

03/08/2023

# Data and Ledgers

Conforming to the rule of law (Version 1.1)



**Gordon Guthrie**

SCOTTISH GOVERNMENT

BIUS WORKING PAPER NO 1

(THIS DOCUMENT DOES NOT REFLECT THE VIEWS OF SCOTTISH GOVERNMENT)

<b>1</b>	<b>INTRODUCTION .....</b>	<b>2</b>
1.A	DATA, LEDGERS AND THE RULE OF LAW .....	2
1.B	WHO ARE YOU? .....	2
1.C	WHY SHOULD YOU READ THIS? .....	2
<b>2</b>	<b>THE BIUS PROJECT .....</b>	<b>3</b>
<b>3</b>	<b>REVISION NOTES.....</b>	<b>4</b>
<b>4</b>	<b>EXECUTIVE SUMMARY .....</b>	<b>5</b>
<b>5</b>	<b>THE EVIDENCE .....</b>	<b>6</b>
5.A	THE CPAG REPORT .....	6
5.B	THE RULE OF LAW.....	7
5.C	TWO STYLES OF DATA SCHEMA .....	7
5.D	READING THE CPAG REPORT WITH A DATA STRUCTURE EYE .....	10
5.E	THE PLURAL OF ANECDOTE ISN'T DATA.....	11
<b>6</b>	<b>ADDRESSING THE ISSUES.....</b>	<b>12</b>
6.A	CHARACTERISATION OF THE PROBLEM .....	12
6.B	SOLUTIONS.....	12
6.C	ADDITIONAL BENEFITS .....	13

# 1 Introduction

## 1.a 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.

## 1.b 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.

## 1.c 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.

## 2 The Blues Project

This is Working Paper No 1 of *Blus - 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* (published)

Working Paper 5 – *Law reform for data* (forthcoming)

Working Paper 6 – *A solera for data cleansing* (forthcoming)

Working Paper 7 – *Experimental digital legislative processes* (forthcoming)

Blus 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](mailto:gordon.guthrie@gov.scot) or subscribe to [Digital Policy | Gordon Guthrie | Substack](#)<sup>1</sup>

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.

---

<sup>1</sup> <https://digitalpolicy.substack.com/>

### 3 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.

## 4 Executive Summary

The Child Poverty Action Group report [You Reap What You Code: Universal Credit, Digitalisation And The Rule Of Law<sup>2</sup>](#) 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.

---

<sup>2</sup> <https://cpag.org.uk/policy-and-campaigns/report/you-reap-what-you-code>

## 5 The Evidence

### 5.a 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 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 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.*

## 5.b 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<sup>3</sup> 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.
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)

## 5.c 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.

---

<sup>3</sup> T Bingham, *The Rule Of Law*, Allen Lane, 2010



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 ORM	Spring Django	Java 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 3<sup>rd</sup> line and correctly entered in the 4<sup>th</sup> 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.

#### 5.d 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.

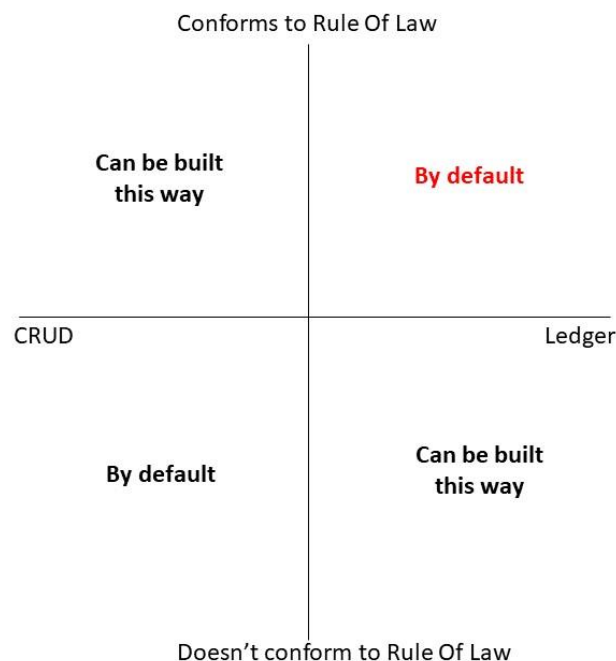
And it should be stated clearly that there is ledgering in the UC system. The payments systems will certainly be full accountancy ledgers. The journal which the systems use for the citizens to communicate with the DWP is a form of ledger – although careful reading of the user stories indicates that it 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.

### 5.e 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.

## 6 Addressing the issues

### 6.a 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.

### 6.b 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).

## 6.c Additional Benefits

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<sup>4</sup>.

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.

---

<sup>4</sup> 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?