Foundations of the Digital State

Working Papers Volume I

An independent report for Scottish Government

Gordon Guthrie

This volume is part of series containing:

Foundations of the Digital State
ISBN: 978-1-0685451-0-8 (paperback)
ISBN: 978-1-0685451-1-5 (epub)
ISBN: 978-1-0685451-2-2 (PDF)

Working Papers Volume I - Foundations of the Digital State
ISBN: 978-1-0685451-3-9 (paperback)
ISBN: 978-1-0685451-4-6 (epub)
ISBN: 978-1-0685451-5-3 (PDF)

Working Papers Volume II - Foundations of the Digital State
ISBN: 978-1-0685451-6-0 (paperback)
ISBN: 978-1-0685451-7-7 (epub)
ISBN: 978-1-0685451-8-4 (PDF)

List of Working Papers

These are the collected working papers referred to in *Foundations of the Digital State* – an independent report for Scottish Government.

VOLUME I

WP X *The heart of the beast*
WP 0.2 *The locus of change*
WP 1.2 *Data and the rule of law*
WP 2 *Rules as code*
WP 3 *The Lego state*
WP 4 *The remixable state*
WP 5.1 *Law reform for data*

VOLUME II

WP 6 *A solera for data cleansing*
WP 7.2 *Experimental digital legislative processes*
WP 8 *An Enabling Act*
WP 9.1 *Reading legislation with a non-functional eye*
WP 10.2 *Immediate hygienic measures*
WP 11.1 *Jeff Bezos' API Mandate, but for government*
WP 12 *A theory of state*
WP 13 *The weak centre*

The appendix to WP 9 is available on request from the author.

These were written when the author was a Research Fellow at Scottish Government under the First Minister's Digital Fellowship programme.

The views in these papers do not represent the views of Scottish Government.

Table of Contents

Working Paper X - The heart of the beast

Version 1.0

Functional and non-functional specification in the digital state

Introduction

A short story about Enron

Enron is an interesting tale – a paradigmatic tragedy, a 3 act play marked out by hubris and nemesis, the high-flyer brought down by their own flaws.

In the first act Enron is a traditional pipeline company with a slightly-shady taste for coorying up to elected pols and regulators. In the second act it is a high-flying, market-making, energy-trading company. In the third act the denouement unfolds, criminal conspiracy and fraud flourish and finally the great bust flowers.

All the focus is on the nemesis, the creation of the first off-balance sheet entity to pour failing projects into, but the hubris is fascinating in its banality. The transition from pipelines to trading comes at a board meeting. A powerpoint deck is presented, the board argue for 8 hours, the ship changes direction, *le jeux sont faites*. There is a detail, a wrinkle tho, a hook. The slide deck consisted of a single slide.

This paper, with its aberrant name, flows from my long fascination and speculation with 'an Enron slide'. What is there that can be summarised so, that can cause you to spin on a sixpence? Could I, should I, would I ever write one?

It turns out the answers are, yes, yes and yes, even tho I didn't intend to. Whilst preparing diagrams for Working Paper 7 – *Experimental digital legislative* I doodled one up and it struck me with the force of revelation. After 20 years thinking about these problems, my Enron slide had just wandered over and pronounced itself to me, as the unicorn to the pure, the chaste, the Virgin Mary, mother of god herself. The key that unlocks digital transformation was in my grasp.

To my great joy the slide itself is a masterclass in powerpoint banality: unassuming, modest, as chaste as the virgin herself.

Who are you?

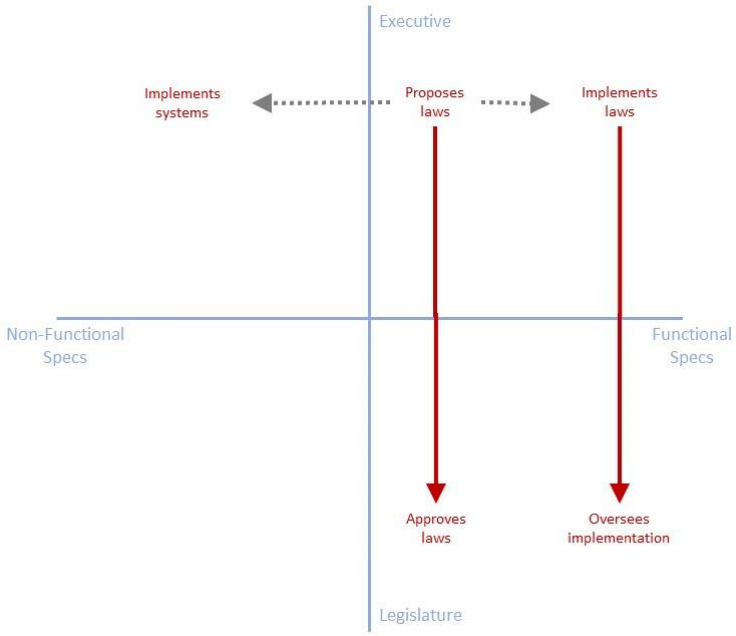
You are an elected member, Minister or Spad, a think-tanker or policy person, somebody in delivery trying to build out or drive joined-up government.

Why should you read this?

You should read this to help understand the structural problems of the emerging digital state and their origin in the analogue state – and the steps governments will have to take to address them.

The heart of the beast

You will be wanting me to cut to the magic slide. So here it is – prepare to be underwhelmed.



In this slide I am mapping the journey from policy to system and its supervision on a quadrant.

The upper side is the executive side, the government, the side of agency: what we plan to do and how we do it. The lower half is the supervisory side – where the legislature keeps an eye on the executive.

The right hand side is the functional specification, and the left the non-functional specification. Non-functional here is a term of art in software development – meaning *everything-not-in-the-functional-spec* and not *it-doesn't-work*. You could think of it as the Context Specification or the Foundation Specification – the things you need to do to be able to do the thing itself.

Functional specifications of computer systems are the things that make it distinctive. A social security system will collect details of people's personal circumstances and pay out (or not pay out) benefits accordingly. An alcohol licensing system will collect data about pubs and their location and staffing and allow or deny the sale of alcohol on a premises.

The non-functional specifications of computer systems are not distinctive. The people who administer via social security systems and licensing systems alike must log in. The data will be stored in databases and backed up in both cases. There will be specific volumetrics (10,000 pubs Vs 1,000,000 people) but generally you need to squint to tell them apart.

So the government wishes to do something – it proposes a law – which is

itself a functional document – it speaks of the criteria and processes of social security or of licensing. It is largely silent on the non-functional specification.

The parliament scrutinises the legislation – and this scrutiny must be functional only – there are no non-functional elements to be considered. When approved the government moves to implementation – functional and non-functional alike. In the case of a social security bill, a social security department or agency is created (a functional body) and the operations of that body are monitored by a parliamentary social services committee (a functional body).

So the challenge, the light bulb, the "oh!" is that the non-functional specifications are just... there. At this stage I expect you to be scratching your head and thinking "why <oh!> and not <what?>?"

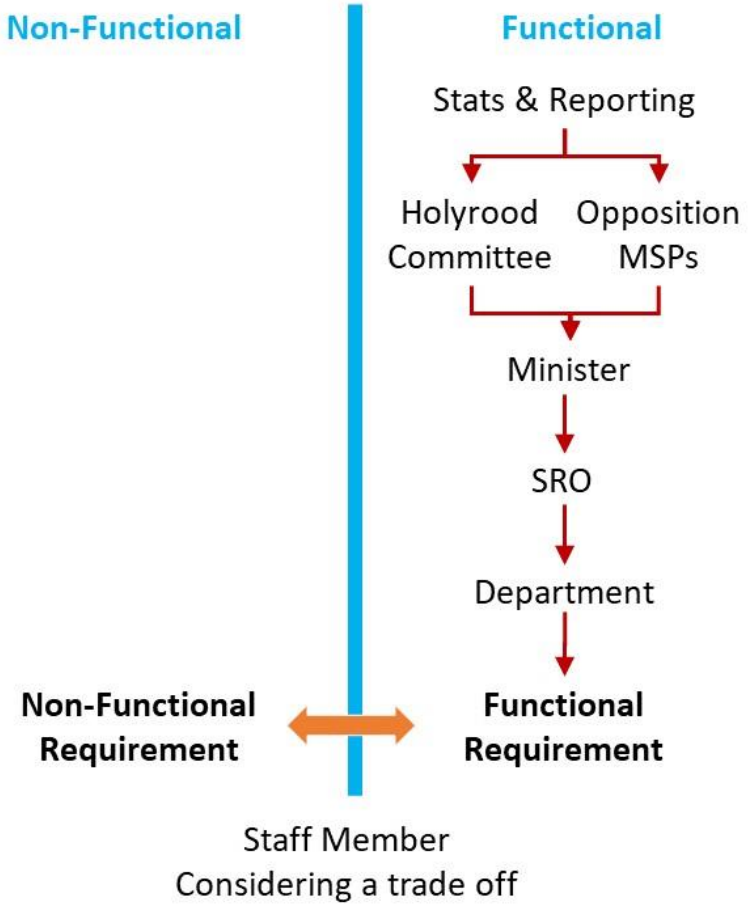
The thing is the non-functional part contains such elements as: *joined-up government*, *data sharing* and *public sector transformation*. Oh.

These things are *willed* in manifestos and ministerial declarations but the organisation of the state lacks the *means* to deliver them or supervise them.

If you have every wondered why the UK has had 12 major public sector transformation programmes and 25-odd data sharing ones, well here you go. The *will* without the *means*.

To understand how multiple initiatives over many years, with the highest level of political backing, all eventually melted into the sands, it is best to turn the problem on its head. You are a civil servant in the heart of the beast, making calls about technical issues in an implementation and you hit a trade-off between implementing a functional requirement and a non-functional one.

You are wrestling with what should be done, what is the right thing, what pressures there are on who to resolve it and all the absolutely normal questions of digital implementation. This is what your pressure and management lines look like:



For the sake of argument let's say you work in the department of Education – a functional agency. Your senior responsible officer is a functional officer responsible for delivering Education. Their minister is a functional minister for Education, who in turn answers to the functional Education committee at Holyrood. The oppositions MSPs are informed by stats and reports which are issued on a functional basis.

From a non-functional perspective there is a desire to see Education working collaboratively with Social Services.

In the case of a clash that requires a trade-off, how does the collaboration ever win?

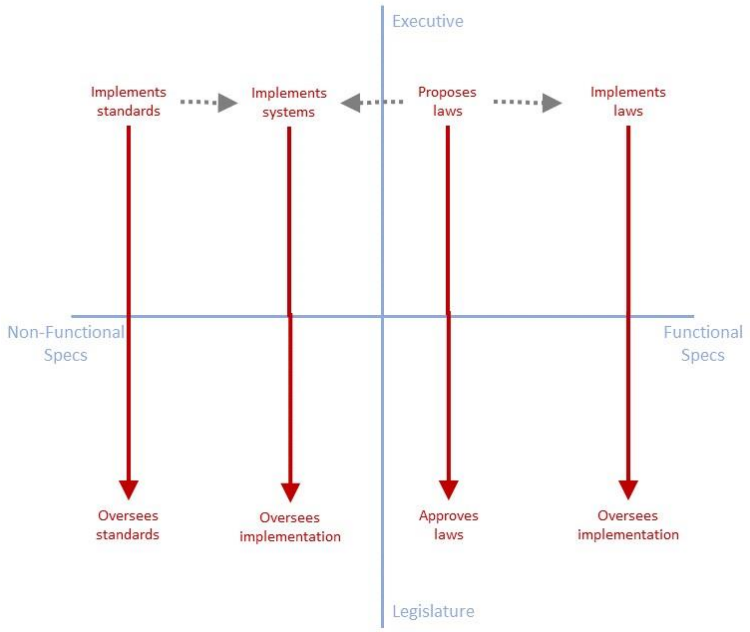
The mystery is not how government programmes swerve off the road into silos, its how anyone expected anything different.

The importance of this slide is that it provides an organising principle for all the work that has hitherto been done. It is the nexus mundi, the ophalion. I now have an analytical frame for intuiting things we should be doing that haven't

emerged so far in the interview/discussion/writing/review process. Every single cog in the end-to-end cycle will need to be looked at again through a functional/non-functional lens. How exciting!

How do we fix it?

Becoming the digital state involves filling in the missing quadrants with the appropriate institutions:



Job's a good 'un, over and out, see ya down the pub, right? Uh, perhaps not. The devil is in the detail and this paper will not go into the detail. Some of the detail has already been gone over in the various working papers listed earlier.

Working Paper 0 – *The locus of change* steps through the institutional architecture – the institution that populates the top left – the Digital Reform Office, and the institution that populates the right – the Digital Scrutiny & Audit Committee. This institutional architecture and the implementation roadmap is obviously Scottish Parliament specific but should be trivially generalisable to other jurisdictions.

Working Paper 1.1 – *Data and the rule of law* lays out some of the generic non-functional requirements for all state data.

Working Paper 2 – *Rules as code* is the outlier – it is the only working paper about functional issues – but, wait for it, it starts from the premise that law is a purely functional specification, and discusses the limitations that imposes on Rules as Code – without this working paper this analytical approach would not have developed.

Working Paper 3 – *the Lego state* discusses how to use common non-

functional requirements across the state to reduce cash/tax cost, transactional time costs for citizens, and improve flexibility.

Working Paper 4 – *The remixable state* builds on the Lego state and looks at what non-functionals are required to make a state that can be reorganised to refocus on different problems and areas.

Working Paper 5 – *Law reform for data* sketches out a legislative architecture, changes to the structure of legislation, to enable data sharing to happen.

Working Paper 6 – *A solera for data cleansing* discusses an implementation process for moving state data from the old world to the new.

Working Paper 7 – *Experimental digital legislative processes* looks at embedding iteration and learning cycles into the parliamentary process to both kill runaway projects and enable more flexible and effective development of new digital systems.

Working Paper 8 – *An Enabling Act* describes a technical mechanism to handle the volume of incremental change that is required – accidental blockers of transformation embedded in law.

So how did we get here?

To understand how we got here we need to be back to the analogue state. Consider social security – the manual, paper-based social security system was also specified in functional and non-functional terms (although that analytic distinction was not used).

The functional specification is familiar from the digital age – collect this information (just on a paper form) and make this decision and record it (just in a paper file).

The non-functional specification – the common elements – is much more prosaic. There will be a building, it will be big enough for the staff, it will have a roof and windows and electricity and be near a bus stop and a car park...

The non-functionals scarcely interact with the functionals – can the building hold the number of people required? can the floor support the weight of paperwork that will be generated?

The analogue non-functional work was all boxed up and implemented. There are legal pattern books for it – a body needs these powers to own a building, and an apparatus for handling it embedded so deeply in government that its hard to remember that it's even there.

When digital came along with its tightly coupled non-functional requirements we didn't change the processes – the non-functionals went mostly

unspecified.

This seems like a contradiction – nobody specified the non-functional requirements, you can't get a system without implementing the non-functional requirements yet we do indeed have government digital systems. What gives?

Well, humans are autonomous beings wrapped in a self-propelled meat envelope and they kinda solve problems and do stuff.

Its worth looking at an under-internalised period of British statecraft and organisation building – 1918 to 1919. On the 14th December 1918 the UK called a general election – the coupon or khaki election and new MPs were elected to celebrate the end of the war. In Ireland the Sinn Féin MPs went to Dublin and declared themselves Teachta Dála of the Dáil Éireann instead.

The Dáil was an organ of *will* and decidedly not *means* – being an underground government. It willed the civil administration and taxation of a new state. Men in slouch hats with guns conjured up the non-functionals – pubs became sites of local administration, a legal/illegal Dáil Loan bond was issued – the real oul Sinn Féin conjuror's outfit.

So it is with the current state. Left to themselves the slouching engineers and technical specialists conjure up the non-functionals each tailored to their functional silo. Adieu joined up government, adieu data sharing, adieu transformation.

Working Paper 0 – The locus of change

Version 1.3

From transformation to capability

Introduction

What is meant by the locus of change?

Up to now digital transformation has followed a common pattern. A set of goals are announced by members of the wider political class: cost savings, targets around some output, social and cultural aspirations. These targets are to be met by rubbing some tech on it: variously e-government, the internet, social media, big data, blockchain or artificial intelligence. There will be a programme do to it (the rubbing) onto the civil service that will in some unspecified way 'transform' government.

The BIus project takes an orthogonal view. 'Transformation' is a by-product of applying technology to administration and communication. This application should be approached as a question of capability – and the task now becomes *how to we increase the capability of the state apparatus for the digital age*.

It might seem counterintuitive, but switching the focus from transformation to capability will see more and not less transformation – because capability compounds like interest. And capability, like muscles, grows with being exercised. The emphasis has to be on the capability of the state in the whole, not in the part. Centralisation weakens.

Digital systems bring opacity and complexity to the heart of the administrative state. That complexity needs to be encapsulated and separated from high politics. This encapsulation empowers technical experts to get on with doing 'the good stuff'. This is the locus of change moving.

But whilst the private sector must woo with honeyed words, the public sector can compel with bayonets. Any and all public servants must be appropriately overseen in a constitutional manner – they are creatures of law. Civil servants may propose changes to *policy effect*, but *policy intent* must remain the preserve of elected politicians. Technical standards are a form of weak law, and they must be developed in public, in the open, in a parliament of standards.

The model of explicit transformation has as its twin a CapEx (capital expenditure) funding model. By contrast the BIus approach has an OpEx (operational expenditure) focus.

There is a saying *data ages like wine and code ages like fish*. Data is a strategic asset of the state – and the digital systems that engage with it require constant maintenance.

In parliamentary terms we have two regimes – oversight of law and oversight of money. And parliament votes government the legal authority to do

something and the money with which to do it separately.

Data (and the digital systems built over it) are strategic assets (and liabilities) and should be managed as such – and part of that management is constitutionally-appropriate oversight.

Moving the administrative state to the digital administrative state is a constitutional moment as well. Governments come and governments go but infrastructure pertains.

There are precedents as to how to manage disruptive and opaque technologies in the public sector – the management of atomic energy being one example considered here.

WHO ARE YOU?

You are someone who cares about the administrative state. You might be minister or parliamentarian, a civil servant or worker in the 3rd sector, a journalist or public intellectual, or simply a citizen and voter.

WHY SHOULD YOU READ THIS?

You need to read this to understand the constitutional and institutional changes required to make the modern state fit for purpose in the modern world.

The discussion is general, but the examples are taken from the UK and Scottish states.

Revision Notes

Version 1.1	The original version of this document referred to the Digital Services Reform Office and the Digital Scrutiny And Audit Commission – they have been renamed the Digital Services Reform Office and the Digital Services Scrutiny And Audit Commission to better reflect their remit.
Version 1.2	New section discussing Ottawa Digital Services and a reference to Working Paper 11
Version 1.3	Ottawa Digital Services correctly names as Ontario Digital Service

The complex state

Back in 1911 Lloyd-George masterminded the general election and created the basis of the welfare state. He wrote the legislation, designed the administration down to the Lloyd-George cards and oversaw its roll-out. That world of a wholly comprehensible major government programme, conceived, designed and directed by a single mind is long gone.

Digital systems are opaque and hard to reason about at the best of times. Under fortnightly releases and constant change, doubly so. *Contra* popular belief this opacity extends to technical experts and people who work in the field – nobody is blessed with some magical x-ray vision to peer into the abyss and see the skeleton, muscles, tubes and organs of large digital systems. It is not for nothing that the dominant technical methodologies focus very heavily on getting things to a state where actual people can start using them as quickly as possible, no matter how limited their functionality. The actions of people when using digital systems are far more comprehensible that the underlying systems themselves. Engineers only know what they are building when they can watch people using them.

Complexity didn't arrive with the digital revolution. But digital takes complexity from the margins of the state into the centre.

The good news then is that managing complexity isn't unprecedented - it is a challenge that modern states have risen to repeatedly.

The solutions and arrangements that we need for the digital state should be easily found. If not exact solutions, certainly ones that rhyme with those we need.

It is worth revisiting the introduction of a semi-tangible discontinuous technology and its associated complexity into the modern world – atomic energy – which is discussed later as a precedent.

On capability

The purpose of this proposal is to increase the capability of the state to do strategic rearchitecting of state institutions to better deliver the desired outcomes of the government of the day by leveraging the opportunities that technology offers.

In its first phase it will involve a large degree of activities to deliver quite low-level technical hygiene. The technological changes will be enabling of change at the higher level, and not as ends in their own right.

The problem of digital in the state currently is not the inability to *conceptualise* how technology might change things, its is an inability to *do* the work, or more properly to *co-ordinate* the doing of the work by taking strategic decisions and methodically implementing them in a programme of continuous improvement.

There is a very strong expectation that at Westminster a new government is à-comin in. The thinktanks of Whitehall are pullulating with plans and schemes for the new regime to implement. Oftentimes they share a presumption that *if*

we can conceive the future then all we need do it is will it – and that the will[1] needs[2] to be stiffened[3] by a strong centre.

By contrast this proposal calls for a *weak centre* with a capability and not a delivery focus – concentrating on *habit*, *tooling* and not *will*. An organisation is what it habitually does.

State digital systems need to be capable of:

- being found
- being understood
- interoperating
- being extendable
- being composable
- emitting desired outputs and interfaces automatically through tooling
- being able to be reasoned about
- being able to be consolidated and improved
- being able to be measured and assessed

The mechanisms for driving these capabilities are standards – which are a mechanism for shaping and making habitual working practices such that disparate and autonomous delivery organisations can achieve harmony without direct communication and control structures. Standards embed ways-of-working in culture. Loose-coupling is a core organisational architectural attribute that we need to work towards.

In precedential terms this is a General Staff[4] model. The central general staff lay out grand objectives and the field commanders (in this case Senior Responsible Owners in the various departments and projects) retain day-to-day autonomy and flexibility in how best to meet those objectives. In career terms personnel move between delivery roles to the centre to learn about planning

---

[1] https://www.institute.global/insights/politics-and-governance/new-national-purpose-ai-promises-world-leading-future-of-britain *Boosting how Number 10 operates, dissolving the AI Council and empowering the Foundation Model Taskforce by having it report directly to the prime minister.*

[2] https://www.ippr.org/research/publications/the-smarter-state The centre of government should be made stronger and redesigned around the missions and Prime ministerial commitment: These Cabinet committees tend only to be successful if there is sufficient PM attention and political capital invested. If attendees know that they need to show up prepared and with results, they can be effective.

[3] https://www.hks.harvard.edu/sites/default/files/centers/mrcbg/files/Final_AWP_216_2.pdf

Many interviewees argued that support for regional growth requires strong and united leadership in Westminster and the enthusiastic backing of the Prime Minister – a high bar to clear, which explains why regional policy has struggled to gain traction.

[4] *pace* Dominic Cummings, ooh-la-la

standards and how-we-do-things and then rotate back as key advisors to the SROs. The proposed central organs more closely resemble the EU or the Holy Roman Empire[5] than GosPlan[6].

This model is critical because the driving motor of continuous improvement must be funded via operational expenditure/daily work (OpEx) and not the old fashioned big-programme approach of capital expenditure/central programmes (CapEx). The centre sets strategic standards and approaches (but can flash steel if required to make the departments comply) but the delivery and in-service teams have responsibility for the *when* – the weaving of change into their daily work. The sheepdog cares that the sheep go through the gate, but each sheep makes it's own path, jostles as it sees fit.

The Senior Responsible Officer has obligations beyond delivering to standards, and work in dynamic circumstances where external events happen that need to handled. At the core of this approach is the recognition that any and every system that involves software and data has a maintenance budget – some organisations recognise this, some try and pretend they don't. The key is to spend that in a way that brings the required harmony.

During the work of the BIus project I have not found any instance of lack of individual or team capability – there are not missing functions in the public sector that are found in the private, nor are the civil servants unqualified for their jobs. The capability that is lacking is on a state/organisational level – and it is this that needs to be addressed.

Precedent 1 – taming atomic technologies

To say that politicians don't understand the details of atomic energy is an understatement. The mysteries of quantum mechanics, the smush of neutrons and protons and the meson family, the charm of quarks, flavoured as up, down, strange, charm, top and bottom, is quite the mystery to them. You might say most people don't even know the αβγ's of it.

So how did we handle this abrupt intrusion? And what can we learn about how to handle digital?

In 1946 the UK passed the Atomic Energy Act 1946[7]. This gave the Ministry of Supply over all radioactive minerals and powers regarding plans for atomic

[5] Beter wordt het niet; een reis door de Europese Unie en het Habsburgse Rijk http://www.carolinedegruyter.eu/#books-2

[6] https://blogs.lse.ac.uk/politicsandpolicy/gove-ditchley-lecture/

[7] https://www.legislation.gov.uk/ukpga/Geo6/9-10/80/contents

energy.

This roughly corresponds, rhymes as you might say, to the relationship of the state to the digital world now. Ministers control it, undifferentiated.

Then with the Atomic Energy Authority Act 1954[8] which created the United Kingdom Atomic Energy Authority, a process of encapsulation began. Atomic energy would not be directly controlled by politicians but indirectly by experts with a skeleton of oversight erected over it.

The UKAEA was to have a membership of between 7 and 10, plus a chair. At a minimum 3 of the ordinary members were to be experts in atomic energy, one in finance and one in organisation of workers - ie a trade unionist.

So in the encapsulation it was made clear: technology is too important to be left entirely to the technologists. Non-technologists and non-technological disciplines must be brought to bear.

And the normal constitutional discipline of separation of powers was applied. The Nuclear Installations Act 1965[9] created Inspectors whose job is not to do, but to monitor.

In this case both the UKAEA and the Inspectorate were creatures of the Government and not the Parliament - appointed at the pleasure of Ministers and acting under their direction. The Minister had only an obligation to consult before acting.

It is worth looking at the timelines as well: 1946, 1954, 1965. The process of managing complexity was a learning process - no discontinuous jump from this world to that world.

Atomic energy is integrated into the energy sector by a set of technical and financial contracts. The money ones cover price per Kilowatt and things like that. The technical contracts handle things like adding or removing a nuclear power station from the grid. These technical contracts are shared with other power suppliers, the hydro, the gas-powered, the wind turbines.

The ability to smoothly ramp up and down power generation to meet load is a critical national function. It is only overseen by the democratic apparatus indirectly - as it should be. The democratic apparatus puts in place the structures and divisions of responsibility and lets the technical experts negotiate with each other under that aegis, and subject to those restrictions.

And this national apparatus of supervision is itself wrapped in an international one.

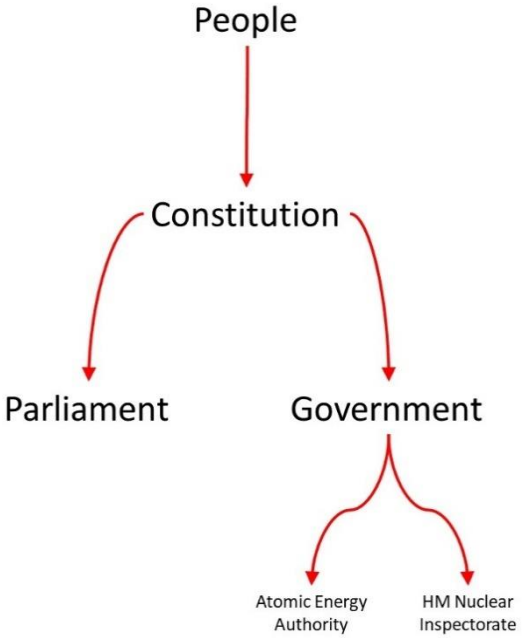
The UK was a founding member of the International Atomic Energy Agency

[8] https://www.legislation.gov.uk/ukpga/Eliz2/2-3/32/section/1/enacted

[9] https://www.legislation.gov.uk/ukpga/1965/57/enacted

in 1957. As an observer but not participant in the Treaty of Rome process the UK didn't join that other 1957 child Euratom until its accession to the EEC in 1973.

The constitutional architecture is:

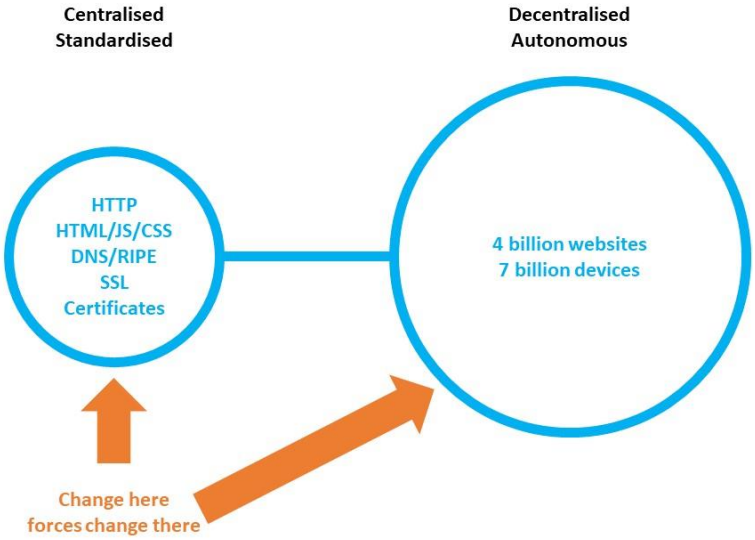


Personally I would put the inspectorate on the other side, but when there's bombs involved governments can get a bit snippy about oversight.

Precedent 2 – web governance, a parliament of standards

Technical standards are a form of weak law – they are clearly not law in the parliamentary and political sense, but they rhyme with it. There are lessons to be learned from internet governance – which is a global paradigm of a weak centre, with a parliament of standards establishing effective governance frameworks and infrastructure that empowers distributed and uncoupled organisations.

The web follows a dumbbell pattern common to decentralised systems.



On one end of the dumbbell there are centralised services and standards – on the other end are a mass of decentralised autonomous websites, services and business.

There is no decentralisation without centralisation – and changes on the centralised end forces changes at the other end. Design decisions at the centralised end are 100 year decisions – the world will be living with them for a long time.

And it is worth recapitulating the size and growth of the *corpus acquis* of the new web developer. The very first version of the world wide web had two standards: HTTP and HTML. HTTPv0.9 was 528 words and the earliest codified HTML was about 4,100 words.

By 1995 the HTTP standard was 17,000 words and HTML grew explosively – in 2023 it comes in at 650,000 words. Other standards grew alongside it - a 280,000 word spec for Javascript and a tangle of hundreds of thousands of words across a maze of documents for CSS.

The vast majority of things that you use your computer for depend on these standards, these weak laws, with no police to enforce them – just habit, convenience, tooling and the value of interoperability.

Web standards are written in public with the circulation of Requests For Comment (RFCs) and consolidation and approval processes. They are consensual rather than adversarial processes. They focus on *effect* and not *intent*.

The critical element to grasp is the necessity to start from the simplest working version. Gall formulated his famous law back in 1975:

A complex system that works is invariably found to have evolved from a simple system that worked. A complex system designed from scratch never

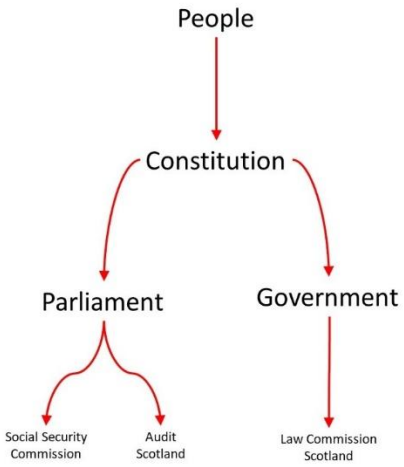
works and cannot be patched up to make it work. You have to start over with a working simple system.

In the development of the proposed structures in this paper the start must be made small by design and from a proper working start rich standards can be developed.

To get a real sense of the simplicity of the early life of a complex system it is worth taking a look at Appendix 1 which describes the technical standards that governed the first 5 years of the World Wide Web and which still structures it today – and will continue to do so over the centuries.

Precedent 3 – separation of powers in the Scottish Parliament

The relevant constitutional architecture of the Scottish Parliament looks like this:



When considering the Social Security Commission and Audit Scotland it is important to understand that these two parliamentary bodies are different in action.

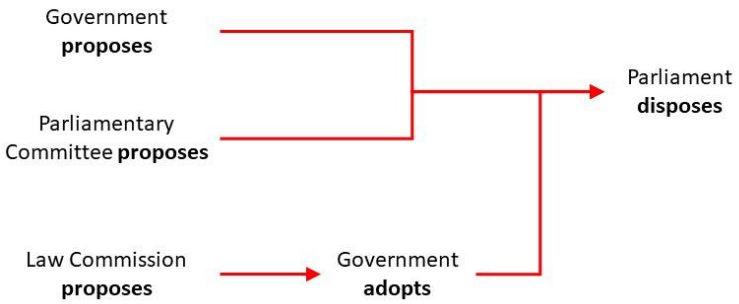
The Social Security Commission is broadly reactive. It examines technical Ministerial Orders on behalf of the parliament.

Audit Scotland by contrast is broadly proactive – it decides who and what it wants to audit, can call for documents and evidence and state bodies have a duty of co-operation – as well charge government bodies for auditing them.

We can see the differences between the three bodies most clearly by examining the legislative routes through the parliament for public bills[10].

[10] There are a range of bill routes not shown here – members public bills, private bills, hybrid bills and so on.

This is the picture for primary legislations (Bills, becoming Acts):



The Law Commission is a body that proposes legislation – the Government has a veto on it – the Commission has a limited right of audience with the parliament and no right to lay legislation. There is no barrier on a Parliamentary Committee freelancing on law reform.

The Law Commission can propose 5 types of legislation:

- Law Reform Consolidation bills
- Consolidation bills
- Codification bills
- Statute Law Repeal bills
- Statue Law Revision bills

Each of these types have their own procedures[11] (9.17a, 9.18, 9.18a, 9.19, 9.20) for handling them in the Scottish Parliament.

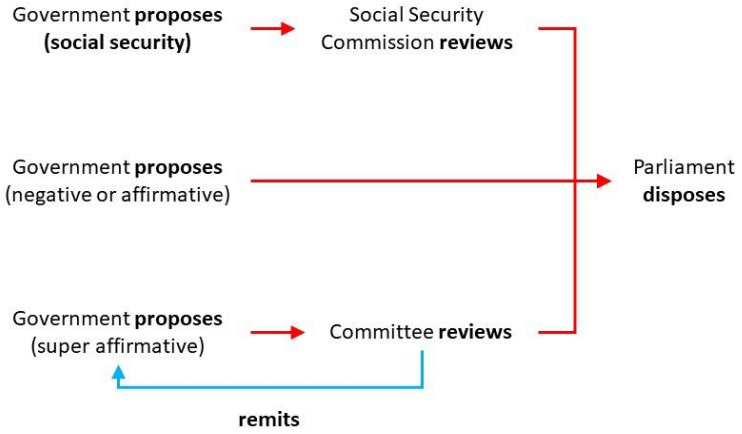
There are pertinent additional restrictions on major bill types. In particular Standing Order 9.16[12] which defines budget bills – these cannot be introduced by committees but only Scottish Ministers – and they have their own proper form and bill pack.

The Social Security Commission has a different role – in relation to secondary legislation (Ministerial Orders).

[11] https://www.parliament.scot/about/how-parliament-works/parliament-rules-and-guidance/standing-orders/chapter-9-public-bill-procedures#topOfNav

[12] https://www.parliament.scot/about/how-parliament-works/parliament-rules-and-guidance/standing-orders/chapter-9-public-bill-procedures#topOfNav

The legislative routes for orders are:



I am drawing a distinction between the operations of lead committees as the *reviewers* for super-affirmative route – which happens pre-laying and their behaviour after laying during the *disposition* of the legislation.

It is worth reviewing the legal basis for these bodies.

The Auditor General and Audit Scotland. The Auditor General is a position under the Parliament and not the Scottish Government under the Scotland Act 1998[13]. The Audit Commission is a corporate body[14] under the Public Finance and Accountability (Scotland) Act 2000 with which public bodies have a duty of co-operation – and which it can charge for its work.

The Public Finance and Accountability (Scotland) Act 2000 also imposes statutory obligations on civil servants – creating the Principal Accounting Officer and departmental Accounting Officers who are responsible to parliament. At Westminster Francis Maude created a parallel accountability structure – adjusting the Osmotherly rules[15] and the Ministerial Code of Conduct[16] - making Senior Responsible Owners directly responsible to the Westminster parliament. In particular, he transferred responsibility for setting go-live dates on major software projects from ministers to civil servants implementing them. This was a major lesson learnt from the debacle of the first 2 times Universal Credit went on the merry-go-round.

The Social Security Commission. The Social Security Commission was created by the Social Security (Scotland) Act 2018[17] to scrutinise Ministerial

[13] https://www.legislation.gov.uk/ukpga/1998/46/section/69

[14] https://www.legislation.gov.uk/asp/2000/1/part/2/enacted

[15] Not an AI hallucination dear readers furth of Blighty https://www.gov.uk/government/speeches/osmotherly-rules-statement-on-updated-guidance

[16] https://assets.publishing.service.gov.uk/media/63a4628bd3bf7f37654767f2/Ministerial_Code.pdf

[17] https://www.legislation.gov.uk/asp/2018/9/section/22/enacted

Orders and Scottish Ministers have a statutory obligation to consult them before laying such orders[18]. The statute makes the point about separation of powers crisply:

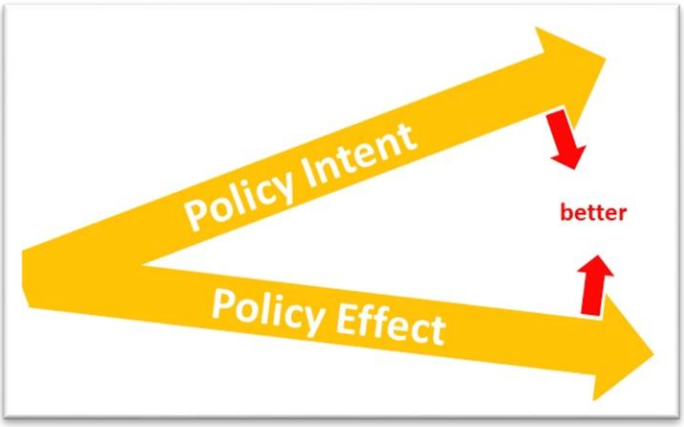
In performing its functions, the Scottish Commission on Social Security is not subject to the direction or control of any member of the Scottish Government[19].

It has an obligation to report to parliament things which it thinks violates the Social Security Charter.

The Scottish Law Commission. The Scottish Law Commission was created by the Law Commissions Act 1965[20]. Its members are appointed by Scottish Ministers. The law commission can propose legislation, and if the Ministers accepts it, it is placed on the legislative programme. The commission must also write a report which Ministers must lay before the Scottish Parliament. Ministers don't have the right to edit the report but do have a right to decorate it with comments.

The Scottish Law Commission is 'constrained' by the policy scissors. *Policy intent* is what the government hoped to achieve with a law, and *policy effect* is what actually happens.

Invariably these are not the same – and the gap varies on a case-by-case basis:



Throughout the BIus project – the word 'better' is used in this narrow sense only – closing the gap in the policy scissors.

The Scottish Law Commission can propose law reforms to meet a particular policy intent at the request of the government and suggest work that would have a policy effect under their own recognisances – and has a right of audience with the parliament for those suggestions.

[18] https://www.legislation.gov.uk/asp/2018/9/section/97/enacted

[19] https://www.legislation.gov.uk/asp/2018/9/schedule/1/enacted

[20] https://www.legislation.gov.uk/ukpga/1965/22/section/2

The distinction between the two blades of the scissors is clear on paper and considerably more murky in the real world. For good constitutional reasons the government has a monopoly of intent and it must adopt law reform proposals if they are to proceed, they are not automatic.

The rational for laying out these architectures, parliamentary routes, accountability lines, rights of audience and remits is that the new bodies, parliamentary processes, accountability lines, rights of audience and remits proposed to handle complexity in the digital state rhyme with them.

Precedent 4 – The Simpler, Faster, Better Services Act in Ontario

Work on institutionalising standards-based governance of non-functionals has already started.

The Simpler, Faster, Better Services Act21 is a Ontario provincial law that puts in place a standards regime.

The Chief Digital And Data Officer is given a statutory duty to make and publish standards. Deputy Ministers are civil servants in Canada. Their work is under the democratic oversight of the Management Board of Cabinet. They must also implement an open data regime, with appropriate security and policy restrictions.

The work must be done in public with a published and regularly reviewed strategy, with all standards and the dataset register published.

The Chief Digital and Data Officer can issue enforcement orders to state organisations directly and must report those enforcement orders to the Management Board of Cabinet.

There is then a reciprocal set of obligations on public sector bodies to follow the standards and make their data public.

The various duties and powers are backed off by a statement of citizen privacy which limits the powers the state has.

During the initial period of the digitisation of Ontario about 100 changes to primary legislation were required to get good, high quality digital services.

Precedent 5 - Amazon

Amazon underwent a transition from being a bookshop to being the major provider of digital platforms and services – and did so by issuing an API Mandate that changed how internal software was built. This mandate was purely non-functional/infrastructural.

21. https://www.ontario.ca/laws/statute/19s07

The story of this is discussed in Working Paper 11 – *Jeff Bezos' API Mandate,but for government* and a draft charter for the proposed Digital Services Reform Office has been written based on it.

Constraints

Any proposals to change how the state creates services and the digital systems they depend on must understand the constraints that exist on the end-to-end systems holistically.

At the moment the Scottish Parliament processes about 22 Bills and 400 Ministerial Orders a year. *Any proposals to change procedures needs to respect those limits*.

We can regard the work of the Scottish Law Commission and the Digital Services Reform Office as maintenance work. Currently the work of the Commission leads to about 1 or 2 bills per session – roughly 5% to 10% of statutes are maintenance work.

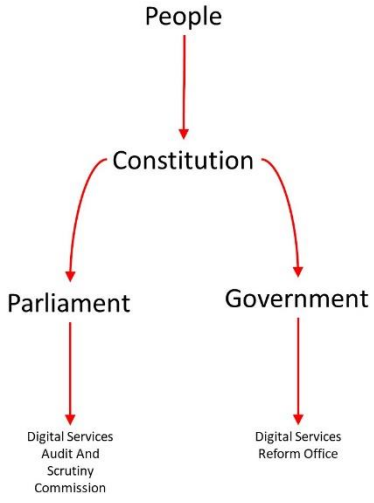
In its early years the Digital Services Reform Office would likely be generating a quantity of changes to primary legislation, either directly to create new institutions or to knock-out barriers to transformation via an Enabling Act. Thereafter the workload should shift to Ministerial Orders.

These proposals constitute a low-touch programme approximately the size of the Law Commission Reform process in the early years before slipping back.

Future state

This paper proposes a schema that rhymes with the current state of separation of powers in the Scottish Parliament.

On the parliamentary side a Digital Audit & Scrutiny Commission and on the Government side a Digital Services Reform Office.



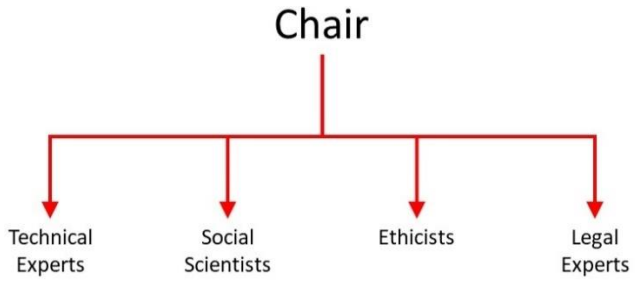
The Digital Services Audit & Scrutiny Commission has two functions (as the name suggests) – scrutinising ministerial orders that pertain to digital and pro-actively auditing activities within Scottish Government.

The Digital Services Reform Office proposes programmes of work on the digital side that rhyme with law reform.

Law reform properly is only concerned with primary legislation, Acts of Parliament (and common law offences) whereas digital reform has a wider remit – covering secondary legislation (Ministerial orders) and what you might call tertiary legislation (regulations and standards).

If we are to treat digital infrastructure as serious national infrastructure then we need parliamentary oversight of digital reform whether it requires primary, secondary, tertiary legislation or just day to day work.

The Digital Services Audit & Scrutiny Commission. Like the UK Atomic Energy Authority, the DSA&SC needs to have technical chops embedded in a wider social matrix:



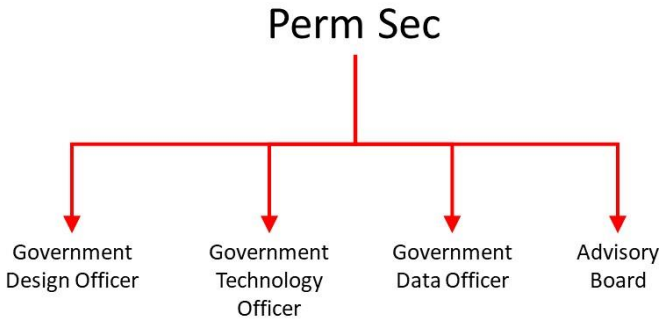
It performs work in both kinds:

- Reactive scrutiny of government instruments as they pertain to technical issues (particularly standards and legislation[22] pertaining to data)
- Proactive scrutiny of government digital systems – particularly with respect to cybersecurity and resilience

The Digital Services Reform Office. Like the Scottish Law Commission the DSRO has a measure of autonomy, limited right of audience with the parliament and a focus on policy effect. Like the Scottish Law Commission its legislative proposals must be adopted by the Government before they can progress.

Unlike the Scottish Law Commission though, the DSRO is a core function of Scottish Government:

[22] This will be the subject of the forthcoming Working Paper



It is important to understand what the DSRO is – it is a core function of the civil service. But is *not* a central delivery organ. The creation of the Government Digital Service in Westminster was, how to put this, constitutionally organic. No great consideration was given to its impact on the structure of government, it grew from its successes. As an unintended consequence GDS turned out to be (yet another) act of centralisation in an otherwise massively over-centralised state. DSRO is a standards body, and a technical leadership body. It provides a career route for the specialist to the heart of the civil service (shades[23] of Fulton, of Haldane and indeed of Maude[24]).

Whereas the Scottish Law Commission only deals with primary legislation, the programme of work that the DSRO might propose may involve primary, secondary and tertiary legislation or regulation. In order to bring this under the supervision of parliament it must be under an obligation as a body to lay an annual report.

In addition, the various officers as functional leaders of their professions need to answer for the work of those professions to the parliament – via mechanisms analogous to those for Senior Accounting Officers and Senior Responsible Owners.

There has been a certain amount of cosplaying in Whitehall and the Cabinet Office with the creation of posts like COO (Chief Operating Officer) and CTO (Chief Technology Officer) that are not analogous to their Silicon Valley equivalents but give the appearance of it. In my naming I have eschewed that dubious pleasure. These are civil service and public sector jobs with all the culture and responsibilities that entails.

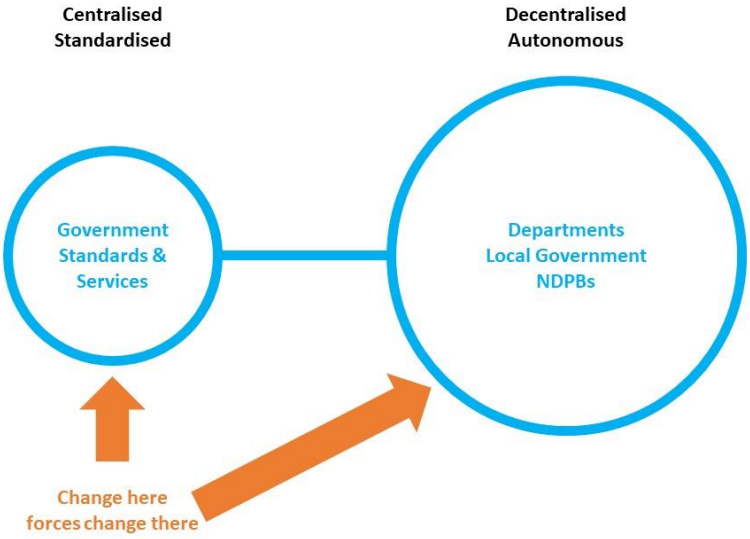
As a standards and leadership body the DSRO is responsible for ensuring interoperability, joined up systems, data management. It should operate in

[23] https://www.civilservant.org.uk/csr-fulton_report-background.html

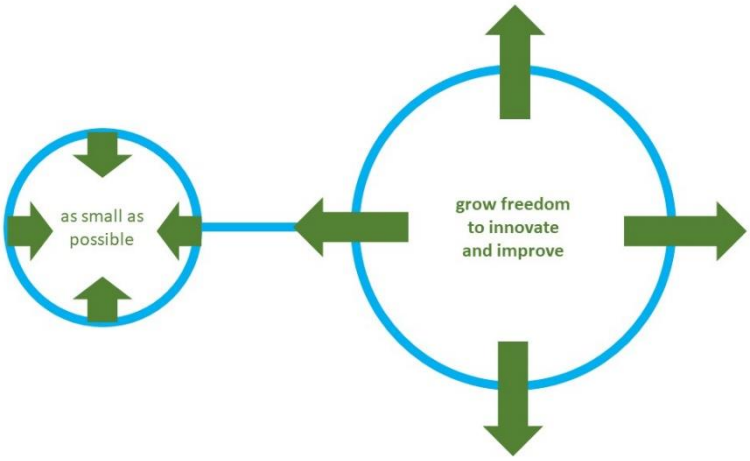
[24] https://www.gov.uk/government/publications/review-of-governance-and-accountability/independent-review-of-governance-and-accountability-in-the-civil-service-the-rt-hon-lord-maude-of-horsham-html

public, like an internet standards body, issuing Requests For Comments and having open public discussions about technical standards, and generally working openly. It will be a parliament of standards – a parliament in the sense of the old St Kilda parliaments[25] - all the working people assembled and participating in the allocation of standards work.

The DSRO's relationship with the departmental bodies follows the dumbbell pattern:



The challenge for government standards is the same as that for web standards:



The goal is to carefully design and restrict the core, shared standards, services and infrastructure to maximise the freedom and autonomy of the entire state apparatus to innovate.

Life in modern Germany is still structured by the social insurance reforms brought in by Bismark in 1889 during the Second Reich – these foundations

[25] https://www.ambaile.org.uk/asset/38917/1/

survived Weimar, the Third Reich and the GDR/BND cleavage into the modern united BRD.

Similarly, some of the proposals[26] of this research project are 100 year decisions – which is why the parliament of standards needs to be brought into the overall constitutional settlement with bodies placed on a statutory basis and with defined parliamentary oversight.

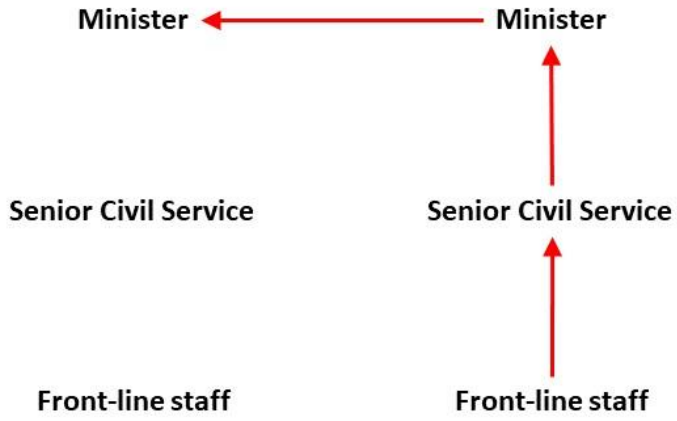
As well as issuing RFCs and standards, the DSRO needs to have a role in the oversight of open source software. Government needs new tooling and components[27]. Pushing regular working processes and outputs into tooling to automate them away is a super-power of the best of the big internet companies. A single open organisation that manages government standards and open source projects makes sense.

A critical element of this is closing the continuous improvement loop. Much is made of empowering and enabling people on the front line, the real experts. Genuine innovation rarely comes up a chain of command in the internet era. Organisations rightly have their delivery imperatives to focus on. Cross-cutting suggestions that require trading a negative short term impact against a super-positive long ones rarely survive going up against the flow of delivery imperatives raining down. A parliament of standards, a St Kilda parliament, where the various technical specialists can raise and thrash out cross-cutting ideas and express them as consensual standards is critical. And a 'parliament' that can enforce long-term behaviours – albeit with generous allowances of time to conform – is necessary. The parliament of standards needs to be backed by a genuine parliament that grants it enough muscle.

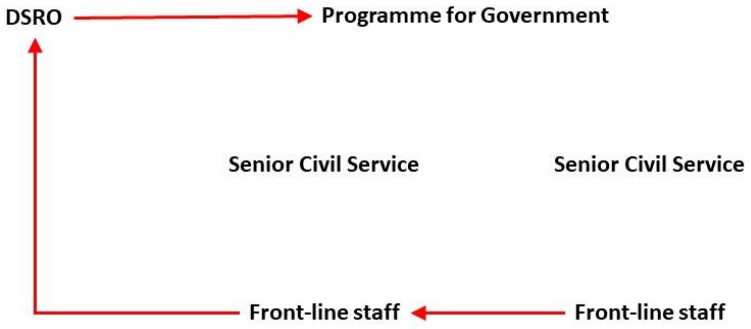
[26] See Working Paper 4 – *The remixable state* for the most obvious example, but other working papers contain 100 year recommendations too

[27] See Working Paper 1 *Data and the rule of law* for an example – the replacement of CRUD ORMs with ones that support ledgers.

The current standardisation and innovation loop looks like this:



The future loop flows the other way round:



If we are to get serious long-term strategic technical work coming from practical experience and front-line work, we need this flow.

The DSRO should co-opt external technical experts from anywhere in the world under its own recognisance. It needs a strong and defined charter with explicit measures to prevent corporate and vendor capture – an all too prevalent problem in the internet standards community. Anatole France once said *the law, in its majestic equality, forbids rich and poor alike to sleep under bridges, to beg in the streets, and to steal their bread*. A standards regime of majestic equality that lets all participate provided they live in Edinburgh and devote 5 self-funded working days a week to it will not cut it.

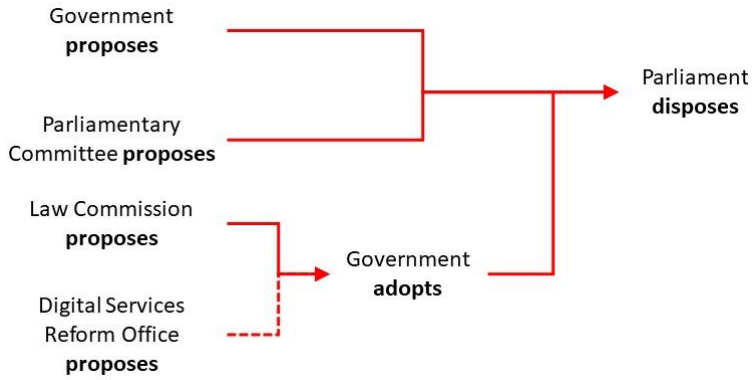
There are also clear points where the DSRO and the Scottish Law Commission will not rhyme. The Commission is an non-departmental public body and its members are subject to a public appointments process, the DSRO officials are civil servants in post. The Commission is mostly backwards looking, tidying up the garden of law. The DSRO will initially be backward looking putting in place clarity and the necessary infrastructure to do genuine transformative things. Thereafter it will gradually switch into a more future facing orientation. It is expected that the first phase will result in a higher proportion of changes to primary legislation than the latter which will tend to shift to secondary and tertiary legislation.

The Scottish Law Commission proposes programmes of work that tend to the *vertical* – an area of law at a time. The DSRO proposals will tend to the *horizontal* – all use of address data, all APIs, etc.

The Scottish Law Commission stands at arm's length from the government – the DSRO is much closer – one of its roles will be to work with bill teams and the programme for government team in shaping proposals pre-legislatively and helping shift the location of the centre of design (in the widest sense) from post-parliament to pre.

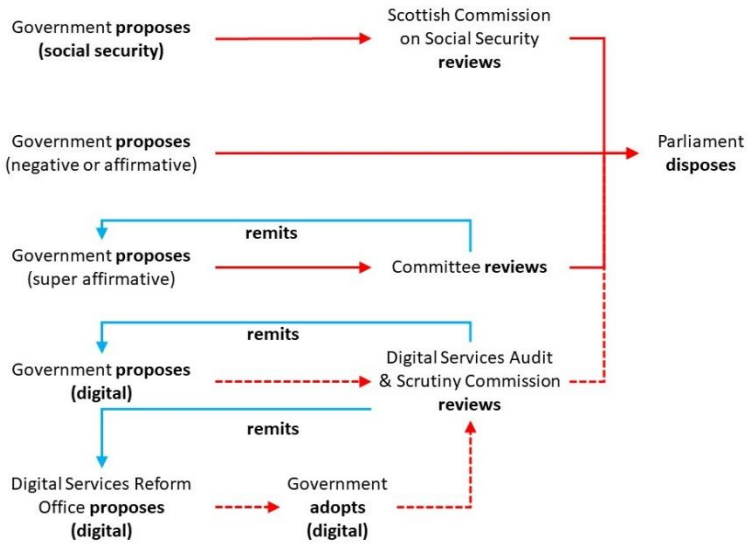
In addition, the remit of the Scottish Law Commission is very broad, covering all aspects of the law, whereas the remit of the Digital Services Reform Office is very narrow – concerned only with bills, or sections/clauses of bills, that touch on digital systems in state administration.

The new legislative routes for primary legislation (Bills into Acts) are shown here (dotted):



Budget Bills are reserved to the government to initiate. Until the format and shape of digital legislation is understood in greater detail it will not be possible to determine if this precedent should be applied to digital bills.

The secondary legislation (Ministerial Order) routes are:



It all seems clear. Set up the bodies, job done. Unfortunately, this is far from the case. The institutional work outlined in this paper is embedded in a wider programme of recommendations (33 at time of writing, but subject to extension, breaking out and consolidation as the research and review process comes to an end).

The territory is not the map, and these proposals are just a map. The work of actually exploring the territory and confirming (or not) that these proposals have the intended effect has to be done.

How to do that is the subject of the following sections.

On quality

Before laying out a roadmap it is perhaps time to take stock and consider quality. Artefacts (documents, reports, etc) do not have intrinsic quality – they get their quality from the process that produced them (who was consulted and how, how the conclusions were generated, how those conclusions were validated, etc, etc).

The BIus research followed a seagull process. A scope was defined – from manifesto and thinktank, through the programme for government, bills and bill packs, parliamentary process, design, testing, delivery and in-service. A wide range of participants have been interviewed and their interviews reviewed in context of the extant literature. The interviewee selection aimed to be a complete hand-to-hand cycle in Scotland and interesting people with things to say elsewhere. The range of skills and competencies involved means that no single person can be an expert in the complete end-to-end process.

In summary, fly in, eat your chips, shit everywhere and fly out again.

Seagull processes cannot produce outputs that are 'correct'. The implication being that they are 'wrong'. The question is in what kind and degree. Some of the recommendations of my work will simply be wrong. Some will be wrong by emphasis *we should do a, b and c* when it turns out *a is massive, b is middling and c is tiny-tiny*. Some will be wrong by interaction *when you do this to achieve this good thing, this bad thing also happens*.

If this seems very gung-ho to you, then you need to understand I am a software developer to trade. Thanks to Panko's ground-breaking work[28] we know how bad software developers are at writing software. In code reviews, co-workers will find a defect in about every 10 lines. Filming software developers writing code (as I did to myself during lockdown) shows that even with over 40 years' experience a software developer is incapable of writing more that 3 lines of code from a clean start with making an error that needs to be corrected.

The art of software development – of all the digital trades - is in the correction or errors, not their avoidance. Plan to fix. This is a map, with defects. The correct approach here is for practitioners to explore the territory and correct the map's errors.

Roadmap

So how to do that? - with an experimental process that systematically tests elements of these proposals, by:

- building shadow organisations inside the parliament and government
- choosing an uncontentious but appropriate sample bill from the programme for government
- using existing standing order powers to create an experimental legislative path confined to that chosen single bill.
- executing the experimental process
- rinse-repeat until there is consensus on a final state between the Minister for Parliament and the committee members

Having agreed a final state the shadow bodies can:

- instruct parliamentary counsel to prepare the final statutory form for the various components which the Committee can propose to parliament
- define the final state Standing Orders for the Corporate Body to adopt – this includes the standing committees that will need to supervise The Digital Services Audit & Scrutiny Commission and be on point for digital

[28] https://panko.com/ssr/index.html

bills

- propose modifications to the 'proper form' for Bill Packs being introduced to parliament to the Corporate Body

The motto of the BIus project is *Explicite*, *Constitutionnalité* and *Simplicité*. In the spirit of that the development of the new ways of working should be as conservative and non-disruptive as possible, using existing mechanisms and not introducing new ones.

The shadow work can be organised by using existing powers.

Shadow Digital Audit & Scrutiny Commission

The work here touches upon the work of the Public Audit committee as defined in Standing Order 6.7[29], and the Delegated Powers and Law Reform committee defined in Standing Order 6.11[30].

The parliamentary bureau should consider whether to establish a wholly new committee, a joint committee under Standing Order 6.14[31] or a joint subcommittee of one of the standing committees (Standards, Procedures and Public Appointments, Public Audit and Delegated Powers and Law Reform).

This new committee can then appoint advisors who are technical experts, social scientists, ethicists and legal experts under Standing Order 12.7[32]. It is then equipped to play the role of The Digital Services Audit & Scrutiny Commission.

Shadow Digital Services Reform Office

The civil service can identify staff members who have the skills and experience to be appointed to the roles of Government Digital Officer, Government Technical Officer and Government Design Officer and then assign them to the project – and use the existing provisions of the First Minister's Digital Fellowship[33] to co-opt external experts to an advisory board.

Experimental legislative path

Standing Order 17.1a[34] allows the Scottish Parliament to create a temporary amendment to standing orders based on a motion introduced by the Standards,

---

[29] https://www.parliament.scot/about/how-parliament-works/parliament-rules-and-guidance/standing-orders/chapter-6-committees#topOfNav

[30] ibid.

[31] ibid.

[32] https://www.parliament.scot/about/how-parliament-works/parliament-rules-and-guidance/standing-orders/chapter-12-committee-procedures#topOfNav

[33] https://digitalsupporthub.service.gov.scot/s/article/first-ministers-digital-fellowship-programme

[34] https://www.parliament.scot/about/how-parliament-works/parliament-rules-and-guidance/standing-orders/chapter-17-miscellaneous#topOfNav

Procedures & Public Appointments Committee - see for example Temporary Rule 5 Proxy Voting Pilot[35].

Rather than expand on the experimental legislative paths here – that work will be explored in a further discrete working paper – No 7 *Experimental digital legislative processes*.

In conclusion

This paper proposes a structured mechanism for changing the capabilities of the Scottish state in relation to digital. It takes a precedential, prudential, iterative, consensual and constitutional approach to building out the institutional structures and processes required. The goal is long-term strategic institutional change. This mechanism will not work without a host of other activities not discussed here. They can be summarised as giving as much power and autonomy to the delivery and in-service engines as they require to do the job.

Appendix 1 – the Web 0.9 specification

Connection

The client makes a TCP-IP connection to the host using the domain name[36] or IP number , and the port number[37] given in the address.

If the port number is not specified, 80 is always assumed for HTTP.

The server accepts the connection.

Note: HTTP currently runs over TCP, but could run over any connection-oriented service. The interpretation of the protocol below in the case of a sequenced packet service (such as DECnet(TM) or ISO TP4) is that that the request should be one TPDU, but the response may be many.

Request

The client sends a document request consisting of a line of ASCII characters terminated by a CR LF (carriage return, line feed) pair. A well-behaved server will not require the carriage return character.

This request consists of the word "GET", a space, the document address[38], omitting the "http:, host and port parts when they are the coordinates just used

[35] https://www.parliament.scot/about/how-parliament-works/parliament-rules-and-guidance/standing-orders/annexe-temporary-rules#topOfNav

[36] https://www.w3.org/Addressing/BNF.html#5

[37] https://www.w3.org/Addressing/BNF.html#7

[38] https://www.w3.org/Addressing/BNF.html#1

to make the connection. (If a gateway is being used, then a full document address may be given specifying a different naming scheme).

The document address will consist of a single word (ie no spaces). If any further words are found on the request line, they MUST either be ignored, or else treated according to the full HTTP spec.

The search functionality of the protocol lies in the ability of the addressing syntax to describe a search on a named index[39].

A search should only be requested by a client when the index document itself has been described as an index using the ISINDEX tag[40].

Response

The response to a simple GET request is a message in hypertext mark-up language (HTML[41]). This is a byte stream of ASCII characters.

Lines shall be delimited by an optional carriage return followed by a mandatory line feed character. The client should not assume that the carriage return will be present. Lines may be of any length. Well-behaved servers should retrict line length to 80 characters excluding the CR LF pair.

The format of the message is HTML - that is, a trimmed SGML document. Note that this format allows for menus and hit lists to be returned as hypertext. It also allows for plain ASCII text to be returned following the PLAINTEXT tag .

The message is terminated by the closing of the connection by the server.

Well-behaved clients will read the entire document as fast as possible. The client shall not wait for user action (output paging for example) before reading the whole of the document. The server may impose a timeout of the order of 15 seconds on inactivity.

Error responses are supplied in human readable text in HTML syntax. There is no way to distinguish an error response from a satisfactory response except for the content of the text.

Disconnection

The TCP-IP connection is broken by the server when the whole document has been transferred.

The client may abort the transfer by breaking the connection before this, in which case the server shall not record any error condition.

Requests are idempotent[42]. The server need not store any information about

---

[39] https://www.w3.org/Addressing/Search.html

[40] https://www.w3.org/MarkUp/Tags.html#18

[41] https://www.w3.org/MarkUp/

[42] https://www.w3.org/Protocols/HTTP.html#13

the request after disconnection.

Working Paper 1 – Data and the rule of law

Version 1.3

Using ledger to conform to the rule of law

Introduction

Data, ledgers and the rule of law

This document looks at the different styles in which one can design a database. The rule of law requires that the state be able to adequately describe its decisions (a synchronic perspective – where we are at a point in time) and how the decisions came to be made (a diachronic perspective – how we got here).

Systems with all styles of database design can be made to conform to the rule of law. But not all database design styles map well to it.

Using a ledger style of database design will make systems development easier – and make conformance to rule of law a property of the system and not a feature that must be built.

Who are you?

You are a programme manager, service designer, data or technical specialist in the public sector.

You wish your team to deliver software that conforms to the rule of law and also save money on doing it.

Why should you read this?

You should read this to understand the issue, how pervasive it is, and how to address it in your team or organisation.

Revision Notes

Version 1.1 has additional content arising from a conversation with Stuart Roebuck of the Scottish Government regarding both the composability of ledgers and their ability to have per record permissioning and access control.

Version 1.2 has an additional section on additional benefits added which discusses evidential ledger-based data and its role in simplifying system decommissioning.

Executive Summary

The Child Poverty Action Group report You Reap What You Code: Universal Credit, Digitalisation And The Rule Of Law[43] makes a convincing case that there are structural issues in the design of Universal Credit that impact on the systems ability to conform the rule of law.

This working paper explores the possibility that the design failures lie not in the software layer, but the data schema.

If this is the case, the bad news is that these design failures are structural, not accidental, and almost certainly occur across many state computer systems.

But the good news is that there is already a well known solution to this problem - use of ledgers.

Such a solution was identified by one of the claimant advisors who was interviewed:

Amelia But those were your records. It's not their's to do that.. it's like changing a bank statement. You don't go and change bank statements.

Ledgers are certainly implemented in *some* government computer systems – an indeed are imperfectly implemented in Universal Credit itself – in the Journal.

Their use is easily learned, and simple, comprehensive training could eliminate most of the effects (over time – systems will need to be changed).

In addition ledger entries keyed off the same master index key are trivially aggregated (subject to some technical caveats) – and critically the aggregated data can retain the sharing consent of the original dataset – enabling single repositories to serve a wide range of users with different permissions to inspect the data – each of whom is served with an appropriately filter view.

The Evidence

The CPAG Report

The report is an important read – but it is worth quoting the first paragraph of the conclusions here:

Our research found that in UC, the reasons decisions are taken not in accordance with the law include digital design and implementation choices systematically producing the wrong decisions for claimants in certain situations; the digital architecture not accurately reflecting the legislative decision-making framework; and certain digital design or implementation

[43] https://cpag.org.uk/policy-and-campaigns/report/you-reap-what-you-code

choices contributing to repeated errors in human decision making.

It is inappropriate to summarise a 201 page report in a working paper – but it was an extensive piece of work with semi-structured interviews of claimants and advisors and examination of case studies from groups set up to to monitor issues in the administration of the social security system.

The report details 160 individual user stories that describe issues in the encoding of the rule of law in the UC systems. Sometimes for a particular issue there might be 1, 2 or 3 individuals stories attesting to it. This working paper focusses on analysing those user stories.

The overarching research conclusions have this subhead:

Rule of law principles have been undermined by the design and implementation of universal credit, but this is not an inevitability of digitalisation

and makes the following particular observations:

Simple design choices when implementing a digital-by-design benefit can significantly affect the extent to which a system complies with rule of law principles, and the extent to which it can result in negative consequences for claimants.

The DWP appears to prioritise simplicity over legality, which is not a choice available to it if the system is to comply with the rule of law.

At CPAG we observe the same mistakes in decision making occurring again and again in relation to individual claims and awards, and despite investigations using freedom of information (FOI) requests and other methods, it is very difficult to find out whether these errors are solely caused by human error, due to a programming error, or due to a digital design feature which encourages DWP officials to repeatedly make the same mistakes.

The rule of law

Lord Bingham was the first President of the UK Supreme Court Lord. The CPAG report's framework (which this working paper accepts) is based on Bingham's eight principles of the rule of law[44] which the CPAG researchers grouped into 3 categories:

Bingham's eight principles of the rule of law

1. The law must be accessible and so far as possible intelligible, clear and predictable.
2. Questions of legal right and liability should ordinarily be resolved by application of the law and not the exercise of discretion.

[44] T Bingham, *The Rule Of Law*, Allen Lane, 2010

3. The laws of the land should apply equally to all, save to the extent that objective differences justify differentiation.
4. Ministers and public officers at all levels must exercise the powers conferred on them in good faith, fairly, for the purpose for which the powers were conferred, without exceeding the limits of such powers and not unreasonably.
5. The law must afford adequate protection of fundamental human rights.
6. Means must be provided for resolving, without prohibitive cost or inordinate delay, bona fide civil disputes which the parties themselves are unable to resolve.
7. The adjudicative procedures provided by the state should be fair.
8. The rule of law requires compliance by the state with its obligations in international law as in national law.

As it relates to social security they aggregate these down to three core principles:

- Transparency (Bingham 1)
- Procedural fairness (Bingham (4, 6, 7)
- Lawfulness (Bingham 3, 4, 5)

Two styles of data schema

There are roughly two schools of data schema operations.

The traditional commercial model is based around CRUD operations (*Create*, *Read*, *Update*, *Delete*). With CRUD changes to data are made in-place – the old value is overwritten with the new one. Oftentimes the record is decorated with a couple of datetimes – a creation one and a last-updated companion.

This way of working is baked into standard development frameworks.

Data Handling Library	Web Framework	Development Language
Ecto	Phoenix	Elixir
ActiveRecords	Ruby On Rails	Ruby
Eloquent	Laravel	PHP
Hibernate	Spring	Java
ORM	Django	Python
Gorm	Grails	Groovy

It's a habit that goes back to the olden days when memory was super-expensive, as was disk space – and disk access times were slow – or systems

even used tape storage.

By contrast banks and other financial institutions use ledger-based systems where the only data operations used are *Create* and *Read* and where data is immutable.

Essentially when data is 'changed' a new record is written that supersedes the old one.

We can write down how these two styles differ. Consider a record of a persons marital state.

In CRUD each time her status is updated it overwrites the previous status:

Annie Beacon, born, 03/05/1978
Annie Beacon, married, 13/04/1999
Annie Beacon, divorced, 01/02/2003
Annie Beacon, married 11/11/2004
Annie Beacon, widowed 04/05/2016

So when you query the database you can only get the current status and when it happened.

By contrast in a ledger each record is maintained with a date:

Annie Beacon, born, 03/05/1978, 01/01/1998
Annie Beacon, married, 12/04/1999, 27/04/1999
Annie Beacon, reverted, 12/04/1999, 28/04/1999
Annie Beacon, married, 13/04/1999, 28/04/1999
Annie Beacon, divorced, 01/02/2003, 14/02/2003
Annie Beacon, married 11/11/2004, 18/11/2004
Annie Beacon, widowed 04/05/2016, 09/05/2016

In the ledger the date the data was recorded (in the first line 01/01/1998 is different from the date the event happened – here the new system wasn't set up until 1998 and the birthday is retrospective. Note also that the 2 line has a date entered in error, reverted in the 3rd line and correctly entered in the 4th line. (It is possible to have ledgers that have more than 2 dates, indeed it is common.)

Ledgers are designed to hold a synchronic and diachronic view of data:

- Synchronic
    - what is the position now (or at any point in time) - *what you ask the judge to rule on*
- Diachronic
    - how did we get to the decision - *the evidence you present in court*

The key to ledgers is double dating systems – a date-on-ledger field and an effective date field, which allow events to time travel.

She told me today that she got a job last week – the date-on-ledger date is

today, the effective date is last week.

This means that two different time lines can be trivially created for evidential purposes:

- how the citizen interacted with the officers of the state
    - when they called the office
    - when they updated their journal
    - when their letter arrived
- the sequence of events in the real world
    - when they got made redundant
    - when they sold their house
    - when their kid went away to university

The use of ubiquitous double-dating means that data elements from different sub-systems and database tables can trivially be combined into a single timeline:

- customer comms
- back office decision making
- actual electronic payment logs

Time travel is also used to correct errors. If my salary was put in as *£100* last week and corrected to *£1,000* this week, today's entry would time-travel back to when the error was made and correct it then.

Time travelling can also be used to pre-register activities *I will be starting my job next month on the 6th*.

Typically ledger entries are decorated with other data that is critical in adherence to the rule of law:

- the name of the person who supplied the data
    - the citizen
    - an employee of the state
    - a default value baked into the system
    - etc, etc
- if it records a decision then a code indicating which law or regulation was used to make the ruling

As the name suggests – ledger-based data structures are based on paper ledgers and traditional paper administration required immutability – things were logged. When my Dad went to Trinidad to work by oil tanker, and my Mum to marry him, they both signed on as Supernumerary passengers in the ships log as required by the Board of Trade when they embarked in the UK, and both signed-off on arrival in Port of Spain.

All those ink stamps that put RECEIVED 08/08/2023 on letters and

ACTIONED 09/08/2023 on when it had been dealt with are part of a historic administrative legacy of data immutability and rule of law.

Ironically having embraced mutable data, all software systems at scale now back their operational data systems with data warehouses based on logs - and logs are immutable data with strict time stamps, so the wheel is being reimplemented but partially and badly.

Reading the CPAG report with a data structure eye

By carefully reading the report and considering the details of particular user stories highlighted an approximate sense of the scale of the problem can be sketched out.

Section	Total Stories	Issues resolved by ledgers	Percentage
Claims	32	12	38%
Decision Making	60	54	90%
Communicating	25	20	80%
Disputes	43	26	60%
Total	160	112	70%

It is important to take these figures with a pinch of salt.

Some of the issues raised would be totally alleviated by use of ledgered data – but in other cases ledgering is just a part of the correct solution of them. Some of the stories marked as not affected by ledgering might have had their problems designed out if the design and implementation team had a deeper sense of what the rule of law entails irrespective of the underlying data regime.

Caveat Lector! Reading this you should rightly be anxious that this is based on indirect observation and not direct review of the underlying code and data. My reading is informed by many years of experience, but that on its own is not enough.

And it should be stated clearly that there is ledgering in the UC system. The payments systems will certainly be full accountancy ledgers. The journal which the systems use for the citizens to communicate with the DWP is a form of ledger – although careful reading of the user stories indicates that it is not immutable – or that the presentation of all the information in it doesn't respect underlying immutability.

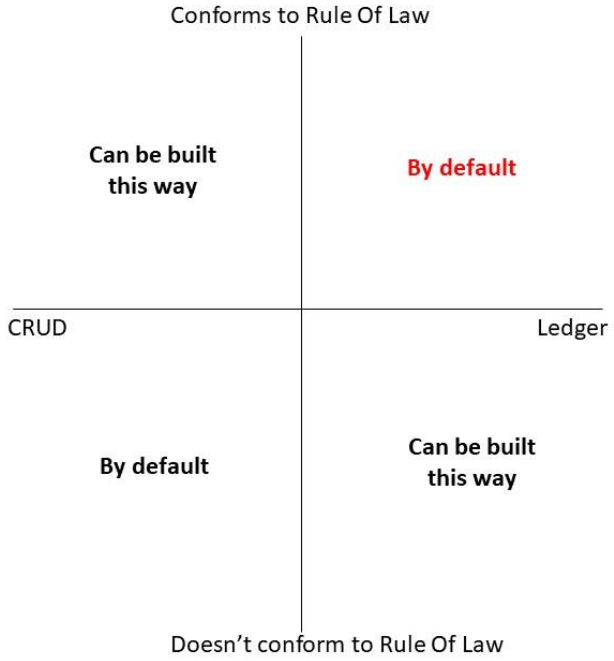
The user stories mention Subject Data Access Requests as being a key part of

escalating and resolving miscalculation issues. Based on experience I suspect the data being returned in them is a mixture of operational data from the live system and updates pulled from logging and reconciled in to a coherent data pack/storyline.

I suspect that the journal design has come from user engagement and fulfilling user needs – which would only imperfectly capture the design principles to conform to the rule of law.

Systems that use mutable data persistence layers can be designed to conform to the rule of law – to have adherence to the rule of law as a *feature* – and this is clearly visible in the UC system – its not a wild west yolo affair – *the intention of the developers was compliance with the rule of law*.

But systems built on immutable ledgers have adherence to the rule of law as an inherent *property*.



All software is imperfect and imperfections in the UC software is not a personal criticism of the people that developed it.

This analysis is a strong indicator that a deeper examination of the understanding of the rule of law as it related to the structure of state administrative data is required.

THE PLURAL OF ANECDOTE ISN'T DATA

Since embarking on this work I have asked civil servants who work in tech and data if they have been trained in the Rule of Law – and have received only 1 positive answer which was about rule of law in general, and not the data implications.

Addressing the issues

Characterisation of the problem

The problem is an accidental one. Service designers (and their private sector counterparts: product managers) live and operate in the world of their users, for whom software systems are merely surfaces. The user has no sense of the underlying plumbing. And so the designers leave technical and implementation decisions in the hands of the engineering staff. Engineers have been trained to use mutable data and libraries that enforce mutability and simply turn to the tools of the trade to do their work.

There is no gap in the law – state administration systems must already conform to the rule of law – the problem is enforcement. Failures and defects will be corrected, slowly and painfully, by litigation and appeals on a case-by-case basis and the subsequent amelioration by change control. It could be better, cheaper and more effective to not do this.

Solutions

Potentially this is problem whose solution could still be having significant impacts 100 years from now – we should not think of this as trivial in its import.

Adherence to the rule of law is a fundamental non-functional requirement for all state administrative systems that record decisions made for, about, or on behalf of citizens and incorporated bodies (in the widest sense). Simply telling software developers that, and outlining how to use ledger-based data structures to embed that in their work is the start of the solution.

Likewise service and other designers who shape the structure and interfaces of the system need to be told to ask their software developers how they intend to persist administrative data. Given that service designers are positively trying to build systems that are user friendly and accessible, the expectation must be that they will welcome making the underlying foundations of their systems being optimised for clarity of exposition about how the systems work.

There are already systems in place to do mandatory training, for sexual harassment, cyber security etc. A similar short course on the rule of law and the development of software with a particular emphasis on data structures would suffice to inject the idea into the wider state development community.

This should be backed up by written guidelines – and backed by blog posts and other learning materials being commissioned and published. Ideally a book.

But, in addition the structural problem needs addressing. If the tooling is a golden path to sin, other tooling must be built. The rule of law is shared

between governments and jurisdictions – and an open source project dedicated to delivering roughly-comparable ledger-based ORMs and new table skeletons for a range of web delivery frameworks would be the most effective way to do this.

The use of ledgers needs to be mandated for state systems that serve citizens or organisations. The ideal mechanism to do this would be via a declaration in an Interpretation Act, backed off against a gazette of data standards. A more detailed description of this will be published in Working Paper 5 – *Law reform for data* (forthcoming).

Additional Benefit – data sets filtered by permissions

If we have multiple ledgers – say a ledger of health records for a patient and a register of social work contacts and a register of police contacts – these ledgers can be zipped together to provide a holistic view of the persons interactions with the state. This zipping is subject to some caveats around causality[45].

Each ledger entry can bring the data permissions and lawful access controls of its originating source – and then individuals who wish to access a particular persons records can have only those disclosed that they have permission to see.

This is tremendously powerful.

Additional benefit – better decommissioning

If a CRUD approach is used, the status and decisions of a particular system require both the software and the data to be understood – and this means the systems needs to remain up and available – with all the associated hardware and data centre costs.

Returning to ledger based data and ensuring that data structures have evidential quality will enable evidential data to be taken off systems no longer in use and them to be decommissioned.

---

[45] In the absence of all the supplying systems being in Google 'Spanner' style data centres with atomic clock time synchronisation the zipping will be done on local machine clock time – the expectation would be that ledger entries from a particular source (police, health board, social work) would be correctly ordered, but that time drift might mean that an event in one ledger is inaccurately timestamped relative to another. Perhaps the correct and ambitious-for-Scotland response to this caveat is for Scottish Government to bite the bullet and build Google-style data centres?

Working Paper 2 – Rules as code

Version 1.0

Definition, delivery and testing

Introduction

What is Rules as code?

Rules as code is an emerging discipline that aims to bring computational discipline to rules-making, particularly as to rules that are defined in legislation.

There are many variants and approaches in this emerging discipline, but this working paper is looking in particular an approach whereby the legal text of a law is annotated in a machine compilable way. The compilation process can be used to identify internal inconsistency in different elements of the law. It can also turn the annotated text into a simple computer system which can be used in a variety of ways.

Who are you?

You are a programme manager, software developers, tester, service designer or parliamentary draftsman with an interest in the creation of digital systems that implemented laws.

Why should you read this?

This report outlines a number of approaches that might help you:

- develop new legislative proposals
- model the impact of draft legislation before it is adopted
- generate components for use in live calculatory systems
- generate test suites for sytems implementing legislation
- remove barriers to entry into regulated sectors of the economy

Executive Summary

Rules as Code is a movement looking at annotating legislation in machine readable formats that enable various technical transformations and tests to be performed on it.

The goal is to variously make law into one or more of the following:

- executable production code
- basic expositionary systems to drive shared understanding and iterative development
- systems that have had formal consistency proofs applied
- navigable information architectures that are machine traversable
- executable inputs into macro-economic statistical models
- as a data feed to be interrogated by AI and made more comprehensible to

people without legal training

- reference systems for regulated industries that can reduce their compliance costs

paper identifies two new possible outputs which are important for digital transformation:

- test first development using property-based system tests
- a shared catalogue of data sources

The significance of this is that test-first development can dramatically reduce development times and costs – savings in the 10's of percents and not single percents. See Appendix 1 for worked examples with measured costs.

Context

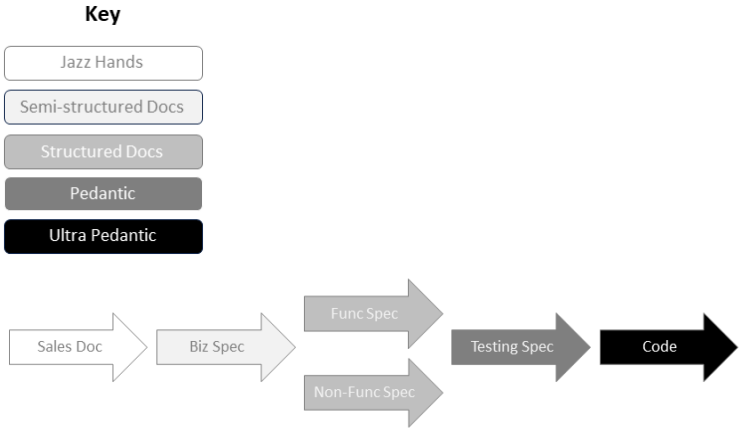
I have been talking to Denis Merigoux and his Catala team at INRIA, Bridget Hornibrook at the DWP and Adrian Kelly who has been working on LogLaw, Matthew Waddington and Flora Leather in the Bailiwick of Jersey and Pia Andrews in New South Wales

Adrian and Pia were part of the OG Rules As Code endeavour - the New Zealand Government's *Better Rules for Government Discovery* Report - which is an oldie-but-goldie that is well worth your time reading.

However, no comprehensive study has been made of legal tech by me. This working paper is observational.

Background

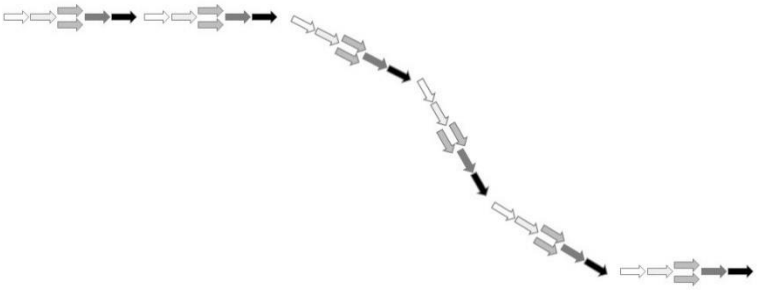
In the private sector the process of specifying a new computer system can be regarded as a process of empendantifying a series of documents, taking them from being fairly free form to very strictly structured:



Code is very pedantic – consisting as it does of 0's and 1's in a strict order – swapping any pair of them can cause systemic meltdown.

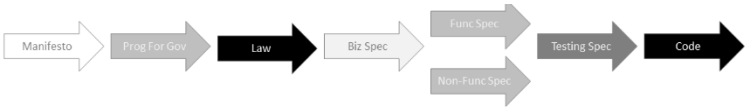
In this diagram I split out the technical specification into 2 parts – the functional specification and the non-functional one – for reasons that will become clear later.

Agilists might here be looking knowingly and saying “ah but that’s waterfall!”. All software development is waterfall – agile is just lots of them.



The quantum of work is not affected by doing it agile – the benefit comes from early course correction, eliminating rework and fix-ups and arriving at a better outcome faster and cheaper. Each agile sprint is a waterfall in its own right.

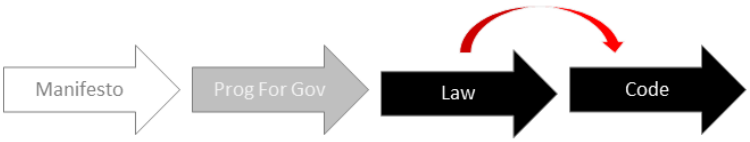
The public sector by contrast can be doubly-pendantifying:



Note: in the context of the Scottish parliament the Programme for Government (which contains a yearly Legislative Programme) is a 5 year living programme which includes all the primary legislation (Bills/Acts) passing through the parliament. Law pertaining to state systems is also defined in secondary legislation (ministerial orders).

Law, like code, is ultra-pedantic – a comma will support enough rope to hang a man.

The *fever dream* of Rules as Code (or to be fair, as it comes across to me) is to capture the core essence of the law in a machine-readable form and transform it into code, a great leap forward:



When we look at what Rules as Code people are doing with their tools – it isn’t this.

The initial work in New Zealand was a lawyer-led approach to rethinking the development of law to enable simpler and better development of regulations, entitlements, calculations.

They pioneered cross-team working with parliamentary counsel, policy

makers, service designers and delivery people working in cross-disciplinary teams.

The project also identified and struck down key barriers between policy intent and deliverability and demonstrated value and velocity by using rules as code.

Rule as Code tech can be used as way of building quick prototyping tools that:

- enable fast design feedback loops in the development of policy and legislation – having a common 'surface' that members of different professions can engage with is an excellent tool for collapsing getting-on-the-same-page discussions and associated costs
- provide plug-in entitlement and calculation elements for financial modelling covering take-up and impact of benefits, monte-carlo exploration of tapers and better design of hardship/compensation schemes, tax base modelling etc, etc
- are a very useful quick'n'dirty first pass usability tool that can be used with co-design communities

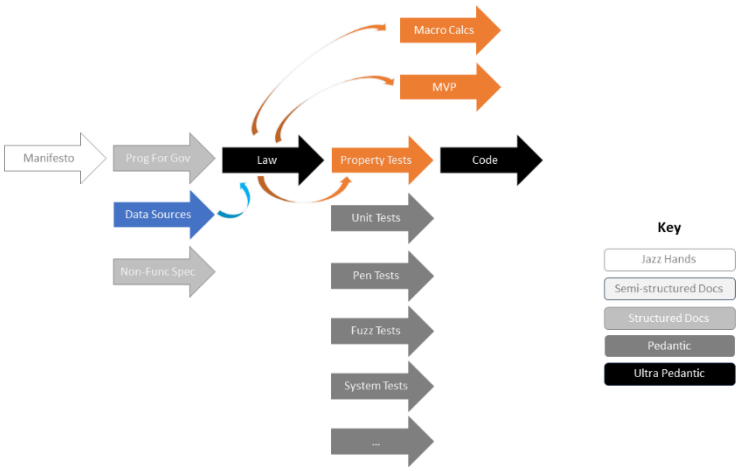
But the leap to write-once/deploy is never going to happen. And its obvious why looking at the flow of pendantification. The law only covers the *functional spec*. In the case of social security that would be who gets what money in what circumstances.

It doesn't cover the *non-functional spec* – things like: you need to log in, the data needs to go into a database, there has to be rules-based-access-control, and dashboards, and cloud deployment and it must work in browsers and disburse actual payments to actual bank accounts.

I think a more realistic approach should be called *Rules As Tests*. In this world the machine-consumable annotated law consumes existing data sources (and their legal and regulatory definitions). It then possibly generates three testing outputs:

- macro-economic calculations – code that can be deployed without the normal non-functional requirements because it is used in economic modelling systems where individuals are treated statistically and not as individuals (economic testing)
- an MVP (single user, basic GUI) which can be used to iteratively seeking consensus during the development of the law – saving time and effort in developing regulations whilst increasing quality and effectiveness is a worthy goal in its own right
- as a property test generator for the final live production system

Lets look at these 4 products in context:



The use of these technologies to make MVPs is proven by the work of Bridget Hornibrook and Adrian Kelly at the DWP and elsewhere. Denis Merigoux and his team have demonstrated its possibilities in Macro Calculations.

Consequently the rest of this paper will focus on two things:

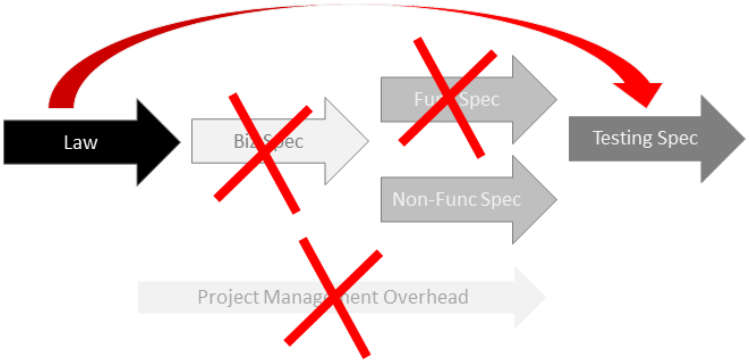
- Test First Development
- the consumption of Data Sources
    - (this will include the pull-back of the Non Functional Specs in the diagram – not strictly related technically, but organically related organisationally)

Test First development brings tremendous gains – the fact that the test type is Property Tests is just an added bonus.

Future State

Test First Development

Test First Development is a common technique to improve software delivery and reduce rework (and hence costs). However in this instance by generating tests it adds additional value by eliminating work:



With the right tooling and appropriate care at the design-of-the-law level the entire business and functional specification steps can be eliminated. Well designed, comprehensive and appropriately annotated system tests can perform the majority of this work.

Project management in major IT deliveries imposes huge reporting demands on software developers and other professionals to assemble a picture of progress that is comprehensible by non-technical managers. Much of this can be expressed in terms of *simple tests passing Vs tests not passing – if and only if* the test suite is known to be functionally complete and *can run and fail for all tests – including those that test as-yet-unwritten features*.

See Appendix 1 for a measure of the size of the cost savings – but be cognisant that previously measured costs in different circumstances can't be simply read across.

For systems like social security and taxation there are additional benefits. It is possible to generate property tests. A property test takes a set of inputs and states that the output must have these properties. In the case of a social security system you could generate a test along the lines of:

Caroline is 42, has two children aged 11 and 7, one with special needs. Her husband earns £2,400 a month and she is entitled to X in benefits.

If the end system produces the same value of X – good to go.

But these tests are generatable – there is also a *Caroline with 3 kids, and an income of £1,200* and so on and so on. And Caroline can give birth, and her middle daughter can be paralysed suddenly in a car crash, and her man can get cancer and rolling on and over all the edge cases endlessly.

Generative property-based tests are unbounded in number. Most software testing problems can be summarised as "not enough tests" – test first property generative tests have the opposite problem – "too many to run".

A huge proportion of the cost of major software programmes consists of a number of things:

- driving agreement and understanding amongst all stakeholders that they are talking about the same thing
- building a model of progress towards a goal that can be used in communication with stakeholders
- testing that the developed software actually does what it is supposed to go – and delivering confidence to stakeholders and team members that it is reliable

Rules As Code/Tests can significantly reduce costs by addressing each of these areas – by generating MVPs that professionals from different disciplines

can share, by generating completion figures in the form of tests passing/not-passing and by actually writing huge and flexible test suites.

(Readers should *beware* of going over the top – the annotations to law that Rules As Code uses are not in themselves justiciable – the fact that a system passes the system tests does not, in itself, mean that the system is legal or complies to the rule of law. As long ago as 1970 Djikstra told us *Program testing can be used to show the presence of bugs, but never to show their absence!* It follows that tests suites generated by Rules As Code can only tell us when the system under test violates the law, not that it conforms to it.)

Consumption of data sources (and common non-functional specification)

Moving towards using code annotation in law opens up two other potential savings. The functional specifications describe what the software does – what makes it a social security system as opposed to a tax system. They are different for each system.

By contrast the non-functional specifications usually are similar (if not identical) for different systems. It is perfectly possible for a tax and a social security system to have identical log-in mechanisms, reuse the same payment rails, work in the same browers, use the same underlying database technologies.

So simply splitting functional and non-functional requirements enables the partial Lego/IKEAisation of systems. The expectation that systems run by government have common and standardised non-functional requirements eliminates rework in its own right – this is another world tho and won't be discussed here – but in Working Paper 3 – *The Lego state*.

Law and any Rules As Code language typically define entities in an abstract sense 'a person', 'a taxpayer' or 'a child under 16'. What we want in government is not the abstract person – but the reified one – 'a person with an identity on the government identity service' or 'a person who has a medical certificate issued by a recognised national health system'.

If the definitions in different parliamentary acts can be harmonised and systematised, then the implementations of that data can be merged. The move to services that encapsulate and expose single sources of data under an appropriate API will become possible.

How this process would be embedded in a language like Catala needs to be determined. It might be a simple case of using include/header files – so common definitions are stored in their own legislation which is annotated in the usual way – but lacks process or rules and contains only entity definitions.

These definition can be included in other legislation and their Catala entity relations imported. There is then (outside the language/law definition) a presumption that these entities are implemented as services somewhere and offered as APIs.

These two problems merge because some of the core barriers to merging data sets don't lie in the functional or calculation aspects of a particular statute but in regulations that are formally non-functional – so access controls (who can see what data) or data retention policies (how long the data must be kept for) and a myriad of other seemingly insignificant things that act as a barrier to consolidation.

Data consolidation is a key activities for two reasons:

- it brings simplicity to the user – if only one system holds your address and surname then changing them when you get married is much simpler
- it reduces cost – every instance of data has to be maintained

Barriers to uptake

Testing as a discipline is fairly low profile in Scottish and UK governments. GDS doesn't have explicit testing standards much beyond "test things" in the Service Manual[46] and there isn't a service community for quality[47]. So jumping from here to Property Based Testing is a not-insignificant leap and would require a training/education programme. (This is on top of the migration of policy and legislation people from the old world to the new.)

For the Scottish Government with its stated policy aim of independence, building these capabilities now in order to support the creation of new national institutions (central bank, main tax office, etc) would seem to be a sensible option.

But there are also outstanding technical issues – I have explored them in the context of Catala in Technical Appendix 2 – Outstanding technical issues.

In summary there are both skills and technical barriers to moving in this direction.

Further work

And there is a second element of the modern state that Rules As Code can help in. We substantially live in a regulated world. *Rules As Tests* could also potentially be used to publish compliance test suites for regulated organisations

[46] https://www.gov.uk/service-manual/technology/quality-assurance-testing-your-service-regularly

[47] https://www.gov.uk/service-manual/communities

to use. It has been reported to me that Angus Moir at the Bank Of England is exploring this use.

In addition Pia Andrews reports[48] that publishing a reference example of banking regulations using Rules As Code saved a single regulated bank $16m a year (not sure if that's $US or $AUS there...).

As long as the strategic goal of the Scottish Government remains independence then there is a strategic requirement to have the capability to build a new tax system, central bank, full social security and pension system, etc, etc. Rules as Code and Rules as Test both would simplify, reduce the cost and delivery time of transitioning to this new world.

Technical Appendix 1 – Test First Development Worked Examples

Test Case 1 – Hypernumbers

Hypernumbers was a startup that aimed to build a web-native spreadsheet (Google Sheets is an open source desktop spreadsheet under the covers). Every cell, every page, every range would have its own URL and these URLs would be composable in functions (making a functional programme of the web).

To that end a goal of Excel 95 compatibility was set and a test framework was developed that could convert Excel spreadsheets into system tests.

A function would be inserted into a cell and Excel would resolve that function (and any dependency tree it was involved in) in the usual manner returning a value.

A programme was written to traverse all the test spreadsheets in a directory and then make of every populated cell a systems test.

The tests were hand written – but that was simply a large set of spreadsheets. After about 2 weeks there were 100,000 of them. At the time of the first test run the results were *6 tests passing, 99,994 tests failing*. (The 6 passing tests had the formulas 1, 1.0, 1.1e+1, -1, -1.0, -1.1e+1).

During the development some thousands of unit tests were written alongside the hundred thousand of end-to-end system tests.

The cash-equivalent costs (what we would have spent if we had paid market salaries and had an office, etc, etc) were between £1.25 and £1.5m – and the COCOMO II Estimate (based on the Open Office spreadsheet source code) was £8m to £20m. The work was done with a team of 4 engineers. These figures

[48] https://docs.google.com/presentation/d/1uUYTlmsj05KjUrto2U1u1RK08WeBorKIEbm4K3MRMTU/edit#slide=id.p19

need to be treated with appropriate caution. Reading across savings on this scale to the public sector would be a mistake – but the possibility of very significant costs savings is very real.

This is an example of naïve property-based testing, using the identity property – the same as Rules as Test would generate.

Test Case 2 – bet365

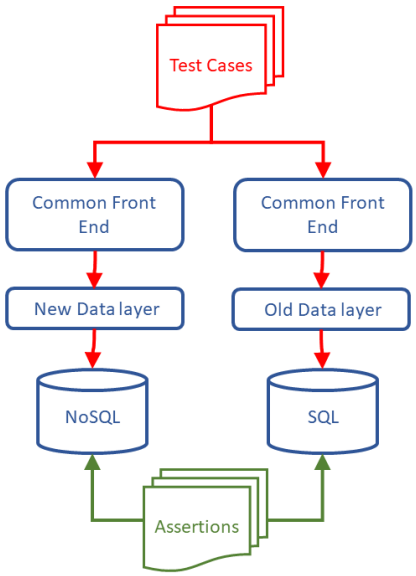
bet365 is the largest and most successful internet company in the UK. bet365 had its growth blocked at about £30bn turnover because its data layer (based on Microsoft SQL) just couldn't scale at peak. Betting is a very bursty business with lowish daily traffic, weekend peaks and Himalayan traffic at the Grand National, the World Cup etc.

We put a slip into production and logged the Germany-Brazil semi-final in the World Cup and captured over 8 million discrete events.

The logs were processed to anonymise them before copying them down from the production zone in the data centre. They had to be post-processed to create setup activities (create all the users, create all the markets in all the fixtures that they betted on, price those markets, etc, etc).

These events were then replayed side-by-side into one version of the code with the existing data layer and one with the new data layer.

The test case was run twice and the results from the two systems were compared.



If the two systems returned the same results for all posts then the tests themselves passed. As belt-and-braces a job that ran over the post test databases was created. It asserted that the persisted data was consistent (same number of users, same number of bets, total wagered, total won and totals lost

identical in both cases, and so on and so forth).

I do not have working cost estimates for this work, but the live/no-down-time replacement of the data layer was a small team of less than 10 working for less than a year. At the time bet365 was turning over $600m a week (it is now $2bn a week) – so a zero down time/failure free swap out really mattered. The property being used for property testing here is also the naïve identity property – the same as Rules as Test would generate.

Technical Appendix 2 – Outstanding technical issues

Apart from the training issue – there are three outstanding technical issues that Catala would need to solve to be fully useful for this approach:

- generating the tests
- making the tests anti-fragile
- decoupling the system under test from Catala

I am using Catala as the example because it's the one I understand best, and the one with a fully articulated parse chain[49].

Issue: generating tests

Testing is substantially about 2 things:

- applying a defined payload to a point of application
- matching the response to an expectation

Its clear that Catala has enough information to generate the data payload and it is already capable of generating code to calculate the expected response – the problem is the point of application – and this in itself brings fragility issues.

Both the examples in Technical Appendix 1 – Test First Development Worked Examples have one thing in common – the test suite and the system under test share a routing table – *by design*.

In the online spreadsheet we can express a route to a particular formula/value pair in the Excel spreadsheet that contain the test definitions as a generalised path (which by definition is directed and acyclic):

directory -> file name -> sheet -> cell addressed as row/column

In writing the tests this can be transformed into a URL which has the self-same properties:

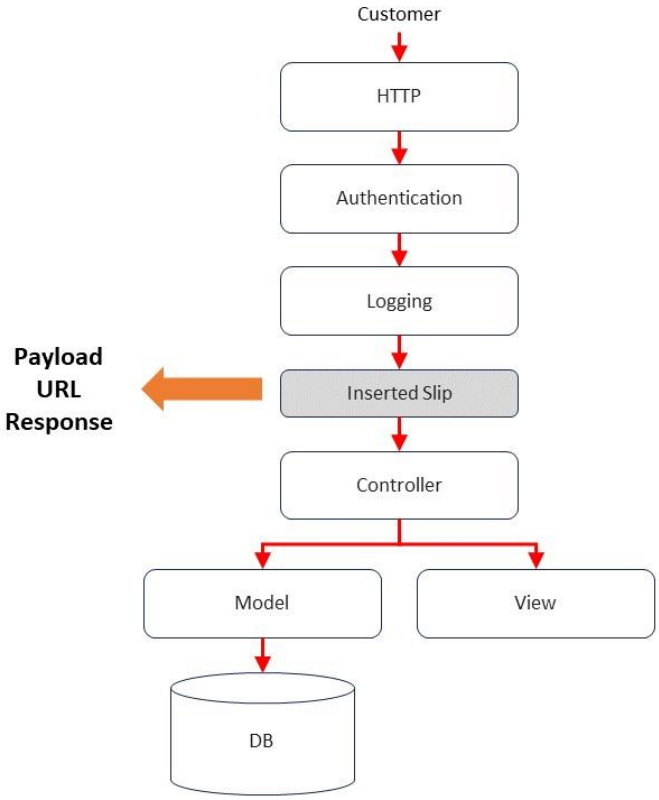
http://testsystem.local/directory/file name/sheet name/cell address

It is demonstrably trivial to apply the payload at the point of application and get back a result with can be compared. The entire test suite can run and where the formula isn't parsable (as =1+1 wasn't on day 1) the system under

[49] https://catala-lang.org/ocaml_docs/catala/index.html

test returns an error value and the assertion, and hence test, fails.

In the case of bet365 the same capability arose from where we recorded the test case:



Because the test was a side-by-side test with the same front end and only the DB layer (ie the model and below) changed out, we were able to capture the payload and the URL (the point of application) as well as the result. This meant tests could be generated.

The problem with Rules as Test is not that a routing table *cannot* be constructed from the law – the sort of law under consideration – propositional logic – can be expressed as a directed acyclic graph (Catala will throw a consistency error if that is not the case) and thus the application of data items and the retrieval of calculated values *could* be done via a URL structure which *could* be generated. The problem is that that URL structure *is unlikely to have the appropriate affordances* for a user-friendly system – very unlikely.

A second problem is that there are actually two sorts of tests that we can generate:

- simple property tests
- hysteresis property tests

A simple property test scenario is *Caroline has a child* and applies for benefit.

By contrast a hysteresis one is Caroline is childless and applies for a benefit, then she has a child, then the child gets a terminal disease, then the child dies.

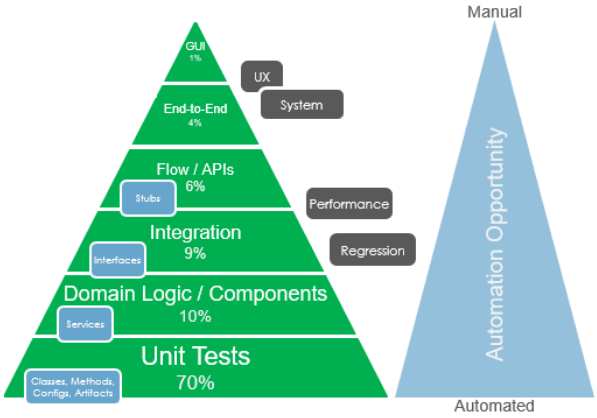
These two tests capture different aspects of systems behaviour and ideally we want both of them – writing tests for both of them automatically in an anti-fragile manner will be a challenge.

It would be trivial to have Catala generate a skeleton set of tests and a complete set of generators and then let software developers assemble a test suite as they went along and implemented features.

The problem with this approach is that the day 1 progress report would not be *6 tests passing, 99,994 tests failing* but *6 tests passing, 0 tests failing*. The lack of insight into the missing *99,994* would in and of itself conjure up a complete project/progress management apparatus whose elimination we are seeking.

Issue: fragility

The second issue arising is fragility. The property-based tests are a sub-class of system or end-to-end tests which are frowned upon in most software shops. Here is the test pyramid from the UK Hydrographic Office[50]:



The strategy here is to test discrete components and then do the minimum amount of testing to ensure that they are plumbed together. It builds on the principle of shadowing.

Shadowing is when the failure of one test guarantees the failure of another.

Consider the following spreadsheet formulae written as tests:

Formula	Expected Value
=1e+1	10
=1e+1*10	100
=sum(1e+1, 1)	11

If the first test fails because the spreadsheet under test doesn't not yet

[50] https://github.com/UKHO/docs/blob/main/quality-assurance/test-strategy.md

understand scientific notation then the 2nd and 3rd will also fail – they are shadowed by the root test.

A test suite where one error causes a cascade of failing tests is a fragile suite – fragility is experienced by software developers when a small change to the code causes a much larger amount of work to make the test suites pass again.

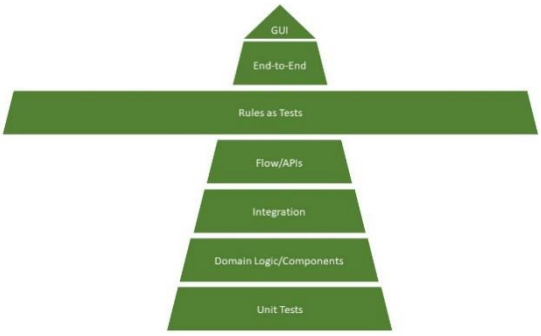
The UKHO pyramid deals with test fragility in two ways.

Firstly it tries to reduce shadowing to a minimum – which is why it's a pyramid and not a column or a vase.

Secondly is it uses the knowledge of shadowing to triage the tests. The pyramid roughly is a hierarchy of shadowing: if there is a unit test failure in the bottom layer, there is an expectation that might be failures higher up in the domain logic/component tests, and integration tests, and flow/api tests and end-to-end and gui tests.

In the event of a regression failure and a sudden massive amounts of tests no longer passing the developer has their failure triaged by the shadowing structure – first fix all the failing unit tests and then retest. Usually that will fix it. If not move on to the failing domain logic/components test and systematically work up the pyramid.

By using generated property-based end-to-end testing we don't see a simple pyramid but a table:



(Generated property-based tests would augment but not replace normal testing protocols.) With property-based generators there might be 10,000,000 rules as tests generated – 99.99% of all tests.

This brings a couple of issues:

- when to run the test suite
- what the developer does when confronted by 3,000,000 failing tests

The first is a set of practical problems arising from how long it takes. Where in the build chain does it run – not on the client side pre-comit, but:

- on each commit?
- daily overnights?
- weekly over-weekends?

If it runs daily or weekly then it runs against a basket of commits – so who is responsible for fixing the faults?

The second problem is our old friend shadowing – generated tests are slight variations on each other and the amount of shadowing is enormous. If the tests use a reducing framework like QuickCheck – it will triage for you. In the absence of that the test system will need to have some simple-to-complex naming convention (at Hypernumbers we had test suites named with prefixes a_, b_, c_ etc, so fix the a's first, tests were also arranged simple-to-complex within our hand-written test suites too – fix the top ones first and work your way down).

Developing a test system that can generate self-triaging tests is not insoluble, but it requires someone to do it.

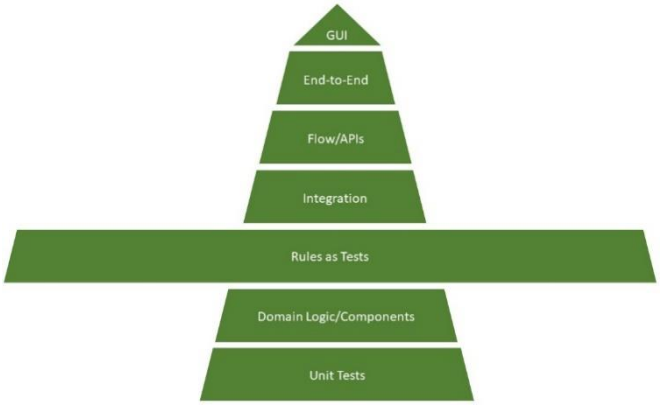
Issue: decoupling

Catala is a programming language – its parser chain is written in OCaml, but it has hooks for transpilers to produce outputs in a range of languages other than OCaml, most notably Python and Javascript[51].

If it is to be used to generate test suites it needs to respect the fact that the systems under test may or may not be written in any particular language. The mechanism that Catala uses for the calculations (transpile them to your target language) may or may not be appropriate for testing (generate a test runner in your target language) – or it may make sense to expect the system under test to expose URLs to which payloads can be applied – which would enable a single test runner executing generated tests for all development languages. As noted earlier this brings its own problems.

If Catala were to output not calculations but state machines it might be possible to rejig the property-based tests as component tests:

[51] I had a pop at writing first a transpiler first to Gleam and then Elixir (both Beam languages) for fun, but only having 2 or 3 days, and not speaking either OCaml or Catala (or Gleam) proved a bit of a barrier to making a lot of progress – but adding transpilers to other languages in Catala is, as they say, *only a small matter of code*.



This would solve the problem of the routing table – you would be applying state transitions at some internal level – the state transitions (sans logic) could probably be generated too as stubs. I can see a way to do this for Elixir/Erlang and the Beam in general, where there is strong and native support for state machines – but this violates the principle of decoupling.

In Summary

There needs to be a substantial programme of work to address and work through these issues before Rules as Test could be production ready. Building a test runner with the right affordances to fend of project management demands, and also be integratable in automated build processes on GitHub is non-trivial.

Working Paper 3 - The Lego state

Version 1.0

A guide to thinking in components

Introduction

What is a Lego state?

A lego state is one where the administrative, regulatory and financial state is no longer a large group of unique, hand built (and perhaps hard to navigate) systems.

It's one where the digital state is built from common, and shared, components. Some are stand-alone services: identity, payments. But some are occluded from the citizen: shared organisations, shared digital systems, shared databases, shared legislative solutions.

It's a state where the thinking that has made Lego and IKEA and car manufacturing and high street coffee chains such successes are applied to government. The twin paradoxes of standardisation:

- choice by assembly, where standard components are assembled into tailored solutions
- flexibility from rigid composition

Who are you?

You are a policy person, a service designer, a data architect, a delivery manager, a member of a project team, an operational manager, an elected representative. You are in government or opposition. You work at a thinktank, or in parliament, or government, as a civil servant or political advisor. You care about how we build an efficient and effective state in the digital age – one that has the best outcomes, is easiest and unobtrusive to use and has the lowest costs possible.

Why should you read this?

There is a thinktank-to-government-services assembly line that takes ideas, turns them into policies, and legislation and onto departments and systems and service delivery. You should read this to help you think about this as an improvable process, one that you can shape to get the better outcomes you want, to help you speak to your colleagues in a shared language that will let you change processes and make the trade-offs in an informed manner.

Context

How to think about componentisation?

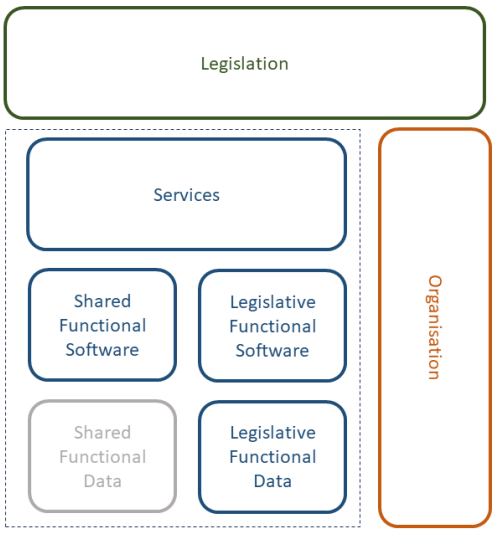
Developing complex propositions should usually follow the think big/build small model. This document unabashedly *thinks big*.

Consequently it is appropriately shallow. It is about sketching end-state patterns in broad outline.

When developing your lego strategy you should do the opposite – *build small* – and identify the smallest programme of work that will implement components. Legoisation is about capability more than it is about outcomes – if the capability is grown the outcomes will follow.

Legoisation described

Legoisation is the process of moving from hand-building state systems to assembling them from components. It can happen in many places.



Legislation

The law matters. Ultimately the computer systems have to do what the law says. Consider licensing schemes. Governments issue lots and lots of licenses to do things. The more custom the laws for various licensing schemes are, the wider the range of behaviours of a single deployed licensing software system would have to implement. So legoising legislation should shrink the amount of work. It should also reduce the costs of developing legislation and policy.

Services

The citizen experiences the state as services – provided by people or digital systems or a mixture of both. Things like tax and social security are experienced as web pages or call centres, as data to be provided and hoops to be jumped through, as success of failure and appeal against the decision, finding, navigating and logging in, getting paid.

Software

Services are the implementation of the services in code – either built as a single system or composed from multiple services.

Data

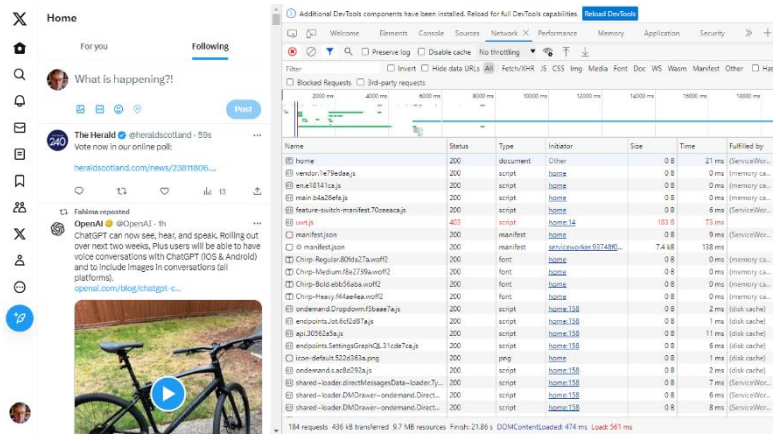
Code and services typically change and mutate, but underneath them is data which is the persistence of customer information in a database or file system. Data comes surrounded and hedged with procedures: who can do what to it when and under what circumstances.

Service Composition

Typically a software deployment will consist of a service, or services, implemented in software and with its own data persistence/database layer. These three elements tend to be fairly tightly coupled which is why they are shown linked in the diagram.

The service is usually a composition of many different tech and data components – what you might call technical services. This composition (and encapsulation) is a key feature, lots of small components become one larger one at a higher degree of abstraction.

For example the home page of my Twitter account on the web calls 184 different end points to retrieve the data and layout to display my Twitter.



Organisation

The final wrapper is organisation. Software service don't run themselves but are wrapped in people. And organisations can be layered. One organisation can develop, deploy and manage software that is used by another organisation. Development, deployment and management can themselves be split out by organisation.

Organisation is different to the other domains here – because ultimately human beings are the determining factor – all the other components and the contracts that we use to make them interoperate are creatures of the imagination, things human beings tell other human beings to help them organise their work co-operatively.

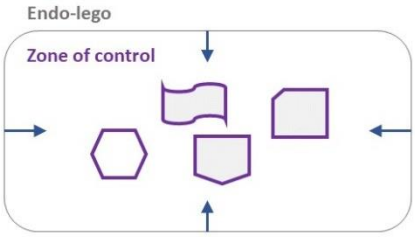
A Lego state has capability

A Lego state is a state that has the capability to create digital, legislative and organisational components and compose those components in more effective ways.

Legoisation is not something that can be bought, or applied, or completed. Becoming a Lego state involves a host of changes. Mindset, process, organisation, training and incentivisation, career and financial flows to name a few.

Modern states and civil service organisations are already a long way on the journey – albeit mostly as *endo-legoisation*. Different professions are producing pattern books within their own domain.

Endo-lego is made by professionals within their zone of control – data experts making standard data components, UX experts making HTML controls with JS and CSS libraries to support them and so on.

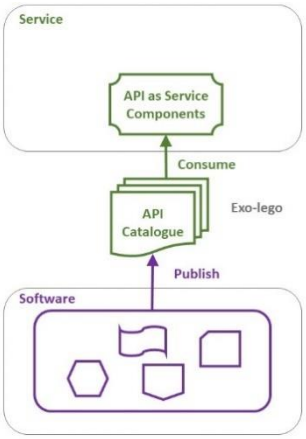


Here is a pattern book for common web components from the Government Digital Service.uk gov. The Scottish Government's Parliamentary Counsels have a pattern book *Common Legislative Solutions*. Two wildly different professions, web developers and legislative drafters.

And sometimes these patterns are crossing domain boundaries – the web components design pattern book has been implemented in code at GOV.UK Frontend. This is the first step in *exo-legoisation* – where patterns in *my*

discipline start being promoted as canned solutions with catalogues in *your discipline* and we started seeing alignment across organisational boundaries.

Exo-lego is related to endo-lego by a translation and a promotion. An example would be written standards (this is how we build APIs) becomes a framework for making APIs. An instance of that API is exposed in a catalogue for Service Designers to use – and the API is consumed as a Technical Service Component. It is a promotion because the Service layer is at a higher level of abstraction than the software layer.



That process of how you take identified patterns in one world and translate and align them in another to simplify delivery is what this document is all about.

And the fact that it is an exo-process and not an endo-one should tell you a lot about how to think about it. If you are only talking to your team, your peers, your profession, then you are only talking inward facing - endo. Sorting out your professions view is an important part of the game – you can't align with anyone else if you don't, but it is simply the beginning, getting to the start line, the base camp. The finish, the summit still awaits you. When you are componentising your team, like your output must be endo, outward facing, including people *NOT* from your zone of control, *NOT* from your zone of comfort, *NOT* from your zone of expertise.

PROMOTION OF COMPONENTS

One of the core concepts in component thinking is promotion – and the key thing is that there are lots of routes to promote. The endo/exo example took a well understood one software patterns to API-as-a-service but there are plenty more.

A standardised component is promoted from a pattern, a way of doing things, to a thing to be used, to be selected from a catalogue of things.

This is best explained with an example. Data on property was

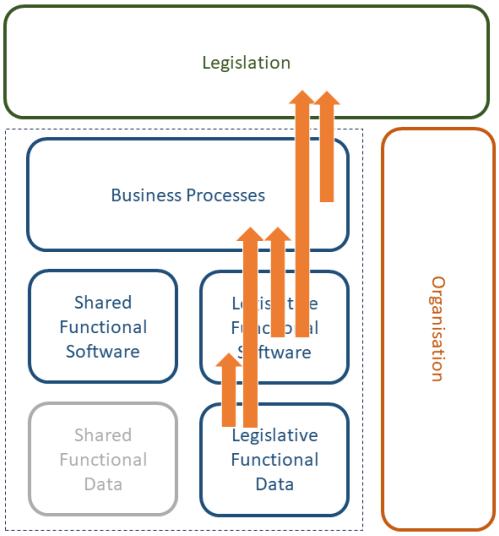
standardized[52], with the creation of UPRNs (unique property record numbers) and USRNs (unique street record numbers). This was a written standard – you should use their attributes in your data schemas.

This didn't standardise postal addresses though – my flat can be described equally well as:

- 16-5 Hart Street
- 16/5 Hart Street
- Flat 5, 16 Hart Street
- Top Left, 16 Hart Street
- TL, 16 Hart Street
- 16-5 Hart St
- 16/5 Hart St
- Flat 5, 16 Hart St
- Top Left, 16 Hart St
- TL, 16 Hart St

The standard (UPRNs and USRNs) was *promoted* to a web service AddressBase[53]. Before it was a description of how to do something (but you had to build the thing), now it is a web API (which you can just consume).

Generally legoisation is about promotion – up the stack – and there are lots of routes:



In this case data has been promoted to a web service – the left hand, smallest and lowest orange arrow on the diagram.

All promotions, irrespective of their 'from' and 'to' have the same characteristics:

[52] Identifying property and street information - GOV.UK (www.gov.uk)

53 Access free address data using AddressBase - GOV.UK (www.gov.uk)

- take something that is standardised in one domain and translate and transpose it into a more abstract but different domain
- make it technically available in some fashion
- publish it in some sort of catalogue so that a range of different organisations can find and use it

Critically publication must be a co-operative act with the consumers and not something imposed on them. Components are only components when they are used – otherwise they are just white elephants

Remixability

Remixability is a property of a Lego state – the ability to reorganise institutions and refocus attention from one area of society to another. It is part of the world of componentisation because it involves reassembling components into different organisational structures, but, but...

There is a paradox of decentralisation – in order to decentralise digitally there are core components that you need to centralise first.

Remixability falls squarely into this world – and for that reason it is the subject of a different working paper *Working Paper 4 – the remixable state* (you can find it on SubStack at DigtialPolicy[54]).

Tradeoffs

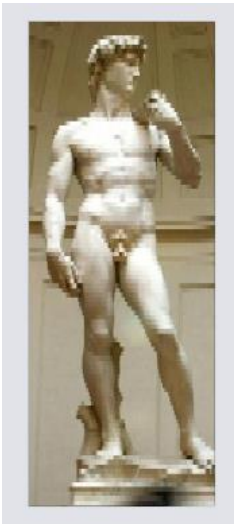
There are a range of trade-offs that need to be made.

Legoisation is a lossy process, you lose the ability to hand craft details. The degree of loss depends on the brick size. Imagine a statue of Michelangelo's David made in lego. Would it be good enough? Imagine it in duplo (the big lego for toddlers). Imagine it in breeze blocks.

[54] https://digitalpolicy.substack.com/

So there is a brick-size dial that you can twiddle to trade off.

Marble



Lego

Duplo

Breeze blocks[55]

This loss of resolution is the consequence of the rigidity that enables the flexibility. During the design of your components you must focus on the rigid constraints, they should be as small and as few as possible, but no lesser and no smaller. If they stop being rigid they stop being able to define composable components.

The trade-offs will be described in terms of ladders as much as possible. Options on the lower rungs are suitable when there is low levels of functional alignment and the more alignment the higher up the ladder you can go.

Exactly how much do you want to componentise, and how much must be custom and how do you match the two? At the highest level each organisational unit might only need to be able to skin services into its branding with its logo and colours, but there will be other reasons for customisation that might require more structural divergence.

A good way to understand componentisation is to look at the Volkswagen family of cars. Audis, VWs and Škodas share about 75% of the same components, engines, chassis, gear trains, electrics and so on. But the variation allows the creation of product lines for premium, family and cost-conscious

[5555] Modern art is rubbish, innit? Just a lot of old breeze blocks and yeah that's me being polite, you can't even see if he's in the nip

customers. (The book *The Machine That Changed The World* [56] is the classic read about this and the birth of lean production).

Once upon a time there was a separate car factory per brand – one for Audis, one for each VW marque, one for Škodas. By componentisation wider ranges of cars could first be built on the same line, then lines could be merged until one factory can churn out lots of different types of car. Gradually the cost base was shrunk and squeezed. And this restriction is experienced by the customer as an explosion of choice, making things swap-out in the macro makes them swap-out in the micro: seat fabrics, in-car music systems, colour, lights, windows, engines, wheels, all now offered in a range unavailable 20 or 30 years ago. "Any colour they like as long as it's black" it surely ain't no more.

As we go through the layers the same rough'n'ready reckoning will be used based on a VW percentage. The lower the VW percentage (low % of shared components) the lower down the ladder you will componentise (and the more operational systems you will end up with). The higher the number, the higher the componentisation and the less systems you end up with.

At a software level there is a centralised/decentralised trade-off. A software system for each of the many thousands of license types that Scottish Government issues is definitely excessive. But one software system for them all is probably not enough.

Centralising might reduce deployment cost at the expense of slowing down systems maintenance and increasing operational costs. Very similar systems can be economically supported by a common system, but too much variation, even if it is implemented in pluggable blocks, doesn't necessarily make for the best result. A single Scottish government system would impose massive cross-departmental communication overhead.

Legoisation is about making choices and trade-offs. There is no one correct answer – and the best answer is determined not by where you want to go on its own, but also where you are starting from – the current landscapes: legal, organisational and technical.

There is another aspect that needs to be considered – organisational maturity – how many rungs are we capable of going up at a time. Legoisation is a learned skill and we are finding our feet at the bottom at the moment.

---

[56] https://www.amazon.co.uk/Machine-That-Changed-World/dp/1847370551/ref=sr_1_1?crid=2DJ9X7YQUH79&keywords=machines+that+changed+the+world&qid=1696506260&sprefix=machine+that+changed+the+world%2Caps%2C72&sr=8-1

MOST OF THIS SIMPLY ISN'T NEW

All of the components and componentisation choices discussed in this paper are well known, well understood and have considerable literatures. This is about extending techniques that are tried and tested – and which in many cases are already implemented in the public sector. *You probably know more about how to do this than you think*.

ORDER OF CONSIDERATION

Government computer systems and their development fundamentally differs from private sector ones because the private sector has to seduce people to become users whilst the public sector can compel.

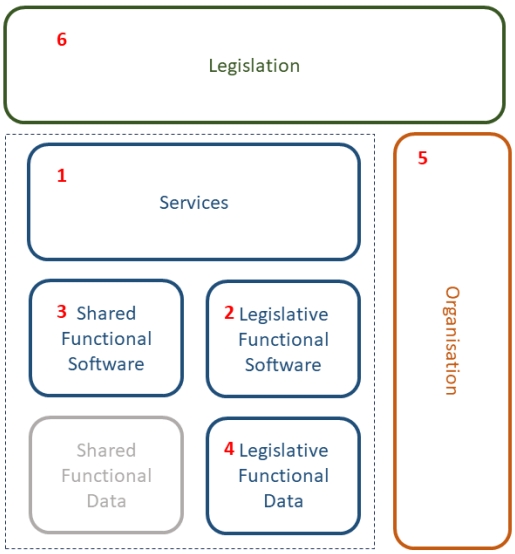
In thinking about legoisation we should start not at the top (with legislation) but in the middle (with services) and the citizen and their experience. The law should be shaped to make systems easy for citizens to use, and the technical and data architectures fitted to that world.

After services it's time to look at the software architecture and then the data architecture. The organisational architecture is how these are composed into services that can manage themselves operationally: costs, staffing, operational reporting and all the rest.

It is critical to remember that the organisational level is different to all the rest – the models and components and all the rest only exist in the minds of people. Fundamentally, unlike lego which has physical components, all of these components are "contracts" that need to be lived by human beings. Fundamentally at scale all software problems are human problems[57].

[57] hat tip to my old gaffer Mahesh Paolini-Subramanya for that zinger

Legislation is the final part – once there is an appropriate organisation and service model that meets our cost and flexibility requirements it becomes possible to make of the parliament a machine for stamping out bricks that we can build with.



Componentisable Layers

Introduction

This section will step through the various layers that we can componentise and sketch out the options in each. Critically we can understand where promotion of components (say from a service to an Act of Parliament or a Ministerial Order) can pre-bake systems and reduce costs and implementation times.

Services

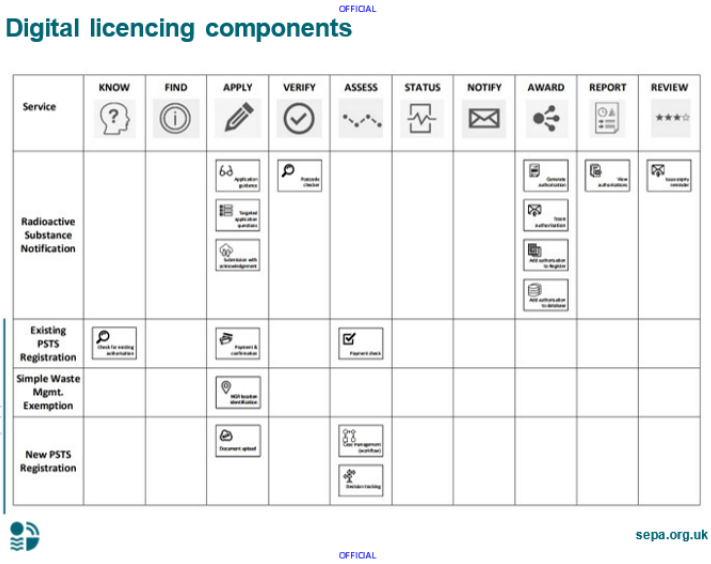
The Scottish Environmental Protection Agency issues thousands of licenses, in a variety of flavours to a wide range of organisations across Scotland. They have been through a couple of rounds of legoisation – building software systems and approaches that can consolidate the business processes of different licensing services whilst supporting the full range of required licenses.

In this slide from SEPA the licensing team have componentised their systems based on verbs extracted from service design – cataloguing what the user has to do to successfully apply for a license to perform some action. The software development process was then to build reusable systems that could support multiple licenses of which they have several thousands.

Their specific licences are themselves legoised – a custom license for a particular installation might be assembled from a selection of sub-components

– think of a ‘coffee’ from a chain coffee store – *double decaff frappe hazelnut latte with oatmilk and 2 sugars* is both ‘custom’ and ‘standardised’ – back of the envelope calculation Costa has north of 1,600 variations of coffee alone – choice through assembly of components – the paradox of standardisation.

We can see this in their licensing diagram:



Common business processes are necessary but not sufficient for effective componentisation. They represent the functional specification of the system, but there might be non-functional requirements that prevent different things sharing the same platform.

When looking at ‘Verbs’ it is worth assessing the VW percentage of each verb. The verb *Pay* has a VW percentage of nearly 100% across all government systems that make payments (Social Security has a duty of care to release lump sum payments to vulnerable clients in a trickle, but apart from that bog standard).

Functional specifications are things like paying benefits in Social Security, or calculating tax in the Revenue – things that are specific to a particular system, that make it what it is.

Non-functional specifications are things like: must work in a browser, must have data saved to disk, must use 2-factor authentication. Things that are not specific to what the software does – more about how it does it.

Typical non-functional barriers to legoisation relate to rules around data in the widest sense. To access any police system you need to be a vetted police officer or police civilian worker, and for the police to access any non-police system they usually need a warrant from a sheriff. For good reasons you won’t be seeing common platforms between police and non-police functions anytime soon.

A verb with a super-high percentage is a good candidate to be pulled out into a standalone service (as *Pay* already is).

There are a couple of things to look out for:

- does the presence of a handful of 99% verbs pull up the overall VW%?
- could one set of verbs with a low VW% be turned into 2 or 3 each with a much higher one?

Imagine a proposed set of (partially randomly generated) component verbs:

Verb	VW%
Find	99%
Identify	99%
Overlook	20%
Realise	31%
Generate	19%
Benefit	23%
Interfere	32%
Pay	100%

Overall it looks componentisable – but *Find* is a CRM, *Identify* is the digital ID service, *Pay* is the payments rails – pull these out as stand alone systems and you don't have much left.

Take them out and then split the candidate systems into 2 sets and you might get:

Verb	Set 1 VW%	Set 2 VW%
Overlook	62%	N/A
Realise	71%	N/A
Generate	78%	54%
Benefit	N/A	65%
Interfere	N/A	72%

This sort of juggling of business process is the way you design your ideal future state. Identifying where you would like to be. The next challenge is to figure out if you can, should or would get there. It is technically possible? is it legal? is it easy? do the necessary people have the capability to do it? is it cost-effective to do it? what else would doing it unlock? of all the things you could do now, is this the priority?

Legislative Functional Software

There are a range of models for legoising *legislative functional* software –

that is to say software that implements a particular specialist requirement. The actual provision of an actual service (say social security) requires many different software solutions: the main social security system, a call centre system, HR, payroll, desktop support. This table applies to the social security system only – other shared components have their own consolidation ladder.

Software as a service	
Rung	3
Description	A configurable online platform that a particular government department can use to set up a service that conforms to a certain type
When To Use	When there is legislation designed to be implementable on the government SaaS platform

Hosted Software	
Rung	2
Description	One unit of government runs a software platform and when another one wants to implement a service on it they contract with the hosting unit to build out an implementation of their service
When To Use	When software components or add-ons need to be developed to support a particular new requirement

Open Source or Shared Software	
Rung	2
Description	One unit of government builds a system and publishes the software for other units of the government to use 2 flavours: • keep in synch – where each user contributes changes to the central version and the software organically grows to handle more types of government service fork – where you start with a copy of the software and customise it into your own system

Open Source or Shared Software	
When To Use	When you need to run your own instances (you have sensitive data and you can't let other state employees see it) It is possible to move Open Source/Shared software to Software As A Service if there is sensitive data but it requires secure data multi-tenanting and a hosting organisation that has the highest levels of authorisation.

Source of inspiration	
Rung	1
Description	One team has done a good job and the other team comes and studies their design processes, implementation and service model
When To Use	When the two business processes are not closely enough related to make a common system worthwhile – perhaps one system is legoised already for all the Xs and the other group want to build a lego systems for all the Ys

Notice that Hosted and Open Source/Shared are at the same rung – functionally they are roughly equivalent – but there are non-functional reasons (data access, other operational requirements) that keep them separate.

The higher the VW% the higher up the rungs you can go.

We can also manage components in the business process Verb model by converting the Verbs to Participles.

Just say our analysis of the Licensing Verbs that SEPA identified ended up looking like this:

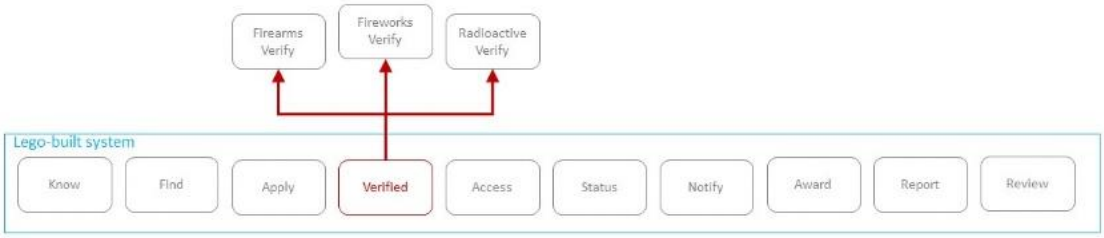
Verb	VW%
Know	70%
Find	69%
Apply	72%
Verify	12%
Access	68%
Status	68%
Notify	71%
Award	70%
Report	69%
Review	64%

So everything looks fine except *Verify* is madly low – Firearms require the police to verify in an interview, Fireworks needs a police record check, radioactive substances needs proof of professional qualifications, and on and on.

We can transform *Verify* to *Verified* with a fan-out architecture. The common components now look like:

Verb	VW%
Know	70%
Find	69%
Apply	72%
Verified	100%
Access	68%
Status	68%
Notify	71%
Award	70%
Report	69%
Review	64%

The core system no longer cares how you verify for a given license – it just kicks out to a system that implements the Verb for each license and the lego system just holds the result *Verified/Not Verified*.



To use the car analogy, you might merge your production line so that you can build Audis VWs and Škodas before you are able to merge your engine lines, so maybe you have 3 of them each producing a low-power, a medium one and a high-performance, or a petrol line making a range of engines, and a diesel one likewise, or some other appropriate split out.

So you can twiddle the Verb-Participle knob to fix up things – but if you whack it up to 100% and make all verbs into participles you smash the whole of Government into thousands of fragments, each running their own tiny webservers to do minute slivers of work in an unmanageable horde.

It should also be noted that basic infrastructure is now commonly rented from major suppliers – software is run in the cloud. This is a form of componentisation (endo-legoisation within the tech world) that is now well

understood and widely adopted. The key take-away about <the cloud> tho is that it is not some magic that you can rub on your software and it makes all your troubles go away (marketing hype ahoy!). Migrating to the cloud is an important first step, and an opportunity to test your organisation's maturity in component thinking but it is not the end by any means.

Shared Functional Software

Shared functional software is things like call centre software, desktops and fileservers, accounting, HR and payroll. There are a couple of reasons why we need to think about it.

Firstly our own government specific software needs to be delivered in a full operational context, staff you work on laptops and log in and take calls and send emails and get paid.

Secondly, the legoisation processes that we want to bring to government are already well advanced in normal commercial software – looking at shared functional software is a good way to get a sense of what our own delivery offerings can and should do. It is a good way to get shared understanding of what our as-yet-undelivered systems need to do by pointing at and learning from work that other sectors have already done.

In particular we can see that already software is moving into multiple layers of componentisation – from cloud provisioning, to fully service platforms-as-a-service to software-as-a-service. Things like multi-tenanting (where many customers share the same infrastructure) will all have their own futures across state systems (if they aren't already being implemented).

There already exist government guides to buying and implementing shared software components so I won't recapitulate them here. The ladder looks very much like the legislative functional one (although there is a plethora of marketing terms and fads).

Data

Data seems the easiest layer to deal with. The first problem is that it can seem trivial. It isn't. Keeping the same data held in multiple places in sync is very difficult and highly technical.

From a cost/simplicity perspective tho it is simple. Either you have the data in one place, or you have it in two or more. If you have two identical data sets with their own procedures and process and you replace them with one then you have eliminated work, and confusion and cost and all the things that we are trying to get rid of. Job done.

But merging datasets is easier to talk about than do.

To successfully merge datasets you need to align a number of operational factors.

Action	Description
Defining	Where data is defined, could be legislation, regulation or ad-hoc
Auditing	This is general looking at the data for data quality, conformance with human rights, and data protection, checking that data is not available to the wrong people, weeding and deletion activities
Appealing	The process and procedures whereby a person or organisation or thing gets onto or gets taken off the database
Partitioning	Where the data entity is partitioned, across local authorities, across health boards, internally within SG and its agencies
Creating	The point of creation - and who, how and why the data must be created
Reading	Access rights to use and see data
Updating	The processes for updating data - specifically updating by overwriting - so not ledgered/immutable data structures (ideally we want to use ledgers as much as possible and not update).
Deleting	The processes around the deletion of a data item (this is generally partial deletes of items and not total delete/weeding of a set of collated data which is covered by Auditing) (ideally we want to use ledgers as much as possible and not update).
Refreshing	Is the data once and done, or is it supposed to be up-to-date?

As a rule of thumb, for 2 data sets to be merged into one, all of these need to be either harmonised or managed. The problem comes from the fact that they are not uniformly specified. Some laws have a set of some of these in the primary legislation (Acts of Parliament) with the rest handed over to the operational implementors to decide. Some pass some or all of them over to secondary legislation (Ministerial Orders). Some legislation (primary or secondary) is handwaving, some is precise.

To make it worse, poorly described rules of managing or accessing data may have been sharpened up in case law where courts, faced with uncertainty or imprecision in the law, have adjudicated and set precedent.

The cost of working out if two data sets are or can be aligned can be expensive and uncertainty itself is a major barrier to joined-up government. Faced with time, cost, immediate impact and promotion or political pressures the default option ‘go it alone’ often seems to make the most short term sense, despite the long term impacts and degradation of services that brings.

Data also needs to be aligned with the Rule of Law at a design level[58].

ORGANISATION

Organisational models have a similar ladder structure to technical consolidation models.

Umbrella Services	
Rung	3
Description	A single organisation provides a host of multiple services using common infrastructure like call centres, website, help desks. It controls software development and how the IT presents to employees and the world
When To Use	When there is a mass of closely related services and transfers between one service and another is common (think health services)

Platform Services	
Rung	2
Description	The 'owners' of the service supply non-technical staff (citizen facing, call centre staff) and purchase or use platform services (managed software) from another government organisation or private provider
When To Use	Where the services being supported are structurally similar but culturally and operationally very different (think firearms licensing and dog licensing).

Infrastructure Services	
Rung	1
Description	One government department or supplier provides raw machines and database servers in an environment that has some basic management (backup, restore, failover, electrical and physical redundancy) and the IT department of the purchasing department install and run software on it

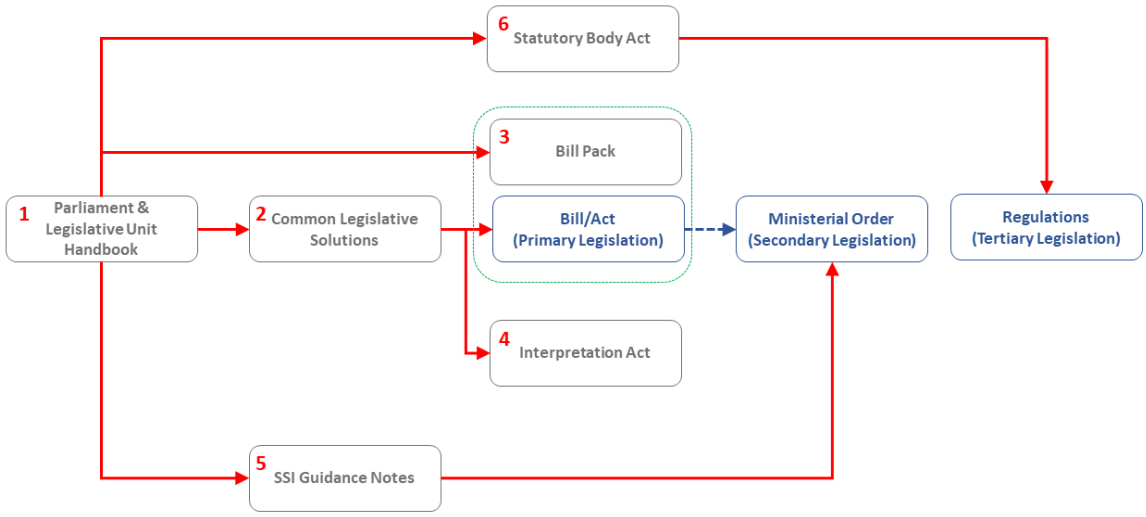
[58] There is a BIus Working Paper that looks at this. Working Paper 1 – *Data and the rule of law* – and another Working Paper 5 – *Law reform for data* will go into the legislative side of it in more detail.

Infrastructure Services	
When To Use	Where there is a need for very specific and custom software environments

Organisational design here is being discussed at a surface level – how it is experienced at the boundaries. As a discipline it focusses heavily on internal structures within an organisational unit. How you build and resource your teams will shape what you deliver. You should design organisational components that encapsulate data, software and service components such that they contain *all the necessary powers* to do their job. Poorly placed organisational boundaries can and will kill effectiveness and performance.

Legislation

Finally, we are at legislation. Before we can discuss legoisation at this level it is important to understand the legislation assembly line. Obviously, this will vary by jurisdiction, but most (Anglo-Saxon) jurisdictions have processes that are closely related to the Scottish system.



Lets walk through this process step by step:

	Description
1	The process of creating primary legislation (acts of parliament) is outlined in the *LPU Bill Handbook*[59]. (The Westminster equivalent is the *Guide To Making Legislation*[60]). Neither of these documents mentions technology, data, services or systems – nor prompts people working on legislation to consider how it is to be delivered at all. These processes are targets for promotion of checklist questions: "have you considered? here is a way of doing this..."
2	The bill process develops policy which is then passed to the Parliamentary Counsels to turn into a Bill – draft legislation for consideration by the Parliament – that process is supported by a pattern book *Common Legislative Solutions*[61]. It has patterns for licensing, creating bodies and a variety of other common drafting problems. It is endo-lego. Again this is a target for promotion of checklists: "write this into the legislation".
3	The law itself is not the only target for promoting into. Every Bill which is introduced into parliament is done so as part of a Bill Pack of which the Bill is just a part. At this point we slide out of the world of government and the civil service and across the constitutional boundary into the world of the parliament. The 'proper form'[62] of a Bill is determined by the Parliament. The Bill Pack contains a variety of elements – impact assessments, financial statements, explanatory notes as well as the proposed law itself. Not all elements of it have the same status. Basically when resolving a point of law, a judge may take into account the Explanatory Notes and the text of the parliamentary debate in forming an opinion on the correct interpretation. Other components of the Bill Pack are not in themselves justiciable.
4	The Interpretation And Legislative Reform (Scotland) Act 2010[63] is the primary pattern book in law. It is a repository of common definitions which other acts then refer. It is in effect a published catalogue of legal lego blocks – and is a promising target for promotion into.

[59] https://www.gov.scot/binaries/content/documents/govscot/publications/foi-eir-release/2022/07/foi-202200306018/documents/foi-202200306018---information-released/foi-202200306018---information-released/govscot:document/FOI%2B202200306018%2B-%2BInformation%2Breleased.pdf

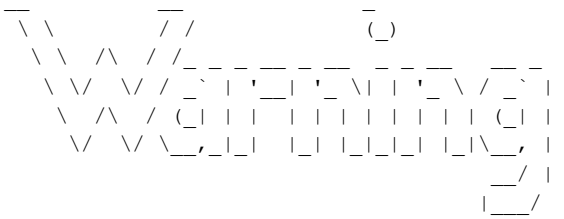
60 https://www.gov.uk/government/publications/guide-to-making-legislation

61 https://www.gov.scot/publications/guidance-instructing-counsel-common-legislative-solutions/

62 In the Scottish Parliament's case by Standing Order 9.2
https://www.parliament.scot/about/how-parliament-works/parliament-rules-and-guidance/standing-orders/chapter-9-public-bill-procedures#topOfNav

[63] https://www.legislation.gov.uk/asp/2010/10/contents

	Description
5	Not all law goes through the Bill/Act (Primary Legislation) route. There are only some 22 Bills per year at Holyrood – as compared to about 400 Ministerial Orders (Secondary Legislation or Scottish Statutory Instruments). The SSI Guidance Notes[64] is pattern book for them, and a promotion target in its own right.
6	The final place for a catalogue is plain old regulations (sometimes called Tertiary Legislation) where a person or organisation has statutory powers to make a determination on standards and people are legally obliged to follow those determinations. This could be a statutory body (created by an Act of Parliament and under the oversight of the Parliament and not the government) or could be a statutory person (something like a Chief Medical Officer but for tech who could say "you must use this, you must do it this way..."). (I have written more extensively on statutory persons and their role in killing duff systems over on SubStack[65]).



Before crashing into this, take a step back and think about the constitutional consequences. The clear, easy and simple route is to create framework bills and push all pattern stuff into secondary legislation and guidance. *This route is wrong*.

Decisions about data and data handling are very long lived – with lifetimes longer than human beings. Careful thought needs to be put in to figure out how to do this.

This is a core topic of my research and is explored in this Scottish

64 I have a personal copy of the SSI Guidance Notes – I don't have access to the Scottish Governments Electronic Document Record Management systems. As far as I have been able to determine it has not be made publicly available by the Scottish Government – I suspect not because it is confidential, but because nobody outside government is interested in it

65 https://digitalpolicy.substack.com/p/stop-the-line

Government blog series[66]: Part 1 – we need a gearbox (blogs.gov.scot)
Part 2 – Frankenstein Bill (blogs.gov.scot)
Part 3 – technical pattern books (blogs.gov.scot)
Part 4 – a legislative architecture (blogs.gov.scot)
Part 5 – testing the proposals (blogs.gov.scot)

The starting point is clear, the first thing to do. The LPU Bill Handbook defines a process to follow – it can be thought off as a high-powered and well-written checklist. The first step is to inject our new pattern thinking into it in the form of a section focussed on the digital implementation that the policy/legislation will need to become a thing:

- have you talked to the data people?
- what existing technical systems will be impacted by this?
- who will deliver this and are they involved in the development of the policy?

and so on.

The process of injecting these patterns must include the owners and authors of the things-into-which-we-are-injecting-them. These professionals have their own patterns and ways of seeing the world – and the reconciliation of these two views of the same pattern cannot take place without their enthusiastic participation.

Creating patterns is all well and good, but unless the policy and bill teams adopt them, use them and see value in them it will be a wasted effort. The LPU Bill Handbook is therefore a critical normative force on future legislation.

There are a wide range of ways in which patterns and catalogues could be expressed in the legislative and policy world. The final arbitrator as to which is the best cannot be an outsider like me, nor is it obvious that currently there is anyone who can make that call.

The only sensible approach is to do this in an experimental manner with the right stakeholders and let the people using the patterns adjust the patterns and

66 https://blogs.gov.scot/digital/2023/08/28/basic-law-making-for-legislative-computer-systems-part-1/
https://blogs.gov.scot/digital/2023/09/04/basic-law-making-for-legislative-computer-systems-part-2/
https://blogs.gov.scot/digital/2023/09/11/basic-law-making-for-legislative-computer-systems-part-3/
https://blogs.gov.scot/digital/2023/09/25/basic-law-making-for-legislative-computer-systems-part-4/
https://blogs.gov.scot/digital/2023/10/02/basic-law-making-for-legislative-computer-systems-part-5/

bed them in.

In that spirit I will romp through a range of different implementation patterns that might be useful. Bear in mind that different elements of the legoisation might need different implementation in different places. This list isn't exhaustive, you should be able to come up with variants and options:

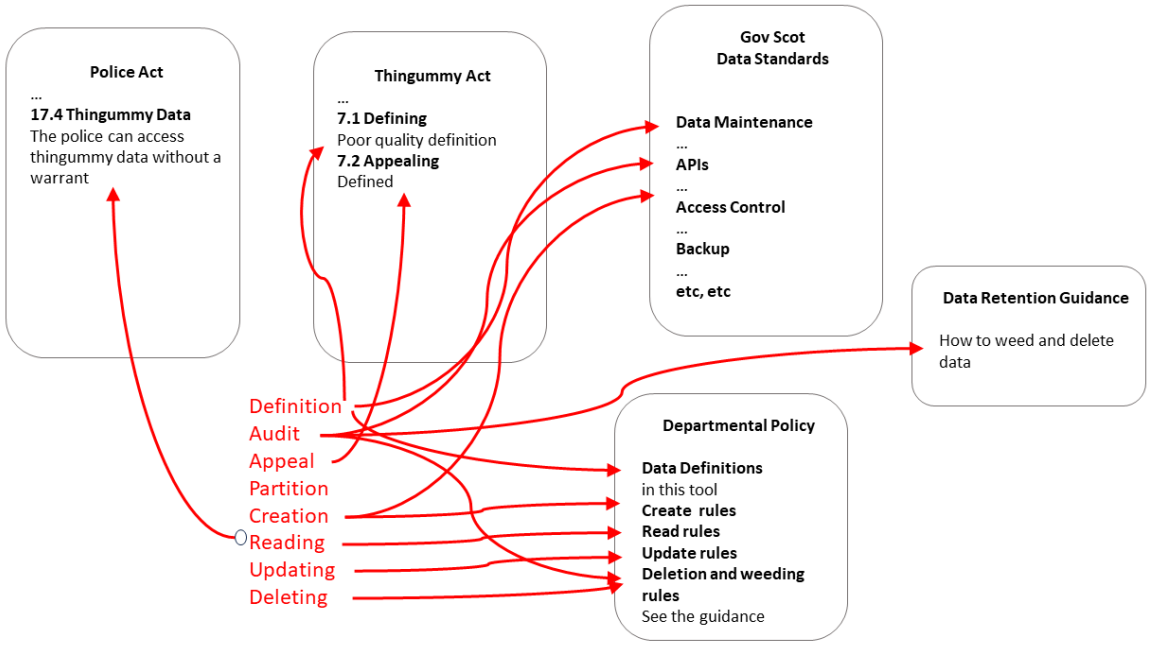
- Legislative tidy-up
- Standardisation of terms
- Drafting checklists
- Fully formed services
- Built-to-be-deployed legislative solutions
- Delivery frameworks

The data aspects of legislation will be discussed in more detail in the forthcoming Working Paper - *Law reform for data*, and the delivery framework will similarly be considered in the forthcoming Working Paper - *Experimental digital legislative processes*.

Legislative tidy-up

If we look at the things which we need to know to define data management we see that for different systems they might be smeared across the legislative and regulatory landscape.

Lets look at where the operational factors for data that were defined in Section 4.5 get applied:



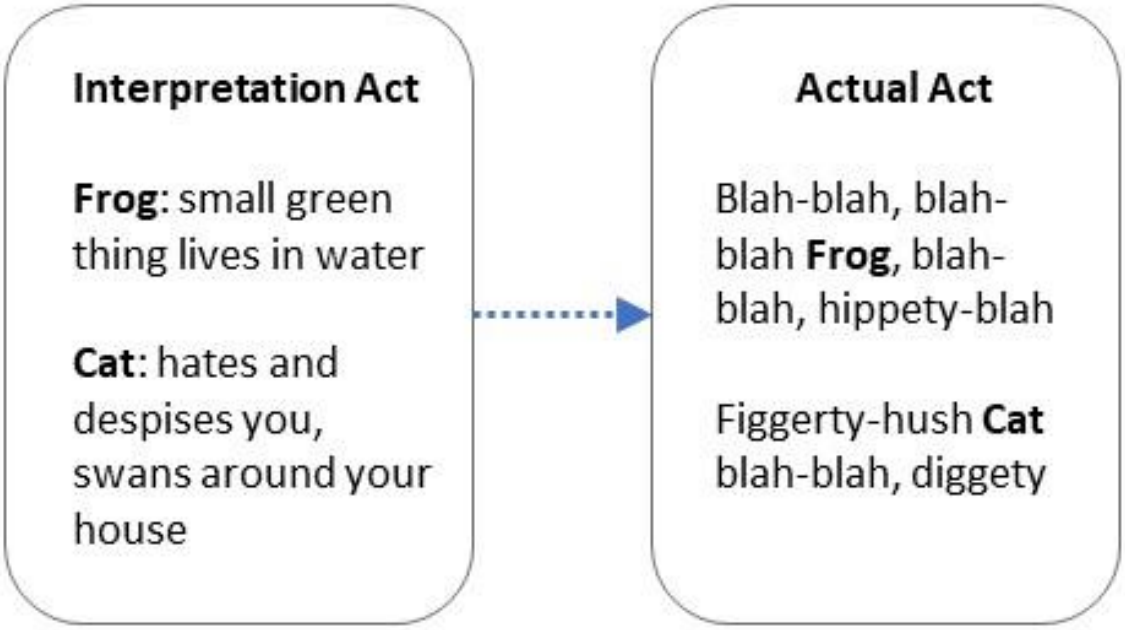
The things we need to know are smeared across multiple places (or simply not defined like partition). This is a world in which one-way is better than the best-way. Making legislation and regulation easier to reconcile with respect to

data management will be a key win.

In a previous life I was Service Architect at Edinburgh City Council as part of the BT/CEC joint enterprise. I did a straw poll among my database folks asking them how many instances of people data they had in separate databases (names and addresses) and the answer was roughly 80. I asked why didn't we just migrate them to a common customer database. Well it turns out that that isn't a technical question, the problem is working out if it is legal – massive job.

Standardisation of terms

Interpretation Acts are great places for standardisation:

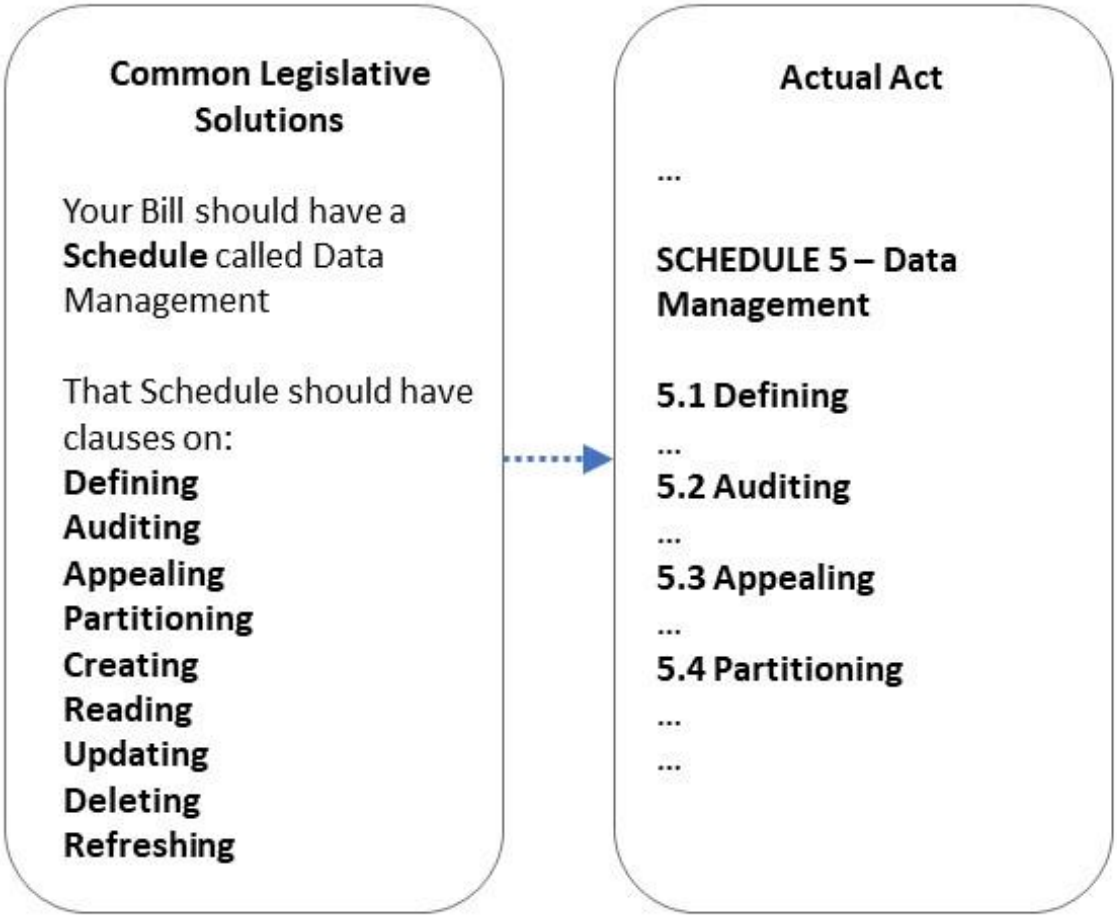


If you get to the point where components exist and there is a shared understanding of them across both government and parliament then promotion to a Interpretation Act is the final boss catalogue.

Drafting checklists

A parallel to the LPU Bill Handbook is the Parliamentary Counsels' Common Legislative Solutions. Structural patterns could be promoted into it.

This would be a catalogue entry for Bill Teams – *I will have an X like wut you have in the Common Legislative Solutions*.

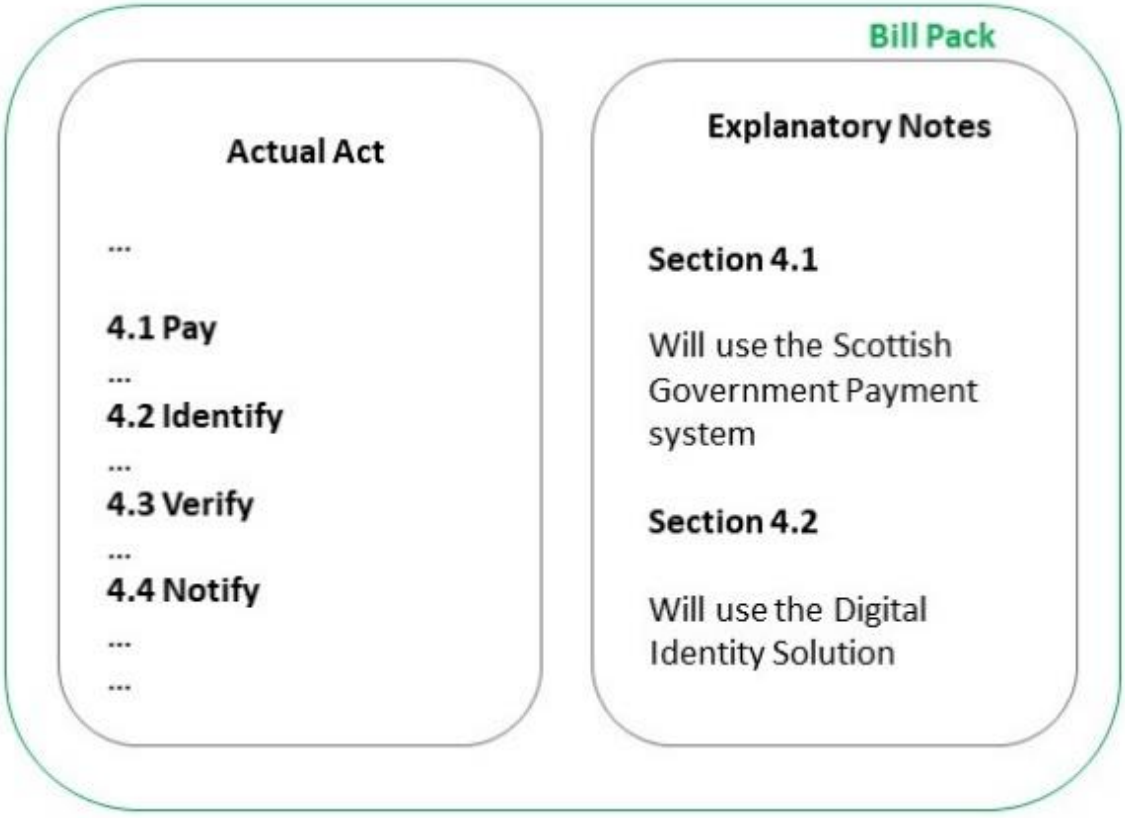


This is a particularly interesting example because it shows the 2 core components of exo-lego. Firstly the standardised object – here in the Common Legislative Solutions – and then the necessity to publish it. By the time it hits the Parliament Counsel for drafting a Bill process with associated policy might have been running for a year or more. The pick-me-from-a-catalogue moment will want to come as early as possible in the overall process. Again this a communications and engagement problem.

Fully formed services

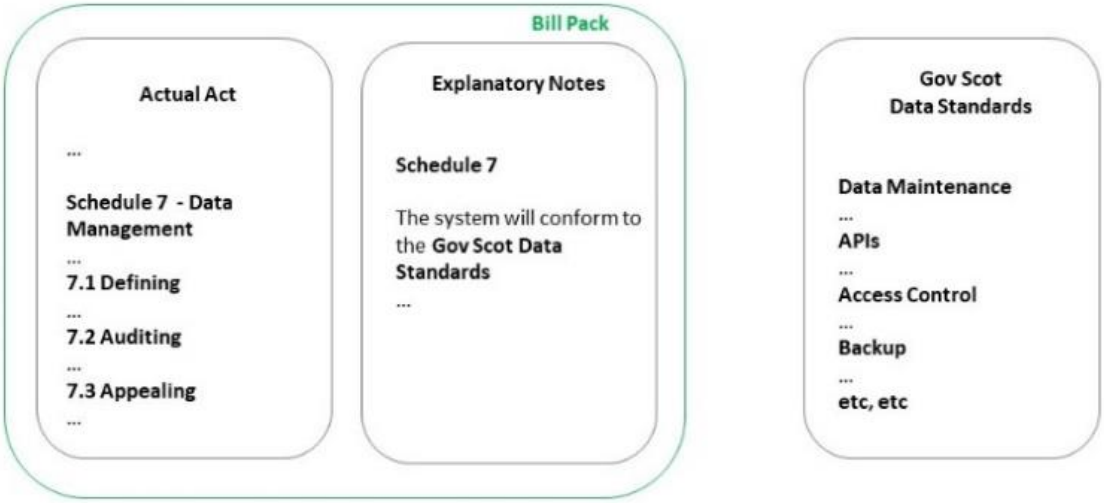
Fully formed systems (digital identity, payment, etc, etc) can be injected either directly or indirectly into the Bill Pack.

Lets look at direct injection first:



You will notice that here I am representing Verbs from the SEPA Service patterns of section 4.2 directly in legislation. And after using Service patterns to structure my legislation I am further embedding APIs-as-service-components (like Pay or Digital Identity) via the Explanatory Notes.

Remember that the Explanatory Notes are in the justiciable part of the Bill Pack. This might not be desirable. In this example of indirect injection regulations are used to embed data standards:



There are a couple of considerations about indirect injection:

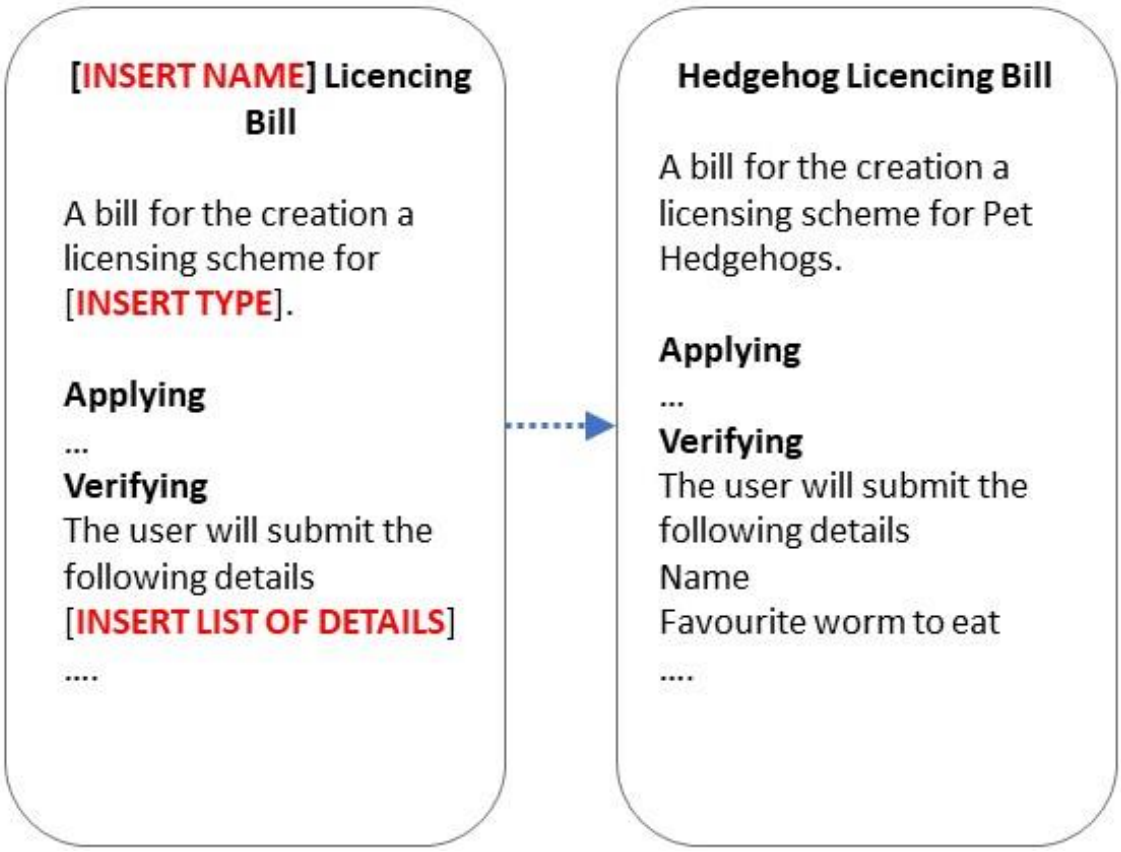
- What is the status of the thing being injected?

- Who issues it? a statutory body, a statutory person or some general function?
- Does the issuer have the power to ensure it is adhered to?
- Is it volatile and likely to evolve and change? (best referred to indirectly) or static and unchanging (maybe better direct?)
- Is it compulsory or aspirational?

Built-to-be-deployed legislative solutions

If the technical and service solution is sufficiently harmonised it might be possible to template legislation and have those templates available in the Common Legislative Solutions pattern book of the Parliamentary Counsel.

This model would work for very well defined and repeated problems (like licensing) where a target software system and associated service and organisation model exists and the government is seeking to get the authority to stamp out another one of them – licensing springs to mind:

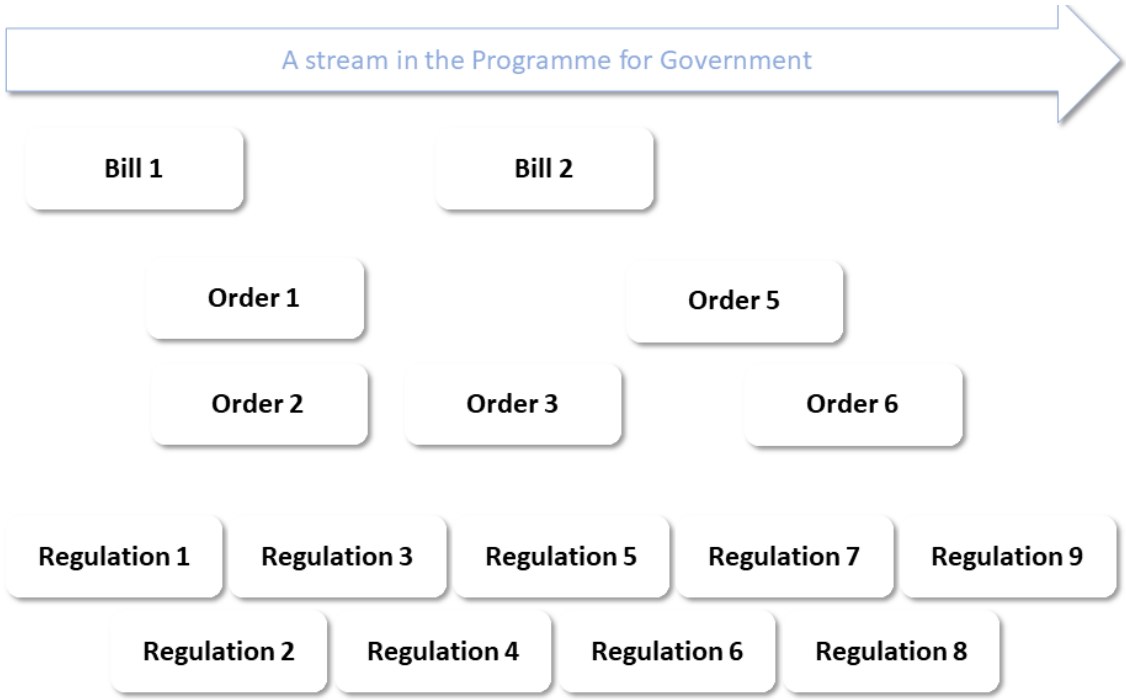


Delivery frameworks

Passing an act of parliament is a point-in-time event – delivering a policy rarely is point-in-time. Something like the new Scottish Social Security system has been in ongoing delivery for over half a decade. Often secondary powers are used to time-smear legal powers over the delivery timetable. The diagram

below shows different sorts of legal instruments being used to move delivery of a major programme – with time running left to right.

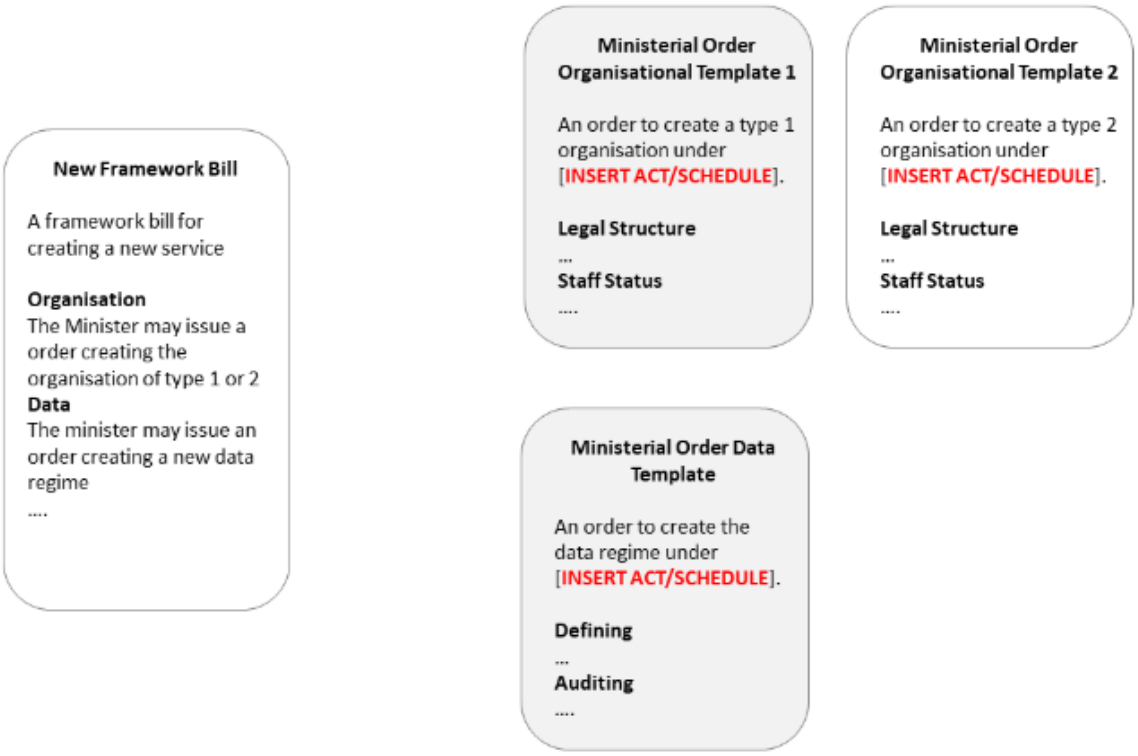
A bill and a follow up. Ministerial powers being used to publish secondary legislation as appropriate and normal/business-as-usual regulations.



The constitutional form of Statutory Instruments (often called Ministerial Orders or secondary legislation) is that there is a full debate at the point of granting the powers – and that the powers be as a constrained as much as possible.

If the broad structure of a delivery programme is understood it might be possible to push patterns into the secondary legislation (as a way for constraining the Ministerial powers and making them more amenable to oversight.

In this model we would make of the secondary legislation a catalogue (of organisational structure, of systems use, of data rules, etc, etc).



This is obviously just the simplest sketch of how that might work.

Summary

This is a complex subject. And it is a subject that no one person can be expected to be an expert in every part of. But I hope that I have shown that within each profession there has been an ongoing process of standardisation and componentisation – and that between professions there have been the systematic promotion of package components from the lower to the higher.

The challenge is to put all these discrete things together – and that starts with a shared vision. This working paper aims to give the interest parties a shared language for discussing it.

It also hammers home the point that moving in the direction of a lego state is primarily *a communications problem* and *a people problem*. Lots of moving parts – lots of professional and organisational boundaries to cross, lots of potential friction.

It is also important to remember that this way of thinking about components is only one side of the story – the civil servants/government/implementors side of it. There is another story which I have made no attempt to tell here – the constitutional side, the story of how parliament oversees, audits and keeps an eye on the lego state. Sufficient to the day is the evil thereof. That needs to be

left to future Working Papers and publications.

There are simple and obvious first steps – getting new checkpoints into existing process definitions – making the process an explicit one and not implicit. There can be no continuous improvement without explicitly.

Remember, the outcomes will come from growing the capability, there is no short cut to results.

Good luck!

Working Paper 4 – The remixable state

Version 1.0

Focus and refocus as a core competency

Introduction

What is a Remixable state?

A remixable state is one where the actions that the state and non-state civil society take can be composed to provide digitally-based services. And critically, these services can be composed in parallel, they can be run-down and replaced with new services. Digital services induce changes in behaviour among their users – changes which in turn exert pressure back on the digital service to change. This mutability is not an exception state, but the normal state – and we need to design a state that is mutable.

In a remixable world there would be an 'official' Universal Credit process online, and a Citizens Advice Bureau version, which would have integrated social work and social housing functionality.

Who are you?

You are a policy person, a service designer, a data architect, a delivery manager, a member of a project team, an operational manager, an elected representative. You are in government or opposition. You work at a thinktank, in the third sector, in parliament, or government, as a civil servant or political advisor or in front-line ops. You are interested in how to make the state more effective at delivering the policy goals you wish to see, as well as more efficient – delivering that benefit for lower costs.

WHY SHOULD YOU READ THIS?

Remixability is fundamentally about decentralisation – enabling lots of different groups – local government, civic society, other parts of central and devolved governments to incorporate core services into their daily workflows.

The paradox of decentralisation is that fully decentralised systems rely on a hard centralised substrate – and the building of that substrate is a common endeavour. Calling for decentralisation without participating in, understanding and internalising the necessary central services is a mugs game.

This paper helps you not be the mug.

Introduction and credits

During my research I had a very interesting interview with my colleague Laura Duarte who is a Senior Service Designer (Strategy) in the Strategic Design & Future Modelling team at Scot Gov – the conversation we had sparked off this set of sketches.

My analysis has been sharpened by disparate conversations on this topic from a very wide range of angles. First by Abby Innes's work[67] on the current UK fad for central planning which rhymes with the doomed cybernetics experiments of the late Soviet Era. And then suddenly in a recent conversation with Richard Pope about the limits of the doctrinal cry in Service Design about *meeting all the customer needs*.

At the heart of this proposal is the proposition that there is no panoptical view of the relationship of the citizen to the digital state. There is no single integrated set of user journeys – nor can there ever be, and most importantly nor should we ever chase that. Each service we design can only be partial. Not just for reasons of time or cost – boundaries and domains must be defined.

The goal then is terminate the user journeys in one system with composable end-points that lets someone else, inside the state or out, continue and extend them, remixing, reimagining and improving them.

The reality is that the modern state is a collective endeavour, covering civil servants in different departments, people in local government and the health service and also civil society and the 3rd sector. Lets embrace that reality.

By moving from a monolithic architecture to a composable, remixable one, one where we break down our existing services into platformed components, we can enable an ecosystem which will, in the round, improve the state for its (plural) citizens by offering new ways for an (individual) citizen to engage

[67] https://www.cambridge.org/core/books/late-soviet-britain/6C375F1A3E6007A1496A52F8BF313277

through multiple versions of the same user journey. It allows for more system death, by critique, and critically by subsumption. Imagine a world where, by and large, people could stop using the DWP Universal Credit online interface if the Citizen's Advice Bureau website, or app was simply better – even tho that is built on the same underlying components.

This is a technical model of the digital state that rhymes with the open source model that underpins the tech sector. Tech companies compete for eyeballs and customer dollars but co-operate in the development of shared software components and standards.

It's a language that technical civil servants intuitively understand but struggle to integrate into the standard models of the state and governance that structure the work of the civil service.

Constraint and composability

But before we talk government nonsense, lets talk Lego.

Lego demonstrates constraint and composability – a small set of common constraints enable a wide world of composability.

So what are the constraints?

- bricks have cross sections built from square units
- bricks have heights consisting of multiples of a unit (or 1/3rds of it)
- on the top a brick has either a decorative layer or one male connector per square unit of cross section
- on the bottom a brick has one female connector per square unit

In the wild, the vast majority of bricks meet these criteria with a small relative population of 'special' bricks where one or more of the requirements is slightly shaved or moulded.

Lego is a trade-off – you can build Michaelangelo's David with Lego but it won't be as pretty. But also an 8 year (with a credit card) can do it. You trade off aesthetics for composability – you can compose almost anything.

53 words of constraints enable a whole world of composability.

So let's look at another example of a composable system – the world wide web.

The compositional proposal here is that if every documentation system used the same simple interface then we could use the same browser to see them all and it would be great.

"Ha-ha", you say, "ya big liar, you have presented a composable system with a tiny constraint set and now you are trying to palm off a massive one on us and kid on it's the same".

And I won't deny it. The constraint set of the web was a whole lot bigger – a whole 10 times bigger – 528 words.

That was HTTP v0.9[68] – by the time the big old fancy HTTP 1.0 came out 5 years later in 1996 it had ballooned to just under 17,000 words. HTTP came with a child standard for HTML which built on previous work and was about 4,000 words in 1992. With the growth of the web, the HTML standard is now just over 620,000 words. And HTML has a couple of children - the Javascript[69] spec is kissing 280,000 words, and the fistful of CSS specs[70]? – best leave counting them as an exercise for the reader – but I wouldn't like to meet that burly crew on a dark night.

But all of them are but the child grown up. And the child grown up illustrates one of Gall's famous maxims perfectly:

A complex system that works is invariably found to have evolved from a simple system that works.

At first blush, the digital era offers us a host of opportunities for a better state with better outcomes, using better data to *focus*. But on reflection that is a chimera. States that can only be described as failed have demonstrated extra-ordinary focus – the North Korean ballistic missile programme being a case in point – famine cheek by jowl with bombs. True success in the digital age is a state that can *refocus* – having tackled something, can reconfigure itself, reallocate resources and take aim anew – a state that can remix its operations.

Where stands focus in the digital state? The point was well made by Laura Duarte regarding the Scottish Social Security programme – the large majority of users of social security are one-and-done, can come online, apply and use the social security system without human touch. But the users who are most in need, are most in distress, are most in poverty, are most stuck in cycles of deprivation, can't. The challenge of the focussed state is to concentrate resources and activities around those in most need and break long-running patterns – and having done that move on.

Let us conjure some constraints that might enable recomposability of state activity.

These constraints must be organisational, constraints on how organisations can compose their activities. Luckily there is a working compositional model, well established.

Back in the dawn of the internet age I was Chief Technical Architect at

---

[68] The HTTP Protocol As Implemented In W3

69 ECMA-262 - Ecma International (ecma-international.org)

70 CSS current work & how to participate (w3.org)

if.com – a UK internet bank. The bulk of the company had come from Direct Line Financial Services via Standard Life Bank and we were wrestling with the operational challenges of migrating the very successful branch-less call centre model of Direct Line to self-service online.

In order to minimise our technical systems build-out we improvised a role-based authentication model to enable our target audience to use our system. There is a trope in accessibility that there are no able-bodied people, just people who used to be disabled, being babies, and who have not yet become disabled again by dint of growing old.

If the problems that social security face now is adults disabled from using technology by chaotic life challenges – the problems we faced in 1999 was that our customers were babies. Veterans of the 1980's internet were rare as hens teeth – the internet was new and being explored for the first time.

To put it crudely our customers were a bunch of internet toddlers. We needed to be hybrid – offering direct services to our customers online, and a back-up call centre. But we also needed to support IFAs (Independent Financial Advisors) – and they were babies also – some of them could go online and apply on behalf of customers but some of them needed call centre support too.

So we ended up with a single browser-based banking system that could processes banking applications under four scenarios:

- direct by the customer (browser)
- by a staff member on behalf of a customer (via a browser in the call centre)
- direct by an IFA on behalf of a customer (browser)
- by a staff member on behalf of an IFA on behalf of a customer (via a browser in the call centre)

You know when you set up internet banking and you have a password and then also a long pass phrase where they ask you the 1st, 3rd and 8th letter when you phone up? That is you revealing a fraction of a password to a call centre person to enable them to log on behalf of you for that session without also getting enough to be able to log on as you when you are not there.

So can we turn this into a set of constraints that enable a composable state? Lets see. At the base of the digital state is getting as many transactions self-service as possible.

- Constraint 1: the citizen can perform a transaction directly

This base level of activity might easily cover 80% or more of user journeys but there are enough serious edge cases that it is wholly inadequate. Lets add some more constraints:

- Constraint 2: someone else can perform a transaction on behalf of a citizen
- Constraint 3: someone else can perform a transaction on behalf of a citizen at the request of a third party
- Constraint 4: a person transacting on behalf of a citizen may do so:
    - (1) at the request of the citizen
    - (2) as an employee of a state body
        - (i) as a condition of service
        - (ii) at the citizen's request
    - (3) under the supervision of the courts

The 4th constraint covers the use cases. (1) covers me doing tax for my Dad or applying for his driving license online. (2) covers calling a Social Security call centre, or a social worker arranging a benefit or housing application for a client, or a crisis team co-ordinating multi-agency responses. (3) covers power of attorney and wards of court.

These constraint addresses the remaining 20% but in too loose a manner, opening up plenty of room for abuse – so let's apply another:

- Constraint 5: transactions done on behalf of a citizen will be logged appropriately and checked for patterns of abuse in a manner consistent with the privacy and dignity of the citizen

These constraints in themselves don't ensure composability. We need a couple more

- Constraint 6: there will be a single source of identity and authorisation for both citizens acting directly and people acting on their behalf
- Constraint 7: services offered by an app or webpage shall be exposed as a published, documented and appropriately managed API

These are the technical kickers that enable composability.

That is a suitably short constraint set in 143 words.

In the next section I will test the constraints for organisational flexibility. In Section 6 I will consider the technology implications of them. Section 7 will look at the implications for law and law making and how it maps to the technical architecture, Section 8 will revisit the theory of the state sketched here.

Stress testing the constraints

In the previous section I outlined 7 constraints that could be used to build a remixable, refocusable digital state.

This section will focus on the first four constraints – the constraints on

roles:

- Constraint 1: the citizen can perform a transaction directly
- Constraint 2: someone else can perform a transaction on behalf of a citizen
- Constraint 3: someone else can perform a transaction on behalf of a citizen at the request of a third party
- Constraint 4: a person transacting on behalf of a citizen may do so:
    - (1) at the request of the citizen
    - (2) as an employee of a state body
        - (i) as a condition of service
        - (ii) at the citizen's request
    - (3) under the supervision of the courts

The first test of these constraints is to test if they really are the smallest set for a digital state.

Every function of the state exists for the benefit of all citizens and a small number of citizens are incapacitated because of illness, age or disability and are simply unable to self-serve. These we cannot abolish.

We can reduce the set of constraints only by abolishing self-service and demanding all services are mediated by a civil servant – which would abolish the very idea of a digital state.

2 is the minimum number of people who need to perform every citizen-facing task. Implementing a single-role system is a false simplification – the other will need to be performed by a back office manual process anyway.

The double delegation of constraint 3 is required because we cannot assume that a person with delegated authority can use direct services either:

```
Citizen -> Service
Citizen -> Call Centre Operator -> Service
Citizen -> Delegated Person -> Service
Citizen -> Delegated Person -> Call Centre Operator -> Service
```

This is a direct cut'n'shut of the banking model:

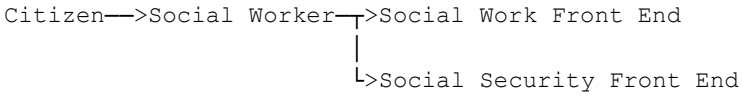
```
Customer -> Banking System
Customer -> Call Centre Operator -> Banking System
Customer -> IFA -> Banking System
Customer -> IFA -> Call Centre Operator -> Banking System
```

In banking delegated authority is implied rather than mandated by law – only regulated Independent Financial Advisers can sell products to customers – which indirectly defines delegation. It is important to understand from a systemic point that the bank is the organisation that maintains the subset of

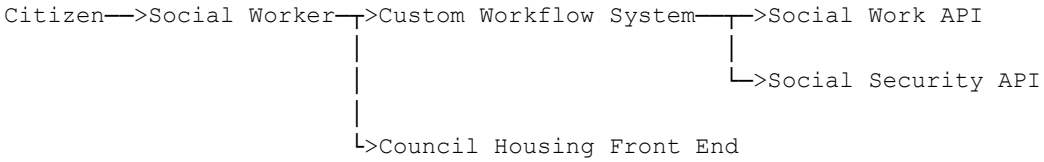
IFA's able to use the system.

(There is a secondary delegation issue where permission is granted not to Jinty McGinty but to the State Body that Jinty McGinty works for but that's properly a technical and not an organisational issue and will be dealt with later.)

So we know that there are not technical barriers to doing it. Where it differs is in its compositional nature:

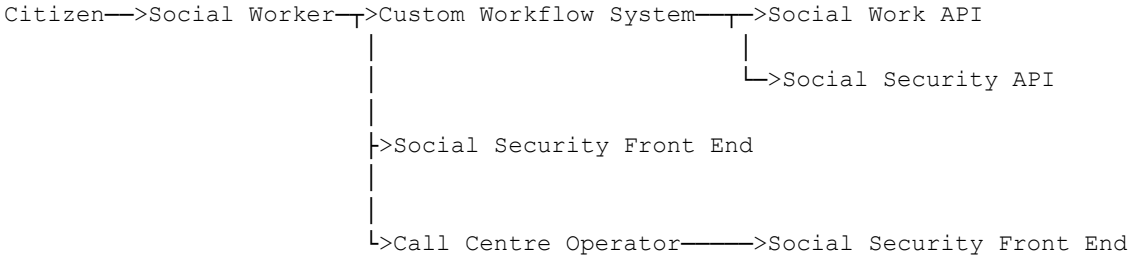


And the constraints set enable composition of systems. If services (web sites or apps) are written to work with APIs then custom workflow and case management systems can interposed between the delegated person and the basic underlying systems:



And this system composition is partial – if you plan to incorporate a common transaction into a custom workflow you are not obliged to integrate all the transactions a particular system offers – you can pick and choose – for the high volume critical social security activities the social worker can do them within their case management system, for the obscure and occasional one they can drop out to using the social work online system – or even call the social services call centre.

We can see this double delegation:



The constraints don't impose any activities on a system implementing them.

The purpose of this construct of using constraint is to decouple systems that provide services from organisations of civil servants who use those services to help citizens and achieve policy outcomes. It is important to understand what is

decoupled.

The organisation offering a service to the citizen has control over:

- which transactions are self-service
- whether a transaction can be can delegated in either an open or closed manner
    - open delegation – the citizen can choose a family member, friend, someone from the church, anyone
    - closed delegation – the citizen can choose to delegate someone from a provided list – a social worker, a health visitor, a district nurse

The law will state whether a service *can* be delegated to a class of people – but operational policy will determine *if it is*. Clearly some state services cannot be made self-service – probation service have a mandatory human stage.

These constraints are designed to decouple the *ability to delegate* which is common to all systems from *the nature and purpose of delegation* which is specific to a particular state body (at whatever constitutional level – national or sub-national/local).

In the next section I will look at the technical issues.

|Identity and authorisation, centralisation and decentralisation

Going back to our original decoupled system – the internet – it is important to understand that decentralisation is always allied with centralisation. You can't have one without the other.

On the internet any browser – and with over 4 billion people using the web that's a lot of browsers – can talk to any of the 400 million active (or 2 billion reachable) websites. This is a genuinely decoupled system. Adding new websites or browsers to the web is a trivial task.

But it is a constrained task. Every device – whether a browser or server needs a unique IP (Internet Protocol) address (as I write this mine is 164.134.2.12). And most websites need a domain name bound to their IP address (and if they are encrypted a signed encryption certificate identifying them as well).

There is a single central authority ICANN (the Internet Corporation for Assigned Names and Numbers) which allocates IP addresses to network providers (who give them to you) and which maintains the DNS (Domain Name Services) system that converts names like gov.scot into the number 13.248.154.230 that is used to connect browsers and servers.

Your web browser comes bundled with encryption keys from trusted

Certificate Authorities which are the keystone of a web of trust that means when you look at a webpage from https://somedomain.gov.scot it actually is from that website – every secure website pays one of these central authorities to sign their web address cryptographically.

The web built on top of the internet. The pattern of IP addresses, their distribution, routing and use was developed up to 1973 and that centralised management structure was built. For 10 years all machines were addressed by number which users had to type it and then DNS was designed to let people use easier to remember domain names.

(They used to say every child could remember their parents co-op number, but it's the SuperJanet address of the RAL cluster from the early 80s for me +j00000000000002– the internet didn't arrive in the UK until 1986 or 1987.)

The world wide web built on top of these with its first server in 1990 in Switzerland and its second in the US in 1990 – exploding to 500 in 1993. By 1995 https – the encrypted web was emerging.

Identity and sign-on is the centralised component of this constraint-based approach to a remixable state.

It is important to talk about it, because the centralised services that support the web are invisible because they work. When a website goes down, some internet users lose some services. If DNS stopped working the web would be bricked world wide. And the centralised services can be massive – any one of the 4 billion users of the internet can easily generate several hundred to several thousands DNS looks up a day.

The centralised component of a massively decentralised system like the web needs to do only a small handful of things, but it must do them a lot and do them invisibly well.

The way they do them well is by delegating. ICANN owns all the numbers but sold the rights to ranges of them. The original allocation in RFC 790 makes for interesting reading. UCL[71] in London had the rights to 011.xxx.xxx.xxx. UCL could then allocated 011.1xx.xxx.xxx to you and 011.2xx.xx.xxx to me and we could delegate down the numbers.

Likewise ICANN allocates the right to sell .com domains to various resellers. They sold vixo.com to me and I can create subdomains like help.vixo.com and email addresses like gordon@vixo.com at will.

I tell the people I bought the domain name off what IP address it goes with, they tell ICANN, ICANN tells everybody including Google, and your phone can

[71] Yes, University College London got 1/256 of the entire internet – a whopping 16.7 million IP addresses

then find my webpages.

The whole decentralised web is underpinned by a centralised web of trust which has to be managed, maintained and policed against wrong 'uns. If you can poison the certificate chain of trust and the DNS chain of trust at the same time you can steal money and information in huge amounts.

Constraints 5 and 6 deal define the centralised component of the remixable state.

Lets look at the latter first:

- Constraint 6: identity and authorisation for both citizens acting directly and people acting on their behalf

The 'me' of the citizen on one system must be the 'me' of them on another – the 'me' of the social worker' must be their 'me' on the social security systems and hospital booking systems.

But if the `me` being delegated is not Jinty McGinty but `a social worker` then we need a delegation mechanism – a web of trust. Jinty McGinty can't be made and unmade as social worker by a central system. Systems participating in the identity system need control of their own organisational definitions.

The constraints that enable decentralisation can only be built on top of another web of trust – an identity chain of trust that opens access to a wide range of state systems. A remixable state with benefits and prescriptions and other services could be exploited by criminals creating farms of vulnerable people.

- Constraint 5: transactions done on behalf of a citizen will be logged appropriately and checked for patterns of abuse in a manner consistent with the privacy and dignity of the citizen

There is a pattern from banking that addresses this – again with a central authority. It might seem odd but banks care more about where you live and if you have access to your phone than who you are. Go to an online bank and ask for a new chequebook or card and they will send it to you. Well they won't send it to you, bad guy, they will send it to the address of the real person (addresses are harder to spoof than people on account of being buildings and kinda hard to move, or disguise). Ask to send money and they will send a code to your phone – if you can prove you have your phone, they think you are you.

The big theft vector back in ye olden days was loan fraud. If, and it was big if, you could 'steal' an address to get post and get a legit identity with a credit score associated with it, you could apply for a load of loans for motorbikes and holidays and gazebos all sitting between £3,000 and £5,000.

The solution to this was a system called Hunter. Hunter was owned by all

the banks collectively but worked for none of them directly. When someone applied for credit a copy of their application, name, address, amount, term, was sent off to Hunter. Hunter didn't share Barclay's applications with the Royal Bank and vice versa. All Hunter did was look for patterns. One house, 5 short term loans, wrong 'un. Hunter didn't take action, its just notified the banks. A remixable state needs a hunter too, that gets notified when permissions are granted to people and looks for benefit farmers and aggrebent coppers collecting vulnerable benefit applicants and other miscreants.

But like the internet, distributed identity systems and single sign-ons have been created, are well known and understood and have rich and detailed threat models that enable them to be policed. The task of building the Scottish single-sign on is well underway. The technical aspects are not novel, no moonshot, nothing Monte Carlo or Bust! about it.

The last and final technical constraint is the runt of the litter, the weediest of the lot:

- Constraint 7: services offered by an app or webpage shall be exposed as a published, documented and appropriately managed api

This is basically an injunction to do it properly. There are good ways and bad ways to build websites and applications – and the good ways lend themselves to making services that are remixable – and the bad ones don't. The previous 6 constraints make it possible to create a remixable state, but this one actually makes it happen, it turns theoretical options into practical ones.

The next section will look at the legal architecture and its relation to technical aspects of this.

Technical and legal architectures

State servants, civil servants, people who work in the health service or for a local authority are rightly limited in what they can do by law. The state is a behemoth, and unchained a tyranny. So if we want agents of the state to be proactive and structure their work to focus on delivering social policy and social benefits then we need to actually create a legal structure that can do that.

And if our goal is that the state flexs, restructures and reforms in a state of constant remixing to use digital technology to better achieve the goals of the government then we need to create a legal structure that can do that too. But as we free them to do that, we need to constrain them too. Power granted to do the good thing cannot be repurposed to do the bad thing. Software and its development must be moulded to the constitutional order and the rule of law

and not the other way round.

At the heart of this is the act of delegation:

- whether a transaction can be can delegated in either an open or closed manner
    - open delegation – the citizen can choose a family member, friend, someone from the church, anyone
    - closed delegation – the citizen can choose to delegate someone from a provided list – a social worker, a health visitor, a district nurse

The law has three separate cycles of iteration at its heart – and one of the key legal tasks will be aligning the desire to delegate with the ability to.

The three cycles are:

- Primary legislation
- Secondary legislation
- Operational policy

Primary legislation is 18 months minimum from bright spark to an oven ready bill of a quality to present to parliament. Once its oven ready it needs a slot in the legislative timetable which might be 6, 12, 18 months out – so the slowest of the three.

Secondary legislation is a lot quicker, but it needs ministerial time, and it needs to touch parliament too, to varying degrees. Scotland has 3 types. The negative process is a mere breath if the parliament doesn't take a vote and vote No the order becomes law. The affirmative is a feather touch – the parliament must positively vote yes. The super-affirmative a gentle prod, but a parliamentary slot nonetheless followed by a yes/no vote. At the very least the Delegated Powers Committee needs to scuff the hair and pinch the cheek of every piece of secondary legislation.

Operational policy is the quickest. Within the broad framework laid down in law the relevant elements of the civil service leadership can set policy.

But whilst software developers and service designers might care about speed of iteration over all things, that's not going to be the opinion of parliamentarians – and quite right too.

There are some delegations that parliamentarians will be hard pressed to care about. My ability to get my personal trainer to book gym slots at Portobello baths. That's clearly a matter for the manager of the swim centre – operational policy – almost certainly open delegation.

There are some delegations that parliamentarians will care about a lot – interaction with the probation system. That will be nailed down in primary

legislation – and forbidden.

And in-between there will be a mass of closed delegations – things that can be delegated to your social worker or health visitor or community policeman, but not your pal from the pub.

In system or service design terms we are delegating user-journeys: a citizen wants to book a slot at the gym.

The process of defining, sorting, aggregating, improving and making available user-journeys is the actual meat of digital transformation – what we are trying to improve in transforming the state's capabilities. And it's at the core of a remixable state – the act of making user-journeys available and remixable will apply pressure for change on the service offering the user-journey. A service that starts as raw user-facing might find itself supplanted by a separate app that subsumes its API and combines it with something else.

The language of the state is a language of patterns. When you are preparing a Bill to go to parliament there are pattern books to help you shape it. When you create delegated powers for secondary legislation there is a pattern book to help you describe them. The people who draft legislation have pattern books that describe what bills and orders must contain, their shape, contents and language. Policy specialists follow their processes and templates. On the delivery side, service design is a language of patterns: personae, journeys. Software developers checkpoint themselves with tick-lists and deployment checkpoints, patterns all. APIs too.

The big task in the IKEAfication of the development of the digital state is the alignment of the pattern boundaries down the entire stack from end to end.

It's kinda like a Russian doll – some APIs representing a user journey *inside* a software deployment *inside* a Service Level Agreement *inside* a monitored service *inside* an operational agency acting *under* law, parliament and government. But at least in traditional Russian dolls, it's the same doll just different sizes – each of our dolls takes a different form – the outside is one shape, the inside a different shape determined by the nature of the thing it contains. We need to be able to unpack the doll set, replace, change or upgrade the innermost doll and repack it without having to break, fix or scour out any of the intermediate dolls.

Almost everybody in the process understands the Russian doll nature of the beast *implicitly* -we pass a social security bill we are going to get a social security system, obviously. The challenge here is to make it *explicit*.

An example would be where we place delegation and why:

Iterative Cycle	Goal	Example
Primary	To forbid	• Probation services shall not be delegated
Secondary	To permit	• Social security powers may be delegated to health visitors, district nurses, midwives and social workers for co-ordination • Social security transactions may be delegated to guardians, family members under these circumstances
Operational Policy	To toggle on/off	• User Journey 1 on for health visitors, district nurses and midwives • User Journey 2 on for social workers • User Journey 3 on for friends • User Journey 4 on for registered family guardians

To allocate these things properly all the actors will need to understand what they are allocating. Ministers, parliamentarians, policy developers, organisation and service designers, coders, testers and operational managers will need to understand the role of the constraints and the way the state needs to work in the new world.

The various pattern books need to be aligned, they need to rhyme, and their users need to understand that they rhyme and why its important.

But there is another critical legal element that we haven't talked about yet.

Back in Section 4 we talked about the constraints that define Lego:

- bricks have cross sections built from square units
- bricks have heights consisting of multiples of a unit (or 1/3rds of it)
- on the top a brick has either a decorative layer or one male connector per square unit of cross section
- on the bottom a brick has one female connector per square unit

This is only a partial description of them – there are deeper, fundamental technical ones:

- bricks are built horizontally in units of 8mm, but the sides are pulled back by 0.1mm to give flex room

- the height is 9.6mm or 1/3 of that, 3.2mm
- the connector holes are 4.8mm wide but the studs are bigger, 4.87mm or 4.88mm to force the brick to clutch and grip (and the brick walls to pop a bit into the flex room left

When we look in detail at how the law specifies computer systems we see elements that are both the floor – built into databases as records and tables – and the roof, defined explicitly in laws. Essentially they are 9 elements of data management that are expressed in law.

There are 4 core elements of data management, how data is:

- created
- read
- updated
- deleted

These map to the familiar CRUD actions that are baked into the SQL database query language – though properly they are here at a higher, human-transactional level.

On top of that are a set of operations at a slightly higher level that pertain to the operation of the organisations that execute create, read, update or delete on data:

- definition – what the data collected it
- audit – how data is maintained, inspected, reviewed, weeded and otherwise managed
- recourse – the appeals process to get on or off a list, by which creation, updating and deletion are triggered
- partition – how the data is partitioned, for example to health boards, local authorities or in a central national system
- timeousness – how and when the data is refreshed or reconfirmed

You can read about these operations in more details in this blog post on the Digital Policy SubStack.

The key point about partition tho is that if the same data can be safely and correctly partitioned among the 32 local authorities and 14 health boards of Scotland it could be safely partitioned if Scotland had the 300 local authorities that Finland have – and those have health responsibilities too.

To get composability these elements need to be brought into line. The police 'create' must have the same definition as the health service 'create' and so on for all the actions and all the statutory bodies.

One part of this is a codification and standardisation process – and there is an obvious vehicle for it in Scotland – the Interpretation and Legislative Reform

(Scotland) Act 2010. This act is one of the great pattern books of the Scottish state – a lexicon of terms that determine their meaning in law. It is a mechanism to IKEAify both statute and case law. If the interpretation defines a `keelie` in law then (in the first approximation) all mentions of `keelie` in all laws refer to the same thing and (in the first approximation) case law on keelies is harmonised.

But there is a further complication – computer systems are not justiciable in themselves – code is not law. The actions of the state in using them, via its organisations, its agencies, its civil servants, is justiciable.

State servants are not interchangeable. This sketch of composability treats health visitors, policemen, my brother, my kirk minister as interchangeable people that we can swap in and out. Different people have different duties and obligations, and these bear down on their permission to do seemingly trivial things like `read` citizen's data.

The operations we are talking about making remixable are, rightly, shrouded in human rights protections based on the dignity and right to privacy of the citizen. The Russian dolls that give the scheme its flexibility might break down due to privacy and dignity violations unless care is taken.

One of the big draws at Pompeii are the bodies of people engulfed by the pyroclastic flow. Except of course there are no bodies, there is only the lava shell surrounding a void where their soft body has leached away.

In The Age Of Surveillance Capitalism Shoshana Zuboff has documented how we encase ourselves in the pyroclastic flow of the internet, building a meta data shell of GPS pings, and purchasing information, of check-ins and image meta-data. Without knowing us, by simply surveilling our meta-data big tech can know about us, and turn our ash shadows into commercial commodities. We are all Pompeiians now. Facebook and LinkedIn both create shadow accounts for people not yet registered. Even if you avoid their services they will infer you and conjure you and your relationships into existence.

If Alice posts a photo with you in it and so does Bob then a shadow you who knows them both is born.

A remixable state is a pyroclastic state – you and your interactions with the state will create a shadow you – and any proposal to build it will rightly meet with serious concern. We need a legal and privacy regimes that covers both the citizen and their encasing data shadow. This is the paradox of the effective state: the more freedom we give agents of the state to act, the more tightly we must constrain them in their actions. It is a balance that has to be found.

In the next section I will revisit the theory of the state.

Theory of the state

Now we have looked at the constraints and their construction from all sides, let us return to the purpose of all this – the sort of state that it would enable.

The government remains the director, the general, the strategic conductor of affairs. Direction is set by the people in an election.

Government and parliament set the rules of the game: who can do what, what resources they have, what outcomes they should seek. It establishes who can work with whom. Social work can work with social security, health and housing. Local authorities and health boards alongside national agencies. And it sets the rules under which they can work. And then it gets out of the way.

The digital age brings complexity to the heart of the state - digital systems, unlike paper administration, are opaque and hard to reason about, and critically they engender behavioural and cultural changes in their uses and mutate the citizens relationship to the state. They are also mutable and in a state of constant flux. This requires not only new ways of working, but a new way of thinking about the relationship between the government and civil servants.

The front-line troops, provided they conform to the rules of the game (common identification, API publication, strategic directives on sharing) can just get on with it.

They can choose their own technologies, their own development methodologies, their own team structure and organisation, their own workflows. They can mould the state to the circumstances of different communities, urban/rural, highland/lowland, island/mainland, rich/poor, big city/hinterland, drug ravaged/drug free.

Civil servants can take the initiative, can reconfigure and improve their own work, can tear down and rebuild bits of the state and adjust and readjust how it works.

But if we look at the legal order we see that the ability of the state to act is governed by two legislative powers – power granted to do certain things in one type of Act and financial powers to finance it granted in another type of Act and subject to formal financial management.

Parliament grants both the power to act, and the money, the means to take the action, and grants them separately.

When we talk about a refocussable state – that focus too must take two forms – the power and the money. Here we run smack into one of the realities of the British state – both devolved and retained. The UK is one of the most centralised countries in the world, with one of the most rigid and uniform

applications of tax rates. The ability of local communities as stewards of local economies to raise and lower taxes, to set priorities is limited. Enabling joint working between parliaments and local authorities by implementing technical systems, must be matched by flexibility and funding. The parliaments need to share political responsibilities with other elected bodies to determine priorities – a refocussed state in Glasgow should look different to one in Inverness. But the parliaments should also share responsibility for putting money behind the push – and that involves devolving control over the tax base. Going back to 1983 when local government raised 85% of its income locally and got 15% from government is not blood red Bolshevism its not even normal-for-Switzerland.

Since the 1920s legislatures in the UK have treated money like a special thing – the government of the day can no longer mark its own financial homework. Assets and liabilities and all properly accounted for.

Well data collected is an asset – as we saw in Covid. As Robert Colvile pointed out in the Times the great successes of the Covid response were built on existing databases and operational systems. Employees furlough was managed by adapting the PAYE system to make tax flow backwards. The self-employed who lacked such a system were hung out to dry. National vaccination and shielding programmes were conjured on top of health databases.

And computer code famously is a liability – every line must be maintained by someone.

But money's value is in the lump. The 50 quid in my wallet, can be chucked in with the 300 under the bed and the 3 grand in my post office account. Not so data – with data the value is the structure, the consistency, the reuse and the conjoining.

The example of bespoke legal management of money flows in parliamentary terms would suggest that we need a bespoke legal management of data and state computer system.s

The differences between data as an asset and lump money would suggest that simply copying the financial legal process would be a mistake – we must develop new parliamentary instruments to manage our digital assets.

The constraints of HTML 0.9 gave birth to a new way of living. Similarly adoption of these seven constraints cannot be seen as a bloodless, technocratic solution. They raise sharp and vivid concerns about the operation of the state. Rightly they should be the subject of vigorous scrutiny.

There are two elements to a constitution. Scotland is familiar enough with the high constitution – should there be a devolved parliament? should Scotland become independent? The Scotland Act and the White Paper on Independence.

But we give much less thought about the low constitution – the Standing Orders of the parliament – the powers and functions of local government. The sketch of a remixable state outlined here is re-ordering of the low constitution and needs to be considered as such.

Conclusion

This is a sketch of a bold proposal. The first step on progressing it is to build a state that is capable of doing it. It is a strategic goal that can inform the construction of capability.

The critical thing is that a project like this be done in public – with a project board that rhymes with normal internet governance – open boards, published roadmaps, Request For Comments and all the rest.

Working Paper 5 – Law reform for data

Version 1.1

Providing legal clarity

Introduction

Why Law Reform for Data?

Data is the foundation of the digital state – and a highly effective digital state must be built on clarity about data. Clarity about what it is, where it is, who may use and for what purpose, how it is to be used. But this clarity conceals wider and deeper problems. The law can only determine the *will* to use data, it cannot determine the *means*.

An example would be the various registers[72] held by Registers of Scotland. The law *wills* that the registers[73], be available to the public in a joined up manner. However the *means* for that to happen are absent – despite parliamentary and ministerial commitments and support going back a decade. The registers are available, but remain unjoined[74].

The second element that this proposal addresses is the rule of law. Working Paper No 1.1 – *Data and the rule of law* focussed on the operation of the rule of law *when a case has got to court*. This paper also discussed understanding data in order to be able to mount a case – *the process of trying to get to court*.

Separation of powers applies to data. The legislature writes law, but the courts, and the courts alone, interpret and apply it. The legislature can define a *thing*, and a government department can *model* that thing in a data structure. But ultimately the model is subject to judicial review (this model violates my privacy, that model discriminates against you, yon model impacts her human rights).

The institutional architecture of the state must enable citizens and their advocates to make the case that a particular data model (and it use in process) violates the law. Obfuscation does not provide a veneer of legality.

There is a relationship between law and technical standards. Processes and procedures which create law and those which create technical standards need to be kept in sync and in harmony. It must be possible to reason both about the legal and technical use of data – and that requires that the legal and technical statements about data be standardised and they be unambiguously findable.

---

[72] https://www.ros.gov.uk/our-registers

[73] The Land Register, the Crofting Register, the Register of SSSIs, the Register of Applications by Community Bodies to Buy Land, the Register of Community Interests in Land, the Scottish Landlord Register, the Scottish Letting Agent Register

[74] https://davidhumeinstitute.org/research-1/2023/2/27/briefing-paper-scotlands-land-information-system-what-is-it-and-why-it-matters

Data is a key building block of a Lego state[75] and it is worth thinking about law reform using the analogy of toys.

The law can state that the child must be able to build things from their basic elements – but it should not state if those elements should be Lego, Duplo, K'nex or Meccano.

But the design of Lego, Duplo, K'Nex or Meccano all embody standards-as-law – a particular thing is either 'legal' Lego or it isn't, and if it isn't, it can't be used to build a Lego model – its just a standalone thing – its utility comes from its conformance with standards-as-law.

These twin parliaments – the parliament of laws and the parliament of standards – must be designed to work with each other. The parliament of standards seems, at first blush, to be a mere bagatelle – some techie nonsense – and yet it will make decisions that the state will be living with for a hundred years or more. The Register of Sasines has been with us for 406 years.

The use of the phrase 'a parliament of standards' is quite deliberate. A parliament is an organisational form that maximises consent: and, in particular, creating losers' consent. And historically parliaments have done that for different communities at different times.

The old Thrie Estaitis of Scotland were:

- the first estaite – prelates
- the second estaite – nobles
- the third estaite – burgh commissioners

Powers in the land all. After Union more communities were brought it – the big city rate payers (or male rate payers rather) in 1832, then in 1918 returning soldiers and older women, before all citizens in 1928 and reaching it's current form in Scotland of all residents in 2020 with the Scottish Elections (Franchise and Representation) Act[76].

The job of the standards body is to maximise consensus and to generate losers' consent (using loser in a fairly loose sense here) within its constituency. That is not the citizenry, or even all civil servants, but the critical technical staff who need to ensure its 'laws' are implemented.

The Thrie Estaitis of the digital world are:

- the first estaite – the data teams from all the departments and NDPBs
- the second estaite – the coders from all the departments and NDPBs
- the third estaite – the designers from all the departments and NDPBs

Like their predecessors these good burghers need to have their voice heard,

---

[75] See Working Paper No 3 – *The Lego state*

76 https://www.legislation.gov.uk/asp/2020/6/contents/enacted

and to see off unreasonable and unpayable demands from their sovereign overload and wanna-be despot, the power, in extremis, to say “that won’t work” – speaking truth to power in the civil service argot.

This paper focuses on the particular mechanisms we need to service the needs of the first estate – the data specialists, and their obligations under the rule of law.

One of the superpowers of the big tech companies is their ability to automate procedures – and it is a superpower that the state must develop.

On first blush this paper proposes putting bureaucratic obligations on data specialists – an obligation to publish detailed data descriptions. But data systems are already self-describing - they have to be for the software developers working on them to do their job. The developers need to reason about data. The challenge for the state is to develop tooling that automates standards – that the obligations on data and discovery to be moved from the civil servants to the systems themselves. A properly structured data systems and API can cheerfully describe itself 10,000 times an hour at negligible cost.

Another reason data law reform matters so much is because of the implications of the data zip. There are a series of causal teeth that engage like, well, a zip.

If two datasets with the same definition (think person data or place data) have the same definitions in each of the 9 categories of Section 5 – then it follows that these two datasets can be merged into one.

If two datasets can be merged it implies that business processes that operate on them can be merged and consolidated (this reduces compliance costs for citizens and organisations and is a net win).

If two sets of business processes are consolidated, then it implies that the organisational units that execute these processes can be consolidated (this generates cash savings and reduces government expenditure).

```
        many definitions          -> one definition
enables many database instances   -> one database instance
enables many business processes   -> one business process
enables many organisational units -> one organisational unit
```

Zipping one lets you zip the next.

This implies that work that starts with consolidation of data definitions ends with machinery of government (MoG) changes. Traditionally MoG is a prerogative of the First Minister (or Prime Minister at Westminster). This is a non-trivial change to the way the state works.

WHO ARE YOU?

This is of direct interest to you if you are a data or technical specialist, or parliamentary counsel, trying to figure out the best representation of data in law. But is should also be of interest if you are an MSP, Minister or Spad, a think-tanker or policy person, somebody in delivery trying to build out or drive joined-up government.

WHY SHOULD YOU READ THIS?

You should read this to help understand how we put in place the basic hygienic administrative measures that will unlock change in the structure and function of digital government whilst protecting the rule of law and respecting the separation of powers.

Acknowledgements

This paper greatly benefited from conversations with Richard Pope, formerly of GDS and the author of the Government As A Platform Playbook[77] and Gavin Freeguard, formerly of the Institute for Government.

Revision Notes

The original version of this document referred to the Digital Reform Office and the Digital Scrutiny And Audit Commission – they have been renamed the Digital Services Reform Office and the Digital Services Scrutiny And Audit Commission to better reflect their remit.

Data in context

WHY IS DATA SO IMPORTANT?

Data is the foundation of digital systems - good hygiene in the management of data is fundamental to high quality digital public services – and data management is poorly handled in the public sector.

This paper focusses on one particular aspect of the story reasoning about data, and it has Working Paper 1.1 – *Data and the rule of law* as a companion piece – both addressing different legal moments.

A proper data foundation is necessary to be able to build out the state as platforms[78].

But a bigger picture is described in Working Paper 3 – *The Lego state* which this paper builds on and Working Paper 4 – *The remixable state*. Without the

[77] https://richardpope.org/publications/2019/11/01/playbook-government-as-a-platform/

[78] see for instance https://richardpope.org/publications/2019/11/01/playbook-government-as-a-platform/

ability to reason about data, the state lacks the capability to improve itself, to reduce the burdens it places on citizens and corporations which have a direct economic costs, to reconfigure itself as circumstances change, both changes within citizens that come with ageing, and changes in society that come from the success of failure of state actions.

Process and procedures, what the state does, are actions on data. Changing data representations enables consolidation of process and elimination of work.

Governing a state without being able to reason about data is like trying to play chess without a board. Moves are more theatrical than purposeful.

What this paper doesn't do.

Data is exceptionally long-lived, data decisions are 100-year decisions and it is critical that there is appropriate constitutional oversight over them.

And data is going to be a coming battleground in politics – who can see what, who can do what with it, how the government uses and shares data are all going to be hot topics for the foreseeable future.

These larger political questions are outwith the purview of this paper.

But these competing issues of citizens' rights and government utility are made more tractable by a data landscape that can be reasoned about.

Why is it difficult to do this?

In *Practical Legislation* Thring[79] wrote:

> *I will venture to affirm that what is commonly called the technical part of legislation is incomparably more difficult than what may be called the ethical. In other words, it is far easier to conceive justly what would be useful law, than so to construct that same law that it may accomplish the design of the law giver.*

There are two sorts of 'law' that apply to our data – from the parliament of law and the parliament of standards – and they differ in their application.

It is important to understand that these are formally different things – to use the language of semiotics the law (mostly) addresses the Signified and the standards only address the Signifier.

To the parliament and the courts I am the corporeal being, *Gordon Guthrie*, flesh and blood and the Signified.

As far as the data systems are concerned I am just *Firstname: Gordon, Surname: Guthrie, etc* – a Signifier.

---

[79] Introduction to the 2nd edition – Practical Legislation, The Composition and Language of Acts of Parliament and Business Documents – 3rd edition, Luath Press, 2019

Notice that there is one Signified but many Signifiers – this is not an abstruse point as it speaks to the rule of law, particularly when the Signified are human persons – and is something that I will return to later.

	Law	Standards
Things vs models	Cares about the Signified, the thing itself. About the Signifier it only cares that it is small enough for the purpose (You don't need to disclose your HIV status to buy a TV license...)	Cares about the Signifier, the model and its ability to capture the relevant attributes of the Signified required to perform some process
Joins	Cares about joining Signifiers that refer to the same Signified – *Parking Gordon* and *Health Gordon* and *Sexuality Gordon*. Focus on yes/no permission to join.	Cares about technical ability to join, and not permission. If *Parking Gordon*, *Health Gordon* and *Sexuality Gordon* have a common identifier (ID card number) they can be joined technically.
Prohibition vs Enablement	prohibits or approves joins.	Cares about *enabling* joins whether they are or are not prohibited. Deals with technical prerequisites – like common indexes, citizen id numbers, etc, etc
Powers of Judgement	Separation of powers - the legislator doesn't decide what the law means, the judge does	The standards setter can sit as judge, jury and executioner in their own court
Localisation of effect	Rule of law, general acts (like GDPR) and case law means legal effect is never entirely localised – comes from a range of sources	Standards are entirely localised – these standards and these standards alone apply to this data repository

It may seem superfluous to say, but the 'parliament of standards' is in no way an equal of the parliament of laws but subordinate to it. Parliaments of standards are technocratic and not democratic.

For different entities (people, companies, properties) there are historic ways of referring to different classes of Signified's in law.

The translation of these into legally acceptable Signifiers is currently a bespoke and somewhat arcane practice – the Business Analysts sit down with

the lawyers and unpick the legal requirements. A process of gap analysis then surfaces missing requirements. There is a back and forth between the operational team and the technical/design team brokered by the Business Analysts and blessed by the legal team. Code is cut.

It is important to note that the two cannot ever be cleanly separated – the principle of minimum sharing of data impacts the shape of the Signifier that we represent the Signified with in our computer systems.

A state servant can propose a particular data structure as representing a person or other legal entity, and someone with standing can legally challenge that and a judge will either bless the data structure and dismiss the case, or declare that it doesn't conform to law and send the state civil servant back to the drawing board. The final adjudicator of state data structures is the court[80].

Critically the various components are smeared across multiple places, a bit of law, some national standards, some organisational standards, some team standards, some developers personal way of doing things.

For many state computer systems it is not immediately apparent what law grants the power for their operation – and this is particularly true for systems that implement non-functional requirements – shared logins for example.

There is a also a tension between the standards work and legal work. If it is technically easier to join datasets there will be political pressure to do so, to generate 'efficiency'. It is important to remember that 'efficiency', 'liberty' and 'privacy' don't naturally align.

There is an apparent paradox here. To maximise reconfigurablility and reduce costs the standards process should lead to any data source being simply consumable by any other system across the entire public sector. The *means* to share data should be as wide and as general as possible. By contrast the drive to standardise the law around data management is driven by a desire to enable the minimum reuse of data proportionate to the legal requirements. The *will* to share data should be as locked down and restricted as possible.

The police do not need and should not get access to *Health Gordon* and *Sexuality Gordon* to deal with *Parking Gordon*.

If the data that various state systems holds is opaque, and the legal basis for

[80] This is in essence the same argument that is made in Working Paper 2 – *Rules as code* about code annotations not being justiciable and that tests generated from them can only demonstrate that a system violates the law but never that it conforms to it. There is no format of law and standards that can ensure state data systems *a priori* conform to the law. The goal is to eliminate egregious violations and make adjudication and justice as simple and painless as possible.

them is opaque and the legal basis for joining them one to another is opaque, then access to justice is impaired and the rule of law is eroded.

The purpose of this proposal is both to drive efficiency, more effective joined up government and also embed the rule of law in computer systems. Clearer processes for converting Signified into Signifiers, publication of data schemas so the data held by the state is open, a mapping of systems to powers that includes the legal basis of joining them up, all these things are necessary if the citizen is to be able to challenge the state in court, if state administration is to be subject to the rule of law.

Defining process and institutions that enable law and standards to run in harmony will not be trivial.

Why only sketches of the future state and implementation plan?

Unfortunately, it is not possible for me to create the final state – I am not a parliamentary draftsman nor lawyer, nor am I a data management specialist, at best I am a marriage broker. My role is as a synthesiser and not an innovator. This working paper is a sketch, some suggestions, of how the final state might look. The law reform process needs to be defined in detail and executed by qualified people.

This is a complex process that ties together a lot of disciplines with plenty of opportunity to go wrong, to make things worse.

The focus on the implementation sketch is therefore an iterative process with all the requisite professionals in the room and the ability to advance incrementally, enter and leave short dead ends, to make and correct small errors.

In parallel, Working Paper 6 – *A solera for data cleansing* explores some technical proposals for building an iterative framework in which the process of executing law reform on data can be executed. Any systemic codification of the law, such as advanced here, will be an extended process, a journey, and not the work of a summer.

Final state – definition of the necessary capabilities

Overview

It is important to define what abilities our final state needs to support so that an analysis of current defects can be made, a final state can be defined and an appropriate roadmap created.

Capabilities involve a range of disparate factors above and beyond law and standards, including resourcing and delineation of responsibilities. And

capabilities span the parliament of law and the parliament of standards, and the executive, often times with a foot in more than one camp – which is one of the core complicators of this work.

The Capabilities

Introduction

This section will discuss all the capabilities:

- the ability to reason legally and parliamentary composability
- the ability to reason technically
- findability
- consumability
- reliability
- auditability and securability
- diachronically and synchronically queriability
- automatability

The ability to reason legally and parliamentary composability

Reasoning about data repositories to see if they can be joined should not require hunt-and-peck through the statute books – each system should have a single point of statement about its legal and technical wrap.

There are 9 aspects of data that need to be known to create a database.

Action	Description
Defining	Where the Signified is defined, could be legislation, regulation or ad-hoc. The definition of the Signifier is always in the system, it is a property of the database. Technical restrictions on the Signifier will lie in standards.
Auditing	This is general looking at the data for Signifier data quality, conformance with human rights, and data protection, checking that Signifier data is not available to the wrong people, weeding and bulk deletion activities for all Signifier data that applies to a Signifed.
Appealing	The process and procedures whereby a Signifed person or organisation or thing gets onto or gets taken off the database
Partitioning	Where and how the data is partitioned, across local authorities, across health boards, internally within SG and its agencies
Creating	The point of creation - and who, how and why the Signifier data must be created

Action	Description
Reading	Access rights to use and see all or any of the Signifier data about a Signified. This spans both full data access and the rights to see depersonalised and/or aggregated data for research purposes.
Updating	The processes for updating an element of the Signifier data in place[81].
Deleting	The processes around the deletion of a data item inside the Signifier data envelope[82].
Refreshing	Is the Signifier data once and done, or is it supposed to be up-to-date, and who is responsible for keeping it so, the state, or a citizen?

Any digital system will have these 9 things baked into it. Either the delivery process leads to a formal definition of each and every aspect of them all, or a software developer, in standing up the system, will make a series of assumptions and encode them in the implementation.

In order for this to have the property of legal reasonability, all nine should be in a single place, standardly expressed. Thring again makes the point[83]:

The same thing should invariably be said in the same words

As much as possible the 9 things we need to know should be built around definitions added to the Interpretation and Law Reform Act (2010)[84]. It is not enough that the same words be used to describe the same thing in a single

[81] this is generally partial deletes or updates-in-place of items and not total delete/weeding of a set of collated data which is covered by Auditing – this is one of the messier parts – deleting all the data, the entirety of a **`Signifier`** that points to a particular **`Signified`** is not the same as deleting or updating a data field within a particular **`Signifier`**. For a detailed discussion of update and delete within a Signifier please see Working Paper 1.1 – *Data and the rule of law*.

[82] see previous footnote

[83] Practical Legislation, The Composition and Language of Acts of Parliament and Business Documents – 3rd edition, Luath Press, 2019

[84] https://www.legislation.gov.uk/asp/2010/10/contents

statute or ministerial order – they should be the same across them all[85].

One of the problems is that while every data system needs these 9 definitions and treats them as equally important – parliament and ministers don't. Legal powers are granted in different ways. Parliament usually cares about how somebody gets on or off a database (but sometimes delegates that to a Minister). It rarely cares about auditing data or is indifferent as to the data being partitioned.

When this single statement about data properties is created to enable data and services to be reasoned about, there must be flexibility so that the whole, the single data definition, can be composed from primary legislation, secondary legislation and general regulation – with power split between parliament, ministers and operational staff if appropriate.

The ability to reason technically

Technical reasonability is about all the things that are required to reason about data, so data definitions, machine-readable meta-data, data schemas (both synchronic and diachronic views), date standards, geospatial standards and so on – everything that is required to asses and maintain data hygiene.

Findability

Data needs to be holistically findable, which means findable as one or more of:

- a raw data set
- a depersonalised data set capable of being securely made available to external researchers (under appropriate research protocols – the Research Data Scotland model)
- publishable open data
- an API capable of being integrated into a product
- a full-blown service encapsulating some data that can be integrated into

[85] one of the great theological debates that roiled left wing politics in the 20th century was the question of *Socialism In One Country* – Stalin's great conception, as opposed to the Trotskyist notion of global socialist revolution. Among the far fringes of Trotskyism was the Argentinian Juan Posadas who (as well as looking forward to the coming nuclear war) wrote the book *Les Soucoupes Volantes, le processus de la matiere et de l'energie, la science et le socialisme* (*Flying Saucers, the process of matter and energy, science and socialism*). In it he pondered if socialism could be built on one planet alone or if we should seek the help of our Communist comrades on other planets. In the true maximalist spirit of Posadas – perhaps we should aim for a shared section of the Interpretation And Law Reform Act across many countries to enable international reuse and joining of data? Fair makes ya think.

a product

It is not only the data that must be findable, but all the attributes required to reason technically about it.

Consumability

Consumability is all about the *means* and not the *will*. Different mechanisms of consumption have radically different costs and opportunities. Emailing a named person to send you a million row spreadsheet is not the same as having a high-volume API that your administrative system can call. Invoking APIs requires you to have defined authentication. If data is personal, but also important for research purposes, depersonalisation needs to be baked into business process. Patterns need to be identified and promoted into law (see Working Paper No 3 – *The Lego state* for more details of encapsulation, exposure and publication of data as services).

Data access needs to be wrapped in a common access control model – which might be at a line level[86].

Where appropriate data should only be visible in a depersonalised form in data safe havens (the Research Data Scotland model).

Nominally the Digital Economy Act (2017) solves the data sharing problem – with Chapter 1[87] granting wide powers to Ministers to enable data to be shared. In theory.

In theory, theory and practice are the same, in practice they are not – as we can see by the story of data sharing during Covid. Gavin Freeguard's expert testimony[88] to the Covid Inquiry makes for an interesting read as to the type and scale of the data sharing problem.

The legal powers to share didn't eliminate the technical ability to share in any way.

Reliability

Data services need to be declared in terms of reliability. The Scottish Government has a single source of Post Office Address (PAF) lookup. Scottish local authorities use it to get a common matchable address format for many different services. It is a critical service. Such an operation needs to be wrapped in a declared set of Service Level Agreements – and process to define

[86] see Section 6c of Working Paper 1.1 – *Data and the rule of law*.

[87] https://www.legislation.gov.uk/ukpga/2017/30/part/5/chapter/1

[88] https://covid19.public-inquiry.uk/documents/inq000260629-expert-report-by-gavin-freeguard-for-the-uk-covid-19-public-inquiry-titled-module-2-political-and-administrative-decision-making-in-relation-to-the-covid-19-pandemic-dated-26-09-202/

SLAs and assign appropriate levels of resourcing to maintain that capability need to be in place.

Auditability and securability

Historically separation of powers is applied to data at a security level – with external Tiger Teams looking for weaknesses and so on and so forth. Data needs to be held securely, and access to data needs to be audited to detect hacking.

Diachronically and synchronically queriability

One of the major problems that digitisation creates for the rule of law is the mutability of software system. When a dispute arises about administrative decisions (whether you get or don't get a particular benefit) it is important to be able to reason about how that decision was made – and that can be difficult to do if the systems, and the underlying data representation is changing underneath the feet of the claimants. Due consideration must be given to being able to understand data diachronically[89] (as it changes over time) and synchronically (holistically at a point in time) – both for individual data items and for data schemas.

Automatability

Putting the necessary information management around the production of necessary artifacts (metadata, change logs, APIs) brings with it a cost. The best way to mitigate that cost is to invest in tooling that generates the artefacts organically as part of the software development and deployment process, to build depersonalisation into systems at the design stage.

Both the consumption and production of data and data standards can be automated, but the state needs to invest in tooling to make this all easy. That tooling should, of course, be open source, reusable and developed collaboratively with other governments in other jurisdictions.

SUMMARY

Enumerating the capabilities makes clear the scale of the problem. Lots of cross-cutting and interacting components. The parliament of laws and the parliament of standards are a pair of Siamese twins and the task of the law reform process is to gingerly separate them – *in as far as it is possible*.

The separation process will be incremental and partial – focussing on the areas that matter most and with generous *de minimis* to exclude smaller and

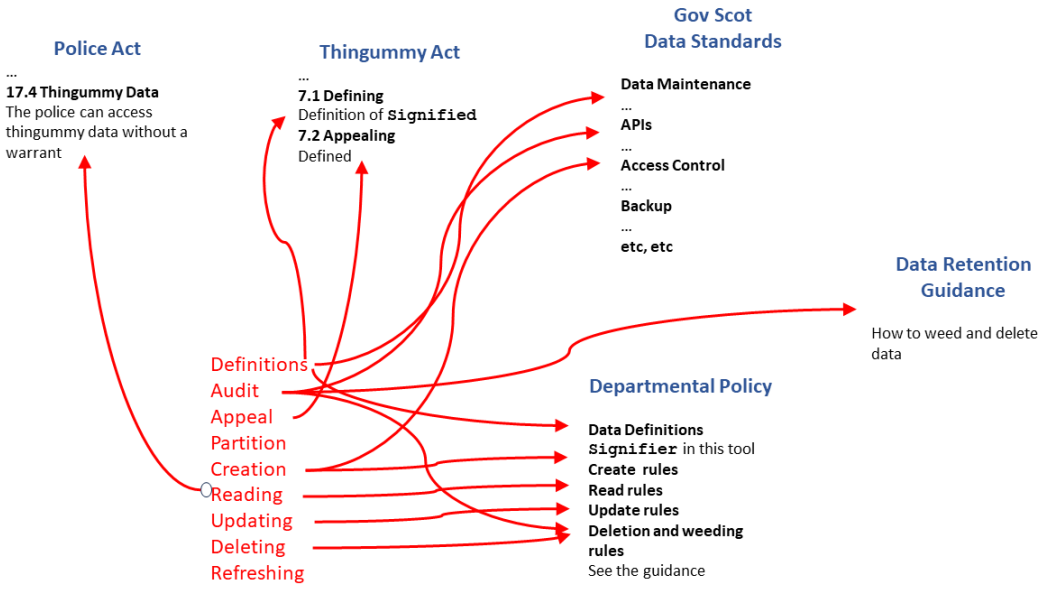
[89] These issues are covered in Working Paper No 1.1 – *Data and the rule of law*

less important systems. A final state will be defined and then extant digital systems will be migrated to the final state over time.

And the separation process cannot and must not be seen as a technical thing – it is also critically a legal and political thing.

Current State

At the moment the 9 core definitions of data are smeared all over the place:



Data access rules are in a variety of places, the power to amend data access rules likewise. For instance, Section 35[90] of the Digital Economy Act 2017 gives very wide powers to vary rules around reading state data. Attempting to reason about whether or not a particular use of data is legal is difficult and expensive.

Sketches of the future state

Introduction

This section will first recapitulate the final state institutional architecture that will support the new world.

This institutional architecture will be supported by a legislative architecture and an information architecture each of which will be explored separately.

The purpose of these architectures is two fold. Firstly it is to support the government in thinking about how it builds it services and how it can simplify and improve them. But the 2nd element is equally important – how to enable the

[90] https://www.legislation.gov.uk/ukpga/2017/30/pdfs/ukpgaen_20170030_en.pdf

citizens and organisations to get justice about administrative systems.

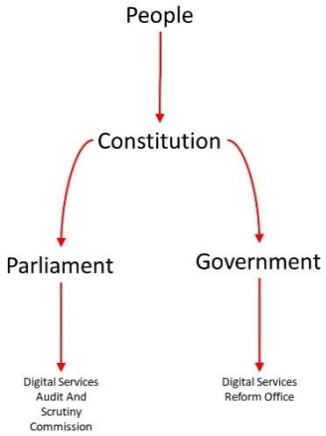
To that end, the recourse to justice cycle that the information architecture is designed to support will be outlined at the end.

Institutional architecture

The future state starts with the institutional structure outlined in Working Paper 1 – *The locus of change*.

The enforcement mechanism proposed in this paper are standards, which are formally non-functional specifications of the technical systems. Working Paper X – *The heart of the beast* goes over the importance of this classification.

There will be a government body called the Digital Services Reform Office and a parliamentary body called the Digital Services Audit & Scrutiny Commission:



These two bodies are key. The DSASC is a scrutiny and oversight body, and the DSRO is a strategy and standards body.

The final state capabilities that are required span two architectures:

- Legislative architecture
- Information architecture

Capability	Legislative architecture	Information architecture
Legal reasonability and parliamentary composability	X	
Technical reasonability		X
Findability	X	X
Consumability	X	X

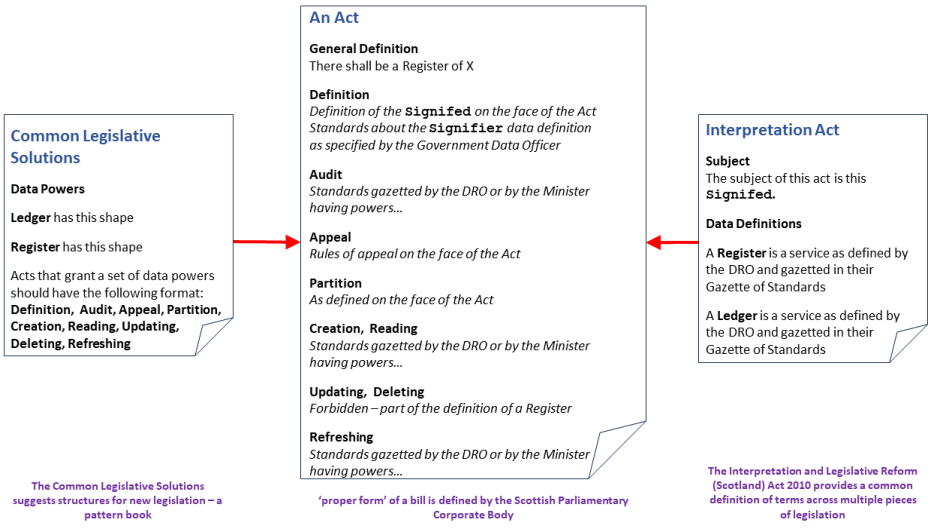
Capability	Legislative architecture	Information architecture
Reliability		X
Auditability and securability		X
Diachronic and synchronic queriability	X	X
Automatability		X

Legislative architecture

The proper form for the section of a bill that grants powers to run a digital system will be standardised to reflect the 9 required attributes.

The discontinuity as to what the legislator and the standard-writer care about as discussed in Section 4 is handled by the double-nature of the *Definition* – the law defines the thing, the Signified, the Government Data Officer specifies the standards that the data definition must match, but the operational team define the actual data structure of the Signifier.

This structure should be backed off by new entries in the Interpretation and Legislative Reform (Scotland) Act 2010[91] as well as entries in the Parliamentary Counsel's *Common Legislative Solutions* handbook[92].



The newly defined entities in the Interpretation and Legislative Reform

[91] https://www.legislation.gov.uk/asp/2010/10/contents

[92] https://www.gov.scot/publications/guidance-instructing-counsel-common-legislative-solutions/

(Scotland) Act 2010 would just be the legislative equivalent of permission masks:

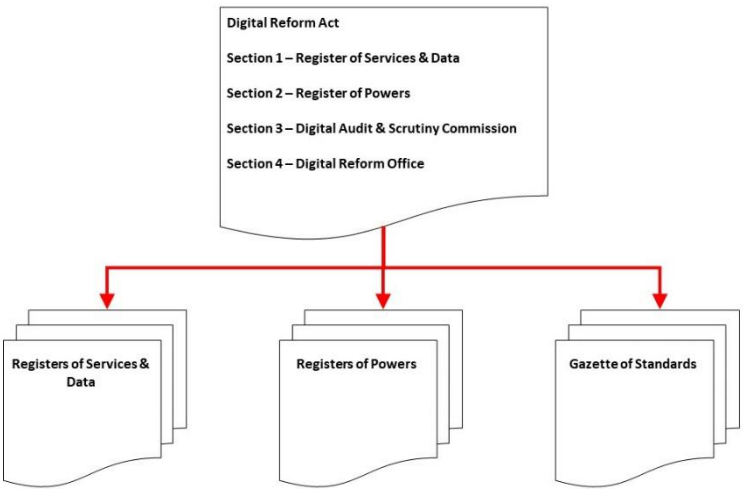
A Ledger	A Register
Definition *blank*	**Definition** *blank*
Audit *blank*	**Audit** *blank*
Appeal *blank*	**Appeal** *blank*
Partition *blank*	**Partition** *blank*
Creation *blank*	**Creation** ***Obligation on citizen/org to register***
Reading *blank*	**Reading** ***Public***
Updating an element in the `Signifier` ***Forbidden***	**Updating an element in the `Signifier`** ***Forbidden***
Deleting an element in the `Signifier` ***Forbidden***	**Deleting an element in the `Signifier`** ***Forbidden***
Refreshing *blank*	**Refreshing** ***Obligation on citizen/org to maintain***

The rationale for these patterns is discussed in Working Paper 1.1 – *Data and the rule of law*.

There will need to be some sort of commencement dance so that existing registers named in extant legislation can be brought in line with the new world. Perhaps using the powers proposed in Working Paper 8 – *An Enabling Act*.

Information architecture

The legislation that sets up the institutional structures of 6.2 can also create the information architecture that we require:



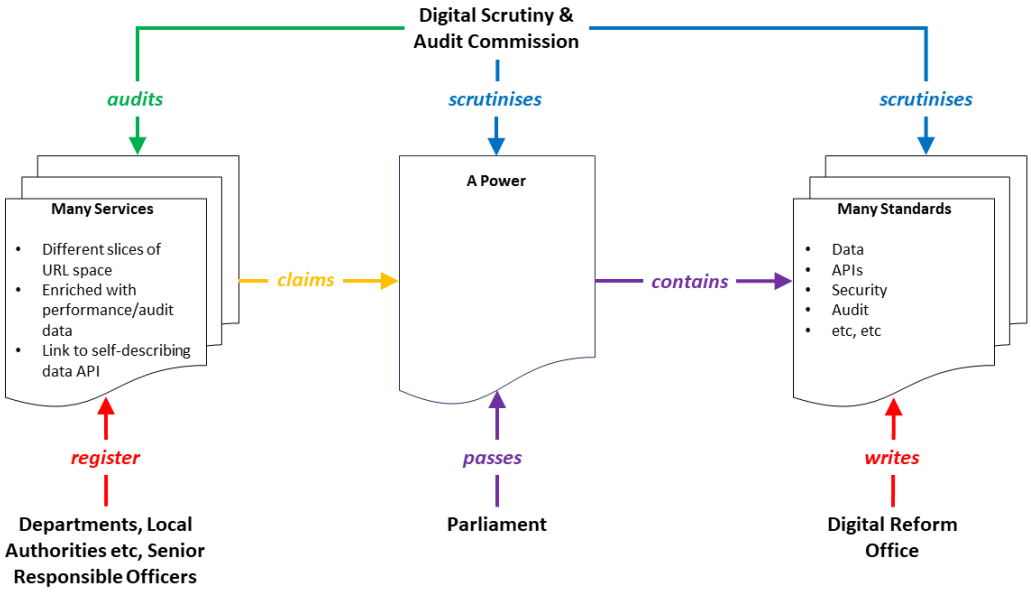
One of the tasks of the DRO (under the supervision of the DA&SC is to

gazette standards) – the gazette above.

It is neither obvious to me, nor my place to say what the correct legal description of the gazetted standards. A maximalist line would be they are akin to the Acts of Sederunt that the Court of Session (with the advice of the Scottish Civil Justice Committee) lays at Holyrood to make the Rules of Court. Laid to be brought to the attention of Parliament but not in anyway amendable or changeable by them – a self-regulating arm of the state.

I suspect a more loosey-goosey legal basis would be appropriate. But an official publication they must be – the sheepdog must have teeth.

Once the registers and gazettes are created they must be populated. Lets start by looking at the final state – and address getting there later.



Let us step through the process:

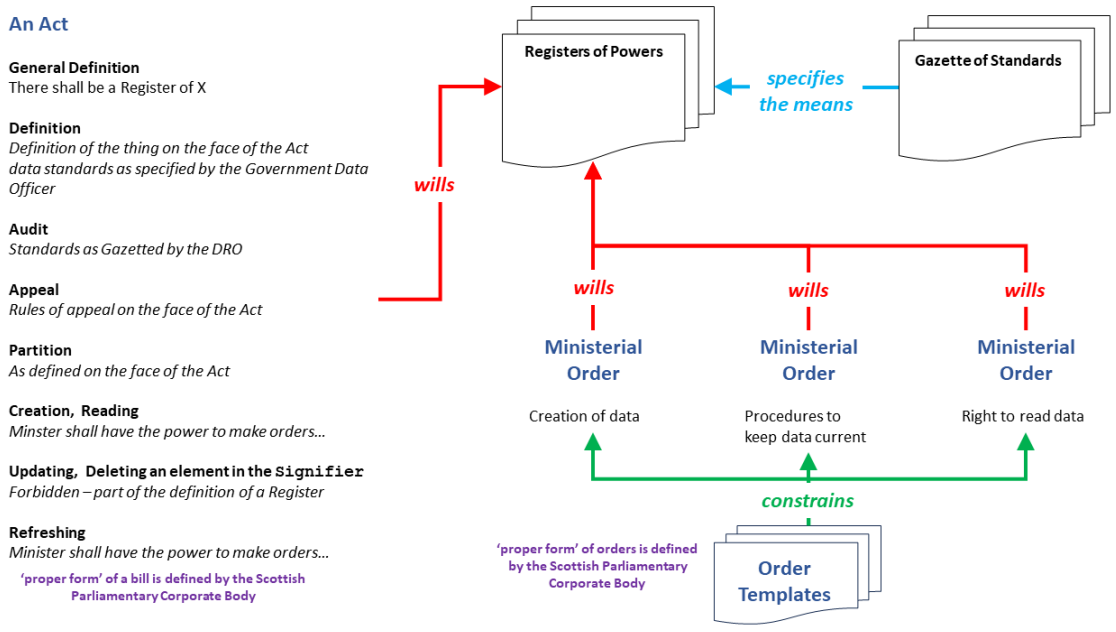
- The Digital Services Reform Office writes standards which (after scrutiny by the Digital Services Audit & Scrutiny Commission) it writes to the Gazette
- The parliament creates a power to run a digital system – with the legislative architecture of Section 7.1 and that is published in the register of powers. This is a machine-readable register. Machine readability is key because the supervision of the registers (who is conforming to what) should be machine driven. Automating processes and turning compliance/supervision from something-that-people-do to something-that-machines-do is one of the superpowers of Big Tech – and it needs to become a superpower of government. Powers may be registered in advance of their commencement dates – this power will come into force at some unspecified time in the future. Under the changed parliamentary

process it is scrutinized by the DA&SC.

- The DSRO attaches standards to the power – anyone exercising this power must adhere to these standards.
- A government department, or sub-state body, or NDPB creates a new digital system (it might be a user-facing service, might be a managed data set) and it registers it with the Register of Services & Data). In the register it asserts a claim to be using one (or more) of the powers. (There may also be one or more system using a single power). The service register entry includes a link back to the system with the URL of the data self-description.
- The DSA&SC audits the service to ensure that it is indeed complying with the standards it needs to follow.

This process enables the harmonisation of two things:

- primary and secondary legislation to give flexibility to the parliament as to the importance of core data operations.
- *wills* and *means* – what the parliament wishes to do, and what is necessary that it be done.



The recourse to justice cycle

Let us step through the recourse to justice cycle:

- a citizen is disgruntled by an administrative decision made with the support of a government digital system.
- they go to the register of services and type in the URL of the service – it takes them to the service entry.
- from the service entry they go to the system data self-describing URL

which tells them the data structure that the system supports – and provides a change log so that they can see the entire release history of the software and how the data schema evolved over time
- from the service entry they go to the Register of Powers which tells them the legal basis of the system – both what laws (primary and secondary) currently apply, and the diachronic history of when regulations changed.
- they can go to the system itself and ask for a dump of their data (as should be their right). The data should be implemented as a ledger - as per Working Paper 1 – *Data and the rule of law*.
- armed with their data, a data description and powers (all in both diachronic and synchronic form) they can then seek legal advice

Sketch of a deployment process

Introduction

A law reform process must have a number of different components:

Phase	Notes
Call for collaboration	See if any other parliament has started, is planning, or wants to collaborate on such a process.
Design and Development	The specification of the formats of all the components: • Interpretation Acts • Common Legislative Patterns • Register of Powers • Register of Services • Standards • etc, etc The Register of Powers needs to be able to cope with <raw> or unrefined statements of powers as well as <cooked> or post-law reform ones
Testing	Taking a bill through the process – might be rerunning an existing bill in a new format, might be a new bill, might be both
Forward looking implementation	Adjustment of parliamentary procedure to bring these components to bear on new legislation
Retrospective implementation	The actual reform of the law – going over existing powers and making the necessary legal amendments. There need to be criteria for bringing powers into line retrospectively – which will be de minimis to the size and scope of systems that claim them

The process of law reform will be entangled with the process of creating the

institutional framework outlined in Section 11 of the Working Paper 0 – *The locus of change*.

The mechanisms in that paper include an Enabling Act as well as temporary Standing Orders that can be used to constrain the application of the new world to a fragment of the statute book whilst the details are being shaken out.

The bodies being proposed in that working party are so critical to the wide range of other proposals that the law reform process cannot but help getting entangled in them.

Call for collaboration

The problem that data law reform is trying to address is not specific to Scotland, and as always an appropriate effort should be made to try and eliminate or reduce the work that needs to be done to achieve it. It would be better if there was a working example to be copied.

To that end, there should be a structure appeal to other democratic legislatures to see if similar transformation has been applied, and to collaborate with any other government that is interested in the topic.

Design and Development

It is important that the elements of this law reform are developed iteratively and in the round. Each part depends on all the others – the various statutory bodies, the associated statutory registers, the technical standards, the amendments to the Interpretation and Law Reform (Scotland) Act 2010, the changes to *Common Legislative Solutions*.

The process of executing law reform needs to be first designed and tested before the actual law reform can begin. This should be a paper exercise that takes a section of the statute book through a law reform process with the appropriate components implemented as dummy systems. The paper exercise should involve all interested parties in the cycle, from policy, Parliamentary Counsel's office, the relevant parliamentary committee, technical specialists and appropriate external parties.

Testing

Once an agreed format and law reform processes has emerged from the design exercises it is important to test the process by taking an existing Act or Acts through it.

Temporary Standing Orders can be used to restrict the first elements but there remain problems with standardising things and amending the Interpretation And Law Reform (Scotland) Act 2010. Bringing existing systems

into the new world will require some sort of commencement mechanism.

It is important that the testing includes both the clean forward-looking implementation process and the dirty backward-looking one. Not least because until the depth and impact of dirtiness can be ascertained it will not be possible to prioritise law reform in the optimal manner.

For instance, a general definition of the word register or ledger is likely to sweep up legislation that accidently uses those terms, or uses them in a looser sense that is now intended.

Mechanisms need to be found to enable the gradual harmonisation of language across the statute book[93].

A Janus-faced implementation

The implementation is janus-faced, one element is forward looking for new legislation and one backward-looking for transforming old legislation.

Forward looking implementation

The forward-looking implementation should be fairly clean – new Acts are born in the new world, designed to be citeable in registers of powers, with the correct format and appropriate clarity.

Retrospective implementation

For retrospective implementation the situation is somewhat messier. It would be appropriate to bring existing systems into the Register of Systems as quickly as possible – and to get a sense of how many systems there are, and some quantification of their size and impact. The retrospective programme should be shaped by size and impact – some systems are already time limited and will eventually die, some are already planned to be replaced and should not be updated.

In order to accommodate this both the Register of Systems and the Register of Powers will need to support 'dirty' registrations – registrations that don't meet the full final state standards – systems without self-describing databases and meta data. Powers that are smeared across many statutes or missing.

Conclusion

The process of law reform for data is complex in both its conception and implementation – but is fundamental to the building the capability of the state

[93] the research team at legislation.gov.uk have a range of corpus-based information tools that can make this task manageable and reasonably quick.

to evolve and change in the digital world. In the overall programme of changes proposed it should be embarked on at the end, after a degree of institutional capability and maturity has been achieved.