Department for Levelling Up, Housing & Communities

Delta Service Mapping

Findings and Recommendations

Contents

Executive summary

Department for Levelling Up, Housing & Communities

Introduction

With over 17,000 registered internal and external users, Delta is a critical system for managing and collecting data, producing official statistics, informing decisions about policy, and making grant and funding payments.

Although Delta enables the department to collect data and process payments totalling billions of pounds each year, its existing underlying technology is a barrier to make changes that would improve use for internal and external users and realise financial and other benefits.

As a result, Delta - in its current state - does not meet our strategic desire for modern, flexible and cost efficient technology, in addition to failing to meet modern accessibility and other technological standards.

This report sets seeks to answer the question should we make fundamental changes to Delta? and, if so, what should those changes be?

We should make fundamental changes to Delta

Delta is a tool,
not a service

but we took a service design approach

Our service design approach

Delta, itself, is not a service. At least not in the sense that service designers or other DDAT professions refer to services. However, we've taken a service design approach to exploring and understanding what Delta does, and the problems and opportunities associated with it.

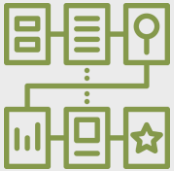
Our approach has followed 5 principles.

Focus on users

Focus on users' needs, identified through user research

Co-create

Include and involve stakeholders in problem framing and ideation



Sequence

Break complex systems into separate processes and journeys

Evidence

Draw conclusions from observed and validated data

Think holistically

Consider all touchpoints across networks and interactions

Adapted from 'This is Service Design Thinking', Marc Stickdorn and Jakob Schneider

We were led by research and data

Key findings I

- Delta broadly meets user needs but changes could create a more efficient system
- Inefficiencies within and around the system > lost productivity
- Processes around Delta, not just Delta itself, could be improved.

a) Processes around Delta could be improved

Grants process is disjointed

Leading to legal and financial risks and loss of productivity

The analysis process is not streamlined

Leading to loss of productivity

Data upload process from data providers is inefficient

Leading to greater chance of manual errors, missed deadlines and increased need for user management

Key

Supported by:

Interviews 

Ethnography 

Survey 

Key findings II

b) There is a lack of appropriate training/tools

Design and editing of Orbeon forms can be complex for non-specialists

Limits number of form creators; Can result in poor quality data retrieved

Guidance is hard to find & needs updating

Time lost & users don't know Delta can do

Limited analysis tools restrict reporting options

Limits types of analysis conducted and presentation of results

c) Users experience some system problems

User interface is unintuitive

Particular problem for irregular users. Users don't know what Delta can do

System stability could be improved

More consistent access to data

Time before timeout is too short

Lost work & time needed to restart work

Key

Supported by:

Interviews 

Ethnography 

Survey 

Key findings III

d) Users report a generally positive experience of Delta

Users are broadly happy with Delta

Delta enables efficiency by allowing information to be stored in one place

Xml bulk upload facility is easy to use for data providers

Data providers would like Delta to be used for data collection by OGDs

Grant payments and collections can be made easier by using Delta

Smaller forms are easy to set up and manage

Key

Supported by:

Interviews 

Ethnography 

Survey 

Use of findings

User research interviews

Internal and External survey

Ethnography

- Research sessions opened to the team to note take and observe
- Survey questions developed in collaboration with members of DLUHC

Analysis and synthesis of findings and development of artefacts

Seeking feedback from show+ tell playbacks and participants

Problem statements and prioritisation

'How might we..' Solution workshop

Findings used to inform other team members' work and workshops

Delta is part of a wider landscape

Mapping the landscape

After initially focusing solely on what Delta does, and thinking about whether it might be valuable to enhance it by adding capability, or moving capability out of Delta, we realised that we really needed to understand what other systems and services already exist in the data collections and payment space.

We created a service landscape map to capture this. We created, iterated, stress tested and validated this landscape map through a series of micro workshops with a range of stakeholders involved in different collections (including collections that don't use Delta).

The landscape map visualises the high level phases in the broad range of data collection and payments journeys in DLUHC, the digital services and systems used in each phase, and their users.

Each collection and grant payment relies on different services and systems, but this map brings them together in a single view, to help us understand potential overlap, relationships and opportunities.

Department for Levelling Up, Housing & Communities

Delta service landscape

Created by **DLUHC Digital** Version 1.2

Purpose

This map visualises the high level phases in the broad range of data collection and payments journeys in DLUHC, the digital services and systems used in each phase, and their users.

Each collection relies on different services and systems, but this map brings them together in a single view, to help us understand potential overlap, relationships and opportunities and implications associated with potential change.

Use

This map can be used to map touchpoints for specific as-is data collections and payments, and provide a tool to support decision-making about new collections.

For our team, it helps us understand how Delta fits in to the wider service and system landscape in DLUHC, and the impact of potential changes to our use of Delta or change in its function, features and capabilities.

Relationship with other artefacts

This visualisation does not, and is not intended to, map out the Delta service, processes, technical architecture, data flows, user journeys or user experiences.

We are creating separate artefacts to visualise these, but we'll make this visualisations consistent so they can be viewed in parallel.

Phase

Initiate and release	Provide and submit	Collect and collate	Cleanse and analyse	Approve, pay and review
DLUHC prepares for the collection	The data provider enters (where necessary) and submits data to DLUHC	DLUHC collects and aggregates data from providers	DLUHC assess and analyse the data	DLUHC approves the data and makes relevant grant or funding payment

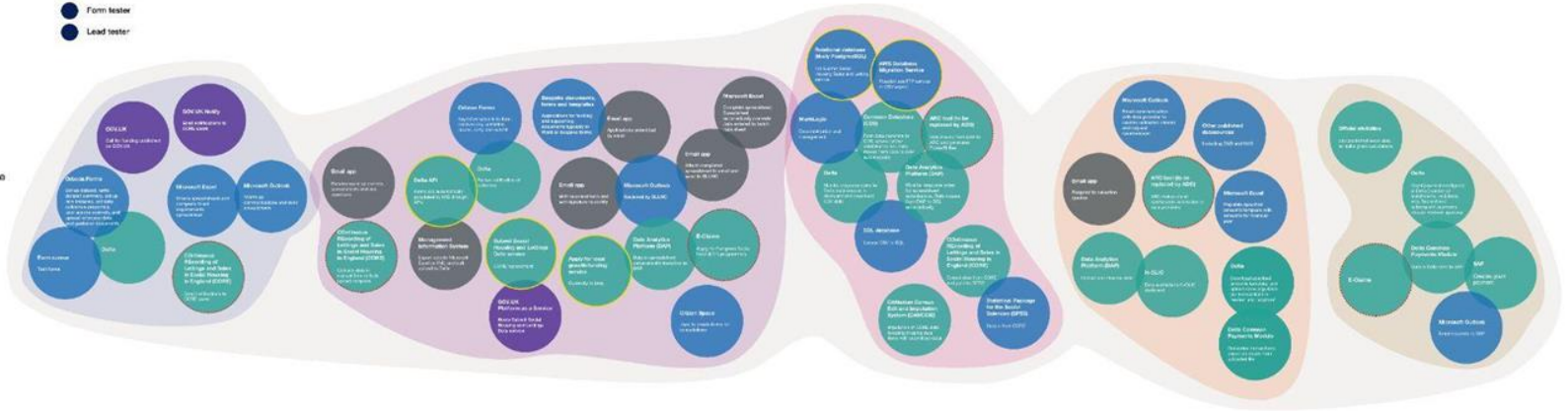
User

- DLUHC user
- External user

Initiate and release	Provide and submit	Collect and collate	Cleanse and analyse	Approve, pay and review
Data owner	Data provider	Analyst	Analyst	Local Government Finance team
Delta admin	Certifier		Grant reviewer	SAP team
Ratifier/grant owner			Grant approver	
Form designer				
Form tester				
Lead tester				

Service and system interaction

- Bespoke DLUHC product/service
- 'Off the shelf' DLUHC product/service
- Government service
- Data provider system
- In development
- Being retired or replaced



We learned from other parts of Government

Secondary research

We reviewed research artefacts created by other teams, like in the Delta Grants Discovery work. We also engaged with a number teams across government whom have faced, or are facing, similar challenges and opportunities to those we face when looking at Delta, and undertook desk research.

These included:

Crown Commercial Service

Report management information service

Government Digital Service

Forms discovery

Local Digital Collaboration Unit

Cyber security discovery and work on future of data sharing

Ministry of Justice

Forms service and analytical platform

National Careers Service

Data sharing service and course directory

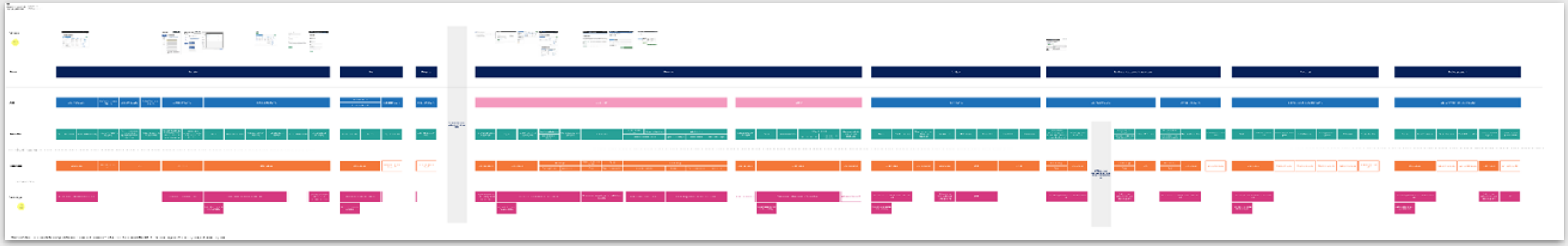
We mapped Delta's visible and invisible touchpoints

Mapping the as-is service

Service maps are a form of user journey maps that help reveal the full spectrum of situations where users interact with a service, detailing every aspect, including those that users don’t see.

Because Delta isn’t a service, it’s not possible to map it in the holistic way we might for a typical service.

However, we’ve mapped where Delta features in users journeys that include Delta, surfacing the ‘invisible’ technologies underpinning each of the visible actions and interactions.



Our findings

Department for Levelling Up,
Housing & Communities

Delta does some things really well

What Delta does well: Observations

Delta is viewed by some as ‘a single source of truth’ where collected data can be held in one place.

Delta has helped streamline the payment process to local authorities.

Delta works well for small forms.

Delta is adequate for data analysis purposes.

Delta works well as a data collection system.

Delta has a good support team.

Delta processes billions of pounds of payments every year.

The advantage of Delta is having all your data in one place compared to multiple Excel documents.

Data collection and analysis

People who use Delta regularly believe it's good and has potential.

Delta has linked up the payment process for us. This has massively reduced the time it has taken to send payments to local authorities. It used to be a very manual process.

DLUHC policy and operations teams (DPOT)

We know there are problems, but we also have ideas

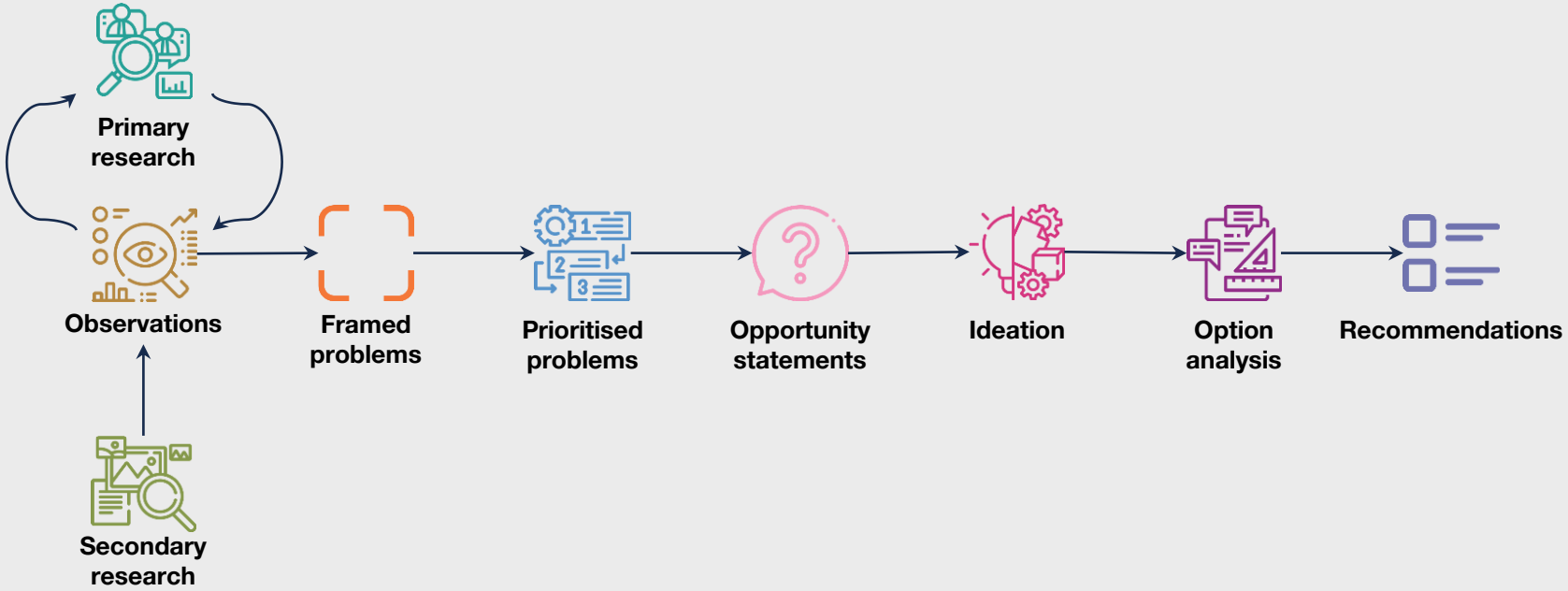
We uncovered problems and discovered opportunities collaboratively

We captured our learnings from continuous primary, and secondary, research activities in an observations log and, where appropriate, framed them as problem statements. We did this using collaborative tools, like Mural, with stakeholders, providing an opportunity to challenge, discuss and validate our problem statements.

We prioritised the problem statements and facilitated co-design activities to ideate potential opportunities collaboratively with stakeholders.

Collaboration

We uncovered problems and discovered opportunities collaboratively



We uncovered 2 broad types of problem

Our work has revealed 2 distinct ‘types’ of problem.

1. Issues with the usability and user experience provided by Delta

1. Technical challenges and constraints related to the technologies Delta depends, and is built, on

Some of these were already-known problems, for which we collected and collated supporting evidence to develop a deeper understanding, and some were previously unknown.

Epic problem 1

Internal and external users need support to complete tasks using Delta.

This leads to high volumes of support calls and means it takes longer for users to complete tasks, causing lost working hours and frustration.

Internal and external users need support to complete tasks using Delta: Problem statements

PS-027

Delta is not intuitive to use. This leads to high volumes of support calls affects user experience and satisfaction, and lost working hours.

PS-028

Using Active Directory for user account management can be complex. This means user management can be time-consuming.

PS-006

Column names are not easy to understand in downloaded datasets. This makes it difficult for analysts to use the data without support.

PS-045

Data providers sometimes find the questions in data collections ambiguous. This can create confusion about what data to input.

PS-001

We are dependent on people with technical expertise to design forms.
This means the people who design forms need to interpret and implement requirements from people who determine the data that needs to be collected.

Downloads in Excel or CSV are harder to analyse and extract. Data dictionary required to help in this process but could made even easier by developing more meaningful names or titles.

Internal survey respondent

Training is a significant commitment for the Delta support team, sometimes as frequently as twice a month.

Delta is very hard to use... even after being trained.

Data collection and analysis

Internal and external users need support to complete tasks using Delta: Ideation outputs

These are some examples of opportunities identified through co-creation activities with stakeholders.

These are included to help contextualise our approach and the ideas generated through those activities, rather than an endorsement.

- Redesign the user interface in the problematic stages of journeys.
- Undertake content review and introduce content design capability to Delta team.
- Provide at-point-in-service support to users.
- Simplify and standardise column names in datasets.
- Design and develop a component that renames or maps data in downloaded files.
- Create, maintain and socialise a data standard and dictionary.

Epic problem 2

There is little documented information about Delta, and knowledge is limited to a small number of people.

This creates significant reliance on small groups, and separation between people who set requirements and those that implement it.

There is little documented information about Delta, and knowledge is limited to a small number of people: Problem statements

PS-009

There is limited documentation about Delta, and knowledge is held by a small number of people, making it difficult to share with teams and suppliers.

PS-037

Building reports in Delta requires XQuery knowledge. This limits the number of people who are able to build reports.

PS-035

There are a high number of servers supporting Delta, with limited understanding of their use or necessity. This is a potential cost inefficiency.

PS-033

Some of Delta's capabilities and features are not used in all collections. This means we are not maximising the value we could get from Delta.

Only 1 out of 20 people in Policy are able to use XQuery.

Data collection and analysis

[We are] trying to have the Funds team being autonomous, but they need technical expertise.

Funding Service Design

You need code expertise to build a form.

Data collection and analysis

Guidance is not easy to find.

Other Government Department (OGD)

There is little documented information about Delta, and knowledge is limited to a small number of people: Ideation outputs

These are some examples of opportunities identified through co-creation activities with stakeholders.

These are included to help contextualise our approach and the ideas generated through those activities, rather than an endorsement.

- Buy or build a simpler form building tool.
- Deliver more training, to more people, more often, so people don't forget between sessions.
- Assign someone responsibility for the goal ‘make Delta something that anyone can run’.
- Help policy people understand how forms are designed.
- Create a knowledge base, or similar.
- Increase the proportion of civil servants in the Delta team.

Epic problem 3

The process of providing data is frustrating for data providers and policy teams.

This provides a poor user experience, relies on extensive manual keying and creates opportunity for error.

The process of providing data is frustrating for data providers and policy teams: Problem statements

PS-038

The time-out period in Delta is short. This can mean that data providers need to sign in frequently and lose what they're working on.

PS-020

Collecting anding send data to DLUHC is time consuming and complex. This increases risk of human error and inaccuracies in data.

PS-047

Data providers find keying '0' to indicate no data time-consuming, unintuitive and potentially misleading. This can cause frustrations and inefficiencies.

PS-041

The time out warning only appears in Delta's browser tab. This means users are often unaware of it and are timed out.

PS-015

Data requested is often not in the format data providers hold it. This means they need manually calculate and key data, creating opportunity for error.

PS-031

Data providers are often unaware of new or changed questions until they start a form. This means they have limited time to get data in the required format.

Bulk upload is okay, but it doesn't tell you which validation failed.

Data collection and analysis

Mistakes are common on each potential input, so we try to use Delta to identify these… but we can’t catch them all. Delta inbuilt validation can't catch all user errors.

Data collection and analysis

Sometimes the questions are not very clear and can be misinterpreted.

External survey respondent

We have automated systems but they cannot transfer automatically into Delta.

Data provider

The local authority doesn't understand what the form is asking. These forms might be different which makes filling them [in] difficult.

Policy and operations teams

The process of providing data is frustrating for data providers and policy teams: Ideation outputs

These are some examples of opportunities identified through co-creation activities with stakeholders.

These are included to help contextualise our approach and the ideas generated through those activities, rather than an endorsement.

- Provide explanatory content at each data entry point.
- Reduce the amount of data entered into single screen.
- Remove reliance on Active Directory to access all components.
- Work with data providers' system suppliers to enable data to move by API.
- Increase use of bulk upload of data.
- Involve a content designer in question design.

Epic problem 4

There are a high number of bugs and usability issues in Delta.

Partly due to a reliance on manual testing; this reduces our confidence in data providers' interactions with Delta, and requires time to fix bugs.

There are a high number of bugs and usability issues in Delta: Problem statements

PS-019

Automated testing practices are limited. This leads to bugs and releases that require fixes.

PS-030

Usability testing of forms is limited, reducing our confidence in data providers' use forms and awareness of challenges.

PS-029

The parameters for searching previous submissions are limited. This can make finding submissions difficult.

41% of the Delta team's tickets in the last 3 years relate to **bugs**.

It is hard to find forms you have submitted because the ‘previous submission’ search is awkward to use, it is not clear or easy.

Data provider

14% of categorised support tickets are requests to **extend submission deadlines**.

Issues with releases, not tested properly.

Data collection and analysis

There are a high number of bugs and usability issues in forms: Ideation outputs

These are some examples of opportunities identified through co-creation activities with stakeholders.

These are included to help contextualise our approach and the ideas generated through those activities, rather than an endorsement.

- Create a testing strategy.
- Implement automated testing.
- Conduct more user research and usability testing in advance of form release.
- Run pilot collections in advance of release.
- Remove the need for users to interact with forms by moving data by API.
- Creation of a new feature allowing deadline dates to be changed without manual intervention.
- Improve Delta navigation.

Epic problem 5

It's difficult to make changes to Delta.

This means we're limited in how we can solve problems which, in other scenarios, could be relatively straightforward.

It’s difficult to make changes to Delta: Problems

PS-040

The technical architecture is highly coupled. It’s difficult to make any change the arrangement, interaction, and interdependence of different elements.

PS-021

There is no long-term technical or strategic direction to inform or if or how Delta should used in the future.

PS-026

There is no design authority for technical decisions. This means nobody has responsibility for the overall creation of the architecture and solution.

We do not have a cohesive team that works together to achieve something.

Delta technical support

‘Firefighting’ seems to happen often and is preventing us from respecting plans and agreements on delivery.

Delta technical support

It’s difficult to make changes to Delta: Ideation outputs

These are some examples of opportunities identified through co-creation activities with stakeholders.

These are included to help contextualise our approach and the ideas generated through those activities, rather than an endorsement.

- Build new front end ‘skin’.
- Rebuild Delta with a more flexible technology stack.
- Create a service owner role for data collections and/or payments.
- Segment Delta into smaller ‘services’.
- Ensure Delta team has required skills for Delta development
- Introduce new governance to make strategic decisions about Delta.
- Move Delta outside the Datamart and loosely couple the new version.

Epic problem 6

Users often need to devise and implement workarounds.

This creates reliance on inconsistent, manual processes and creates opportunity for error.

Users often need to devise and implement workarounds: Problems

PS-044

Delta is not powerful enough to cope with large data flows. This means larger forms cannot be processed.

PS-048

Forms are often not set up to transfer data to DAP automatically. This means analysts need to create manual processes to download and upload data.

PS-005

Delta has little or no automated performance analytics. This means that data reports need to be requested and generated manually.

PS-017

Local authority name changes disrupts unique system identifiers. This means current and past records need to be aligned manually.

PS-004

Analysts extract whole datasets and work on them elsewhere. This means the data is unstructured and risks human error during structuring process.

Downloads of data are cumbersome. An automated transfer would be a huge time saver for collections with vast data volumes

Internal Survey Respondent

We have to do a lot of jiggery pokery to get some data in the format that is being asked for.

Data provider

Delta has difficulty handling the larger forms. There are forms too big for Delta.

Data collection and analysis

There have been times where the categories we're tracking don't align with the categories in Delta which means… [so] we need to re-analyse our spreadsheet.

Data provider

Bringing data out of Delta can be difficult, to find the right datastore. I do this a couple of times a week.

Data collection and analysis

Users often need to devise and implement workarounds: Ideation outputs

These are some examples of opportunities identified through co-creation activities with stakeholders.

These are included to help contextualise our approach and the ideas generated through those activities, rather than an endorsement.

- Introduce automated performance analytics.
- Bring analysis and reporting capability into Delta.
- Increase use of Delta's capability to automate data transfer to DAP.
- Introduce new governance to make strategic decisions about Delta.
- Build new service or data submission capability for larger forms.

Epic problem 7

Delta processes and workflows aren’t optimised for grant payments.

This means Delta can’t be used for some payments, increases manual calculations, and causes delays in payment, which costs us money.

Delta processes and workflows aren't optimised for grant payments: Problem statements

PS-046

It takes 5 working days for a grant set up in Delta to be paid by BACS. This means setting up payment directly in SAP is necessary for more urgent payments.

PS-047

Sometimes payments have been delayed, meaning we've failed to meet SLA commitments, damaged our reputation and needed re-submit, approve, and issue partial refunds.

PS-003

The capability that can be built in forms to calculate grants is rarely used. This manual calculations need to be made, increasing the opportunity for error.

A common pain point highlighted from the external survey was that remittances for grant payments lack detail and need more clarity.

Grant payment processing errors are a risk.

Local Government Finance Team

The main thing is making sure the list HMT and Delta ecodes should be the same. Use the same ecodes. If you guys could merge the ecodes together, that would be great.

Local Government Finance Team

You need to separate out Delta-specific tasks for statistical collections and those for grants from other processes outside of Delta to really understand where the pain points are.

Internal survey respondent

Delta processes and workflows aren't optimised for grant payments: Ideation outputs

These are some examples of opportunities identified through co-creation activities with stakeholders.

These are included to help contextualise our approach and the ideas generated through those activities, rather than an endorsement.

- Include grant name or grant section name in remittance advice.
- "I would like to be able to download the upload sheet from Delta and everything would be 0 – that would be amazing"
- Unify DLUHC and HM Treasury local authority identify codes.
- Create a dashboard for external organisations showing details and dates of previous and future payments.
- Remove previous numbers on the upload sheet.
- Reduce the 5-day approval timeline.

We investigated some observations in more detail

Observation 1: Potential value in reviewing SLAs relating to Delta with Suppliers

SLAs were identified as a financial issue to explore, as partial refunds have been issued in the past due to breach of SLA.

Missed payments via SAP are considered to be the biggest issue.

Three ways of resolving this were identified:

1. Manual entry to make the payment, which is time consuming for the finance team
2. A delayed payment, which breaches the SLA unless agreed to by the receiving authority
3. Agreed amendment to the SLAs to avoid future breach

The biggest frustration is that missed payments could be avoided entirely if the data matching process was better aligned between Delta and Treasury, and if the approvals process was more flexible.

For potential next steps on how to overcome this, please refer to the section under:

- **Financial Opportunity 3: Areas of legal and financial risk**

Observation 2: Validate why grants are managed outside of Delta

Delta is often used instead of SAP, due to a missing feature or component which directs people to Delta to meet the need, then pushes into SAP.

Benefits of using SAP directly:

- If payment is needed urgently (i.e. tomorrow) SAP is fastest.
- 5 working days in delta for BACS payment type as standard.
- You can make other payments in SAP.
- Stream of info from Delta into SAP. SAP pulls data from CPM. See technical pages for CPM diagram to show flow of data.
- Once an organisation is set up, all details are held for ease of making future payments.

Benefits of using Delta interface:

- Easier for more than 10 payments (due to manual entry)
- Delta is much slower, but has a simpler interface and workflow
- Configuring a grant is only a 10 minute job in Delta
- Easier to make payments to Local Authorities are they are already there on Delta.

Recommend that total number of grants vs what was managed in Delta is looked into, followed by a comms and engagement exercise for the Delta Product Manager to encourage people to use Delta.

Observation 3: Conduct user profiling in Delta to understand activity and usage trends

Obtaining data around average number of daily users, peak times and tickets was not possible due to two blockers.

- Jaspersoft reporting capability is available but the development team would need time to build the reports
- IT infrastructure does not allow access to Jaspersoft desktop software

The infrastructure issue a big blocker and could take a couple of months to get approval access. Even if access is given, it is not clear whether the correct skillset exists to do this.

Exports of data are predominantly done via the Data Analysis Platform to meet needs, with statisticians using various data sources. Central Data Store is also referenced here. There is a disjoined approach and an opportunity to evidence this and close the gaps.

Delta HAS the reporting capability, but the access and configuration is not there to get what we need. Has never been enough of a priority problem to get the focus it needs. Two outstanding questions:

- **Is this really the only option?**
- **Can we link PowerBI to MarkLogic to do analysis reports?**

Observation 4: Delta Jira ticket analysis split by issue type and status to identify trends

Overall, 2848 tickets have been created since 6th December 2018. The tables below show the detail broken down into issue type (including bugs and dev defects) and status by year.

Delta Tickets	*Status*							
Issue Type		Done	In Dev	Ready for Dev	To Do	Won't Do/Blocked/QA	Grand Total	Percentage
	0						0	
Bug		1060		3	20	99	1182	41%
Dev Defect		221	1	1	5	24	252	8%
Epic		2	2		72	6	82	2%
Improvement		1					1	
Problem		9			1		10	
Story		756	7	7	56	58	884	31%
Sub-task		273	1	1	5	22	302	9%
Task		110			14	10	134	5%
Technical Task		1					1	
Grand Total	**0**	**2433**	**11**	**12**	**173**	**219**	**2848**	

41% of the Delta ticket issues are relating to bug fixes, whereas less than 1% of tickets were relating to a system improvement. The number of tickets completed has increased by approx. 40 tickets a year since 2019. Lack of system enhancement and/or increase in users may have influenced this increase.

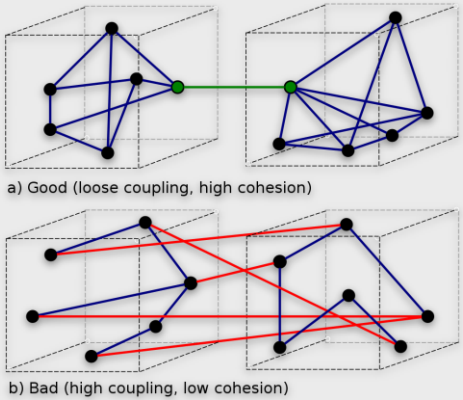
Technical barriers

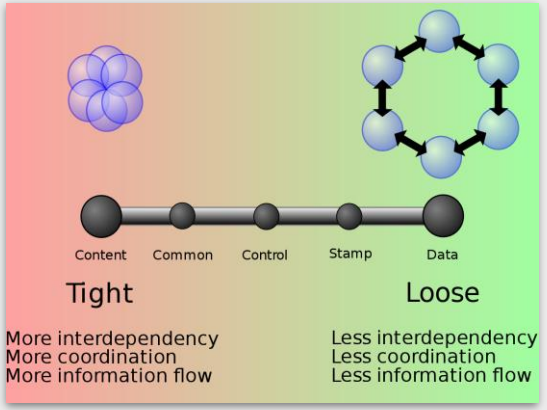
Department for Levelling Up, Housing & Communities

We need to reduce excessive coupling

What is coupling?

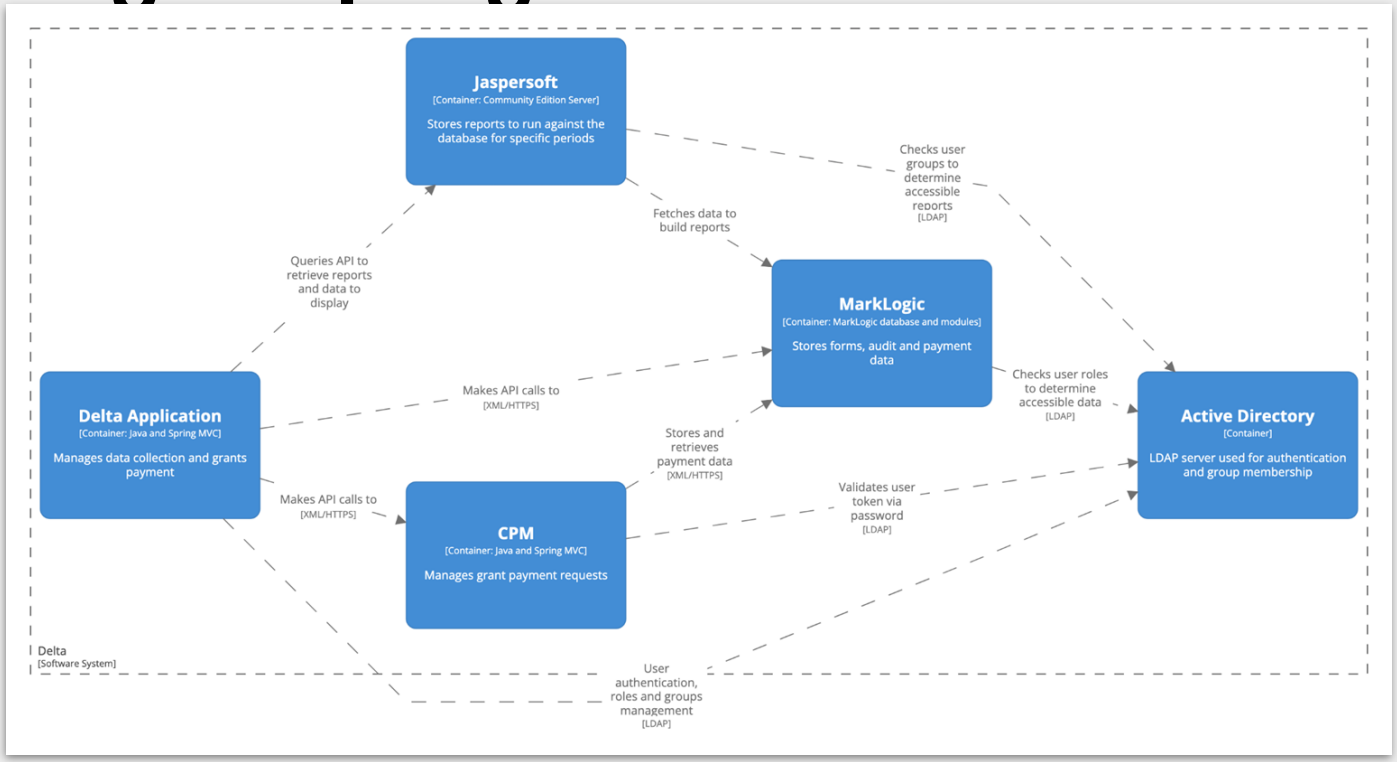
“In software engineering, coupling is the degree of interdependence between software modules; a measure of how closely connected two routines or modules are; the strength of the relationships between modules.”





Source: https://en.wikipedia.org/wiki/Coupling_(computer_programming)

Existing coupling in Delta



Coupling between software components

Coupling as we design software is a necessity

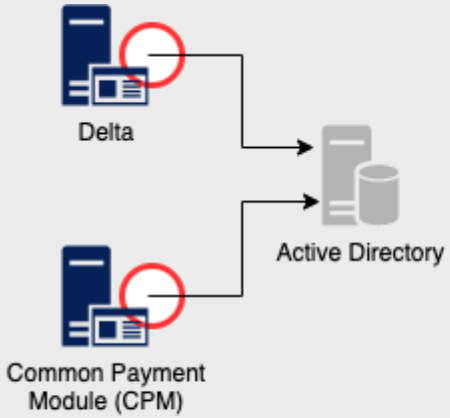
Communication between modules is based on contracts

A contract exposed to many consumers is hard to change

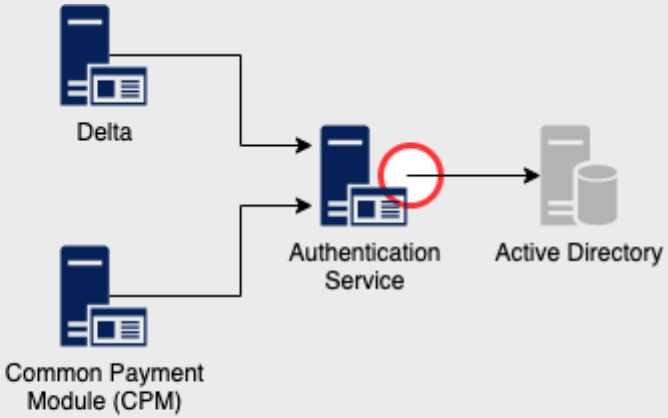
Group and isolate similar business concerns (cohesion)

Being directly dependent on software we do not control, we can’t make changes without impact

How to reduce coupling



Swapping technology changes the contract with both applications!



Now the technology change only affects its related application, the contract with other applications is preserved

Delta coupling with MarkLogic

Known issues

- Poor value for money (compared to alternatives)
- Not supported natively on GOV.UK PaaS
- Hiring suppliers or civil servants is difficult
- Hard to find open source software
- Hard to reuse code across Government

Coupling consequence

- MarkLogic interface has a large scope (low cohesion)

Delta coupling with Active Directory

Known issues

- Group and permissions are modelled as a directory
- Not supported natively on GOV.UK PaaS

Coupling consequence

- Dependencies need to use LDAP for communications

Feedback loops are too slow

No automated testing hinders change

Hard to detect defects in new versions

Cannot guarantee new features are fully working

Cannot guarantee previous features are still working

Dependent on manual testing capacity

We end up revisiting features we already have released

Relying on a release cycle hinders change

Cannot release a feature as soon as it is ready

Many features to test every release

Large changes often need a change board approval

Team knows a feature hasn’t been working many weeks later

Delta’s release model do not allow small and incremental changes to be deployed directly in production

Having basic monitoring hinders change

Miss valuable feedback and recurrent errors faced by users

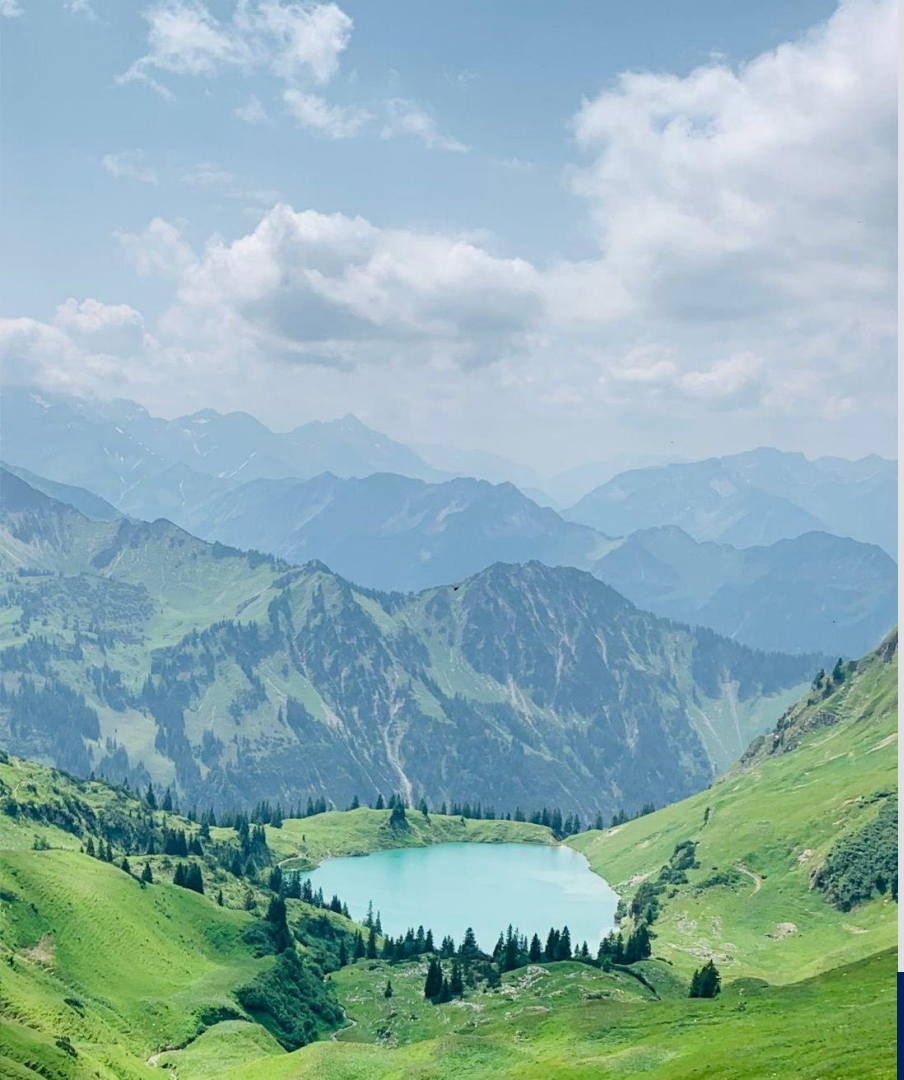
Hard to adjust server resources without performance metrics

Miss business insight based on user behaviour

Not having advanced metrics such as performance, site analytics and requests patterns could be a missed opportunity to improve Delta before issues start to appear

Recommendations

Department for Levelling Up,
Housing & Communities



Delta has a future

Our research and analysis has proven that Delta can play a key, positive role in the department's future.

But, for Delta to be the sustainable, reliable, cost efficient tool the department needs to support and enable its digital strategy and movement toward a 'data-as-a-service' model, it needs fundamental change.

Changing Delta without changing the technology will limit the value of change

We need to incrementally change the technology and then the design

A phased approach

We recommend the department adopts a phased approach to solving the problems we've identified.

The need to migrate away from Delta's existing underlying technology in order to enable other changes needed to solve problems relating to users' interaction with Delta and the value it provides.

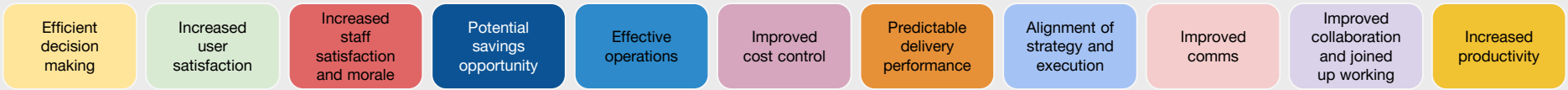
These sequential, intra- and post-migration phases, combined with some organisational change, will enable the department not only to solve the problems we've found, but also support delivery of its wider strategic goals.

These changes will deliver benefits

The benefits include:

- Support of the evolving DLUHC service landscape
- Enables sharing of best practices through development of open-source technology
- Cost savings through change of underlying technology
- Promotes use of Digital Service Standards
- Provision of a robust service that increases user's Trust and DLUHC's reputation
- Reduction of time spent on recurring Delta issues

Identified benefit categories



We'll measure the impact of change

Measures of success

- Reduction in overall Delta running costs
- Reduction in service desks calls with common themes i.e. password resets
- Monitor the sharing (re-use) of components between services
- User Satisfaction surveys (using the analysis in this report as the benchmark)

Migration plan

Department for Levelling Up, Housing & Communities

Migration plan objectives

- **Manage** complexity by tackling the right amount of change
- Build an evolving architecture supporting **incremental** and **guided** change
- Use fitness functions to assess if we are going in the **right direction**
- Use a testing first approach to **detect defects early**
- Use continuous deployment to **release small and often**
- Migrate to GOV.UK PaaS to **reduce costs**, **prototype** applications and **evolve our infrastructure** rapidly
- Start with the assumption that every service should be **open source**

We need to **manage** complexity

Managing change complexity

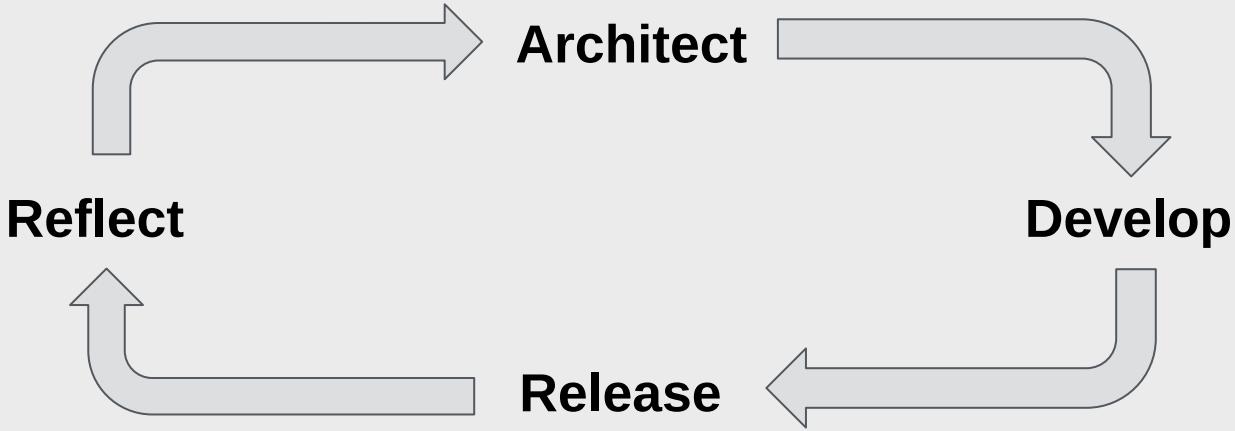
We recommend to replace at most two major components

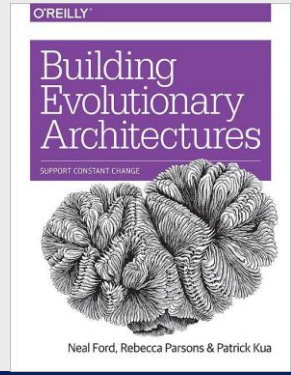
Phase I	**Complexity**
Active Directory	**MEDIUM**
MarkLogic	**HIGH**
Phase II	
Orbeon Forms	**HIGH**
Jaspersoft Reports	**LOW-MEDIUM**

We need to change incrementally

Evolving architecture

*An evolving architecture supports **incremental**, **guided** change as a first principle along **multiple dimensions***





We need to measure the fitness of our changes

Fitness functions

*An **objective** function that measures how close a given solution fits to a particular goal*

Three ways of implementing fitness functions

- Metrics
- Tests
- Process (how are we going to test this architecture is fit for the characteristics we want to optimise for?)

We need to improve testing

Testing First Approach

Why

- Reduce bugs by detecting them early
- Enables continuous deployment
- Enables quick experiments and fast feedback loop on changes

How

- Use test driven development
- Have each released feature paired with automated tests
- Ensure existing tests are passed for every released change

We need to release changes quicker

Continuous Deployment

Why

- Accelerate the time between ideation and delivery
- Have feedback on features as soon as possible
- Exercise tests and deployments

How

- Rely on our complete automated test suite
- Team practices aligned with CD such as trunk based development
- Have release pipelines created before pushing any code change

We need to
prototype
changes rapidly

Migrate to GOV.UK PaaS

Why

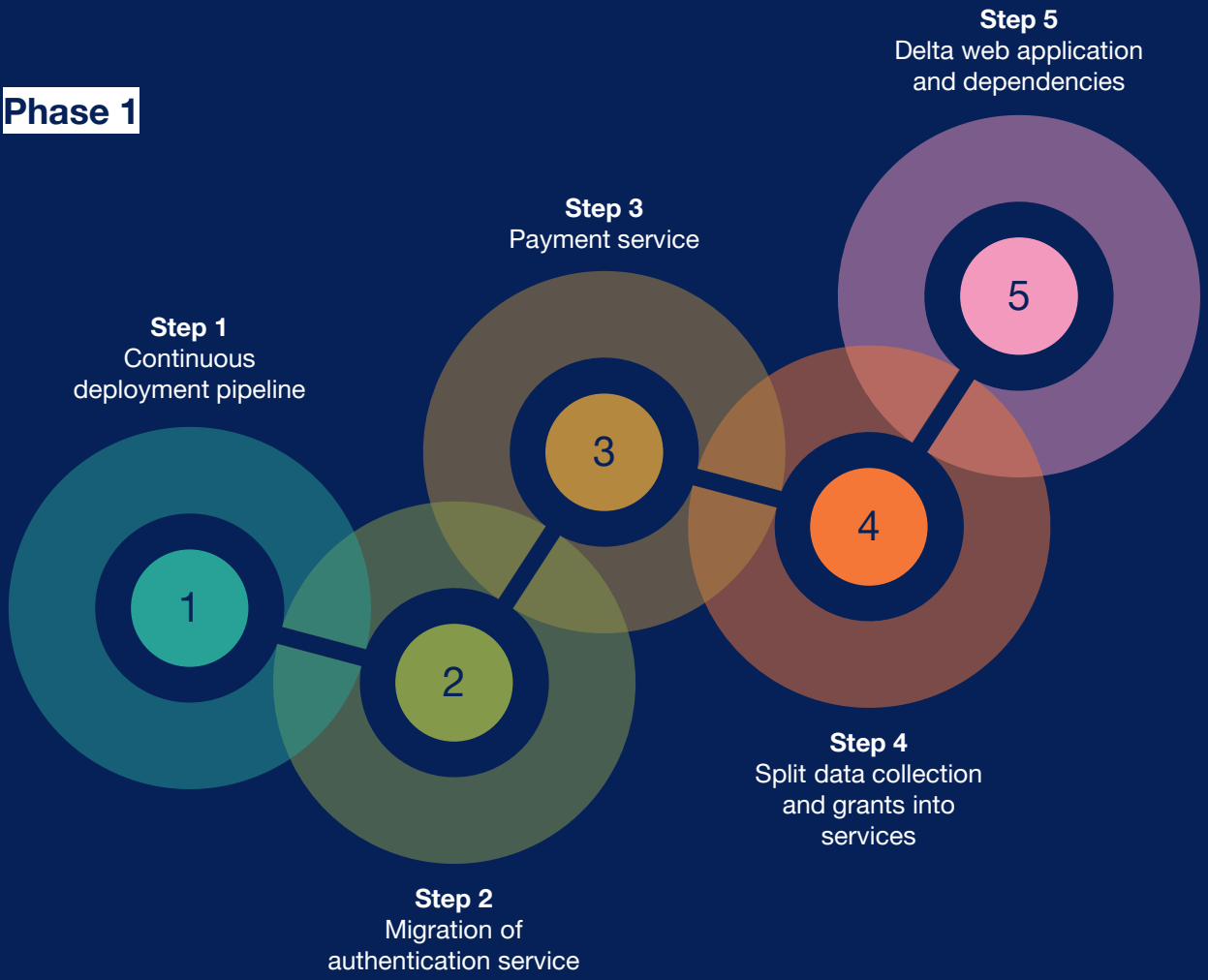
- Cost benefits for Government
- Managed and scalable infrastructure
- Deployments without user impact
- Developers create applications with databases in minutes
- Available metrics for monitoring

How

- Move away from MarkLogic
- Rebuild services with relational databases

We need a phased approach

Phase 1



Step 1
Continuous deployment pipeline

Step 2
Migration of authentication service

Step 3
Payment service

Step 4
Split data collection and grants into services

Step 5
Delta web application and dependencies

Phase 2

Recommendation 1
Revisit Orbeon Forms

Recommendation 2
Improve reporting and business intelligence

Recommendation 3
Improve user experience

Recommendation 4
Consider migrating E-Claims

A two phased approach

Phase 1: GOV.UK PaaS Migration (overview)

- Build continuous integration and delivery pipelines
- Release first improved software module (authentication service)
- Synchronise databases (migration)
- Integration with Delta
- Upon reaching our goal, iterate to new software module
- Move over Delta when all dependencies are migrated

A two phased approach

Phase 2: Post migration changes (overview)

- Consider alternatives to Orbeon Forms
- Reporting and business intelligence using SQL databases
- Review Delta user experience
- Address accessibility issues and compliance with Design System
- Assess if E-Claims could be migrated the same way as Delta

We need focused teams

A two teams approach

Technical team (BAU)

- Maintain Delta to a working state
- Support Delta's users
- Knowledge to help integration of new modules with Delta

Migration team

- Redesign software modules
- Maintain tests and engineering practices to high standards

Ensuring team resilience

Avoid having knowledge held by specific team members

- Pairing to share knowledge
- Technical documentation kept up-to-date (migrations)
- Documents stored on a DLUHC central repository rather than locally

All managed services should have a minimum of two members as administrators (civil servants ideally)

We recommend that regular handover checks are performed with suppliers and that resulting actions are followed up

Phase 1

Migrate to GOV.UK PaaS

GOV.UK PaaS migration

What this plan is

- A series of incremental steps
- A guideline with specific goals to achieve

What this plan is not

- Set in stone and cannot change
- Free of unforeseen obstacles

When presenting options, option 1 is our recommendation

Phase 1 Step 1

Continuous deployment using a service template

Continuous deployment pipeline

Why

- Need to be set up once, mostly consistent afterwards
- Allow developers to focus on changes
- Get features in production as soon as they are available

Measure of success

- Lead time decreasing (time b/w ideation and release)
- Faster feedback (time b/w dev complete and improvement raised)

Option 1: AWS CodeBuild and Terraform

Advantages

- Tested solution being used by EPBR
- Could reuse Terraform scripts adjusted for Delta

Disadvantages

- Require a DLUHC AWS user account for access and diagnostics
- Minor costs in addition to GOV.UK PaaS

Option 1: AWS CodeBuild and Terraform

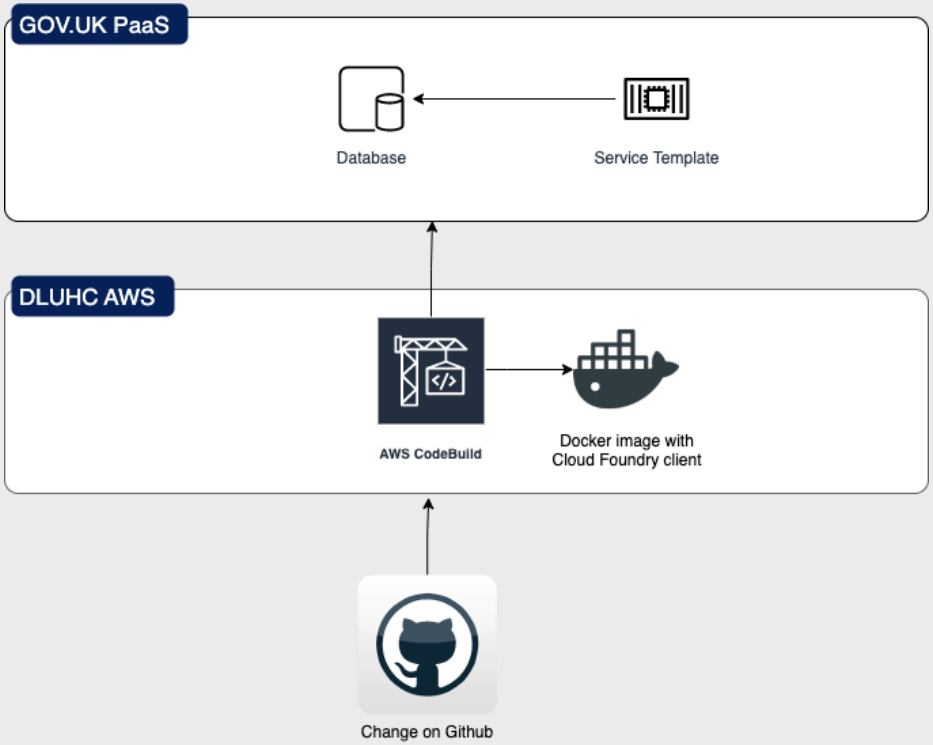
Technical considerations

Create a service template that make a simple query to its database,

Run the Terraform scripts which will create the pipeline on AWS CodeBuild, and set permissions to secrets (optional).

On the next commit (can be an empty one), the service should be deployed on GOV.UK PaaS.

Each service requires an health check endpoint, which can be called for testing.



Option 2: CI/CD managed alternatives

Advantages

- Github actions should already be paid and available

Disadvantages

- Secrets might need to be hosted on the managed platform
- Building the first pipeline might take some time

Option 2: CI/CD managed alternatives

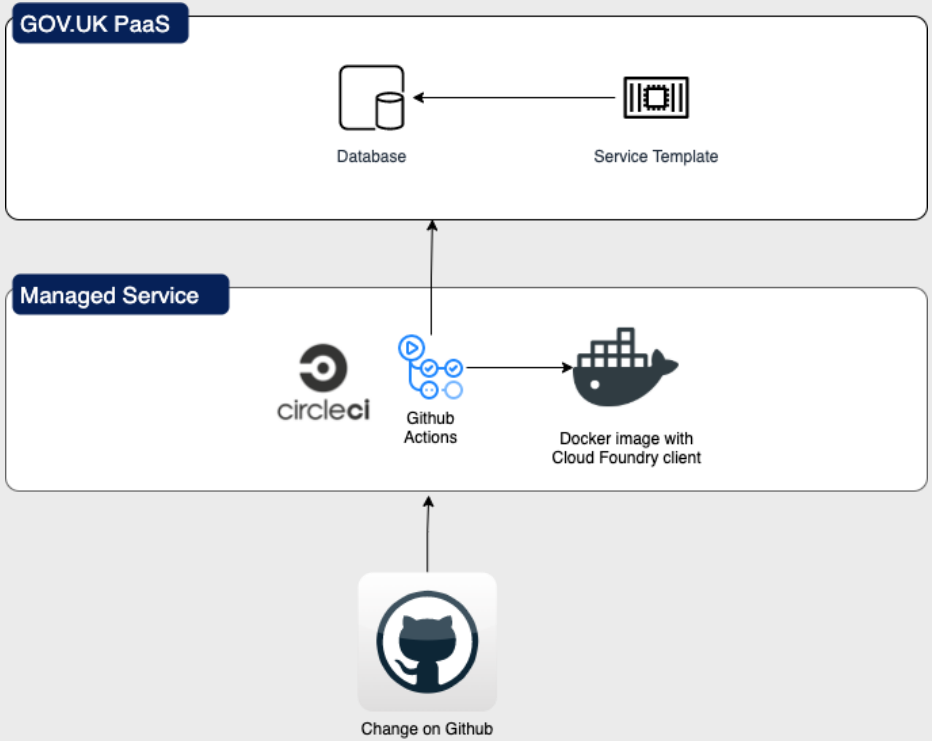
Technical considerations

Create a service template that make a simple query to its database.

Configure the pipeline in the managed service, check if they are code as infrastructure solutions.

On the next commit (can be an empty one), the service should be deployed on GOV PaaS.

Each service requires an health check endpoint, which can be called for testing.



Option 3: Host Jenkins on GOV PaaS

Advantages

- The Delta team already has experience with Jenkins
- We could reuse the existing configuration with some tweaks

Disadvantages

- We cannot leave the instance publicly available on the internet
- We need to use SSH tunnelling to access the instance
- We need to maintain an extra service

Option 3: Host Jenkins on GOV PaaS

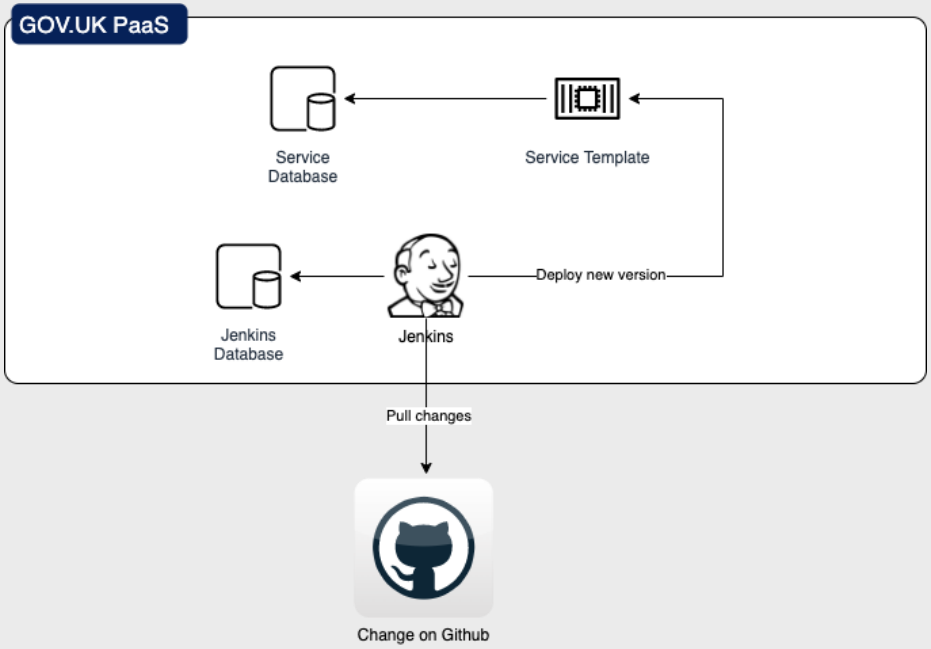
Technical considerations

Create a service template that make a simple query to its database.

Create a Jenkins instance with its database on GOV PaaS (docker image). Configure the service to pull changes repeatedly.

On the next commit (can be an empty one), the service should be deployed on GOV PaaS.

Each service requires an health check endpoint, which can be called for testing.



Phase 1 Step 2

Starting with the authentication service

Authentication Service

Why

- Better control on user groups and permissions
- Address password reset/policy issues
- Session timeouts
- Good service to start with, since it is not coupled with MarkLogic

Measure of success

- Password helpdesk tickets/total issues decreases (currently 70%)

Option 1: Create a new authentication service

Advantages

- Extract user management into a service
- Software could be reused by other teams or departments
- Could be reused from similar service created by CORE or EPBR

Disadvantages

- Complexity of building an authentication solution including 2FA

Option 1: Create a new authentication service

Technical considerations

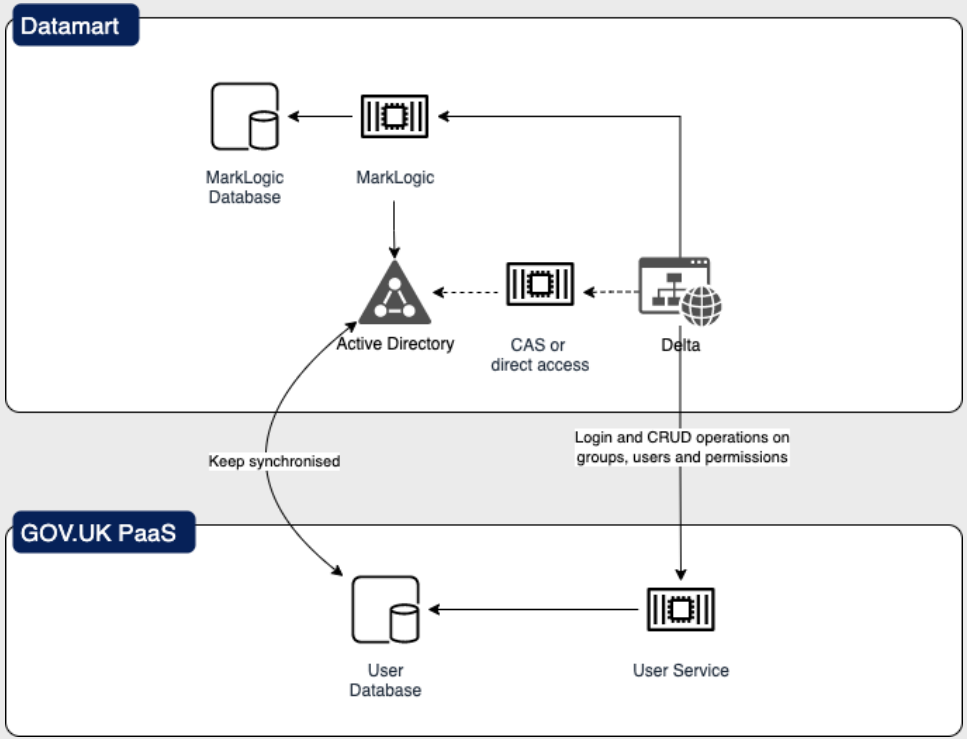
Build a service contract matching current login and user management needs

Synchronise Active Directory and the authentication database

- Dual Write or Change Data Capture (online)
- Database dump (offline)

Delta switch to use the new service

- JWT using asymmetric keys for validation
- SAML support required for MarkLogic
- Consider implementing feature flags



Option 2: Delegate authentication to CAS

Advantages

- Known technology since used by Delta
- Central Authentication Service (CAS) configuration change
- Supports many enterprise authentication methods

Disadvantages

- Users, groups and roles policies would still stay in Delta
- We rely on Apereo maintenance plans (but open sourced)

Option 2: Delegate authentication to CAS

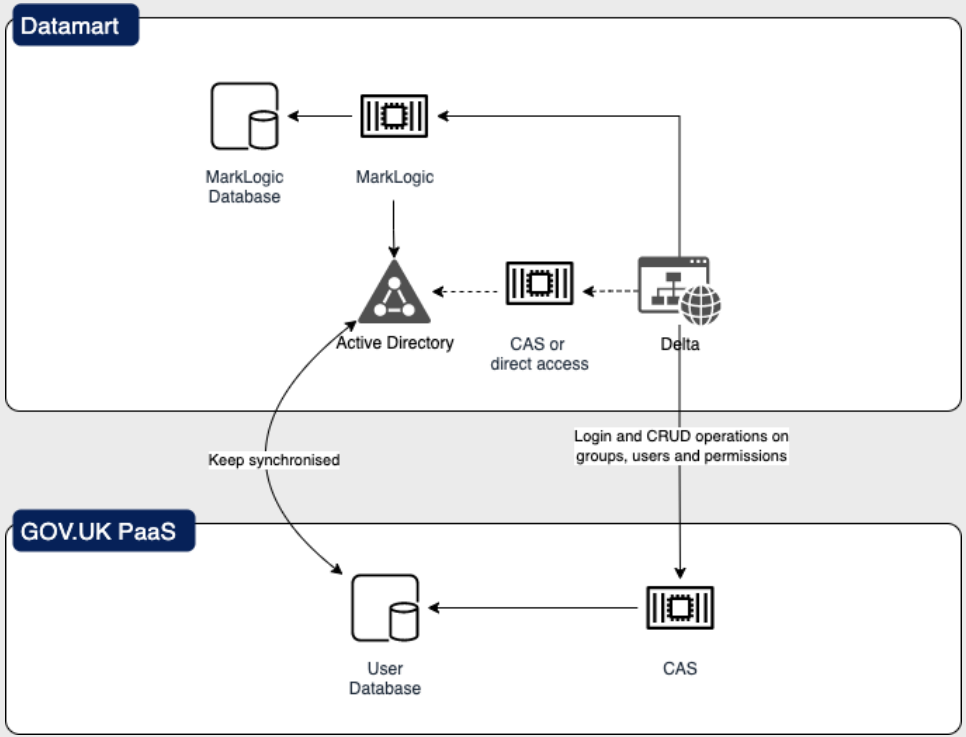
Technical considerations

Deploy CAS in GOV PaaS using database authentication

Synchronise Active Directory and the authentication database

- Dual Write or Change Data Capture (online)
- Database dump (offline)

Delta switch to use the new CAS instance



Phase 1 Step 3

Continuing with the payment service

Payment Service

Why

- Service expected to be fairly independent
- Refactor E-Claims endpoints
- Evolve the service for future uses

Measure of success

- Successful SAP integration on GOV PaaS

Option 1: Rebuild payment service

Advantages

- Introduce testing from the start
- Build a reusable service across DLUHC

Disadvantages

- May take more time than reusing CPM

Option 1: Rebuild payment service

Technical considerations

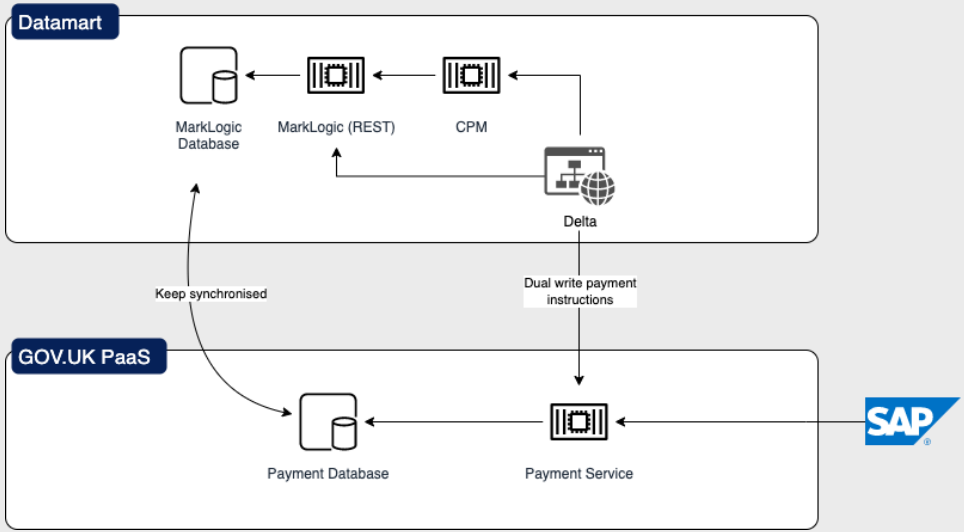
Assess the security concerns of having a payment service available publicly, by default restrict permission scope to strict minimum.

Build a service contract matching existing Common Payment Module (XML for SAP)

Synchronise MarkLogic and Payment databases

- Dual Write or Change Data Capture (online)
- Database dump (offline)

Delta switch to use the new service



Option 2: Deploy Common Payment Module

Advantages

- Work is limited to database changes

Disadvantages

- Introducing automated testing might be difficult

Option 2: Deploy Common Payment Module

Technical considerations

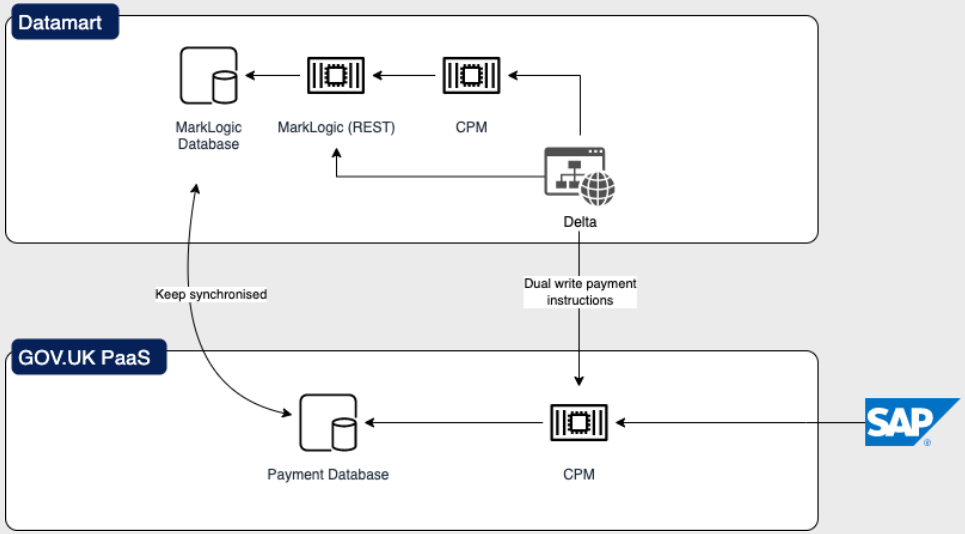
Assess the security concerns of having a payment service available publicly, by default restrict permission scope to strict minimum.

Deploy CPM with changes to use a relational database.

Synchronise MarkLogic and Payment databases

- Dual Write or Change Data Capture (online)
- Database dump (offline)

Delta switch to use the new service



Common technical considerations

There is a risk of creating payment duplication with dual writing

- Always be sure there is always a **single** source of truth

If the payment service cannot be accessible on the internet

- Define a specific API for SAP
- Make the payment service accessible from within GOV.UK PaaS

Phase 1 Step 4

Splitting data collection and grants into individual services

Splitting data collection and grants

Why

- Have services reflecting organisation
- Expose data for consumption via APIs
- Refactor the current Delta API into its dedicated service

Measure of success

- A new feature should only impact its related service
- The H-CLIC imports should still work post migration

The Delta user interface will stay the same for the time being

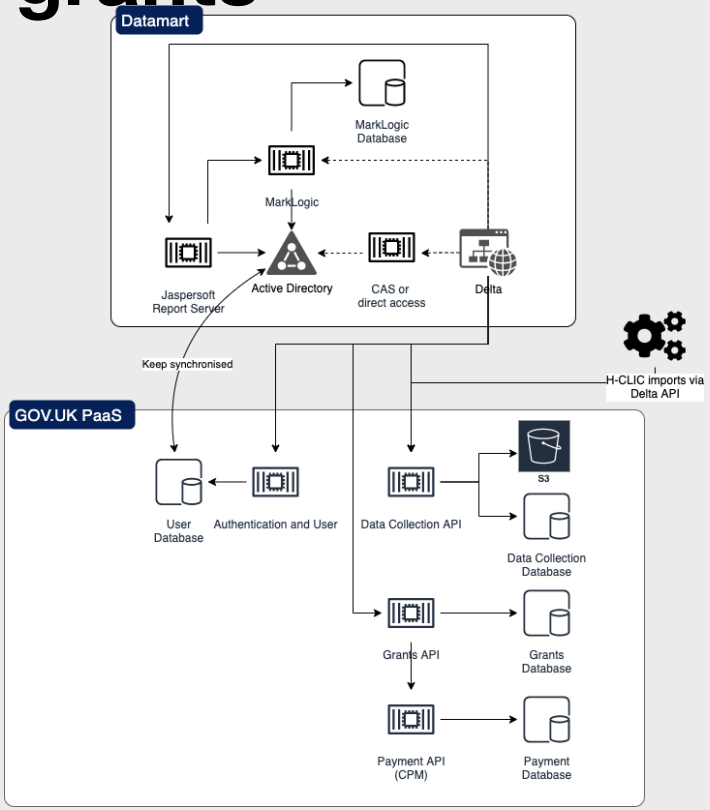
Splitting data collection and grants

Technical considerations

Payment Service already migrated.

The Grant service might have cross concerns with the data collection, this needs to be assessed during this step.

Set S3 bucket and share access keys for DAP exports.



Phase 1 Step 5

Finishing with the Delta web application and its remaining dependencies

Delta Website

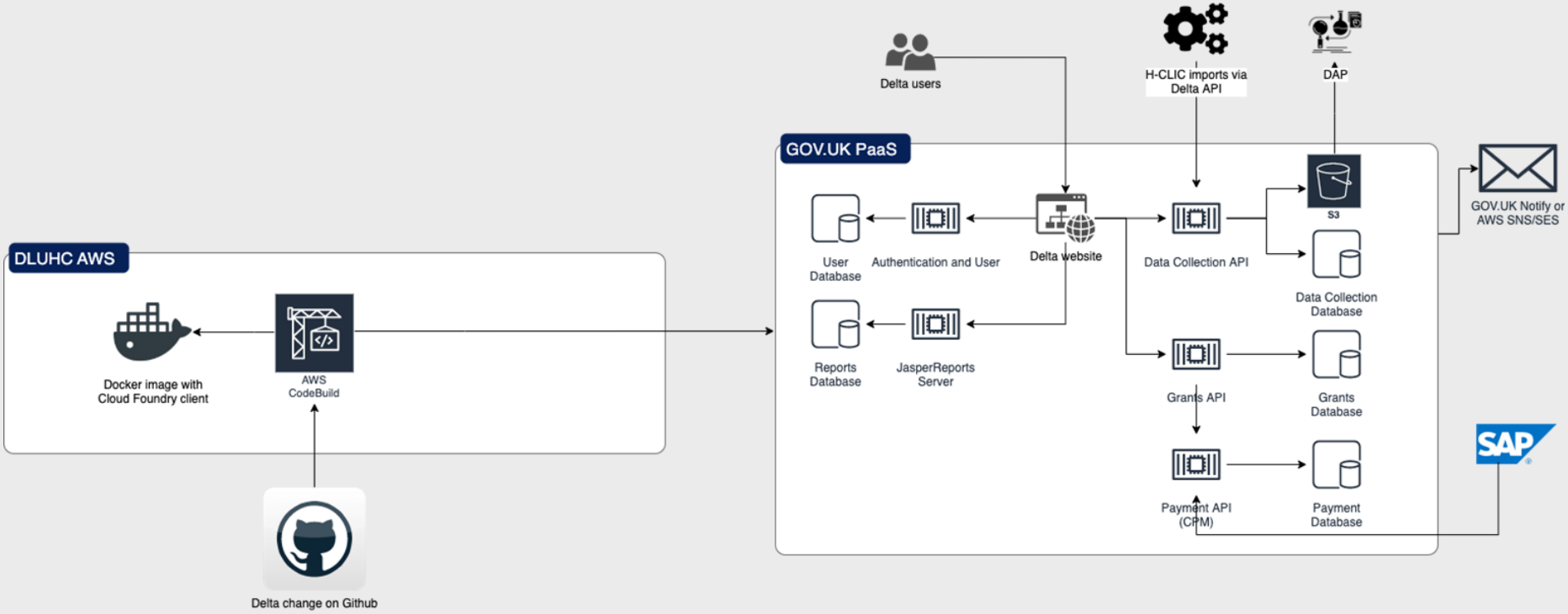
Why

- Complete Delta's migration
- Also includes dependencies such as Jaspersoft and an email service

Measure of success

- Infra + support costs decreases (expected 10x smaller)
- Decrease Orbeon Forms response time
- We can gather insight based on logging and monitoring

GOV PaaS migration final state



Technical considerations

Choose a managed solution for logging (or maintain service)

Extract monitoring data and display it (e.g. Grafana)

Monitor Orbeon Forms performance and assign memory as needed

Orbeon Forms might require direct access to the database

Assess DNS domain changes

Phase 2

Make post-migration changes

Post-migration expectations

Services are migrated into GOV PaaS

Services have a reliable suite of automated tests

Developers can evolve Delta with tests securying deployments

Developers can build prototypes and deploy them in minutes

Delta’s data is stored in multiple relational databases

We can now review the following recommendations in any order

Phase 2 Recommendation 1

Reconsider Orbeon Forms

Reconsider Orbeon Forms

Why

- Issues using Orbeon have been raised during many user interviews
- Using XQuery as a substitute for Excel formula is challenging
- Evaluate if there are simpler alternatives
- Being in control of the form builder design

How

- Create Delta prototypes relying on existing modules
- Assess migrating from Orbeon forms to a generic form structure

Phase 2 Recommendation 2

Improve Delta’s reporting and business intelligence

Reporting and business intelligence

Why

- Delta's data is now available in relational databases
- Developers within DLUHC are more familiar with SQL software
- Availability of open source solutions

How

- Start with tooling approved by DLUHC IT
- Consider replacing Jaspersoft
- Consider the impact on production databases (scale or duplicate)

Phase 2 Recommendation 3

Focus on improving Delta user experience

Focus on improving Delta user experience

Why

- Business logic should be moved of Delta at this stage
- Make Delta compliant with accessibility standards
- Realign with government frameworks (design system)

How

- Pluridisciplinary team with a user experience designer & User Researcher
- Iterations through prototyping

Phase 2 Recommendation 4

Consider migrating E-Claims to GOV PaaS

Consider migrating E-Claims to GOV.UK PaaS

Why

- Decommission the Datamart

How

- Extract business logic from MarkLogic and E-Claims into services
- Migrate all data in a relational database