The Remixable State

Focus and refocus as a core competency (Version 1.0)

Gordon Guthrie

RESEARCH FELLOW AT SCOTTISH GOVERNMENT
BIUS WORKING PAPER NO 4
(THIS DOCUMENT DOES NOT REFLECT THE VIEWS OF SCOTTISH GOVERNMENT)

Table of Contents

1   Introduction

1.1   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.

1.2   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.

1.3   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.

2 The BIus Project

This is Working Paper No 3 of *BIus - Basic Law-Making For Legislative Computer Systems* which is a research project looking systemically at how the state creates the digital systems underpinning its services.

Working papers are being released gradually for comment:
Working Paper 0 – *The locus of change* (forthcoming)
Working Paper 1 – *Data and the rule of law* (published)
Working Paper 2 – *Rules as code* (published)
Working Paper 3 – *The Lego state* (published)
Working Paper 4 – *The remixable state* (this document)

BIus working papers are designed to stimulate discussion about key elements of the relationship of the state to digital systems and their delivery. Your feedback, input, and particularly criticisms of this paper are most welcome. Feel free to distribute it however you wish.

Working papers are published via the *Digital Policy* SubStack.

Author/contact: gordon.guthrie@gov.scot or subscribe to Digital Policy | Gordon Guthrie | Substack[1]

The author is an independent Research Fellow at Scottish Government under the First Minister's Digital Fellowship programme. The views of this paper do not represent the views of Scottish Government.

---

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

3   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[2] 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 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.

---

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

4   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 – over 10 times bigger – 663 words.

That was HTML v0.9[3] – by the time the big old fancy HTML 1.0 came out 5 years later in 1996 it had ballooned to just under 17,000 words – today it's just under 1,000 times larger at just over 620,000 words. And HTML has a couple of children - the Javascript[4] spec is kissing 280,000 words, and the fistful of CSS specs[5]? – 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:

---

[3] The HTTP Protocol As Implemented In W3
[4] ECMA-262 - Ecma International (ecma-international.org)
[5] CSS current work & how to participate (w3.org)

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 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.

5   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
```

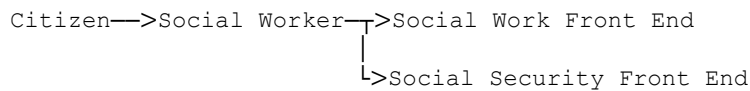This is a direct cut'n'shut of the banking model:

```
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
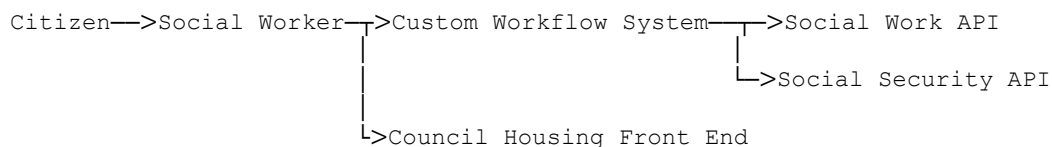
delegation. It is important to understand from a systemic point that the bank is the organisation that maintains the subset of IFA's able to use the system.

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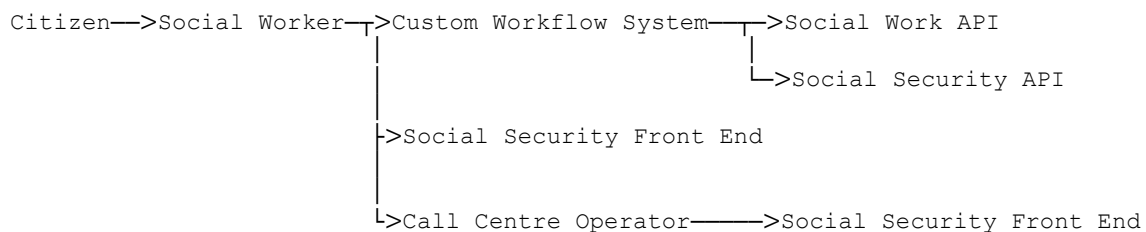
So we know that there are not technical barriers to doing it. Where it differs is in its compositional nature:

```
Citizen──>Social Worker─┬>Social Work Front End
                        │
                        └>Social Security Front End
```

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:

```
Citizen──>Social Worker─┬>Custom Workflow System─┬>Social Work API
                        │                        │
                        │                        └>Social Security API
                        │
                        └>Council Housing Front End
```

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:

```
Citizen──>Social Worker─┬>Custom Workflow System─┬>Social Work API
                        │                        │
                        │                        └>Social Security API
                        │
                        ├>Social Security Front End
                        │
                        └>Call Centre Operator──────>Social Security Front End
```

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.

6   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[6] 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 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

---

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

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.

7 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.

8 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.

9 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.