% of cost)
- Insource only option for ongoing maintenance and enhancements(?)
- Usually high ongoing dependency on individual developer
- Best-practice software development lifecycle unrealistic at small scale

* Technology One is a notable exception

Outside the box

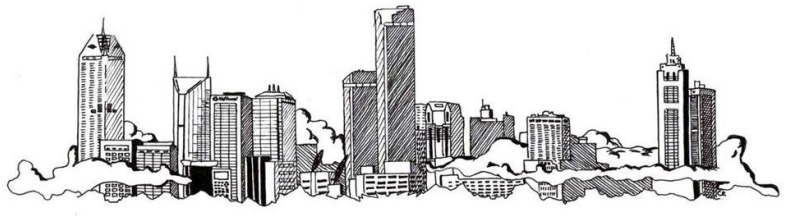


GovCore

The vision of GovCore is an open source platform built on robust scalable big data technology, designed to maximise innovation in both customer experience, data management and business process automation.

It is a form of bespoke build, but one where a large community of developers can contribute openly - sharing ideas and improvements and maximising economy of scale in development talent.

The prize



On the back of an envelope, lets say conservatively in Victoria:

A capital city spends $2million licensing LG mission systems
+ $2 million annual labour and support

The next 5 biggest councils spend about the same combined

The next 10 biggest suburban and regional councils combined spend about the same

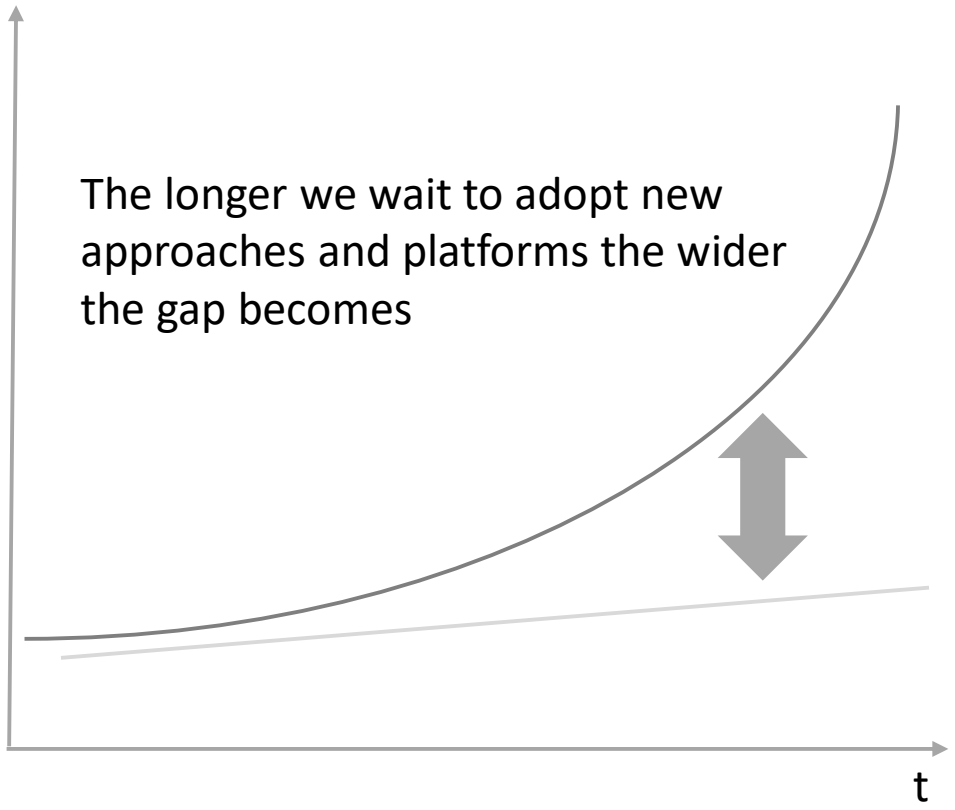
The remaining 60+ suburban, regional and rural councils together spend about the same

It may be even bigger than this? How do we find out?

The payoff: potentially AU$25million+ per annum just in Victorian local government …

… other levels of government… Australia … Asia-Pac … Europe … the Americas …

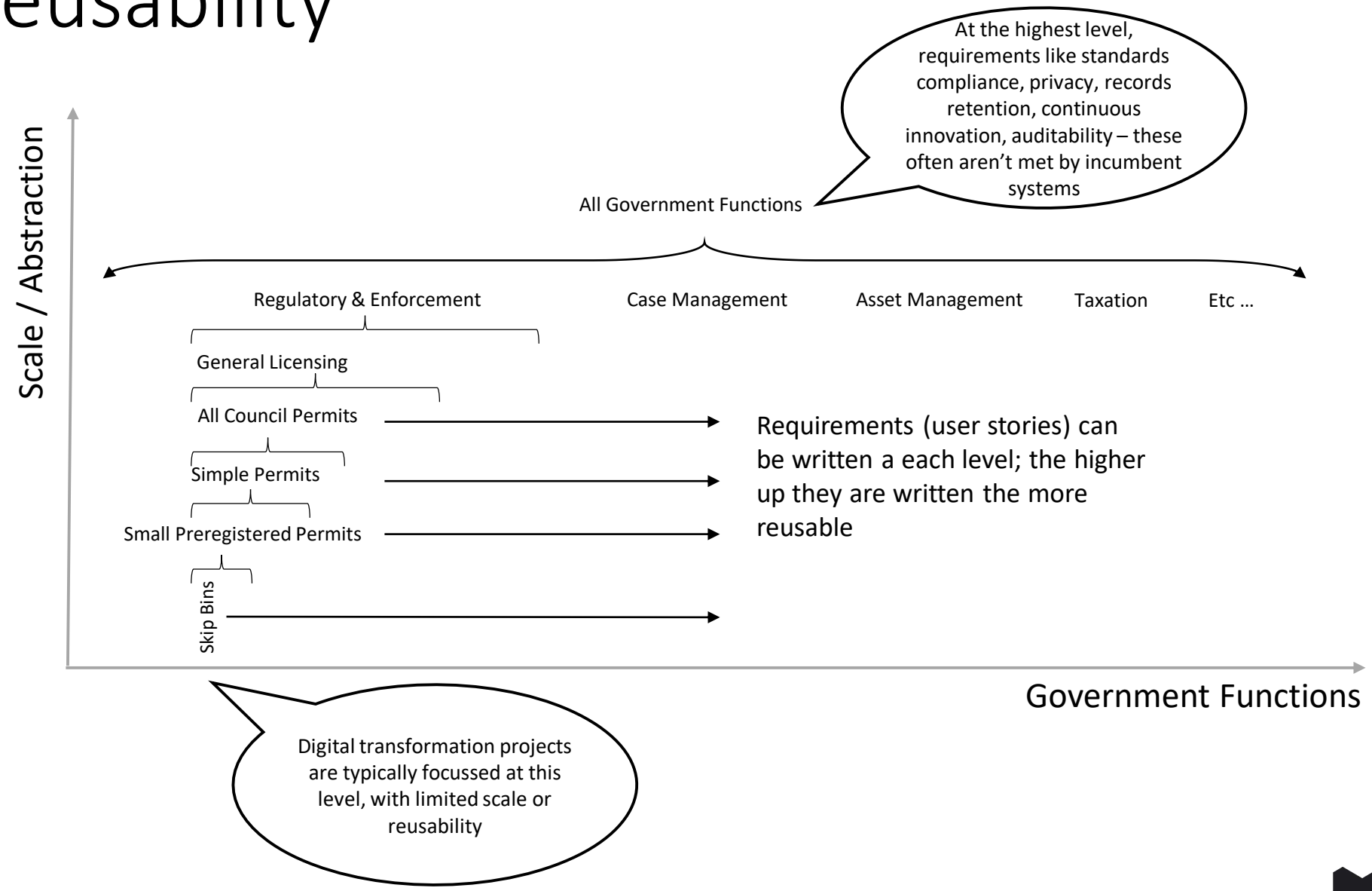
Widening gap



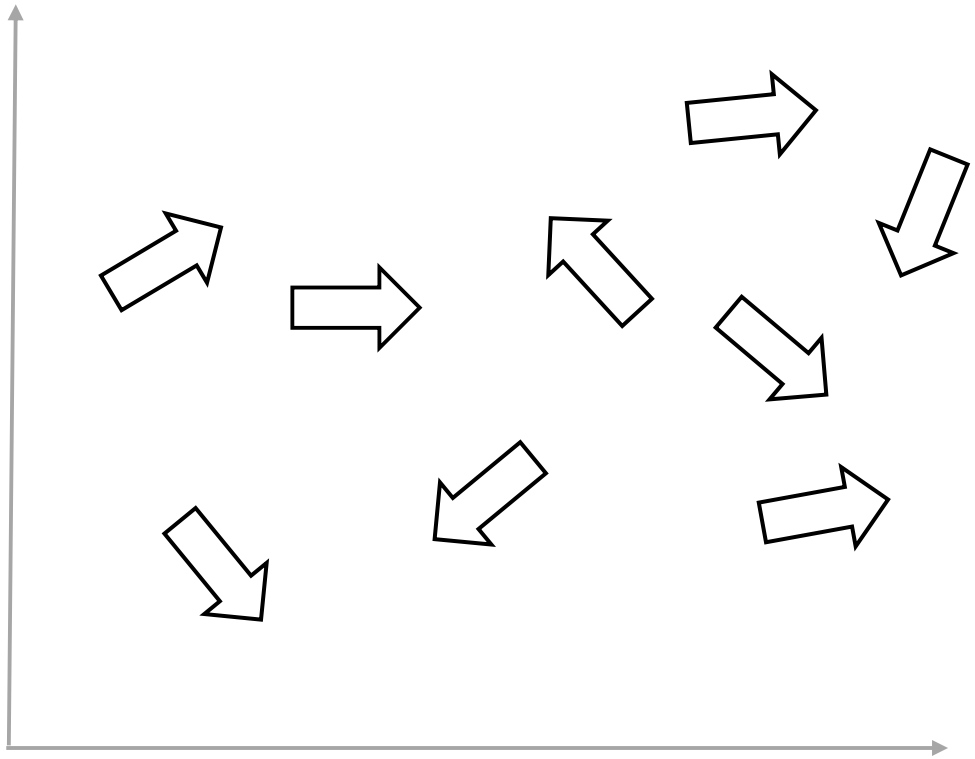
The rest of the world is accelerating its adoption of emerging technologies like cloud, voice, mobility, robotics, AI, ML, DL, IoT, process automation etc

Government continuous improvement

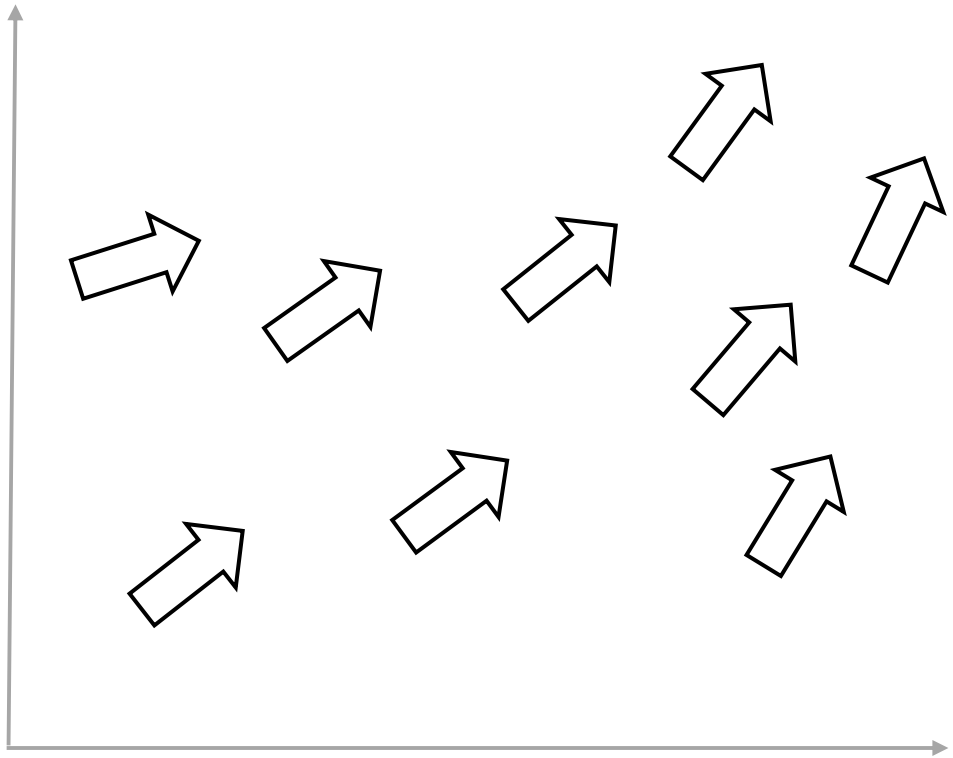
Reusability



A Common Target



IT projects – including (and especially) digital transformations – tend to be independent and don't work towards a common goal



This architecture vision provides a common target to aim for whilst still supporting agile principles of small iterative builds, minimum viable products and early tangible deliverables.

An alternative reality

What would a truly disruptive, and radically different* reality look like?

1. Fully open source option
2. Coordinated global innovation ecosystem
3. Big data at the core
4. Minimum code – massive reuse
5. On-boarding as easy as signing on to LinkedIn

* And much better

What I mean by Open Source

- Free
  - Can run on an entirely free software platform
  - Code is freely available online (Git)
  - Costs reduced to hosting and DevOps & support staff
- Platform independent
  - Entirely portable
  - Runs on any platform
- License ensures software stays free
  - GPL, MIT, Apache, Creative Commons or similar

Is Open Source credible?

The whole Internet runs on open source*

So yes … with the proviso that:

1. There is a large stable global community supporting and developing
2. There is preferably one or more commercial organisations offering hosting, training, support and development leadership
3. Most of the internet and large platforms like Android are open source

*and Android, for e.g.

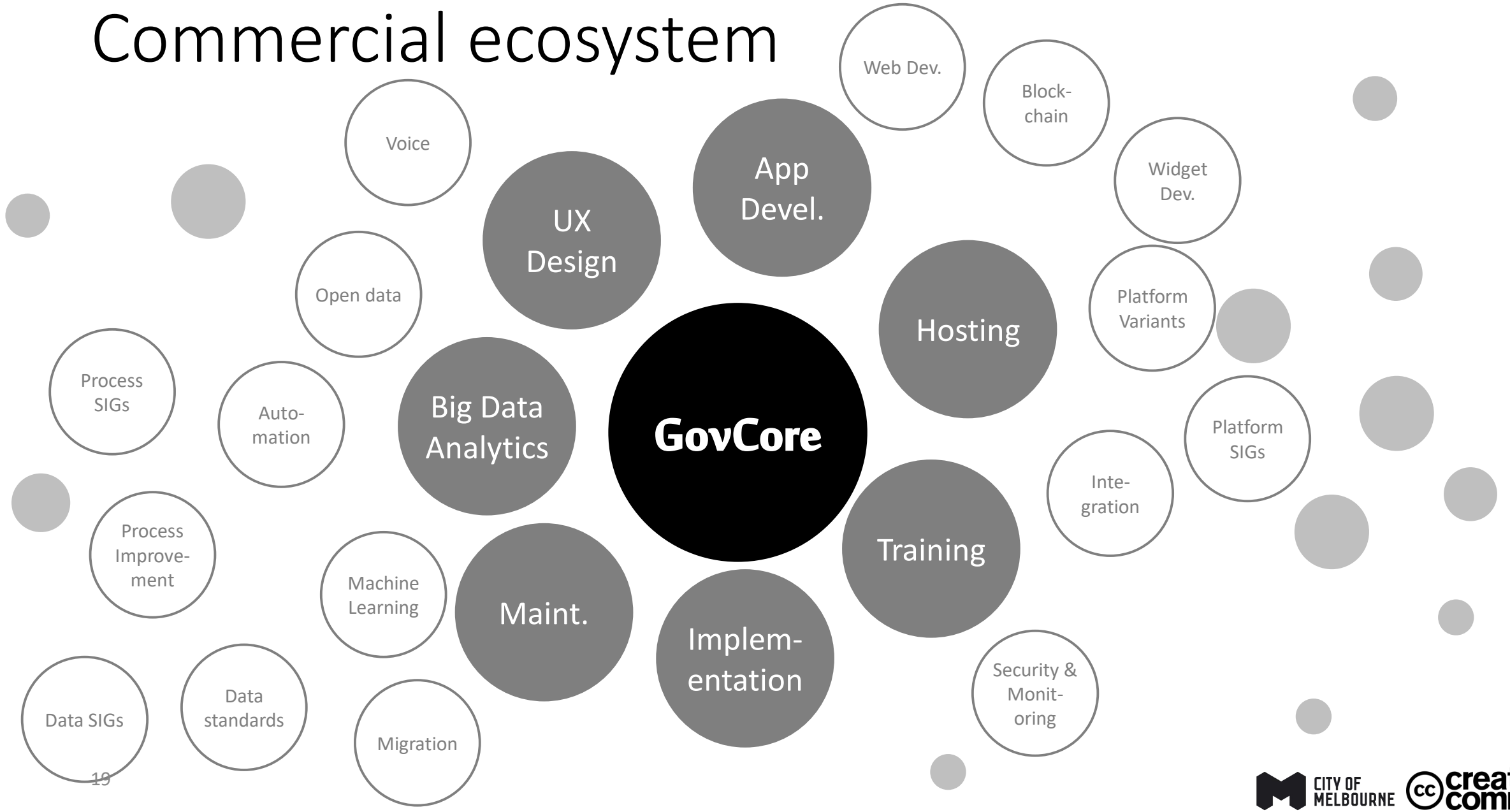
Ecosystem

Where is the commercial opportunity?

Open source examples exist where the software is freely available but commercial opportunities are exploited around:

- Hosting (Drupal is an Open Source Web content management system, Acquia is a hosting service)
- Distribution, Support and Training (Linux is an open source operating system, Red Hat is a hosting, support and assurance service)

Commercial ecosystem



GovCore Marketplace

The goal is to create a two-sided marketplace for developers and users

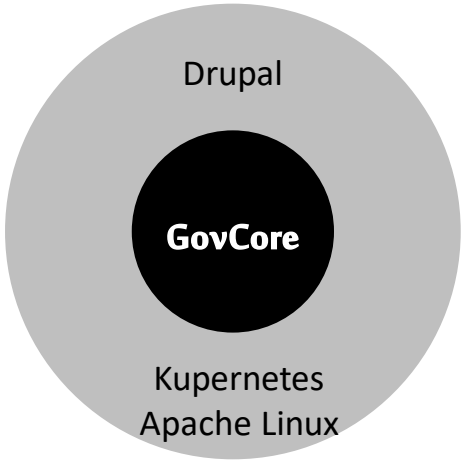
There are 5-6 main elements to the marketplace:

- Application development
- Data
- Hosting
- Implementation and DevOps Services
- Process automation

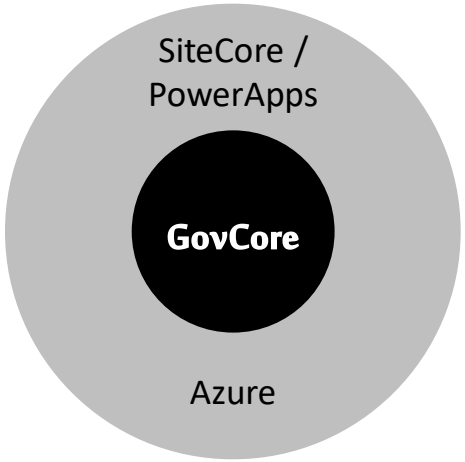
A key consideration is management and orchestration of both the marketplace and the GovCore development effort

Platform independent

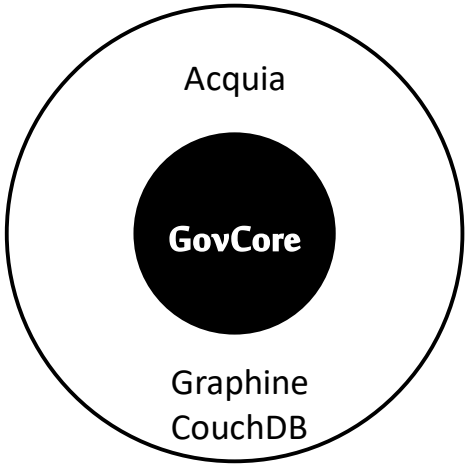
Pure Open Source



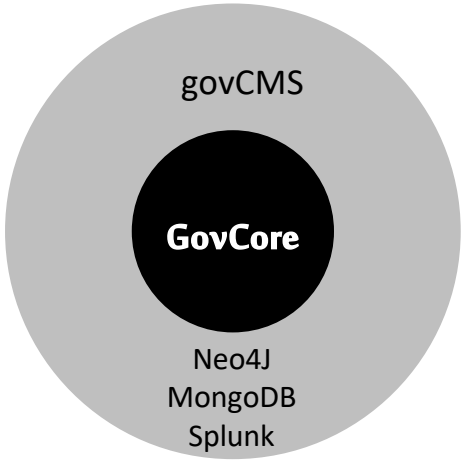
Microsoft



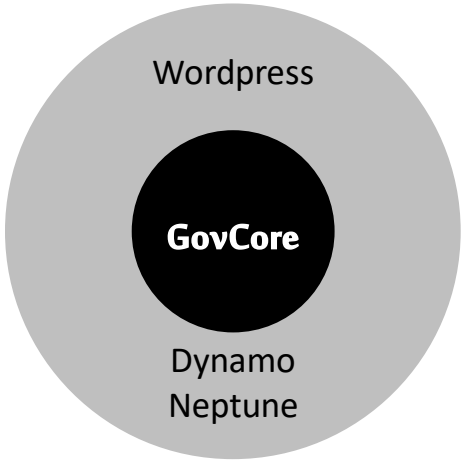
Heroku



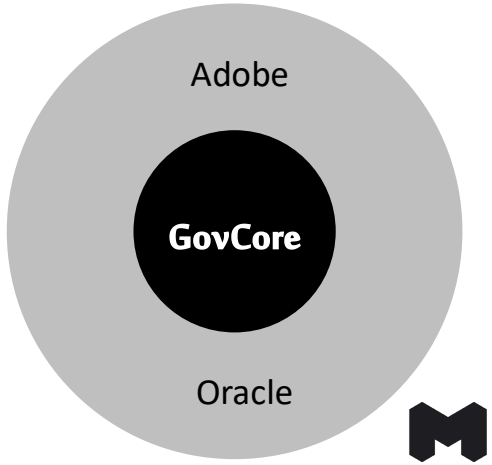
Freemium



Amazon

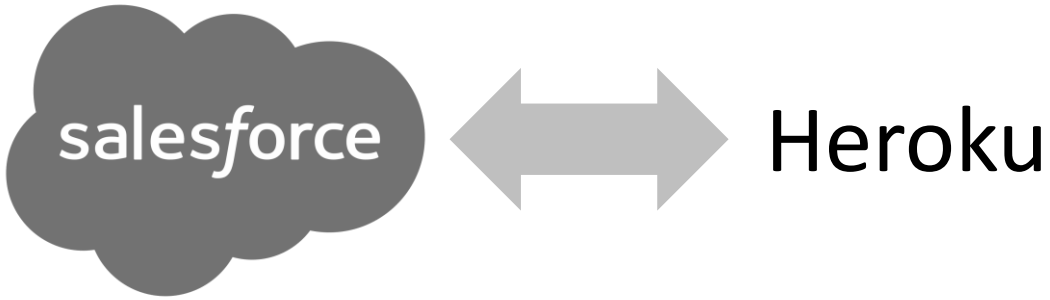


High-end Commercial



Portability is vital to future proof the platform, but also to support the diversity in size and skill-base of local councils (and other agencies)

Salesforce Heroku



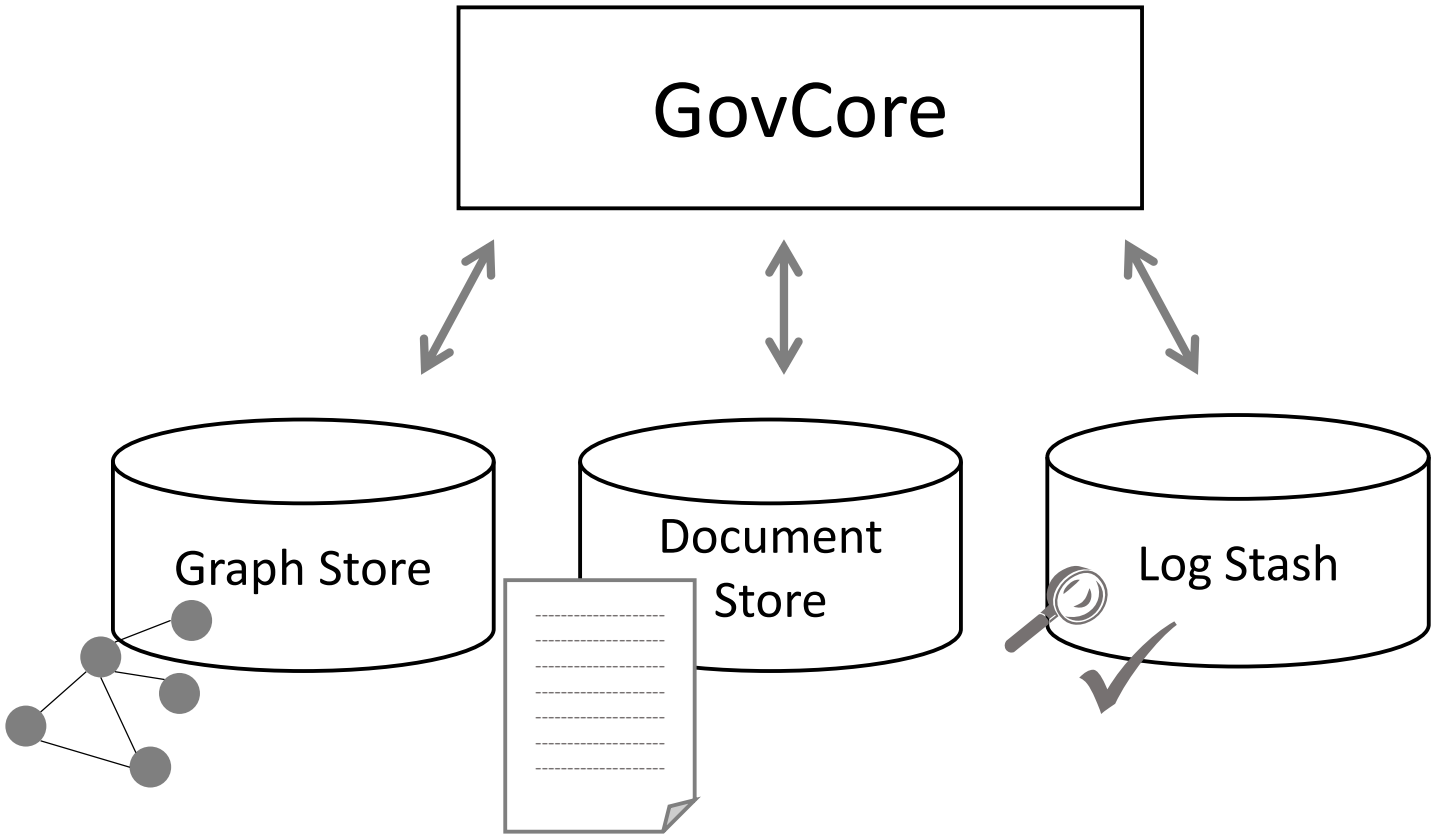
Heroku

- provides a rapid prototyping sandbox
- is a genuine Open Source platform
- is owned by Salesforce
- runs on Amazon
- is subscription based, not user based
- has an unmetered out-of-the-box back door into Salesforce (e.g. Customer objects)

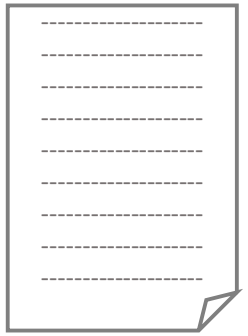
Heroku offers a sandbox platform to rapidly prove the key elements of the GovCore architecture, stay true to open source principles, engage developers in the latest languages and data frameworks and provide DevOps tools for fast configuration, management, resilience and scale.

Architecture

Built on big data



Why three data stores?

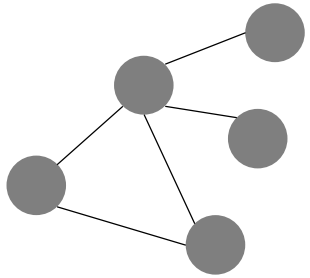


Document Store*

The entities governments manage

Unlimited flexibility

Unlimited mess? Architecture and data management practices are paramount. Design decisions become business-based, not technically constrained



Graph

The relationships between entities

Unlimited relationships

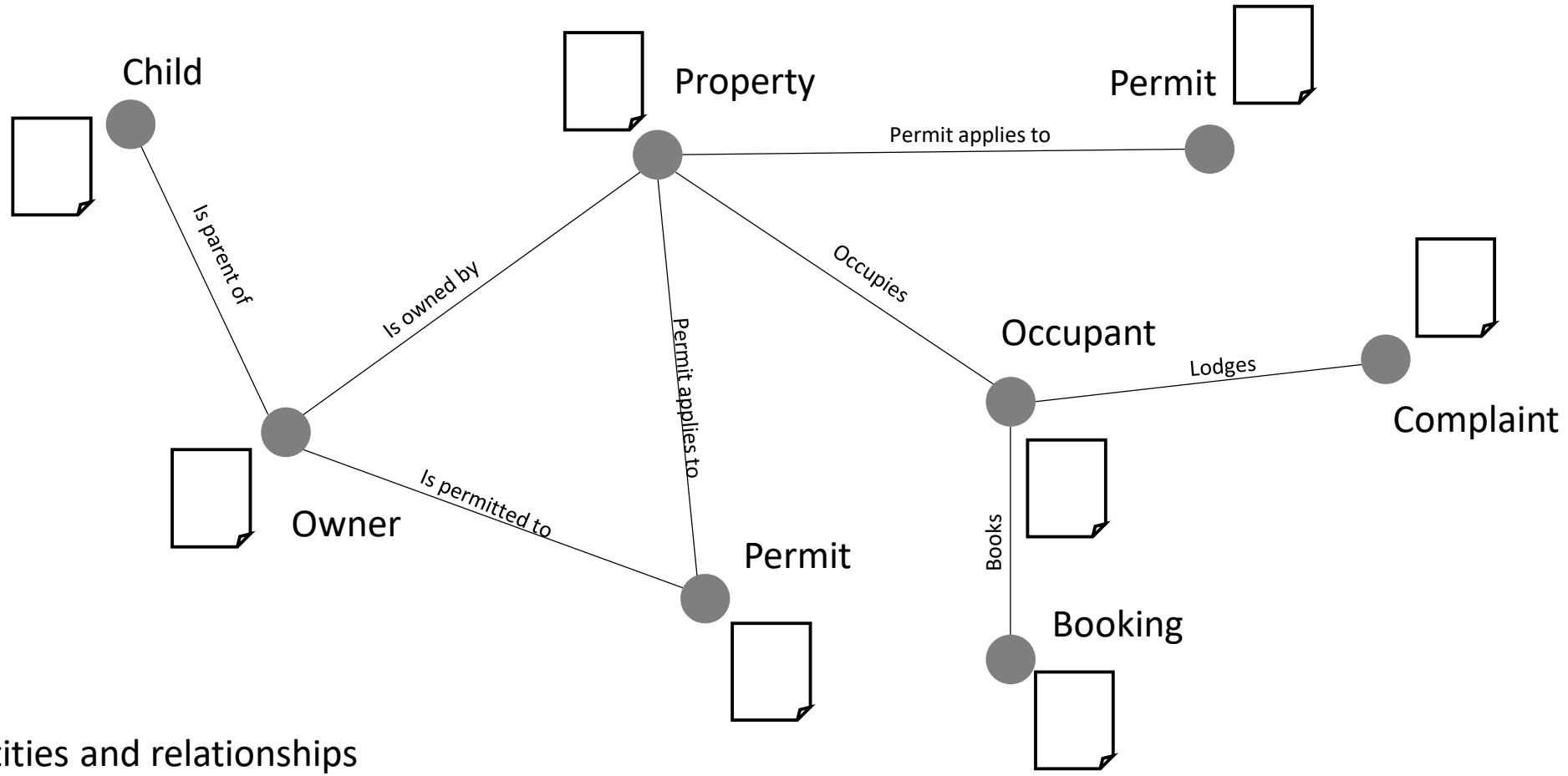
Log Stash

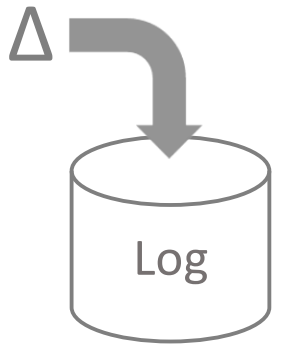
Log every change to entities and relationships

Unlimited scale

* Document store is the source of truth

Combined it looks like this





Real-world entities and relationships reflected much more accurately, flexibly and naturally than relational databases can

What I mean by document (json)


```
{"web-app": { "servlet": [ { "servlet-name": "cofaxCDS",
"servlet-class": "org.cofax.cds.CDSServlet", "init-param": {
"configGlossary:installationAt": "Philadelphia, PA",
"configGlossary:adminEmail": "ksm@pobox.com",
"configGlossary:poweredBy": "Cofax",
"configGlossary:poweredByIcon": "/images/cofax.gif",
"configGlossary:staticPath": "/content/static",
"templateProcessorClass": "org.cofax.WysiwygTemplate",
"templateLoaderClass": "org.cofax.FilesTemplateLoader",
"templatePath": "templates", "templateOverridePath": "",
"defaultListTemplate": "listTemplate.htm",
"defaultFileTemplate": "articleTemplate.htm", "useJSP":
false, "jspListTemplate": "listTemplate.jsp",
"jspFileTemplate": "articleTemplate.jsp",
"cachePackageTagsTrack": 200,
"jdbc:microsoft:sqlserver://LOCALHOST:1433;DatabaseName=goon
", "dataStoreUser": "sa", "dataStorePassword":
"dataStoreTestQuery", "dataStoreTestQuery": "SET NOCOUNT
ON;select test='test';", "dataStoreLogFile":
"/usr/local/tomcat/logs/datastore.log",
"dataStoreInitConns": 10, "dataStoreMaxConns": 100,
"dataStoreConnUsageLimit": 100, "dataStoreLogLevel": {
"servlet-name": "cofaxTools", "servlet-class":
"org.cofax.cms.CofaxToolsServlet", "init-param": {
"templatePath": "toolstemplates/", "log": 1, "logLocation":
"/usr/local/tomcat/logs/CofaxTools.log", "logMaxSize": "",
"dataLog": 1, "dataLogLocation":
"/usr/local/tomcat/logs/dataLog.log", "dataLogMaxSize": "",
"removePageCache": "/content/admin/remove?cache=pages&id=",
"removeTemplateCache":
"/content/admin/remove?cache=templates&id=",
"fileTransferFolder":
"/usr/local/tomcat/webapps/content/fileTransferFolder",
"lookInContext": 1, "adminGroupID": 4, "betaServer":
true}}], "servlet-mapping": { "cofaxCDS": "/", "cofaxEmail":
"/cofaxutil/aemail/*", "cofaxAdmin": "/admin/*",
"fileServlet": "/static/*", "cofaxTools": "/tools/*"},
"taglib": { "taglib-uri": "cofax.tld", "taglib-location":
"/WEB-INF/tlds/cofax.tld"}}}
```


Universal Header (metadata)

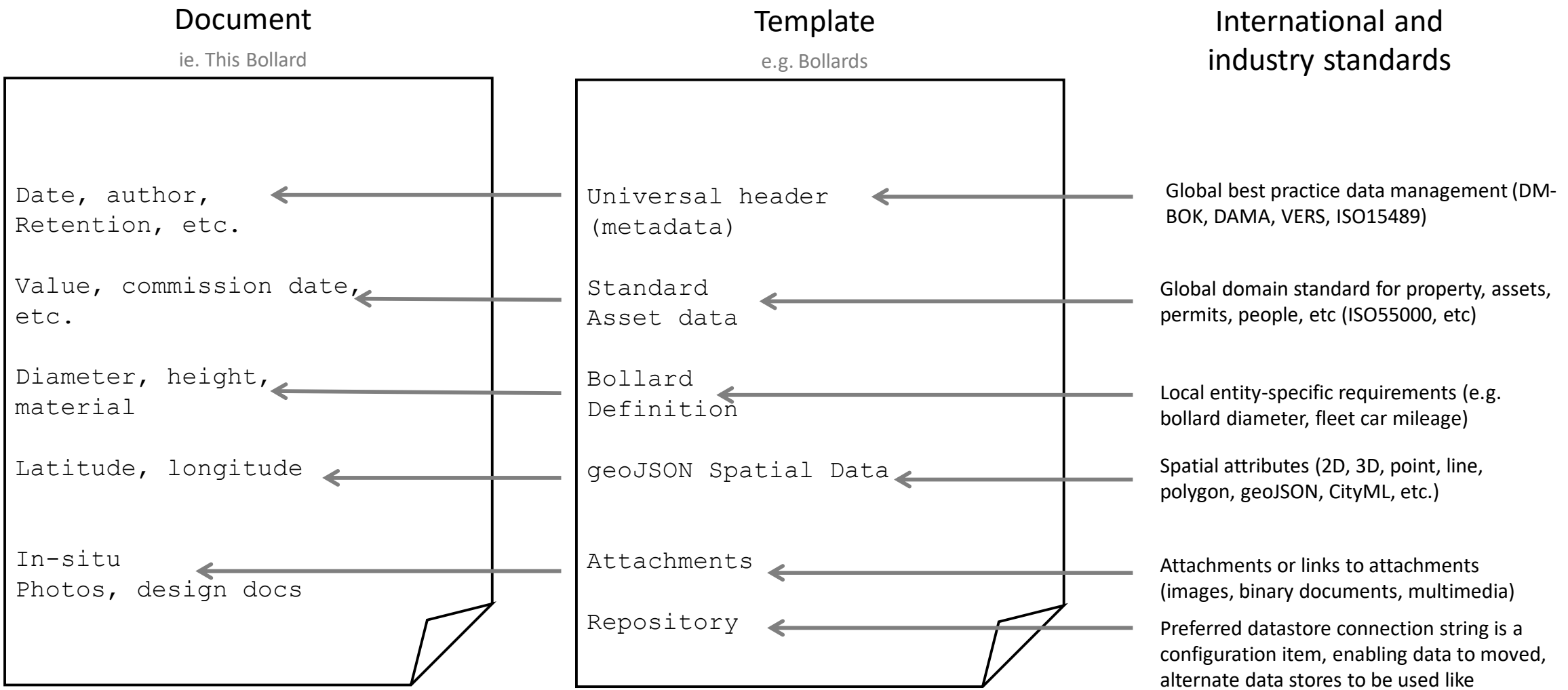
Super Class

Class

geoJSON Spatial Data

Attachments

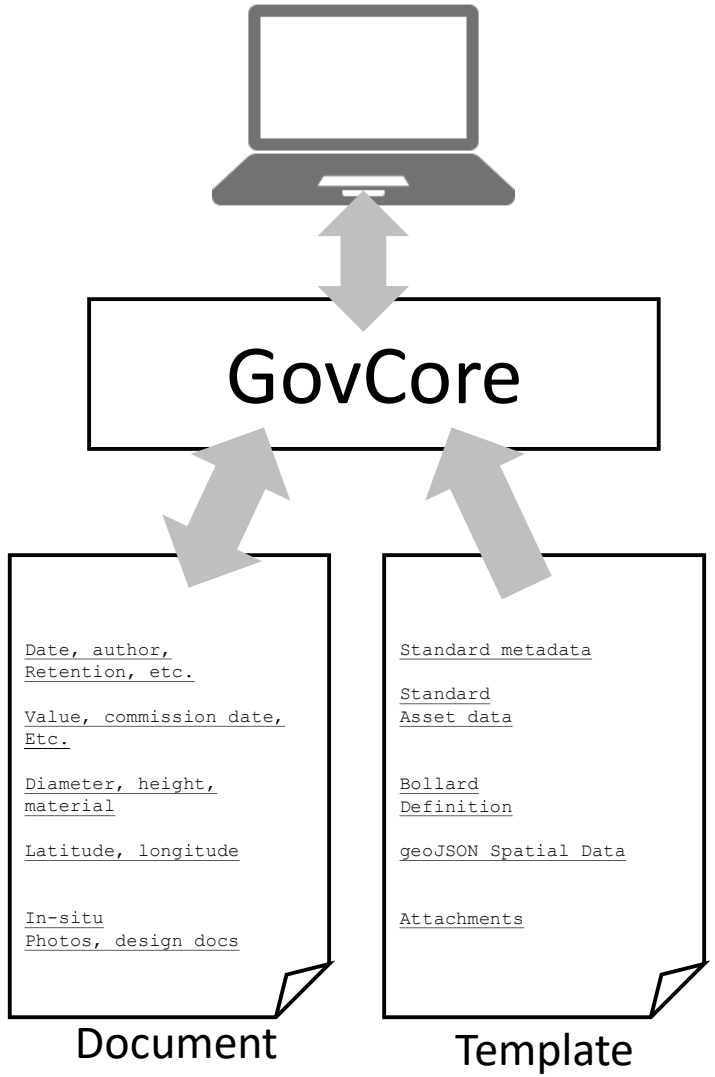
Ensuring data quality



Data in each document governed by template during create/update

Key value pairs in templates define data validation rules, lookups, types

Minimum code 1 of 2

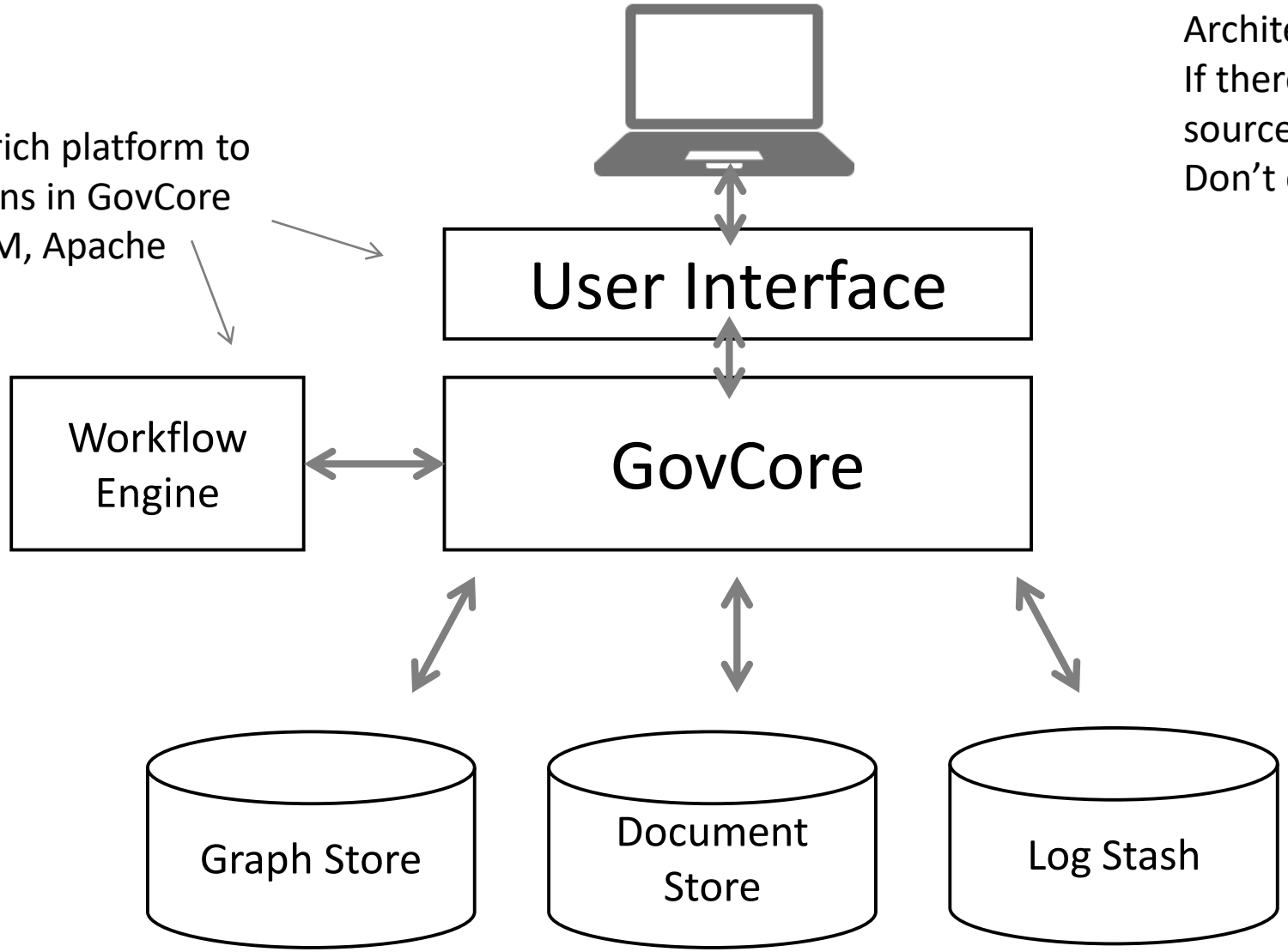


The primary function of GovCore is to combine json documents with the associated json template (with inherent data rules) and present them to a user

The ultimate in reductionism – the only entity (class/object) the code has to manage is a ‘document’.

Minimum code 2 of 2

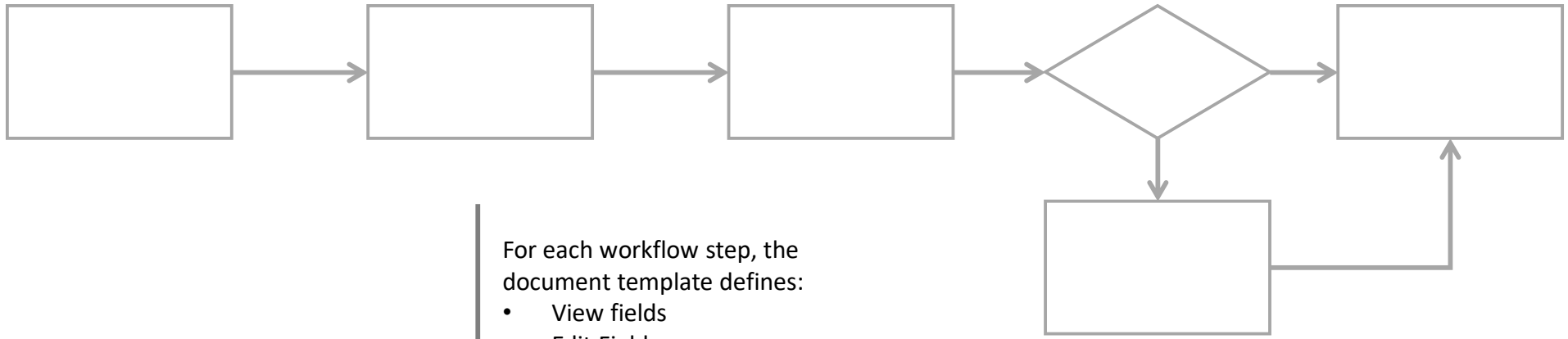
Seek a function-rich platform to minimise functions in GovCore (e.g. Drupal, jBPM, Apache ODE)



Architecture principle –
If there is a credible open source function already, use it.
Don't code if you don't have to.

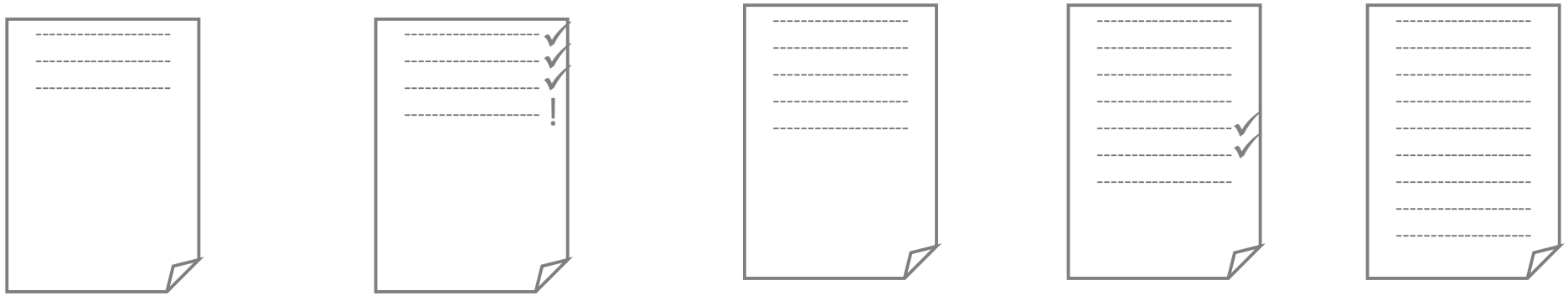
Workflow

Envisaging government as an information processing ‘production line’, taking ‘raw’ information in (e.g. requests and applications), processing it and exporting ‘value-added’ information (e.g. permits, decisions, etc). All processes are born digital – no paper!

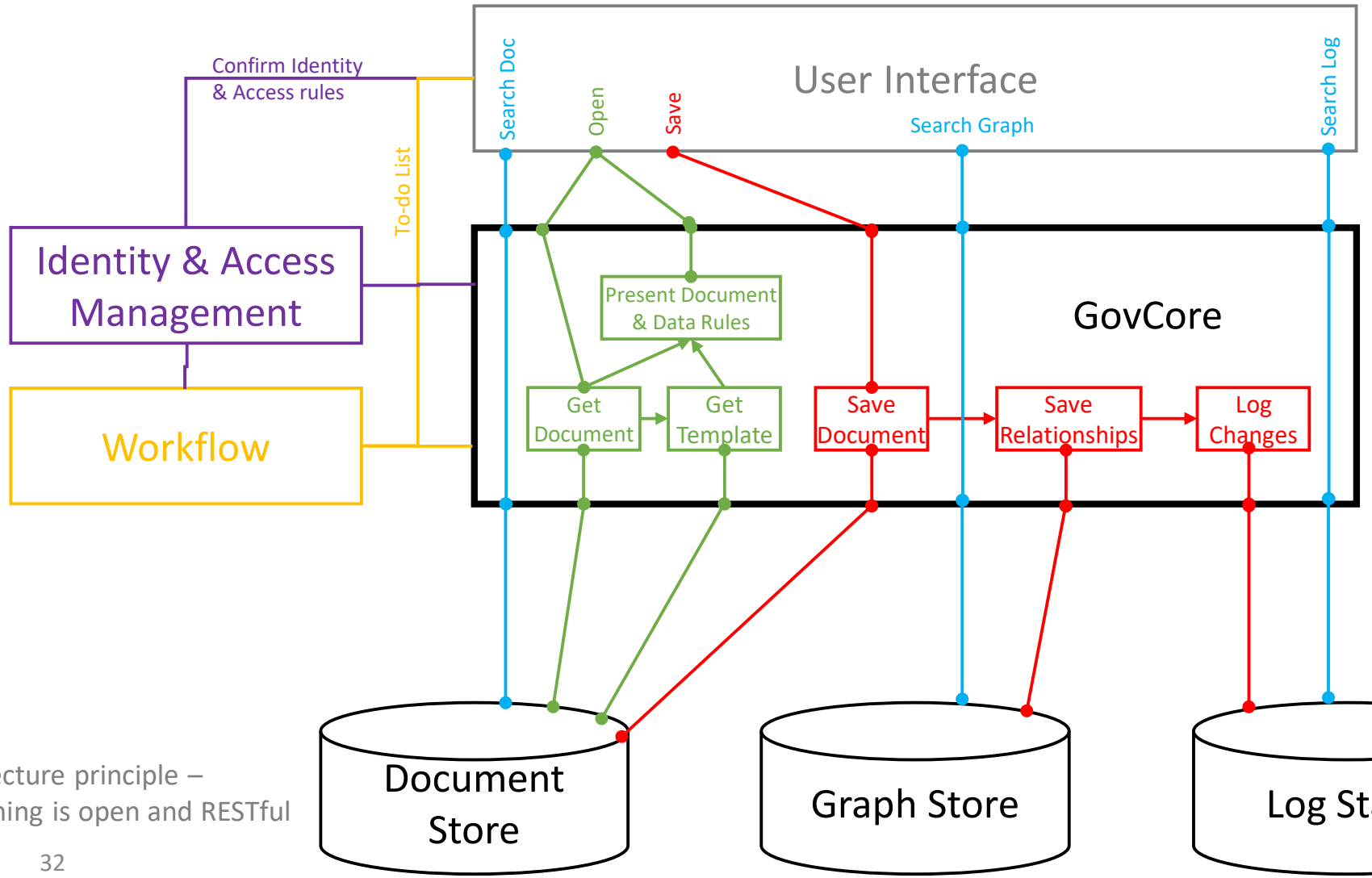


For each workflow step, the document template defines:

- View fields
- Edit Fields
- Mandatory Fields



Architecture



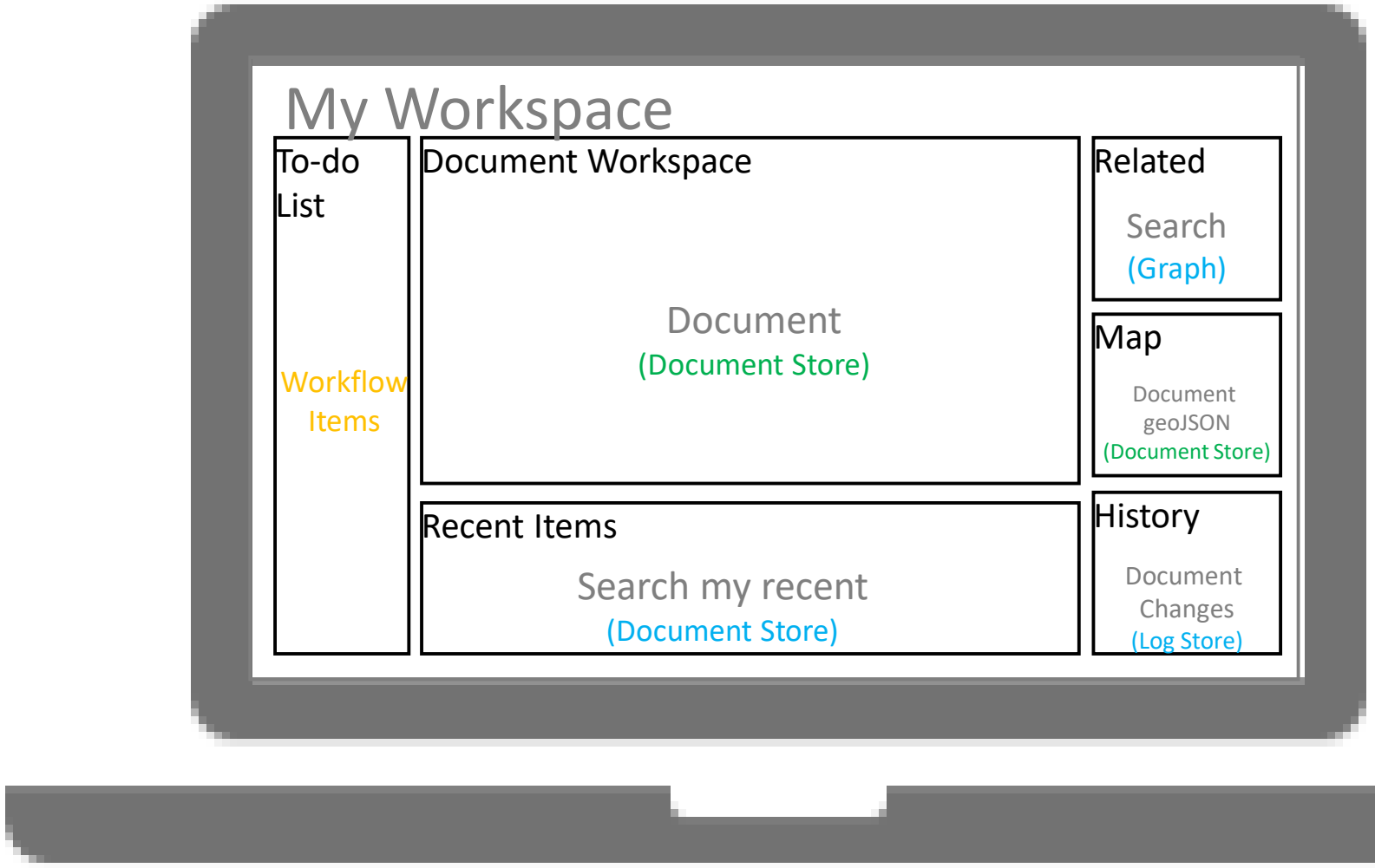
INNOVATE HERE !!

After a flurry of activity hammering out the GovCore code in the early stages, it will become relatively static in the long term (due to its tight scope and basic function).

After that, innovation flourishes in template and workflow design and improvement as well as freeing UX designers and developers to explore and create new user experiences.

Architecture principle – Everything is open and RESTful

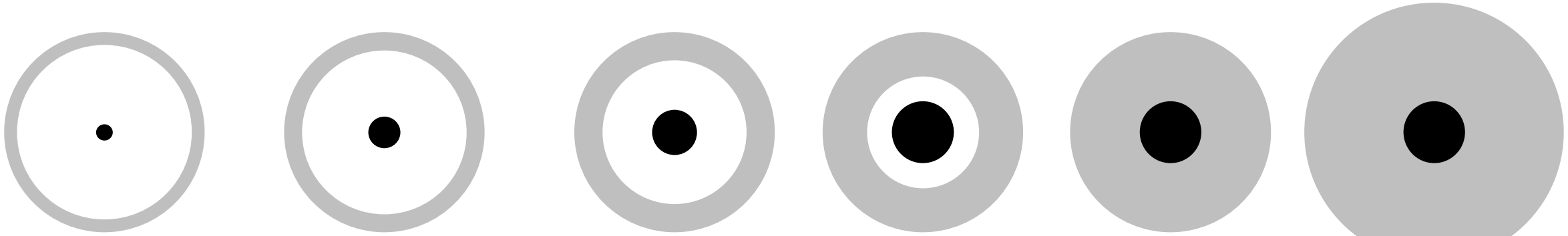
User experience



A fertile ecosystem of user experiences to suit different agencies' needs and aspirations built around a common core.

A basic user-configurable interface with flexible widgets is just one possibility

Agile Build approach



Build approach will be a typical use-case by use-case incremental build. At some point the core will be complete and become relatively static. However, innovation around user experience, workflow, data analytics will continue to snow-ball.

Worth a try …

Curious to know if it would work…

Beyond MVP

Log Stash

Logs in traditional relational systems are huge, unintelligible, drag on the performance of the system and often fall short of ever-increasing auditor expectations. A big data log stash logs orders of magnitude more data and provides searchability and advanced analytics.

Transparency and Accountability	Performance and Improvement
When | Someone | Did something | To what	For whom | Cost | Revenue | Co_2^e | Satisfaction
20/04/2018 12:19 | Officer 123 | Created | Infringement Notice	Licence plate ABC 123 | $43 | $120 | 0.0001 T | ☹
20/04/2018 12:20 | CR Agent | Created | Service Request	Cust ID 1234 | $16.75 | $0.00 | 0.002T | ☺

By logging every single time a document or graph-relationship changes, every conceivable audit request can be addressed.
Combined with universal version control on documents, a complete record of government activity is created, assuring the community of unprecedented accountability and opportunity for appropriate transparency.

These are deliberately the lowest common denominator metrics that apply to every public sector service or activity. Agencies are highly creative in coming up with a wild variety of metrics. Here we start with:

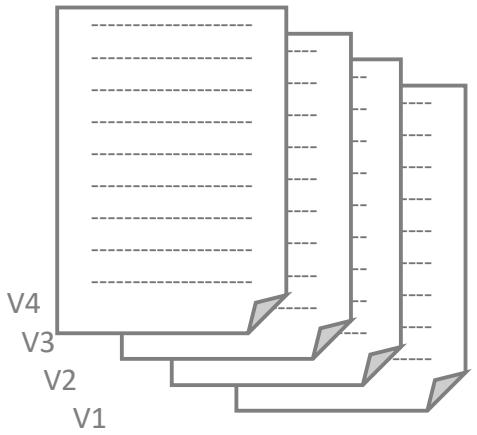
- Number of customers
- Cost of interaction
- Carbon footprint
- Number of interactions
- Revenue of interaction
- Customer Satisfaction (e.g. NPS)

These six universal metrics provide the foundation for standardised performance analysis across all government activities.

Version Control

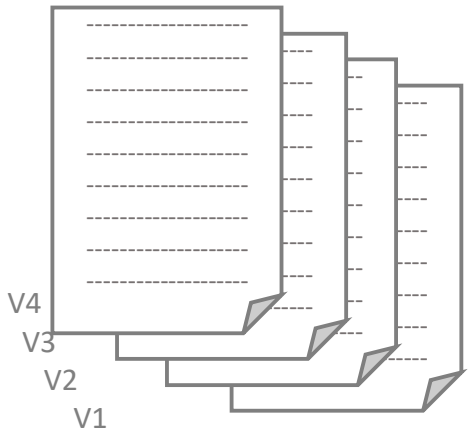
To bake rapid continuous improvement in from the start, we have to assume everything will always change …

Template



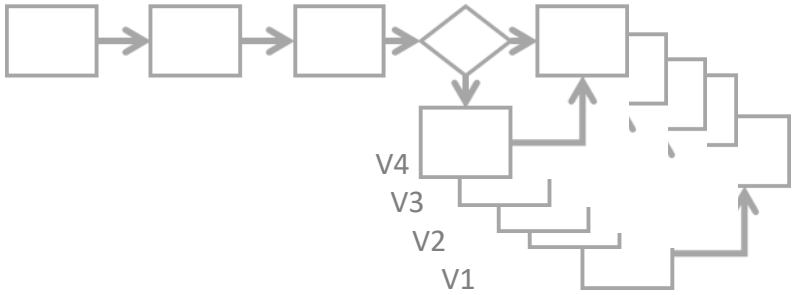
Options are needed to define what happens when templates are changed: i) keep documents on old version, ii) update documents as they're re-saved or iii) retrospectively apply change in bulk

Document



Despite using big data platforms with theoretically 'unlimited' scale, there is a cost implication – hence records retention and archiving policy remains relevant.

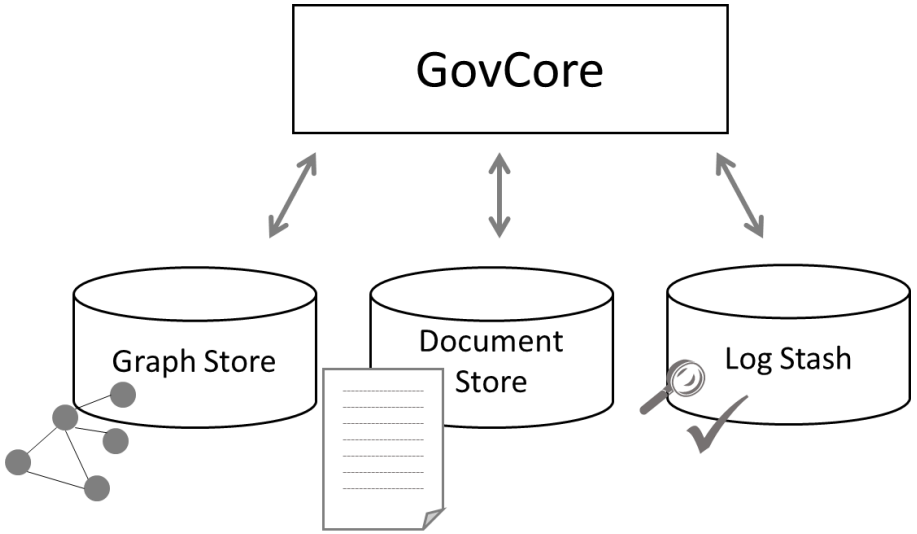
Workflow



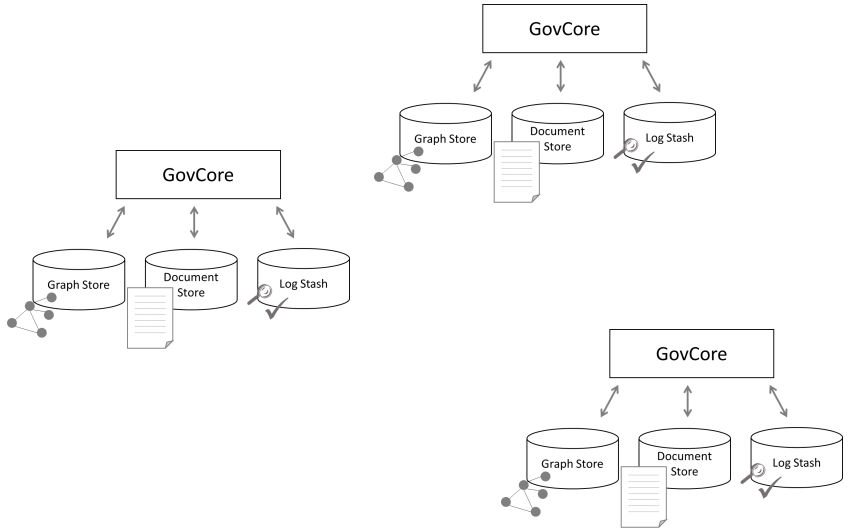
Version control is required for workflow, to enable in-flight work-flow updates; 'open' documents continue with the old workflow, new documents follow the new workflow.

Architecture Principle – everything is versioned

Lakes and puddles – deployment options



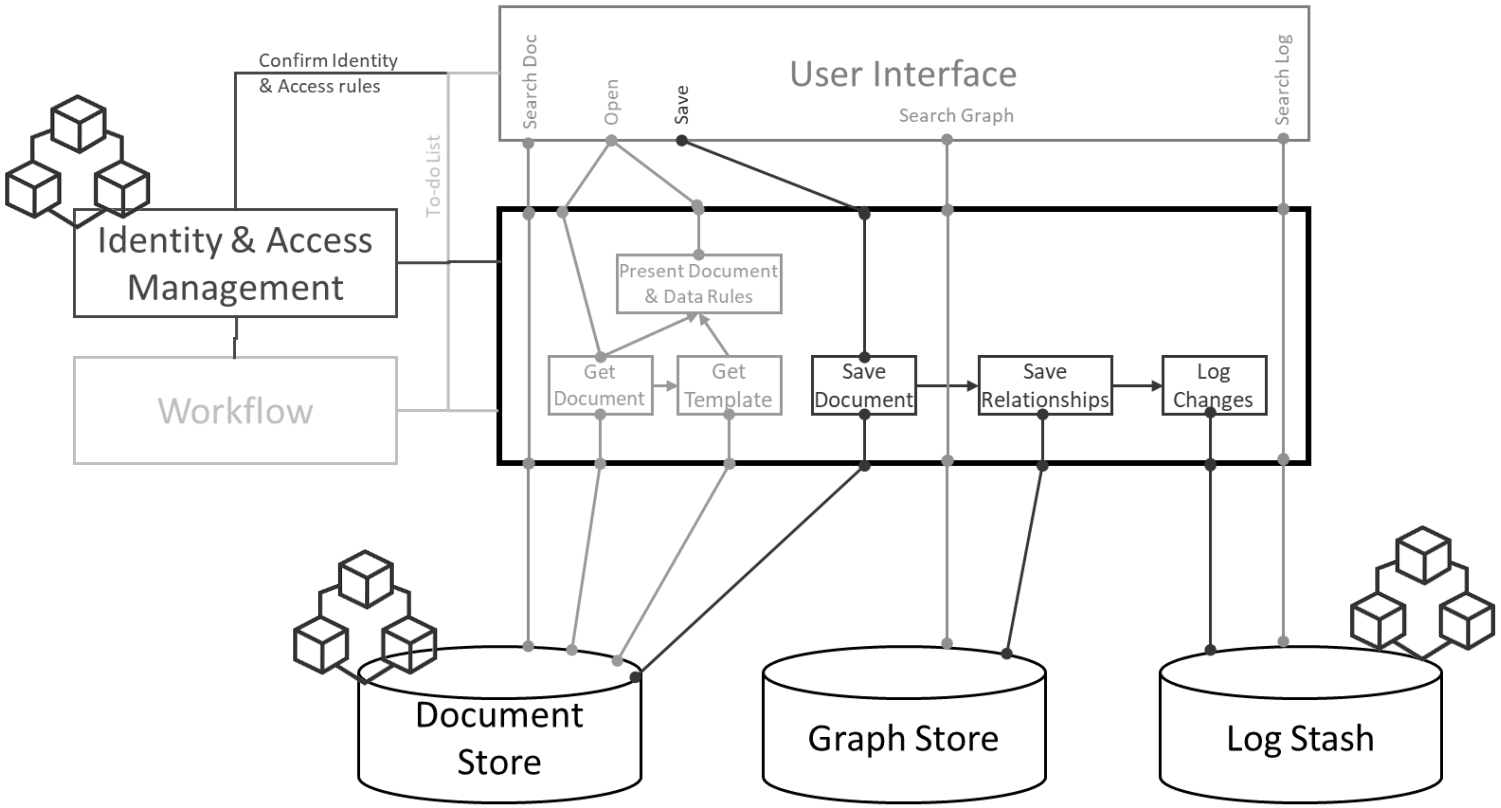
A single monolithic implementation is feasible with a single large document datastore across multiple domains.
A downside of this is a single point for failure and a data honeypot.



Smaller separate implementations for different purposes may reduce risk but increase maintenance overheads and potentially data integration challenges.

Blockchain

Because GovCore is green-fields, blockchain support can be baked in from the start providing options for an immutable ledger …



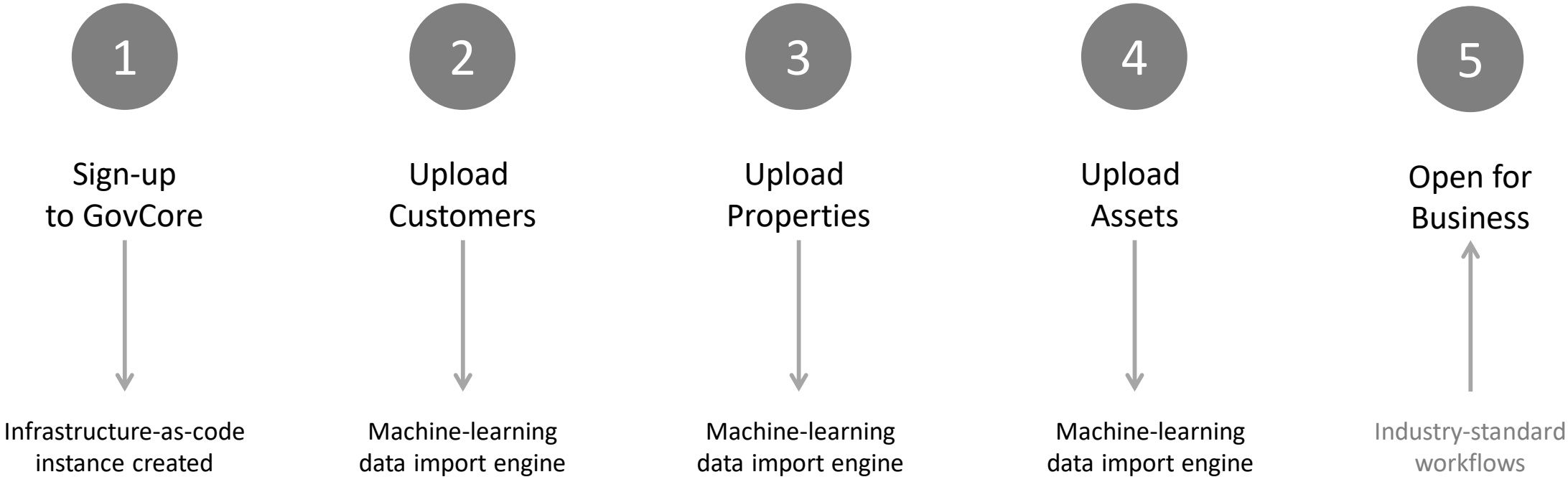
Architecture principle – blockchain is compute-intensive* so use it wisely

Candidates could include:

- Self-managed identity
- Public Registers
- Public marketplace permits
- High-risk audit items
- Entitlements, rights, trades

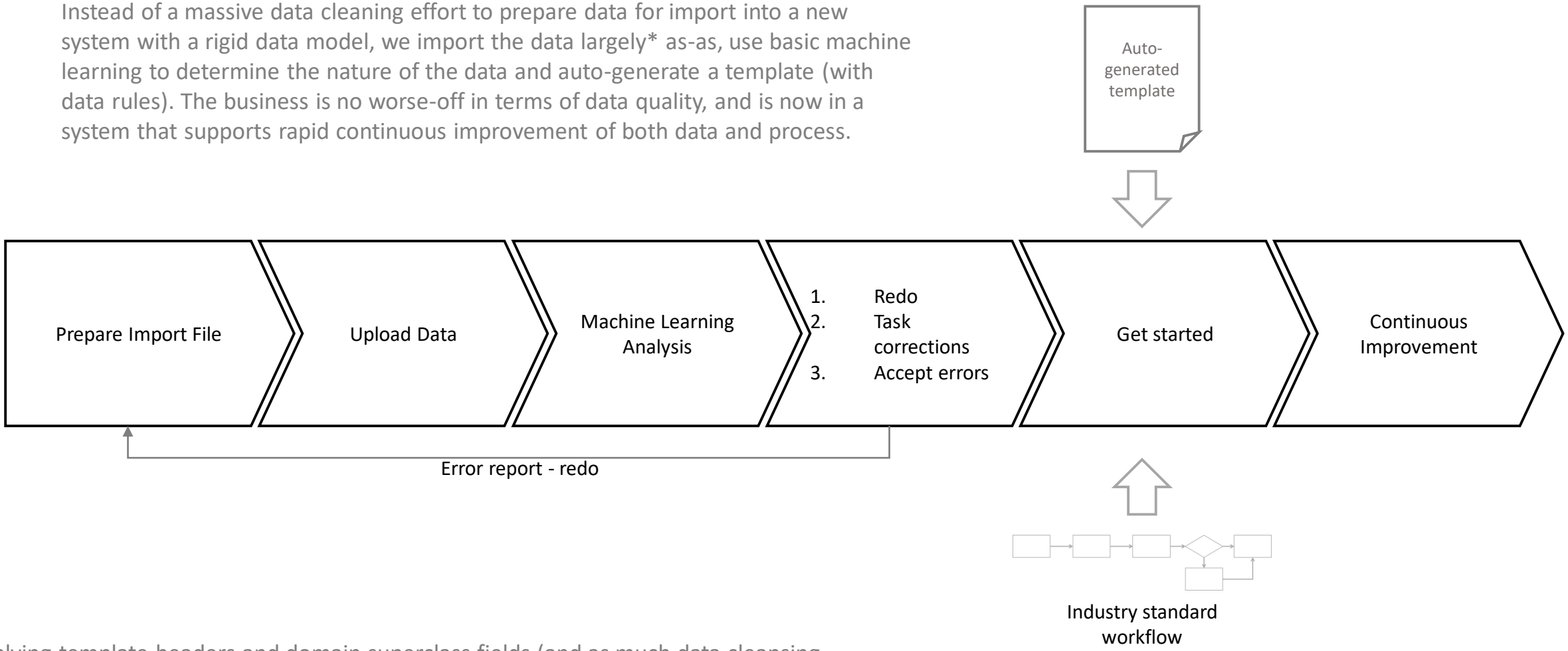
* i.e. for now, blockchain still requires lots of compute resource to encrypt large payloads (documents)

Long-term aspiration: Onboarding Wizard



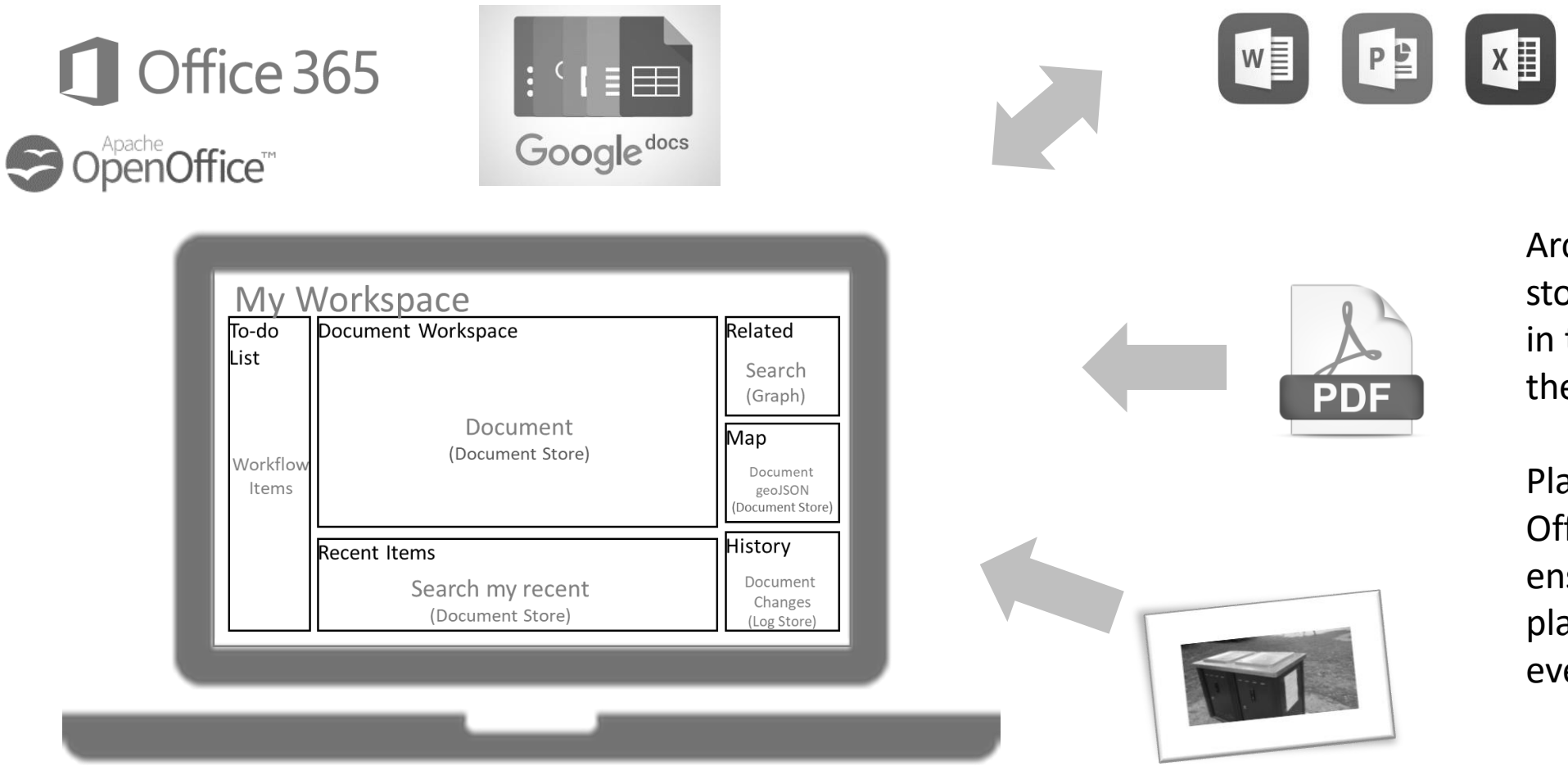
Auto-configuration

Instead of a massive data cleaning effort to prepare data for import into a new system with a rigid data model, we import the data largely* as-as, use basic machine learning to determine the nature of the data and auto-generate a template (with data rules). The business is no worse-off in terms of data quality, and is now in a system that supports rapid continuous improvement of both data and process.



* Applying template headers and domain superclass fields (and as much data cleansing as possible) will yield better results, obviously – but not mandatory because document store can hold 'bad' data until its touched by a user and template rules enforce correction on save. Templates can then be progressively improved and standardised.

Document Editing and Linking

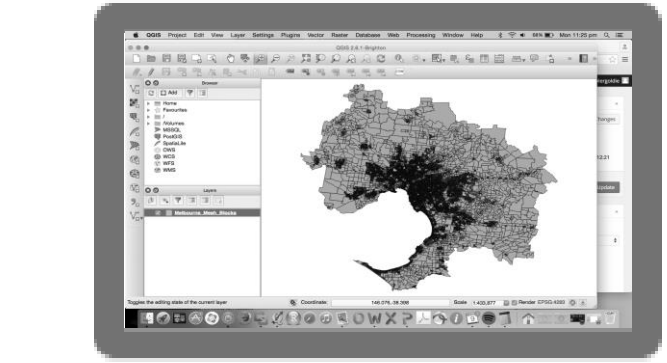


Architecture principle – store (binary) attachments in the system that manages them

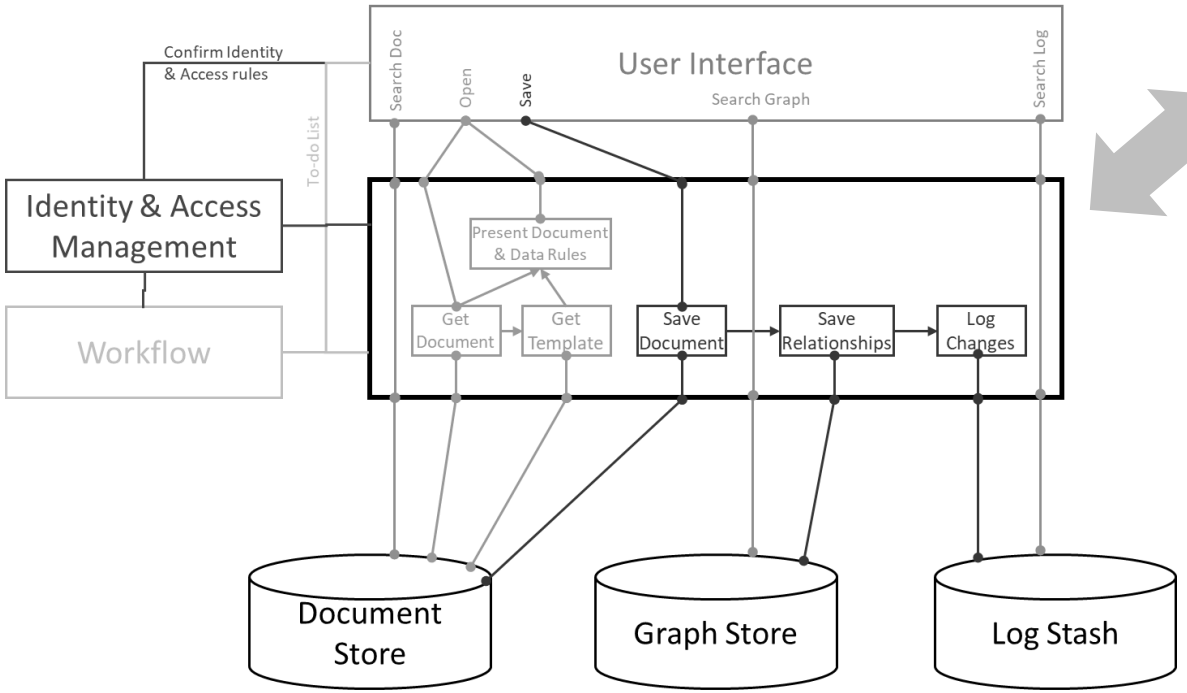
Platform should support Office365 and Google – ensuring portability and platform independence at every layer

Spatial Data Maintenance

Traditional architectures separate the spatial attributes of an object (in a specialist spatial system) from the aspatial attributes (in a typical line-of-business system) making data maintenance vastly onerous, duplicative and error-prone.



QGIS

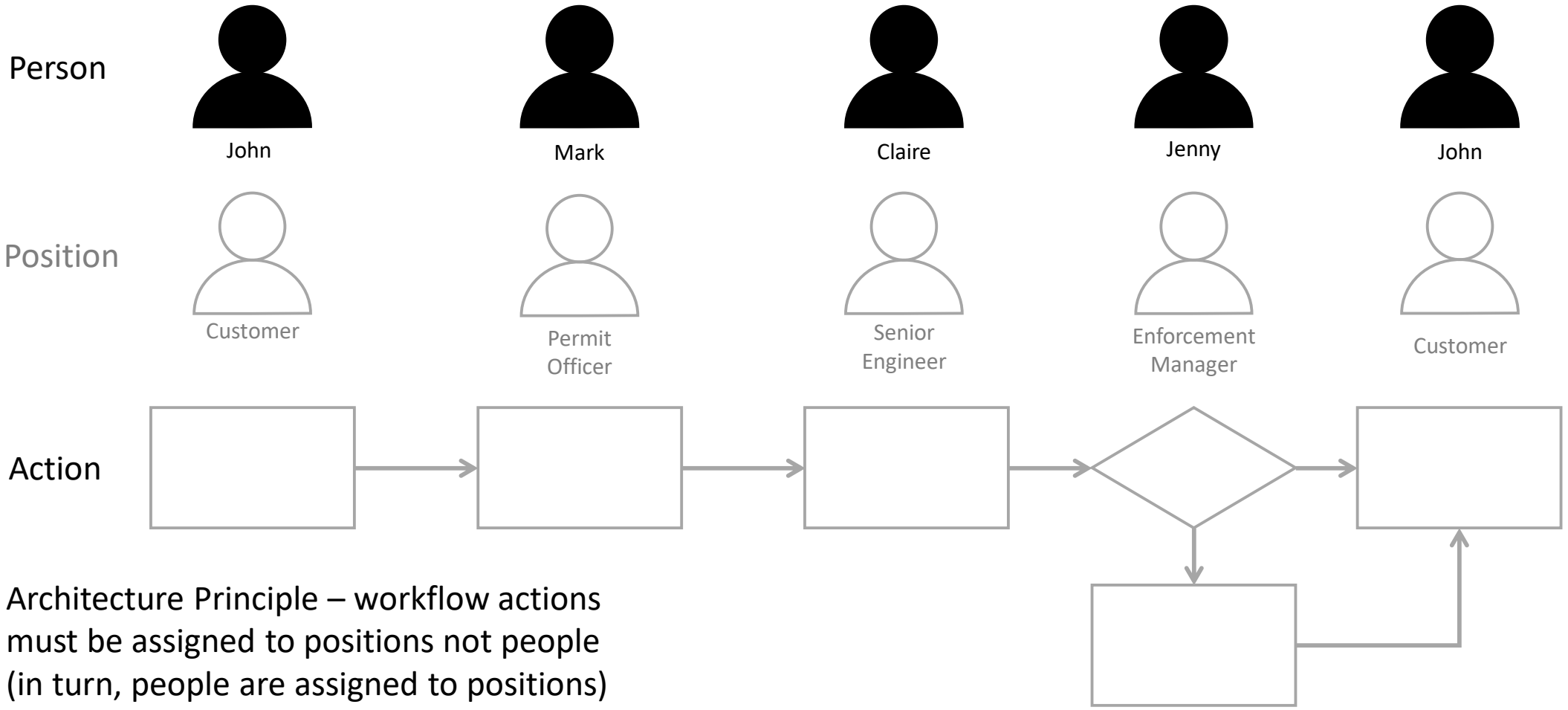


The solution needs to solve the challenge of individual and bulk maintenance of spatial data – combining spatial and aspatial attributes in a common document structure is a huge step forward.

Platform independence …

esri MapInfo AUTODESK. Bentley Advancing Infrastructure INTERGRAPH CITY OF MELBOURNE creative commons

Position

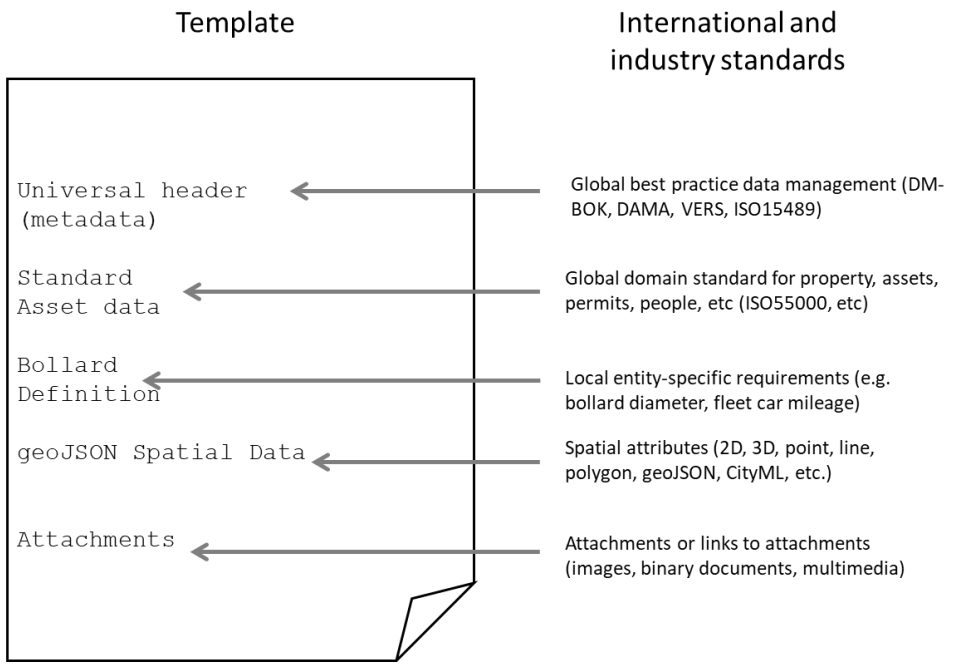


Architecture Principle – workflow actions must be assigned to positions not people (in turn, people are assigned to positions)

Position Description: an interesting artefact is that you could extract the responsibilities of a Position from the actions assigned to it ...

Records and Standards Compliance

Global standards can often be theoretical and unachievable and entirely dependent on the capability of vendors, the capacity of agencies and the will of legislators. Similarly, after 60 years of computing, the gap between records management best practice and actual data management practice seems insurmountable.



In this model international standards for data management and domain best-practice are instantiated in the document templates (as well as workflows).

Moreover, changes recommended by global and regional standards bodies can be directly incorporated into new template versions.

In this way, the gap between aspirations for both business and data best practice and reality can be closed.