FUTURE CITIES
Global Future Cities Programme

UK Government

Cabinet Office

Global Future Cities Programme

UKGDS Workshop - Digital and Technology Standards

7 April 2021
3.00pm – 4.30pm

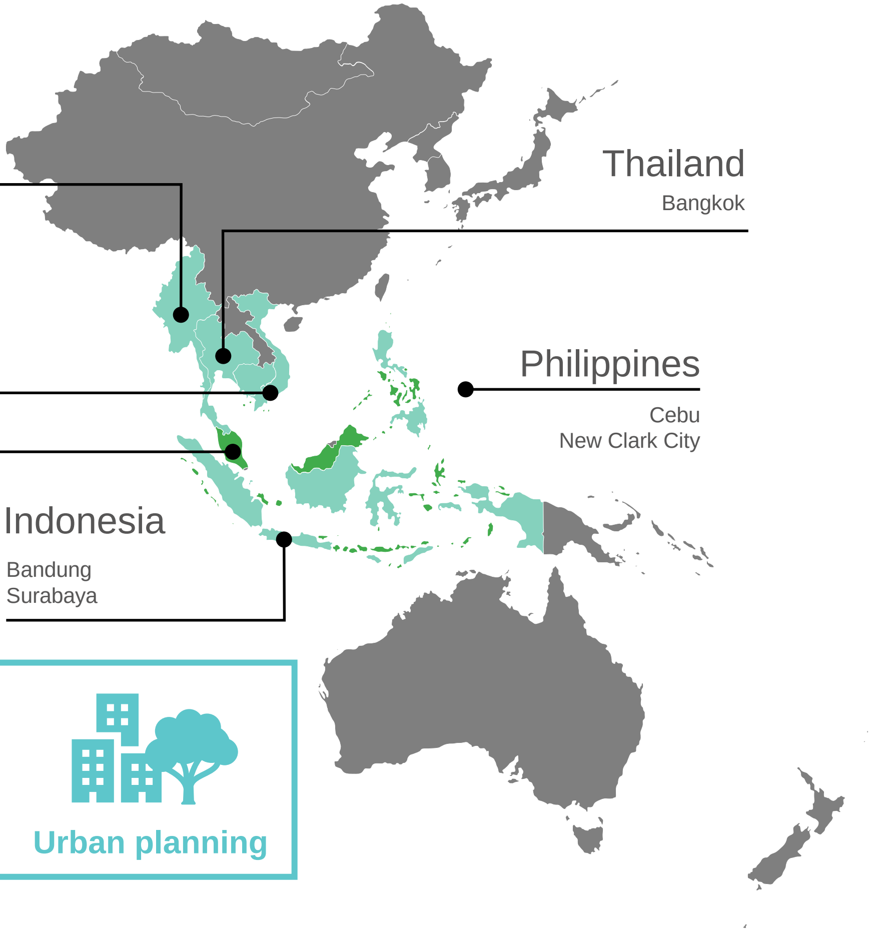
Global Future Cities Programme

13 projects

9 cities

6 countries

3 years



Myanmar
Yangon

Thailand
Bangkok

Vietnam
Ho Chi Minh City

Philippines
Cebu
New Clark City

Malaysia
Melaka
Iskandar Malaysia

Indonesia
Bandung
Surabaya

Resilience

Transportation

Urban planning

Introduction to the Speakers

Tony Richards

- Standards and Assurance lead for the Global Digital Marketplace programme.
- An advocate for the use of digital and technology standards to drive digital transformation.

Cabinet Office

Introduction to the Speakers

Paloma Jain

- Senior User Researcher with expertise in testing of procurement related services.

- Works with international governments on embedding user research practises on all projects.

Cabinet Office

Introduction to the Speakers

Ben Vandersteen

- A career technologist in development and architecture.
- Worked on developing the UK Digital Marketplace to support the UK Government's digital transformation

Cabinet Office

Learning Outcomes

By the end of this session, you should be able to:

1. Understand the importance of standards setting in digital transformation;

2. Consider some areas where standards setting could be of use to their organisation.

Cabinet Office

Tony Richards
Standards & Assurance Lead
Global Digital Marketplace Programme

Working with Digital Penang.

How to add value to standards development.

Global
Digital Marketplace

Who are Digital Penang?

Why is standards creation important?

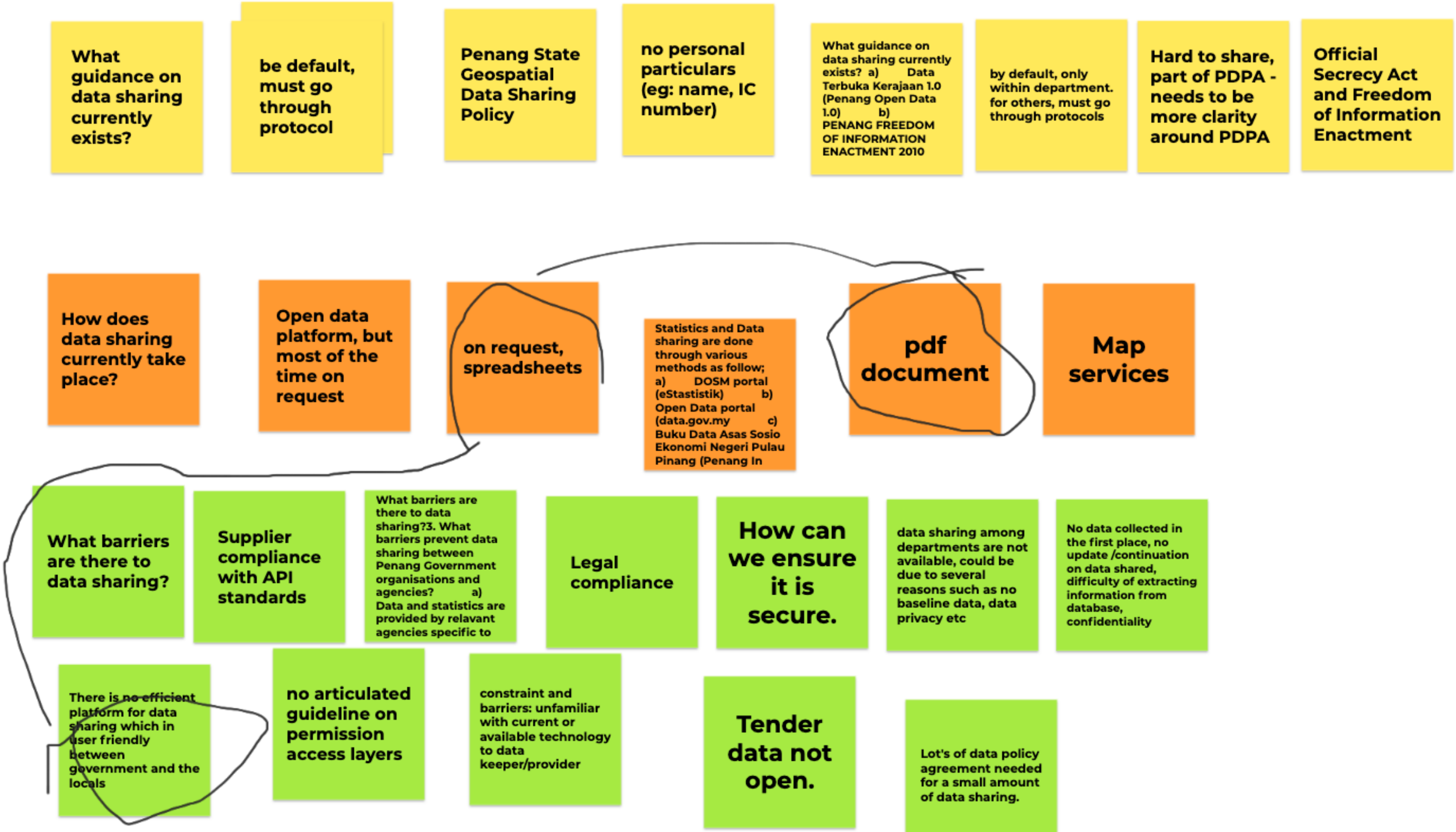
Standards in government help teams follow best practices when designing and building services. They can also be used by a central function with financial controls or other powers to ensure that all services meet a minimum level of quality.

Three tips

Keep it multi-disciplinary up front

Link to partners aims

Research early, research often



What guidance on data sharing currently exists?

be default, must go through protocol

Penang State Geospatial Data Sharing Policy

no personal particulars (eg: name, IC number)

What guidance on data sharing currently exists? a) Data Terbuka Kerajaan 1.0 (Penang Open Data 1.0) b) PENANG FREEDOM OF INFORMATION ENACTMENT 2010

by default, only within department. for others, must go through protocols

Hard to share, part of PDPA - needs to be more clarity around PDPA

Official Secrecy Act and Freedom of Information Enactment

How does data sharing currently take place?

Open data platform, but most of the time on request

on request, spreadsheets

Statistics and Data sharing are done through various methods as follow; a) DOSM portal (eStastistik) b) Open Data portal (data.gov.my c) Buku Data Asas Sosio Ekonomi Negeri Pulau Pinang (Penang In

pdf document

Map services

What barriers are there to data sharing?

Supplier compliance with API standards

What barriers are there to data sharing?3. What barriers prevent data sharing between Penang Government organisations and agencies? a) Data and statistics are provided by relavant agencies specific to

Legal compliance

How can we ensure it is secure.

data sharing among departments are not available, could be due to several reasons such as no baseline data, data privacy etc

No data collected in the first place, no update /continuation on data shared, difficulty of extracting information from database, confidentiality

There is no efficient platform for data sharing which in user friendly between government and the locals

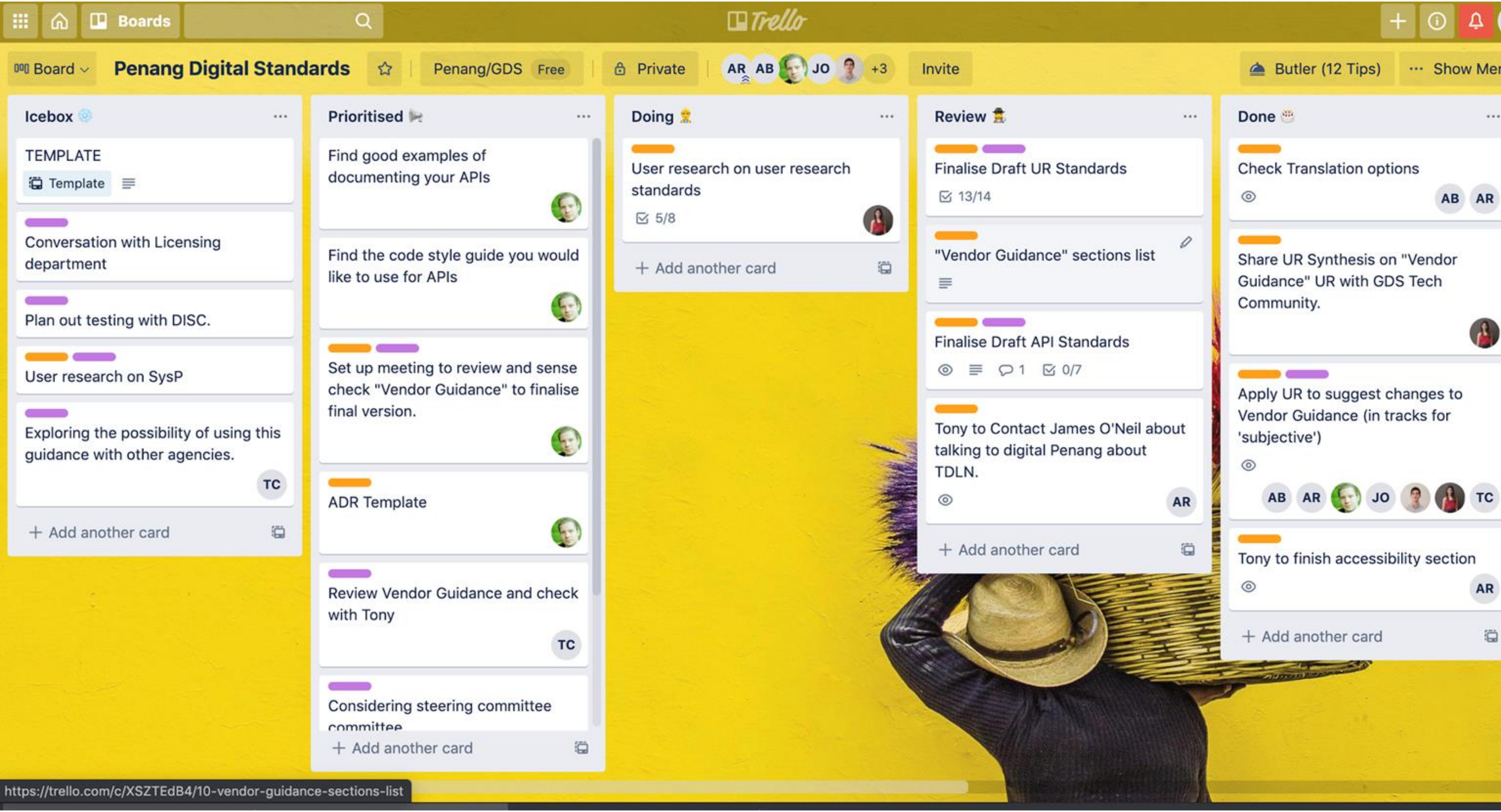
no articulated guideline on permission access layers

constraint and barriers: unfamiliar with current or available technology to data keeper/provider

Tender data not open.

Lot's of data policy agreement needed for a small amount of data sharing.

Bonus: Introduce and embed a new way of working



Penang Digital Standards

Icebox

- TEMPLATE — Template
- Conversation with Licensing department
- Plan out testing with DISC.
- User research on SysP
- Exploring the possibility of using this guidance with other agencies. — TC
- + Add another card

Prioritised

- Find good examples of documenting your APIs
- Find the code style guide you would like to use for APIs
- Set up meeting to review and sense check "Vendor Guidance" to finalise final version.
- ADR Template
- Review Vendor Guidance and check with Tony — TC
- Considering steering committee committee
- + Add another card

Doing

- User research on user research standards — ☑ 5/8
- + Add another card

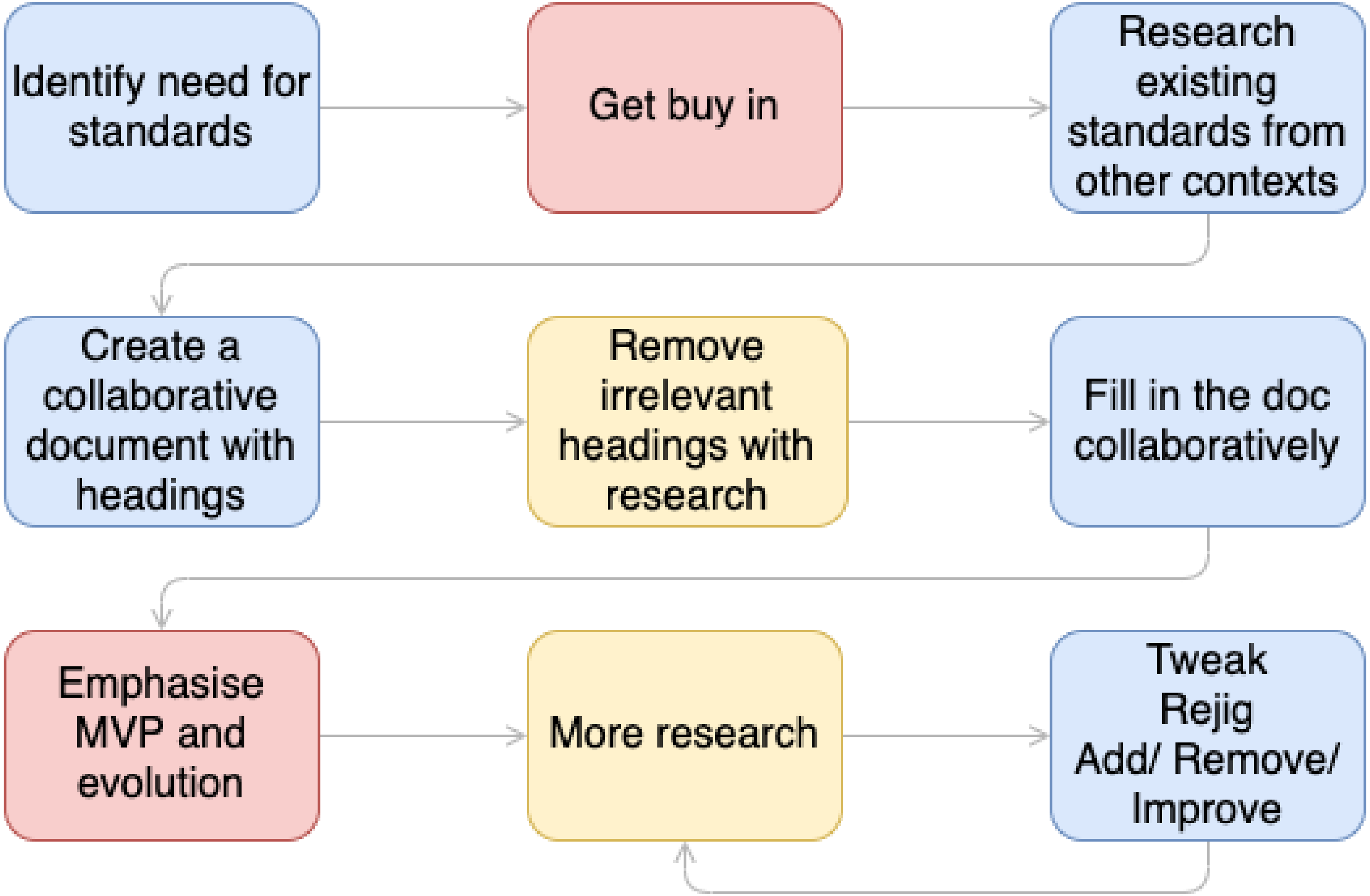
Review

- Finalise Draft UR Standards — ☑ 13/14
- "Vendor Guidance" sections list
- Finalise Draft API Standards — 1 — ☑ 0/7
- Tony to Contact James O'Neil about talking to digital Penang about TDLN. — AR
- + Add another card

Done

- Check Translation options — AB AR
- Share UR Synthesis on "Vendor Guidance" UR with GDS Tech Community.
- Apply UR to suggest changes to Vendor Guidance (in tracks for 'subjective') — AB AR JO TC
- Tony to finish accessibility section — AR
- + Add another card

https://trello.com/c/XSZTEdB4/10-vendor-guidance-sections-list



Identify need for standards
Get buy in
Research existing standards from other contexts
Create a collaborative document with headings
Remove irrelevant headings with research
Fill in the doc collaboratively
Emphasise MVP and evolution
More research
Tweak
Rejig
Add/ Remove/ Improve

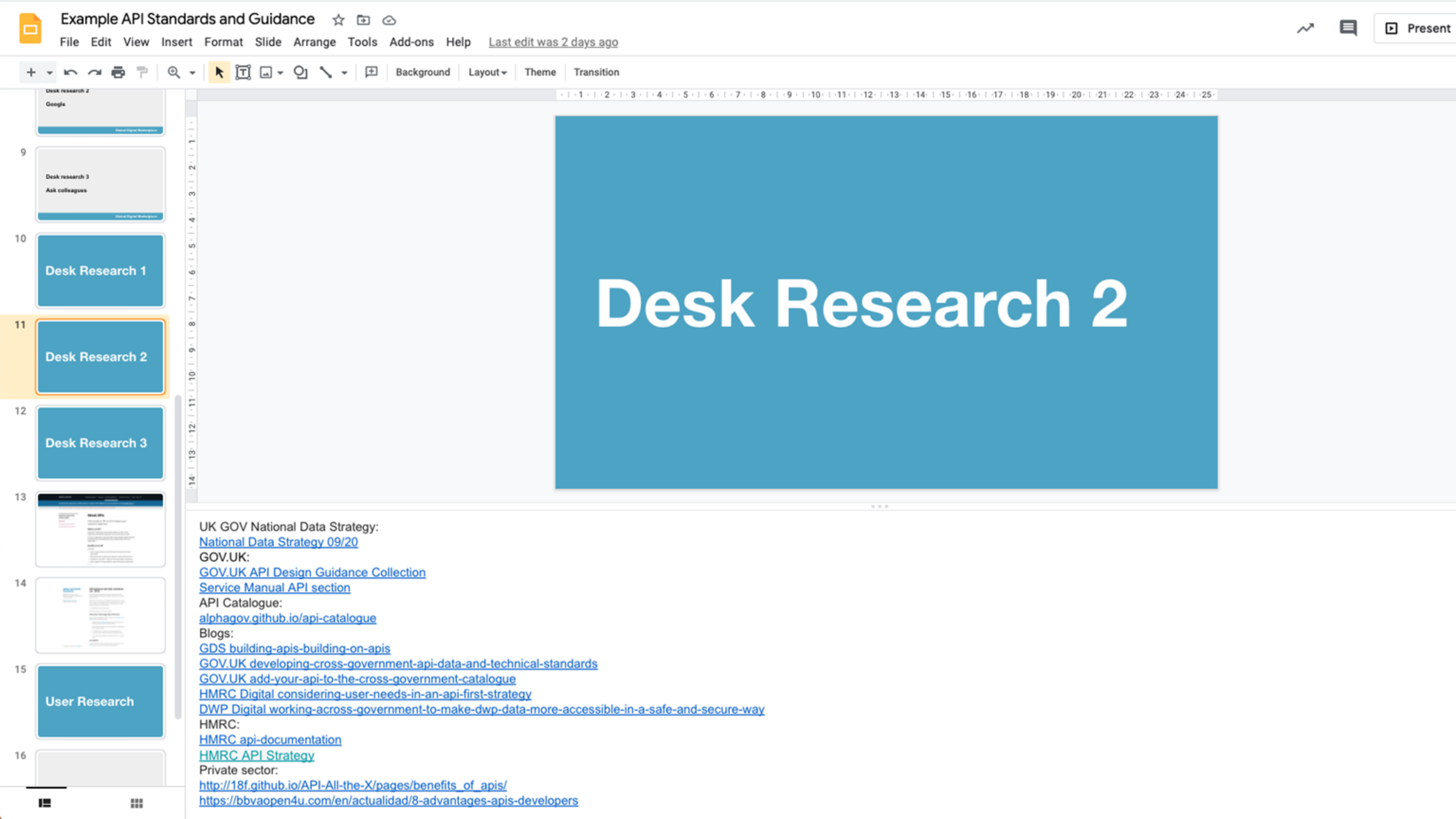
Cabinet Office

Ben Vandersteen

Technical Architect

Global Digital Marketplace Programme

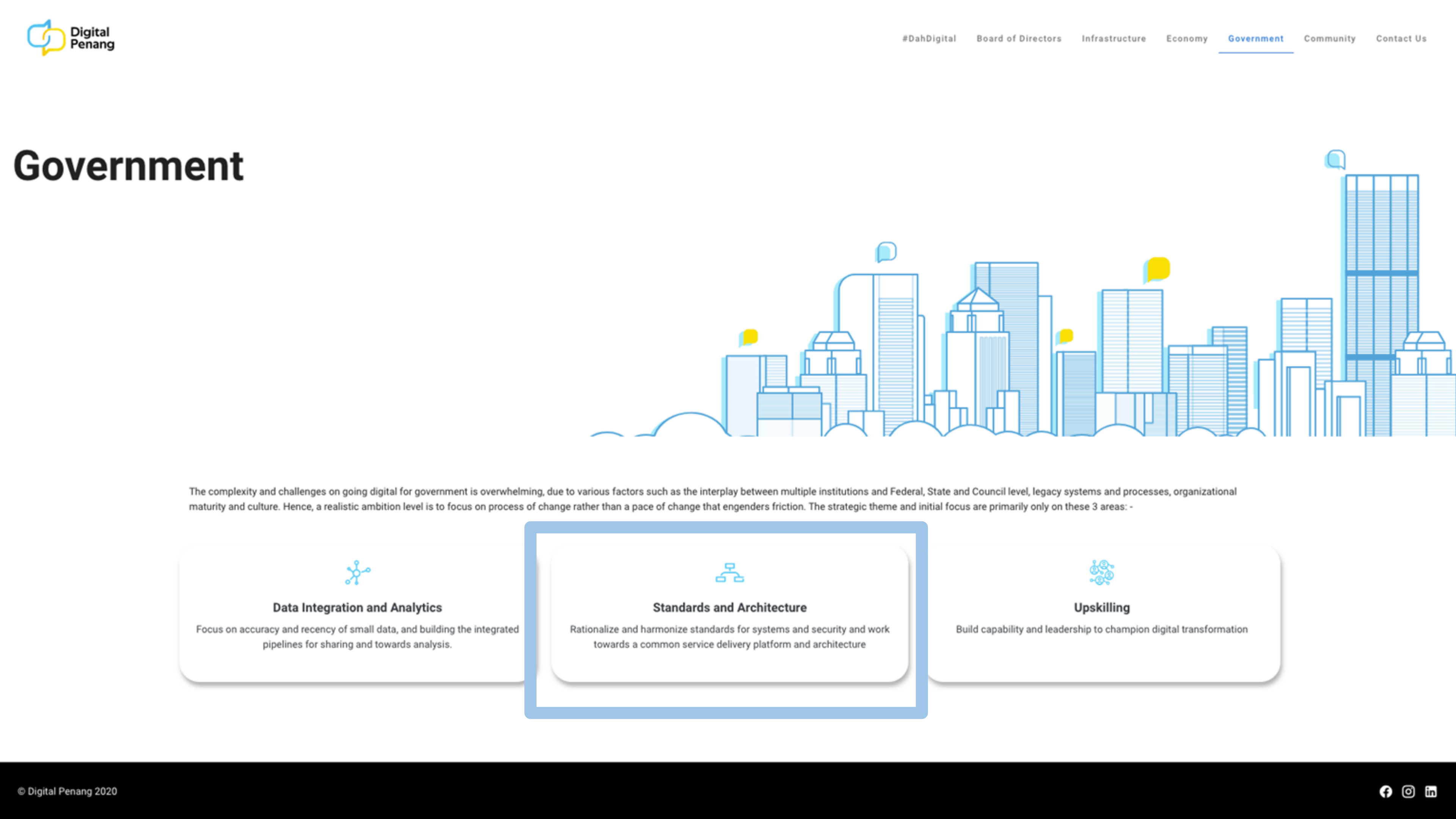
Keep it multi disciplinary up front



Desk Research 2

UK GOV National Data Strategy:
National Data Strategy 09/20
GOV.UK:
GOV.UK API Design Guidance Collection
Service Manual API section
API Catalogue:
alphagov.github.io/api-catalogue
Blogs:
GDS building-apis-building-on-apis
GOV.UK developing-cross-government-api-data-and-technical-standards
GOV.UK add-your-api-to-the-cross-government-catalogue
HMRC Digital considering-user-needs-in-an-api-first-strategy
DWP Digital working-across-government-to-make-dwp-data-more-accessible-in-a-safe-and-secure-way
HMRC:
HMRC api-documentation
HMRC API Strategy
Private sector:
http://18f.github.io/API-All-the-X/pages/benefits_of_apis/
https://bbvaopen4u.com/en/actualidad/8-advantages-apis-developers

Link to partners aims



Government

The complexity and challenges on going digital for government is overwhelming, due to various factors such as the interplay between multiple institutions and Federal, State and Council level, legacy systems and processes, organizational maturity and culture. Hence, a realistic ambition level is to focus on process of change rather than a pace of change that engenders friction. The strategic theme and initial focus are primarily only on these 3 areas: -

Data Integration and Analytics

Focus on accuracy and recency of small data, and building the integrated pipelines for sharing and towards analysis.

Standards and Architecture

Rationalize and harmonize standards for systems and security and work towards a common service delivery platform and architecture

Upskilling

Build capability and leadership to champion digital transformation

Research early.
Research often.

Headings from the GDS Way

- **The GDS Way**
  - About the GDS way
  - How to add new guidance
  - The GDS Way Forum
- **Software development**
  - How to name software products
    - Your product name should be self-descriptive
    - Further reading
  - Choosing a programming language
    - Frontend development
    - Backend development
      - Python
      - Go
      - Languages we do not use for new projects
    - Using other languages
  - Style guides
    - CSS/Sass
      - Contents
      - Whitespace
      - Spacing
        - Use with deprecated libraries
        - Include fallbacks for rem spacing
          - Using rem units with deprecated libraries
      - Vendor prefixing
      - Sass nesting
    - Docker
      - Using tags and digests in FROM instructions
      - Running programs as process ID (PID) 1
      - Subshells
      - Links
    - Go
      - Code formatting
      - Code checking
      - External dependencies
      - Web frameworks
      - Channels
      - Signalling
      - Testing
      - Configuration parsing
    - HTML
      - Browser support
      - Document structure
        - Header
        - Main content
        - Sectioned content
        - Navigational elements
        - Aside
        - Footer
      - Individual element guidance
        - Headings
        - Text emphasis
        - Images
        - Buttons vs Links
        - Visually hidden elements
    - Java
      - Code formatting
      - Dependency injection (DI)
      - Imports
      - Optionals
      - Local variable type inference (the var keyword)
      - Prefer functionality in the Java standard library
      - Comments
      - Testing

        - Why standard?
      - Whitespace
      - Naming conventions
      - CoffeeScript
      - HTML class hooks
      - Styling elements
      - Strict mode
      - Modules
      - Module structure
      - jQuery
      - Supporting older browsers
      - Method arguments
    - Node.js
      - Introduction
      - Node versions
      - Source formatting and linting
      - Project directory structure
      - Language constructs
        - Declarations
        - Functions
        - Classes
        - Asynchronous code
          - Use of async/await
        - Functional programming
      - Errors
      - Node.js's HTTP server
      - Transpiling
      - Frameworks
      - Libraries
      - Node Package Manager (NPM)
        - See also:
      - Starting a new project
      - Further reading
      - Updating this manual
    - Python
      - Code formatting
        - Maximum line length of 120 characters
      - Linting
        - Flake8
          - What is Flake8?
        - How to use Flake8
        - Example usage
        - Plugins
        - Flake8 per file ignores
        - Common Configuration
        - Linting resources
      - Dependencies
        - Applications
        - Libraries
      - Updating this manual
    - Ruby
      - Code formatting
      - Conventional tooling
      - Further reading
  - How to manage third party software dependencies
    - Update dependencies frequently
      - Rebuild Docker base images
    - Monitor for vulnerabilities
      - Use official Docker base images
    - Minimise what you need to monitor
  - Building accessible services
    - How to make your service accessible
      - Consider accessibility from the start
      - Understand that not everyone reads content the same way
      - Automated testing
      - Browser compatibility testing
      - Test content served by third party systems

        - Structure
          - Example
      - Branching/merging conventions
      - Do not use git push -f, use --force-with-lease instead
  - Using Pull Requests
    - Why should you use pull requests?
    - Cautionary notes
    - Guidance for each step
      - Opening a pull request
      - Reviewing a request
        - Guidelines for review
        - Communicate with others who may consider reviewing the PR
        - Helpful things to consider while reviewing
      - Addressing comments
      - Reviewing external pull requests
        - Tone
        - Handling the PR
        - Practical advice
    - Further reading
  - Writing READMEs
    - Length of your README
    - Structuring your README
    - Test your documentation
  - Writing release notes
    - Using code examples
    - Guidance outside GDS
  - Licensing
    - Guidelines for repositories containing code
      - Use MIT
      - Copyright notice
      - Example
      - Guidelines for repositories that are open documentation
        - Example
  - Use continuous delivery
    - Iterate code frequently
    - Low-risk releases
    - Quality software builds
    - Maintainable code
    - Essential concepts
      - Frequent integrations with master branch
      - Automatic build promotion
      - Use production monitoring and alerting
    - How to measure continuous delivery
    - Further reading
- **Hosting and infrastructure**
  - How to host a service
    - Consider vendor switching costs
  - How to manage DNS records for your service
  - Use configuration management
    - Puppet
    - Terraform
      - Versioning
      - Code analysis
    - Further reading
  - Operating systems for virtual machines
    - Amazon Web Services virtual machines
    - Non-AWS and Debian virtual machines
    - Hardening
  - Use a web application firewall (WAF)
    - Why you should use a WAF
    - When and how to use a WAF
    - Managing your WAF
      - Alerts
      - Threat modelling
      - Reviews

      - Structured logging with Logstash
        - HTTP fields
      - Advice for particular frameworks or platforms
        - Dropwizard
        - Cloud Foundry
    - Log shipping
  - How to monitor your service
    - Using metrics-based monitoring
  - How to manage alerts
    - Sending alerts
      - When you should not send an alert
    - Prioritising alerts
    - Further reading
  - Make data-driven decisions with Service Level Objectives (SLO
    - What you'll need to know before setting your SLOs
      - Current SLI levels
      - Your dependencies
      - User satisfaction
    - Setting your first SLOs
      - Use the SLIs as a reference
      - Choose a time window
      - Choose a SLO Target
      - Create error budgets
      - SLO burn rate and alerts
      - Using error budgets to create policy
      - Continuous improvement of SLOs
    - Further reading
  - Run a Service Level Indicator (SLI) workshop
    - Run your workshop
      - 1. Prioritise your most important user journeys
      - 2. Map your user journeys
      - 3. Define what good means to users
      - 4. Map out high-level system components
      - 5. Define your SLIs
      - 6. Create implementation tasks
      - 7. Observe and iterate your SLIs
    - Case study: Reliability Engineering Observe team
      - Prioritising user journeys
      - Mapping user journeys
      - Defining what good means
      - Mapping out high-level system components
        - Choosing a Grafana dashboard: high-level sys components
      - Defining the SLIs
        - Choosing a Grafana dashboard: points of meas
      - Creating tasks and iterating SLIs
    - Contact Reliability Engineering
    - Further reading
- **Operating a service**
  - Understand the risks to your service
    - Model security threats
    - Further Reading
  - How to track technical debt
    - Example consequences of tech debt
    - Example causes of tech debt
    - Classifying and measuring
    - Process
    - Example
      - Whitehall has its own upload management system
  - How to manage access to your third-party service accounts
    - Securely managing account credentials
      - Managing accounts with organisation-level access
      - Managing accounts with individual-level access
    - Managing GitHub accounts
    - Managing Logit accounts
    - Managing Amazon Web Services (AWS) accounts
  - How to do penetration testing

Cabinet Office

Paloma Jain

Senior User Researcher

Global Digital Marketplace Programme

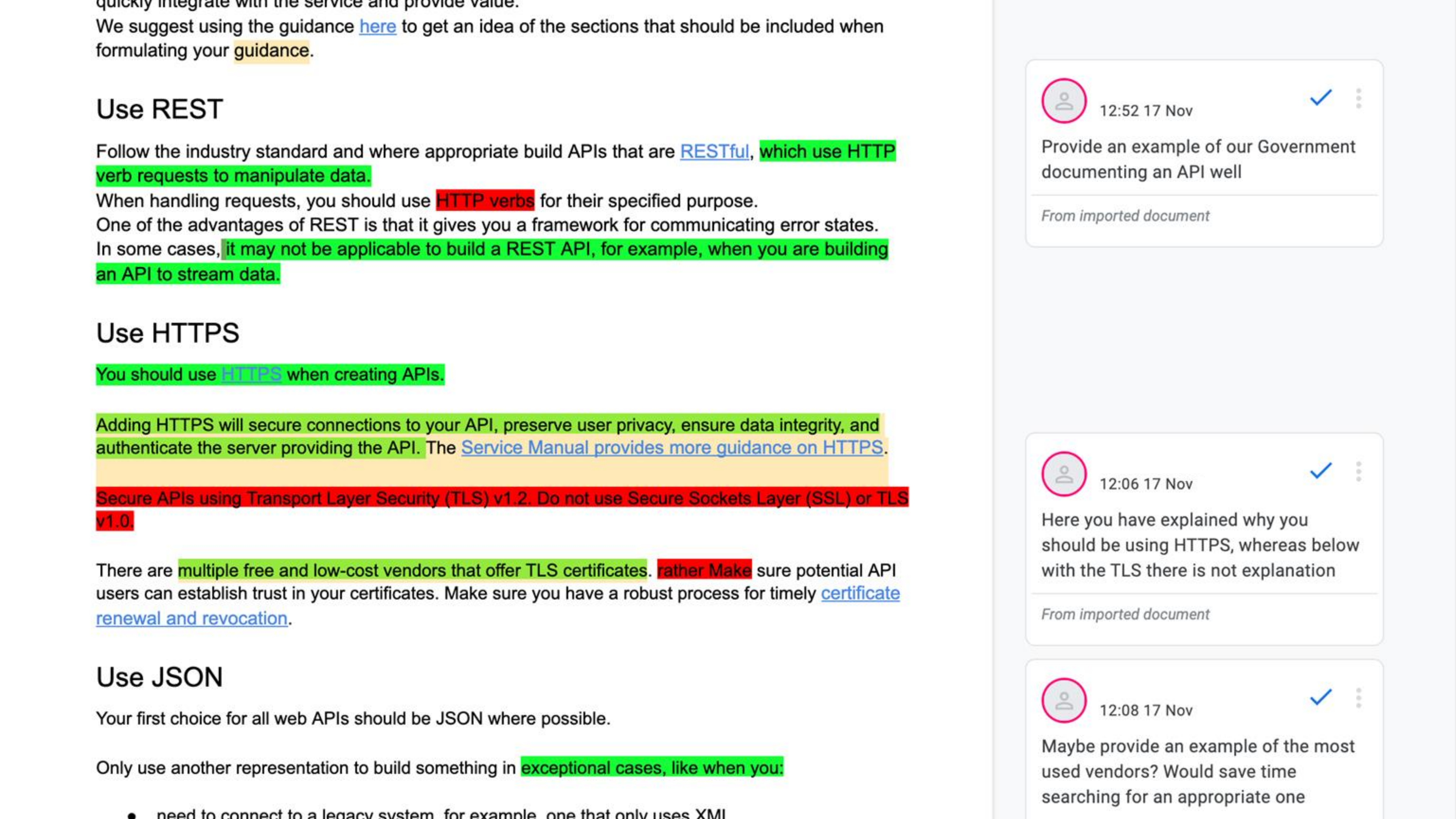
API & Vendor standards

We spoke to 5 participants (5M, 0F)

- 2x Junior developers
- 3x Senior developers
- 2 - 5 years tech experience
- All aware of APIs (not experts)

We did highlighter testing:

1. Read the document in your own time (~45mins)
2. Highlight useful or interesting info in green
3. Highlight unhelpful, strange or difficult to understand info in red
4. Debrief 45 minute meeting to talk through highlights



quickly integrate with the service and provide value.
We suggest using the guidance here to get an idea of the sections that should be included when formulating your guidance.

Use REST

Follow the industry standard and where appropriate build APIs that are RESTful, which use HTTP verb requests to manipulate data.
When handling requests, you should use HTTP verbs for their specified purpose.
One of the advantages of REST is that it gives you a framework for communicating error states.
In some cases, it may not be applicable to build a REST API, for example, when you are building an API to stream data.

Use HTTPS

You should use HTTPS when creating APIs.

Adding HTTPS will secure connections to your API, preserve user privacy, ensure data integrity, and authenticate the server providing the API. The Service Manual provides more guidance on HTTPS.

Secure APIs using Transport Layer Security (TLS) v1.2. Do not use Secure Sockets Layer (SSL) or TLS v1.0.

There are multiple free and low-cost vendors that offer TLS certificates. rather Make sure potential API users can establish trust in your certificates. Make sure you have a robust process for timely certificate renewal and revocation.

Use JSON

Your first choice for all web APIs should be JSON where possible.

Only use another representation to build something in exceptional cases, like when you:

- need to connect to a legacy system, for example, one that only uses XML

12:52 17 Nov ✓

Provide an example of our Government documenting an API well

From imported document

12:06 17 Nov ✓

Here you have explained why you should be using HTTPS, whereas below with the TLS there is not explanation

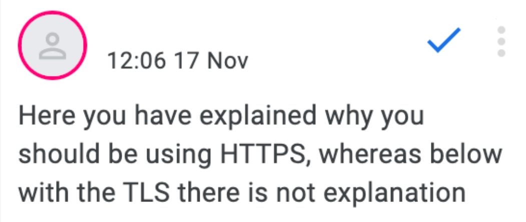
From imported document

12:08 17 Nov ✓

Maybe provide an example of the most used vendors? Would save time searching for an appropriate one

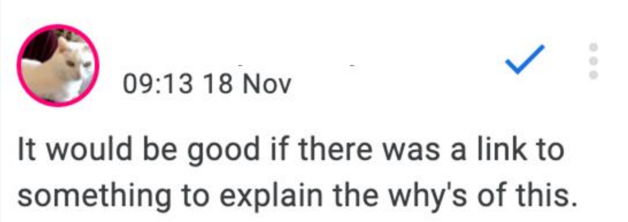
What is good …

Links to documentation (!!!)
Explanations for ‘why’ or ‘why not’
Showing examples



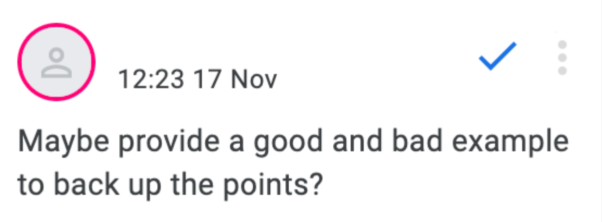
12:06 17 Nov

Here you have explained why you should be using HTTPS, whereas below with the TLS there is not explanation



09:13 18 Nov

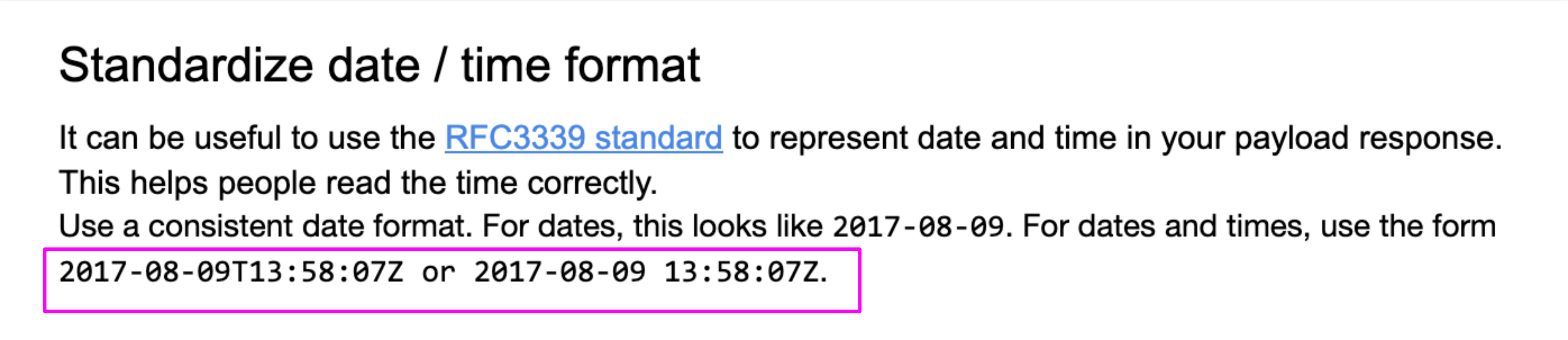
It would be good if there was a link to something to explain the why's of this.



12:23 17 Nov

Maybe provide a good and bad example to back up the points?

What could be iterated for more clarity?

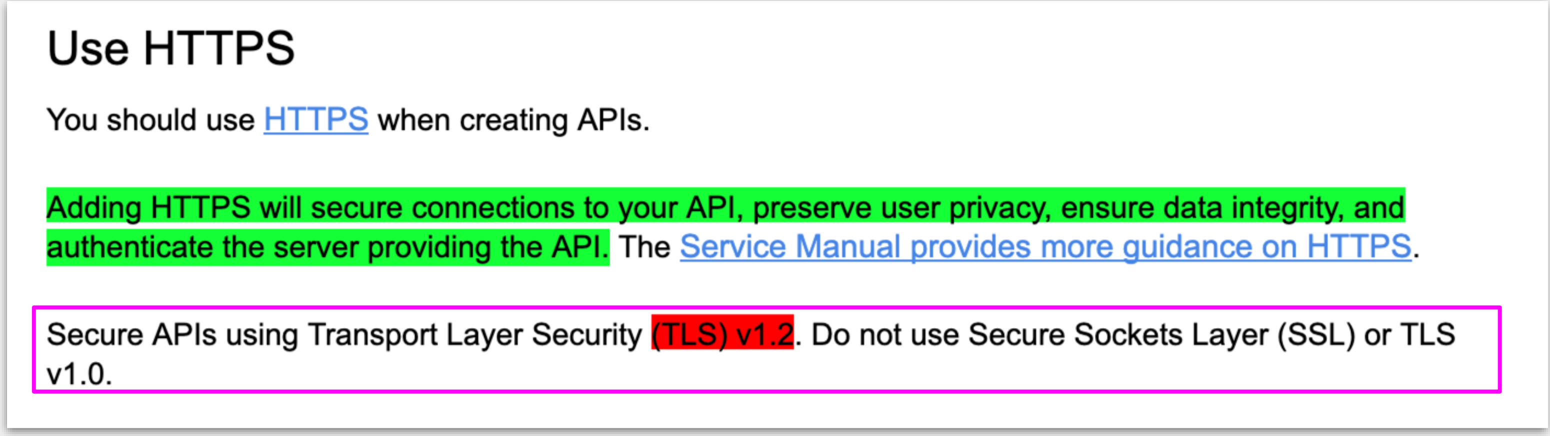


Standardize date / time format

It can be useful to use the RFC3339 standard to represent date and time in your payload response. This helps people read the time correctly.

Use a consistent date format. For dates, this looks like `2017-08-09`. For dates and times, use the form `2017-08-09T13:58:07Z or 2017-08-09 13:58:07Z.`

Re-format the code so it’s clearer to read



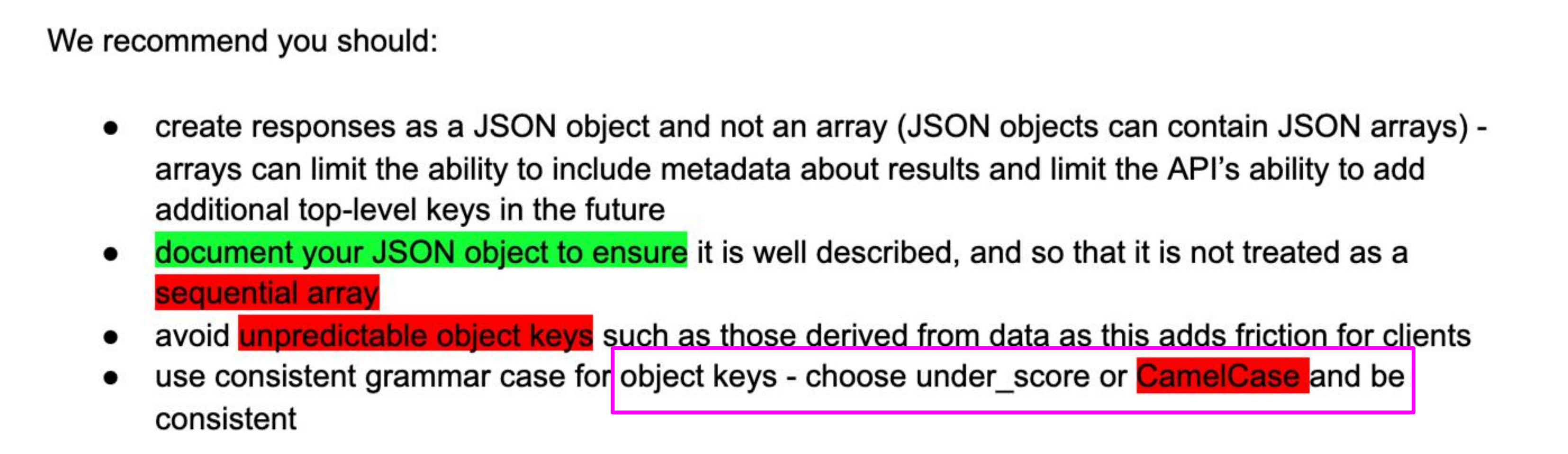
Use HTTPS

You should use HTTPS when creating APIs.

Adding HTTPS will secure connections to your API, preserve user privacy, ensure data integrity, and authenticate the server providing the API. The Service Manual provides more guidance on HTTPS.

Secure APIs using Transport Layer Security (TLS) v1.2. Do not use Secure Sockets Layer (SSL) or TLS v1.0.

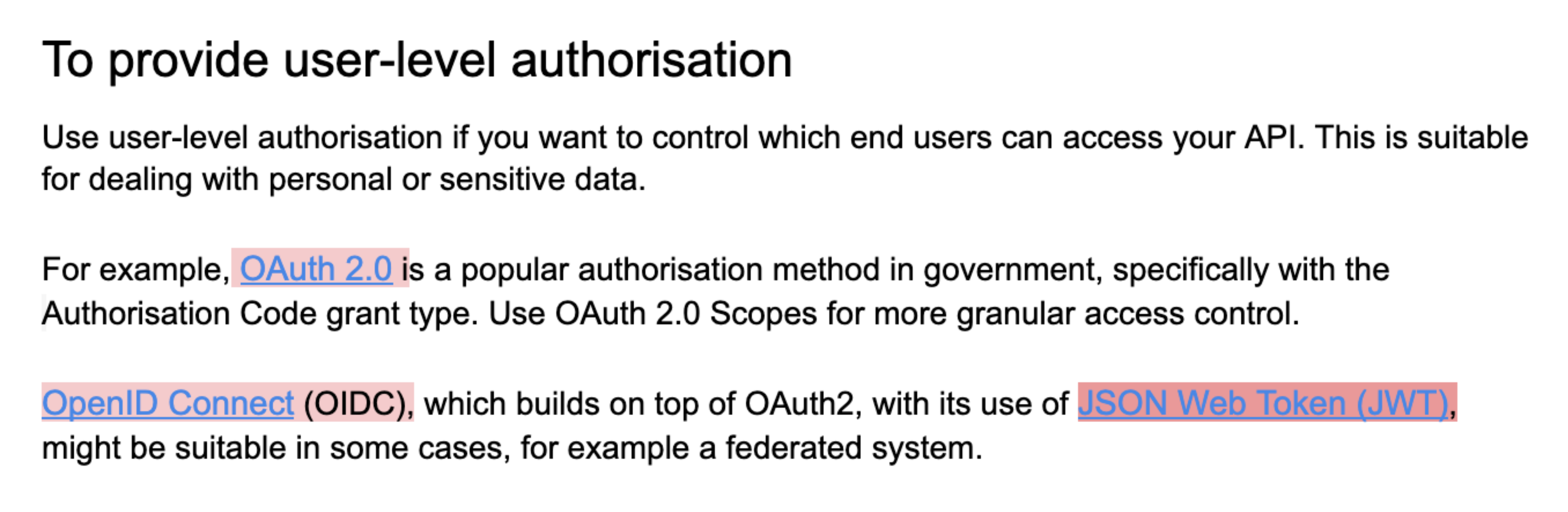
Update the out of date versions mentioned



We recommend you should:

- create responses as a JSON object and not an array (JSON objects can contain JSON arrays) - arrays can limit the ability to include metadata about results and limit the API's ability to add additional top-level keys in the future
- document your JSON object to ensure it is well described, and so that it is not treated as a sequential array
- avoid unpredictable object keys such as those derived from data as this adds friction for clients
- use consistent grammar case for object keys - choose under_score or CamelCase and be consistent

Use consistent object keys to avoid confusion



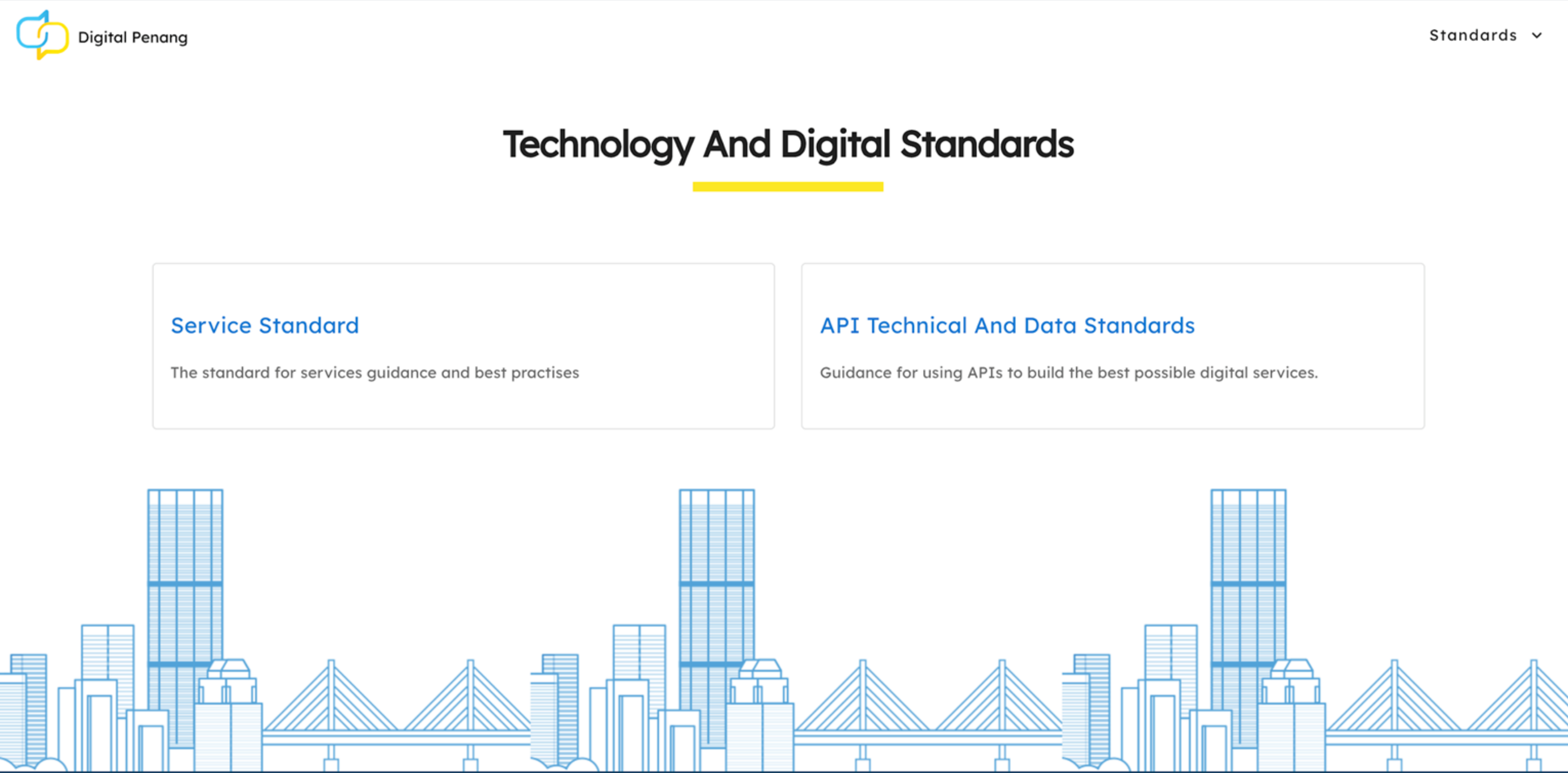
To provide user-level authorisation

Use user-level authorisation if you want to control which end users can access your API. This is suitable for dealing with personal or sensitive data.

For example, OAuth 2.0 is a popular authorisation method in government, specifically with the Authorisation Code grant type. Use OAuth 2.0 Scopes for more granular access control.

OpenID Connect (OIDC), which builds on top of OAuth2, with its use of JSON Web Token (JWT), might be suitable in some cases, for example a federated system.

Give more definition & explanation for words



Technology And Digital Standards

Service Standard

The standard for services guidance and best practises

API Technical And Data Standards

Guidance for using APIs to build the best possible digital services.

Global Digital Marketplace

User Research standards

We spoke to 10 participants in 10 sessions

- All from Malaysia
- 1M, 9F
- 2x from Gov digital service teams
- 8x UX professionals (UR, UI/UX)

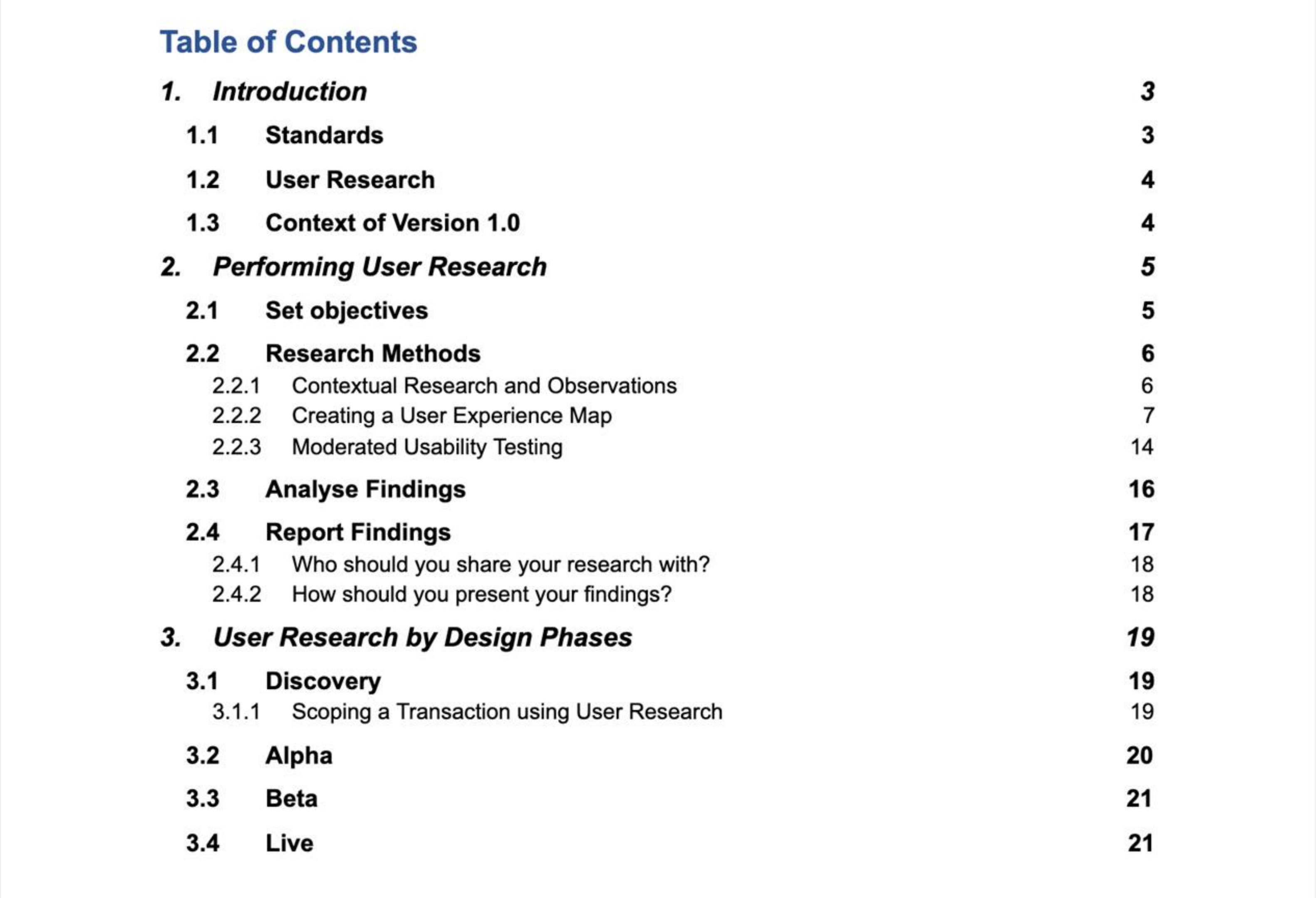
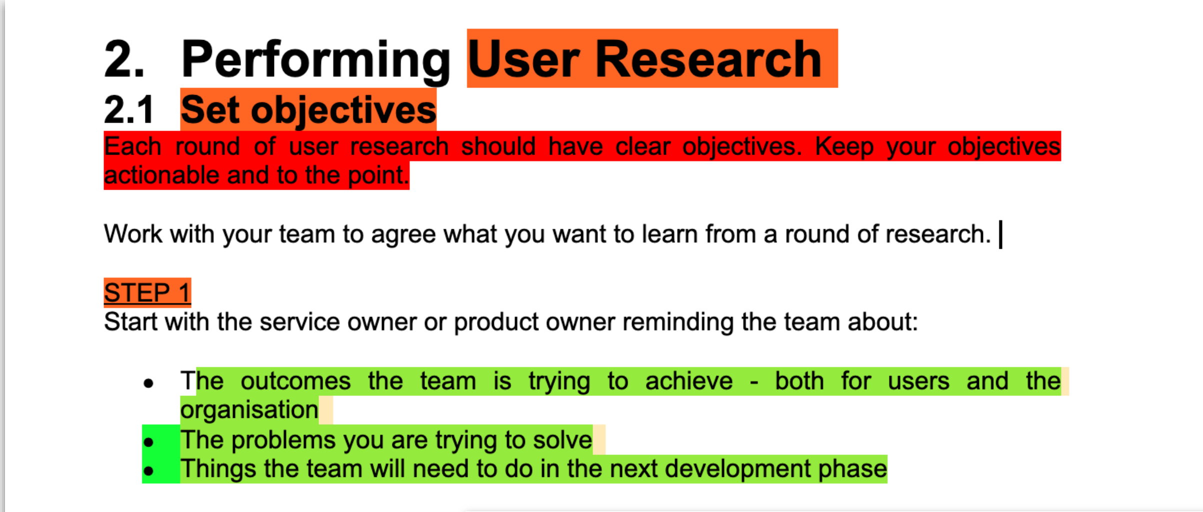


Table of Contents



2. Performing User Research

2.1 Set objectives

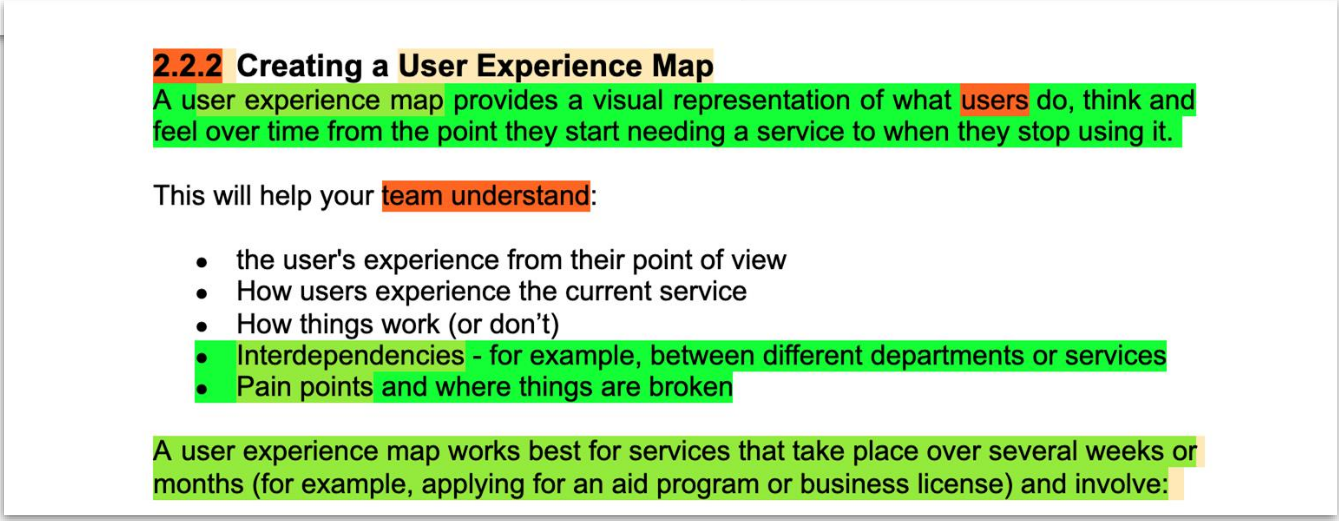
Each round of user research should have clear objectives. Keep your objectives actionable and to the point.

Work with your team to agree what you want to learn from a round of research.

STEP 1

Start with the service owner or product owner reminding the team about:

- The outcomes the team is trying to achieve - both for users and the organisation
- The problems you are trying to solve
- Things the team will need to do in the next development phase



2.2.2 Creating a User Experience Map

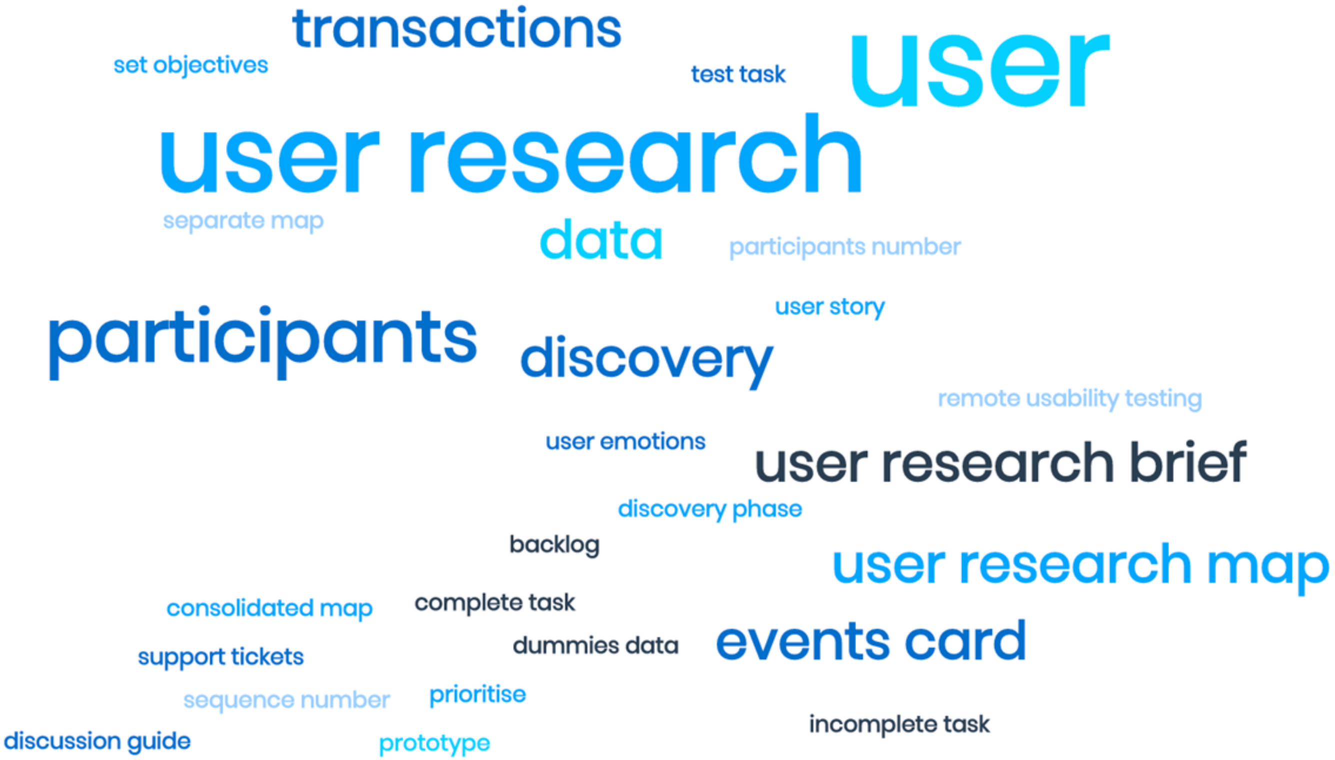
A user experience map provides a visual representation of what users do, think and feel over time from the point they start needing a service to when they stop using it.

This will help your team understand:

- the user's experience from their point of view
- How users experience the current service
- How things work (or don't)
- Interdependencies - for example, between different departments or services
- Pain points and where things are broken

A user experience map works best for services that take place over several weeks or months (for example, applying for an aid program or business license) and involve:

Definitions not understood:



Tone & content

Be more prescriptive.
Clear step by step so ppl can follow.
Thinking critically can be difficult.
Cross reference sections more.

Should anything be removed?

“The document is very comprehensive, I don’t think you need to remove anything but you might want to consider a simplified version for some audiences”

Participant 2

What’s content is missing?

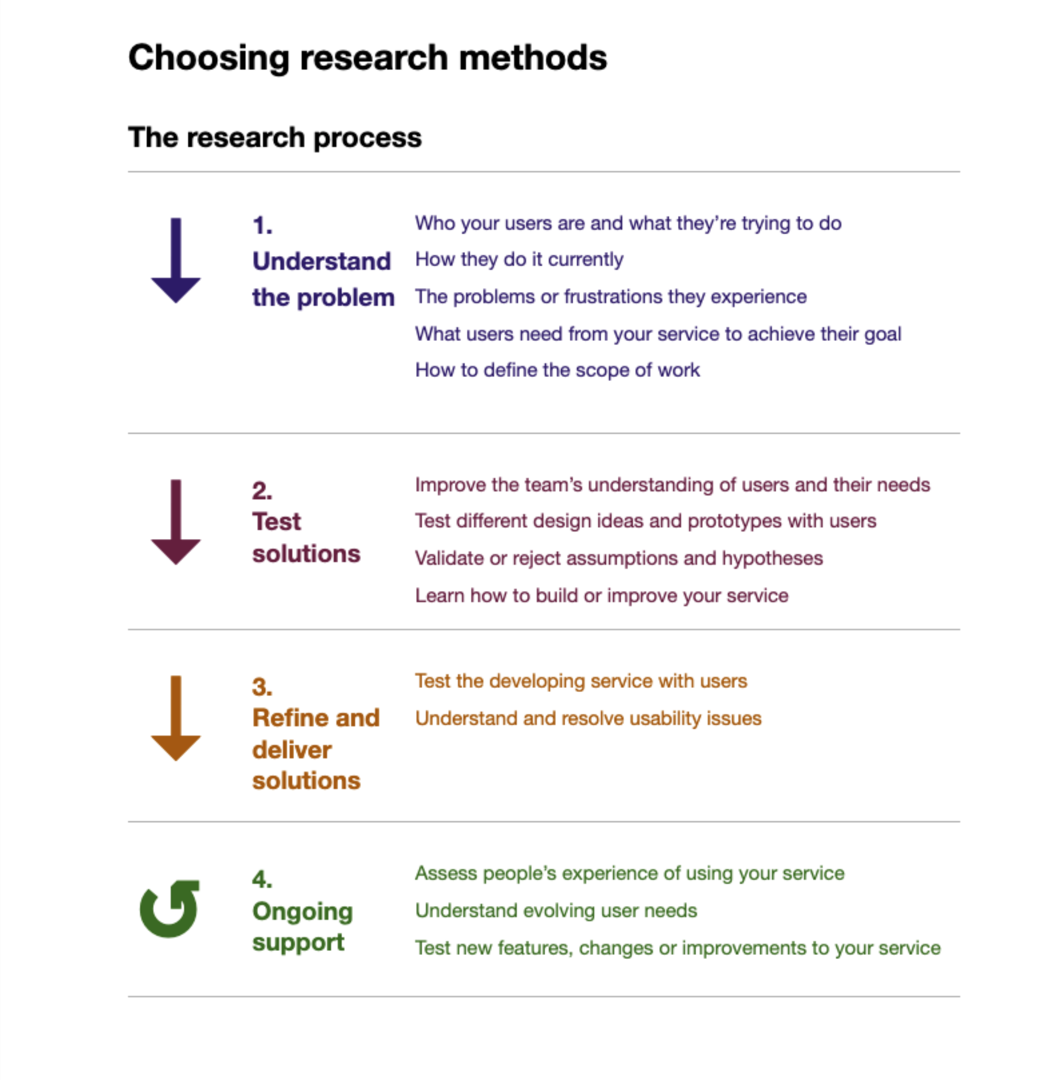
Differentiating qualitative vs quantitative.

Methods for participant recruitment.

Number of participants to test with.

Emphasising empathy.

Personas.



Choosing research methods

The research process

↓	**1. Understand the problem**	Who your users are and what they're trying to do How they do it currently The problems or frustrations they experience What users need from your service to achieve their goal How to define the scope of work
↓	**2. Test solutions**	Improve the team's understanding of users and their needs Test different design ideas and prototypes with users Validate or reject assumptions and hypotheses Learn how to build or improve your service
↓	**3. Refine and deliver solutions**	Test the developing service with users Understand and resolve usability issues
↺	**4. Ongoing support**	Assess people's experience of using your service Understand evolving user needs Test new features, changes or improvements to your service

Lessons

Linking to

67

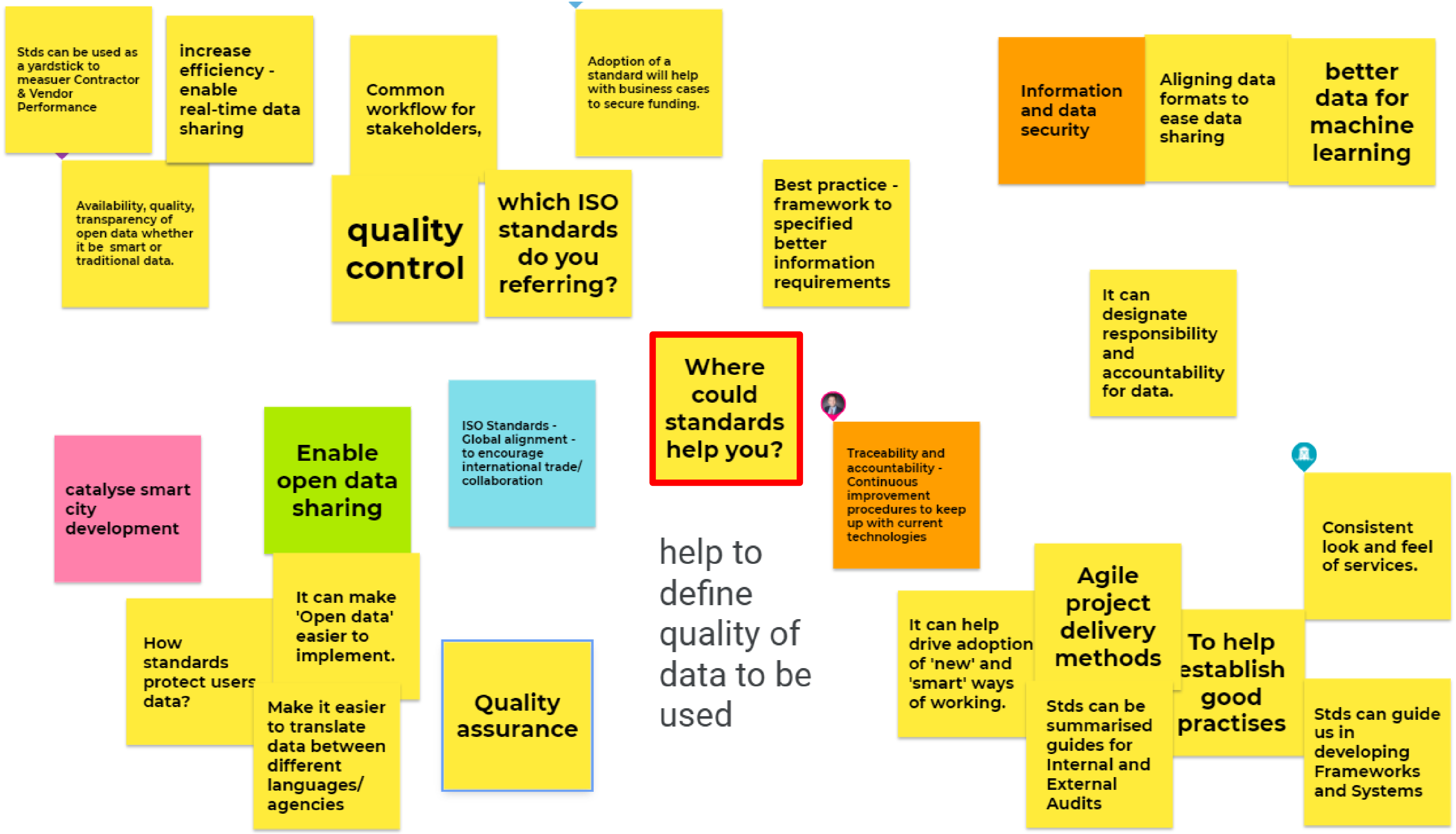
Pieces of MAMPU (National Procurement Agency in Malaysia) standards

Tim Choy from Digital Penang said:

1. Openness to adapt to localised context
2. Access to GDS expertise
3. Consistency in delivery using agile methods

What areas of digital and technology standards do you want to focus on?

Jam Board Activity



Where could standards help you?

Stds can be used as a yardstick to measuer Contractor & Vendor Performance

increase efficiency - enable real-time data sharing

Common workflow for stakeholders,

Adoption of a standard will help with business cases to secure funding.

Availability, quality, transparency of open data whether it be smart or traditional data.

quality control

which ISO standards do you referring?

Best practice - framework to specified better information requirements

Information and data security

Aligning data formats to ease data sharing

better data for machine learning

It can designate responsibility and accountability for data.

catalyse smart city development

Enable open data sharing

ISO Standards - Global alignment - to encourage international trade/ collaboration

Traceability and accountability - Continuous improvement procedures to keep up with current technologies

Consistent look and feel of services.

help to define quality of data to be used

How standards protect users data?

It can make 'Open data' easier to implement.

Make it easier to translate data between different languages/ agencies

Quality assurance

It can help drive adoption of 'new' and 'smart' ways of working.

Agile project delivery methods

Stds can be summarised guides for Internal and External Audits

To help establish good practises

Stds can guide us in developing Frameworks and Systems

Learning Outcomes

- ✓ Understand the importance of standards setting in digital transformation
- ✓ Consider some areas where standards setting could be of use to your organisation.

FUTURE CITIES

UK Prosperity Fund Global Future Cities Programme

Thank you