Public Health Agency of Canada | Agence de la santé publique du Canada

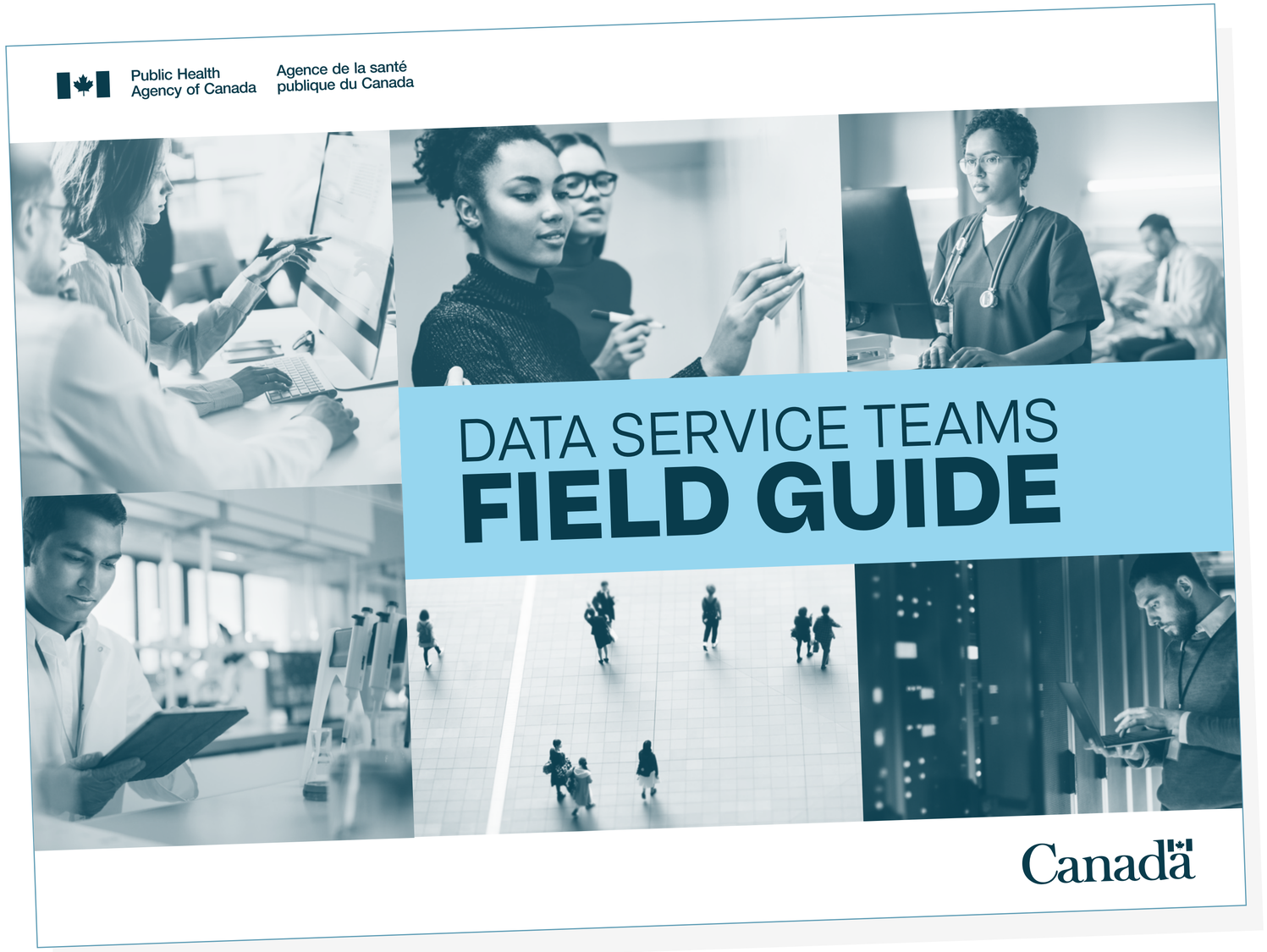



DATA SERVICE TEAMS FIELD GUIDE *in brief*

1.5.0 (E)




Canada



WHAT

A **project-management framework and compendium** of practical advice for the design and development of data services for public health

WHO

Those working in **collaborative multi-function teams** that involve clients and stakeholders as genuine co-creators

HOW

Applying lessons from **client-centred service** designers, “big data” infrastructure, and **rapid-development** operations with a proven track-record of lasting success

WHY

Building capacity within the Agency to get new data services up and running **fast enough** to effectively manage emerging public health threats

1 WHAT ARE **DATA SERVICE TEAMS ?**

DATA SERVICES

Data services are arrangements that ensure the right data is flowing to the right people at the right times in a form that best serves peoples' practical needs. Advanced technologies are crucial enablers but are not the whole story. The substantive subject-matter, social context of use, and ultimate risks and benefits have to be taken into account too.

DATA SERVICE TEAMS

A **data service team** is a diverse group of co-creators who come together to design and develop ways of improving public health through the use of evidence. It is a "platoon on a mission," brought together quickly for a short term project, while leaving behind lasting capabilities amongst those maintaining the services.

DATA SERVICE TYPES

Data can be applied in many ways, based on function and medium, to serve client needs and policy goals

FUNCTION

ANALYSIS

Visual displays of data that help make sense of complex patterns of data, such as data dashboards, interactive infographics, geo-spatial maps, and automated reports

REFERENTIAL

The storage and retrieval of data to complete a task, such as a search or directory services used to look up facts or point to better sources of information

MONITORING

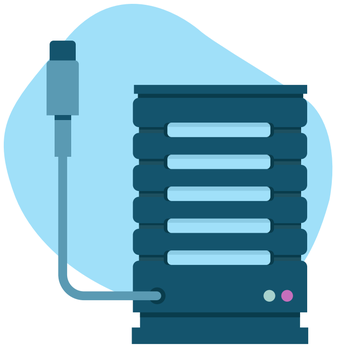
Tracking activities within a domain and drawing attention to points (or patterns) of interest, such as services that alert people to note-worthy events and conditions, or emergent threats

CONTEXTUAL

Services that sense (or are otherwise aware) of the context and are able to provide situation-relevant data, or even execute automated tasks depending on circumstances

VERIFICATORY

Confirming that state or provenance of an object of interest, such as a health-pass service that confirms a person's vaccination status



DATA ACCESS

Allowing people to tap into data for their own purposes, including experts gaining access or third parties building their own services, all while maintaining privacy provenance, and security

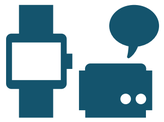
MEDIUM

Open Web

Mobile Apps

Kiosk or On-site Display

Messaging Service



New Consumer Gadgets

Physical Media

Secure Enclave

API (Self-serve Data Stream)

STRATEGIC RISKS

Data usage falls short of expectations because most data pipelines have avoidable shortcomings

✗	✓
Data arrives **too late** to act on or is otherwise stale	**Real-time** data offers insights on pressing topics
Data **underused** because it is not relevant to users	**Policy- and task-relevant** data with clear applications
Data **not in form** easily transmitted or worked with	Self-describing, interoperable, **“productized”** data
Data flows **can-not keep up** with the times	Data flows **evolve** as the science does and new needs emerge
Data **not trusted** due to quality or bias concerns	Secure, verifiable, methodologically **rigorous** data
Inappropriate uses of data an ongoing risk	Data flows **respect** privacy, consent, and fairness

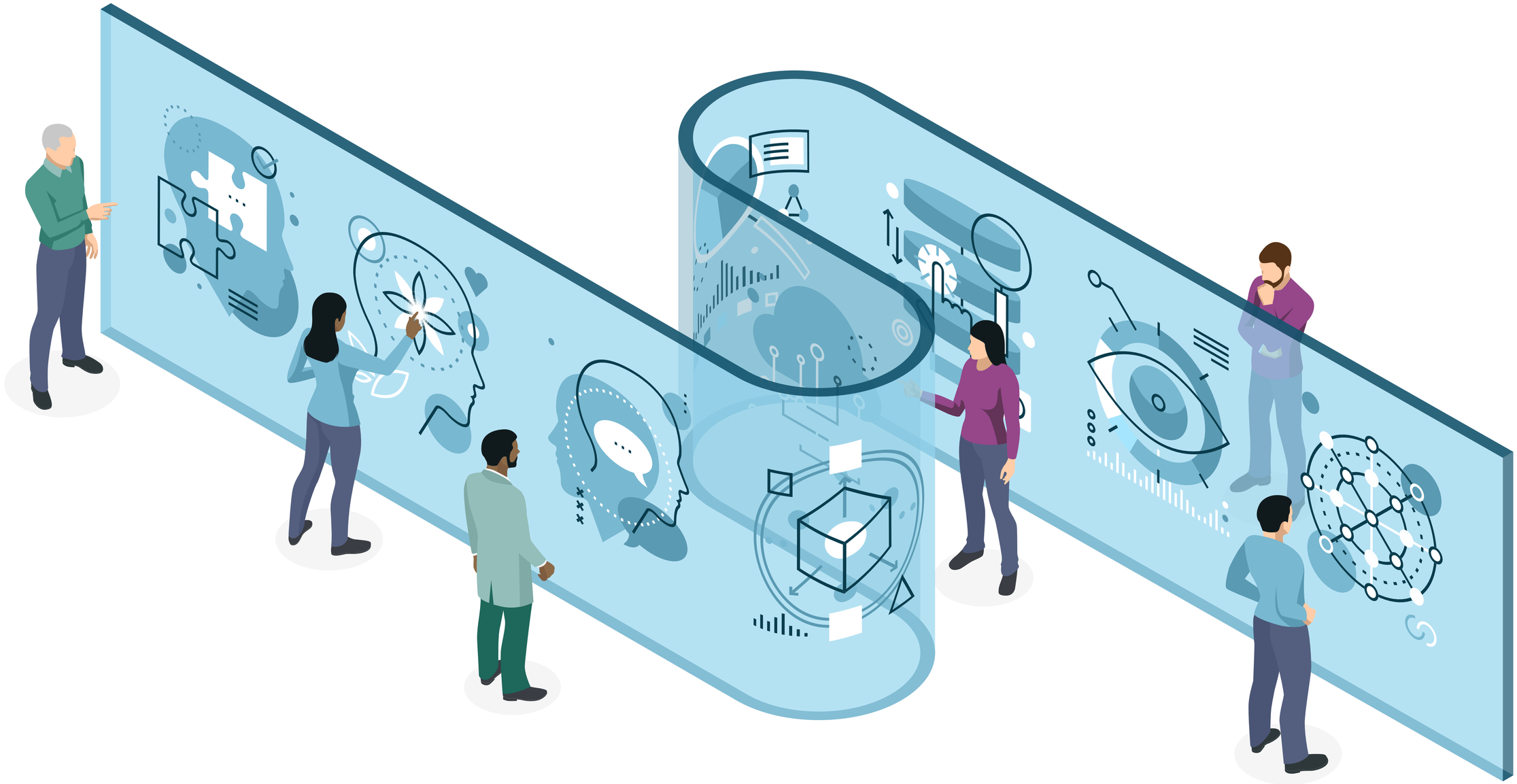
DATA COLLECTION

DATA FLOWS

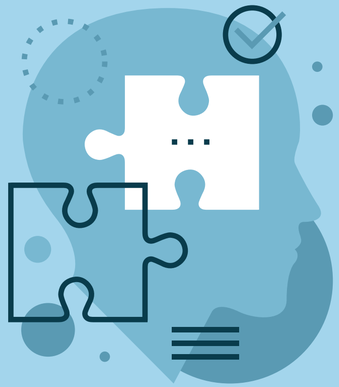
DATA USAGE

ESSENTIAL TOUCHSTONES

Data service teams “design the design process” according to circumstances but are oriented by nine touchstones



A



DESIGN THINKING

Design involves prefiguring how something is made and could be made better. It is:

- a **sensibility**, or a discontent with the merely adequate and enthusiasm for finding elegant alternatives;
- a **way of seeing**, or reflecting on the implications of small details while also being mindful of how the whole works;
- a **body of practical knowledge**, or a collection of methods and insights for discovering what works under what circumstances.

B

HUMAN-CENTRED SERVICE DESIGN

Services are best designed in a way that empathizes with clients. **Human-centred** is:

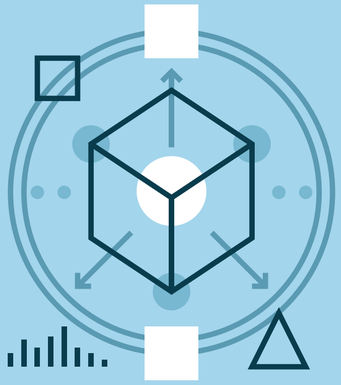
- an **orienting priority**, or designing from the vantage point of the client rather than just doing what is most convenient for administrators of the service;
- a **humility**, not presuming an ideal client but looking at real clients as they are;
- a **sensitivity**, or an awareness of human diversity and varying client needs, wants, situations, interests, aptitudes, social supports and personal quirks.

C

PARTICIPATORY DESIGN

Participatory design involves clients and stakeholders into the design circle as genuine co-creators. They are not merely consulted, but take part in deliberations about the service that matter to them the most. That is done through **dialogue** sessions, or open and earnest conversations in which everyone participates on an equal footing.

D



RAPID DEVELOPMENT

Intricately planned “mega-projects” with exhaustive specifications and elaborate timelines have an abysmal track-record. Instead, small teams break-down services into manageable parts, which are developed iteratively with extensive client involvement. Mock-ups, prototypes, and early versions are generated quickly so that it can be scrutinized and tested in a hands-on way with quick feedback. The process is adaptive, continually adjusting as under-standings of the service evolve. That model of **rapid development** is called *Agile*.

E

OPENNESS

Services are organized with an open spirit:

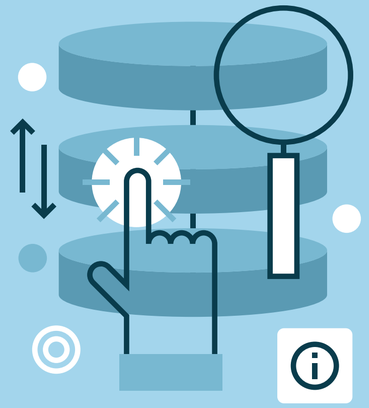
- **open-source software**, or modules of code made widely available for others to reuse instead of having to reinvent;
- **open data**, or since no one controls all the data necessary to promote public health, making it available for all to use;
- **open science**, or making scientific findings available for all to learn from and scrutinize;
- **working in the open** so that progress is clear and lessons are shared.

F

EVIDENCE-BASED POLICY-MAKING

Data cannot be fully understood without insights from the scientific literature and specialist expertise. Many data-mining techniques can lead to false inferences that mistake “noise” for “signal.” Service designers take a substantive interest in the subject-matter and are aware of method-ological limitations. Scientific findings may have to be channeled to users (*knowledge mobilization*) as part of the data service. Scientific literacy is necessary to turn raw findings into evidence that policy-makers can use to craft effective interventions.

G



DATA MANAGEMENT

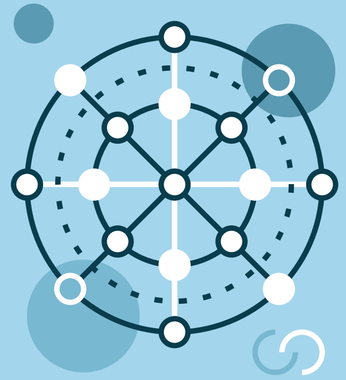
Services of all kinds are becoming more and more dependent on data. Data is treated as a product that is made useful and trust-worthy. Data service teams help put modern infrastructure to make that happen. These infrastructures are more modular, trans-parent, and flexible, as well less brittle and error-prone. Data usage can be inherently secure. Manual processes are replaced by automation. Data is managed according to FAIR principles (Findable, Accessible, Interoperable, and Reusable).

H

DATA STORYTELLING

Data storytelling involves adding context or interpretive aids to the presentation of data in order to highlight findings and make them more meaningful. Otherwise, data becomes a miscellany of “factoids,” open to all sorts of misinterpretations and erroneous conclusions. Visual tools offer suggestive cues and annotations (that is, *affordances*) that guides interpretation, ideally making findings “glanceable.” Not everyone can take advantage of visuals. A big toolkit of multimedia options can offer similar support for those with visual impairment.

I



SYSTEMS THINKING

Policy-making operates in a complex world. A problem cannot simply be fixed in a linear way, with clear-cut results expected from straight-forward interventions. Dynamic reactions create all sorts of unanticipated consequences, including harmful side-effects. Data that captures only a thin slice of reality can cause myopic thinking. Teams compensate by thinking holistically about how data is used. Moreover, teams also build systems that evolve elegantly thanks to corrective feedback and safeguards against systemic bias and vulnerability.

2 WHO IS **INVOLVED**?

STAKEHOLDERS

The core members of a data service rarely have the collective experience to understand the systems being built from all angles. To compensate, others are brought into the design circle. That starts with clients. However, there are a variety of other stakeholders with insights that can make the service better and ease the development process.

INVOLVEMENT

Some stakeholders are involved as genuine co-creators. Others are brought into the process on an as-needed basis. Who should be brought into the team? Who are the clients? What other stakeholders should be invited in? What is the scope of involvement for each? Frameworks are offered to answer these questions.

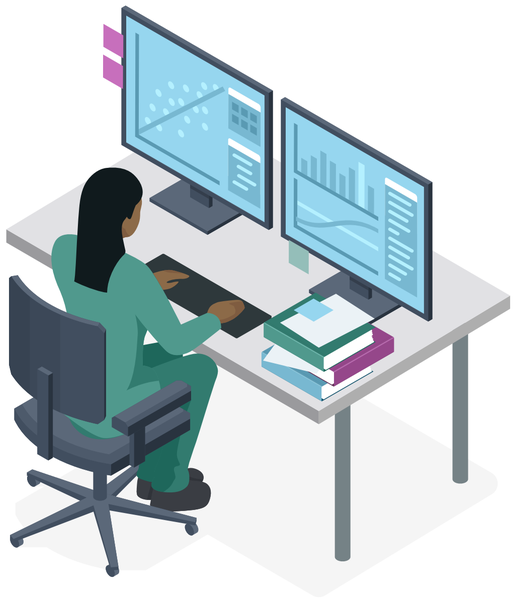
CLIENT-CENTRED FOUR-STEP

Human-centred design for internal data services entails an appreciation for all the ways data is used downstream

1. WHO ARE THE DIRECT CLIENTS?

Who will use the data directly? How many different groups (or *segments*) of client are there? For example, what roles require data inside and outside the Agency? What are their needs, wants, situations, interests, aptitudes, and social supports?

2. WHAT IS THE USAGE SCENARIO?



What tasks will the direct clients perform with the data? What contextual factors complicate those tasks (such as time constraints)? In what forms does the data need to arrive to be most useful within the flow of tasks?

3. WHO ARE THE CLIENT'S CLIENTS?

Internal-service providers should ask: who are the direct clients trying to serve or influence? What are the data needs and goals of those ***down-stream clients***? How can the data service be changed to cater to downstream clients directly?

4. WHAT IS THE END GAME?

What does success look like? What ***public policy goals*** are being served? What do you want people to do? What patterns of thinking and behaviour are you trying to change for the sake of the public good? How do you know the data is having the right influence?

DOWNSTREAM REACH

Designing services for only the nearest client is shortsighted. Seek out downstream users to maximize data use.

ANALYSTS

In-house experts who can make sense of findings

POLICY-MAKERS

Evidence-based decision-making by senior leaders

FRONT-LINE SERVERS

Data at the finger tips of those serving the public

THE PUBLIC

Direct access to data as an information service

OPERATIONS

Emergency personnel who need the latest data to focus efforts

OUTSIDE RESEARCHERS

Experts who can use data to make the next breakthrough

ENTREPRENEURS

Data flows to independent makers of digital services

INVOLVEMENT MODEL

Data services are more likely to fulfill their potential when all relevant stakeholders have input in the design.

METHODS EXPERTS

Data scientists, measurement experts, lab testers, and statistical tool builders

SUBJECT EXPERTS

Experts in the substantive subject matter including emerging trends, practical know-how and scientific knowledge

UPSTREAM PARTNERS

Others involved in data collection, such as NGOs, standards bodies, and other jurisdictions

DATA OWNERS

Collectors, aggregators, and managers of data

DESIGN CIRCLE

Core team plus co-creators

DIRECT CLIENTS

Nearest users and potential users of data

DOWNSTREAM CLIENTS

Potential users of data far from the data source

PLATFORM BUILDERS

Infrastructure developers, security specialists, data architects, and other system technology experts

DEV SPECIALISTS

Those with special research, design, and development skills to help with emerging difficulties

TEAM ROLES

A core team works on service development full time while involving other co-creators in ways that respect their time

INTAKE

Helps initiate new projects

Intake Coordinator

Responsible for managing proposals for new projects

CORE TEAM

Full-time project team (3-7 members) that undertakes main design and development responsibilities

Design Lead

Maintains overall project view, facilitates dialogue, and promotes holistic approach

Design Specialist

User-experience, service, data visualization or related field

Lead Analyst

Public health researcher, design researcher, social scientist, or behavioral scientist

Admin. Coordinator

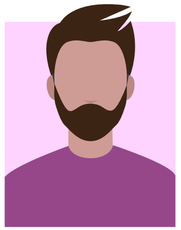
Tasks related to the management of the project

Tech Lead

Software or web developer, or relevant technical expertise

EXTENDED TEAM

Stakeholders and clients who are involved in most major project deliberations and help as they can

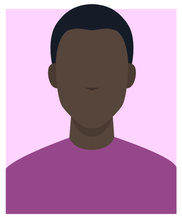


Data Manager

Data collection and database management duties

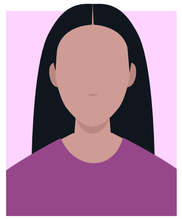
Direct Client

Internal user of data, such as policy analyst, emergency manager, or epidemiologist



Upstream Partner

External partner involved in data collection



Surveillance Practitioner

Subject-specific knowledge and partnership relations

Downstream Client

User of materials from direct clients or ultimate beneficiary of data

Platform Developers

Managing software and tools that all projects can benefit from

EXTENDED NETWORK

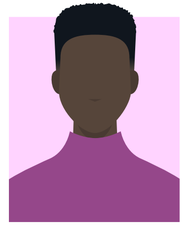
Stakeholders and clients with occasional involvement

Subject Expert

Knowledge about subject-matter, policy, or data analysis within the field

Government Stakeholders

Those with interest in the data, either within the Ministry or the Public Service more generally.



Methods Experts

Developer of methods, models, and technologies of interest to project

Public Health Stakeholders

Members of the broader community with a professional interest in health outcomes

3 HOW DOES **THE PROCESS** WORK?

RESEARCH & TESTING

Instead of rushing to solutions, the team explores the problem space in an open-minded, evidence-based way. Clients, use-cases, constraints, and technicalities are understood through **research and testing**, not left as convenient assumptions nor after-thoughts.

DESIGN

Once a problem is better under-stood, design is the methodical prefiguring of what the service can be. With **human-centred design**, how the service is experienced by actual clients in real-world contexts is the starting point for thinking about how the service should work.

DEVELOPMENT

Development is the actual making of the service using advanced technologies. In real world projects, many design decisions are made while struggling with development challenges, rather than set ahead of time.

DIALOGUE-BASED PARTICIPATORY DESIGN

Design deliberations take place through open exploration of the issues with stakeholders and clients

DIALOGUE

An open and earnest conversation in which everyone participates on equal footing; rank is set aside with quality of ideas judged on the merits

DESIGN CHARRETTES

Extended workshops in which co-creators explore ideas jointly by getting ideas out in the open with sticky notes, sketches, and mocked-up models

CLIENT INVOLVEMENT

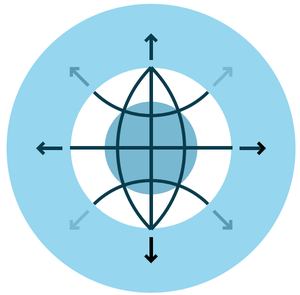
Clients are invited into the design circle as genuine co-creators; not involved in every task and decision, just the ones that matter to them the most

REQUISITE VARIETY

Enough diversity is brought into the circle to think holistically about how the service should work and handle the technical aspects of the system

DESIGNING THE DESIGN PROCESS

Instead of resorting to formulaic approaches, the team designs the process according the specific challenges at hand



END-TO-END DESIGN

Everyone considers how the entire data pipeline works, from upstream measurement to down-stream client usage and all points in between

LEAN PRODUCTION

The process of making happens through swift experimentation and iteration

AGILE DEVELOPMENT

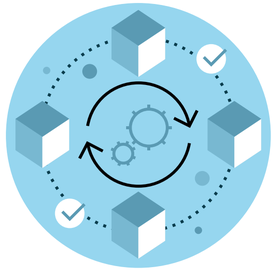
A basic version of the service is built as quickly as possible by working on manageable pieces in quick “sprints”

MODULAR DESIGN

Services are built with discrete building blocks of functionality that can be swapped out and do not have unnecessary dependencies

SOFTWARE REUSE

Open-source libraries are relied on over proprietary solutions, with teams contributing code for others to reuse in turn



TASK MANAGEMENT

Development activities are organized at the task level with scrum boards, kanban boards, or similar task management tool, either in physical or virtual form

AIR COVER

Supervisors free up team members’ time to participate fully, as well as shield the team from distractions that can undermine progress

WORKING IN THE OPEN

Progress is shown by letting others see work in progress, including design concepts, mock-ups, “clickable” proto-types, software modules, and partial builds

AVOIDING PITFALLS

The risk is falling back towards dysfunctional and out-dated ways of managing projects, such as ...

ANTI-PATTERNS

Some practices have a dismal track-record but become the default way of doing things because they somehow get viewed as “normal.”

DESIGN “THEATRE”

Teams may pay lip-service to design thinking and go through the motions, yet fail to engage with actual clients and act on worth-while ideas that are unfamiliar.

SILO-MENTALITY BACKSLIDING

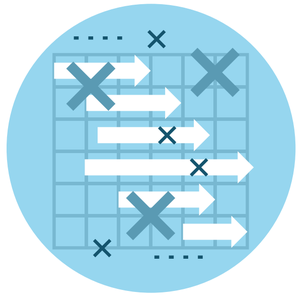
Instead of working collab-oratively as an integrated team, team-mates revert to a bureaucratic division of labour full of hand-offs and turf defences.

TECHNICAL DEBT

Hacking together systems out of short-term conven-ience leads to a hodgepodge of systems that cannot talk to each other, are difficult to replicate, and are a night-mare to maintain.

WATERFALLING

Detailed plans with tightly coupled dependencies inevitably fail to account for everything, lock-in requirements, and cause projects to get bogged down in low-value tasks.



SUPERFICIAL RISK MANAGEMENT

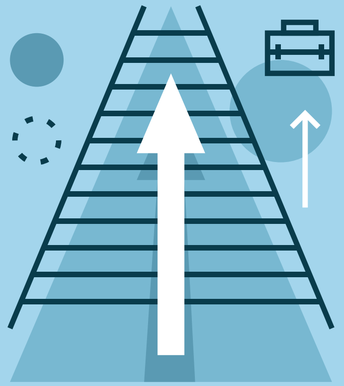
Projects need to learn from setbacks and pivot as needed, rather than pre-commit to a set path without any thought to how to manage risks as they arise.

INNOVATION OPPORTUNITIES

A quick "one-and-done" fix might be called for, but not before exploring the problems deeply and looking for ways to innovate

I'm here to get a new executive dashboard for my surveillance program. I was told that you can help.

I sure can. But let's not jump to solutions right away. This is an opportunity to explore the service options.



LADDER OF INNOVATION

To not pass up opportunities, team members ask themselves:

- How can we **do things better** to improve the way things work (*incremental improvement*)?
- How can we **do better things** to achieve our goals (*major leaps*)?
- How can we **choose better goals** to advance our mission (*comprehensive rethinking and redesign*)?

GENERATING POSSIBILITIES

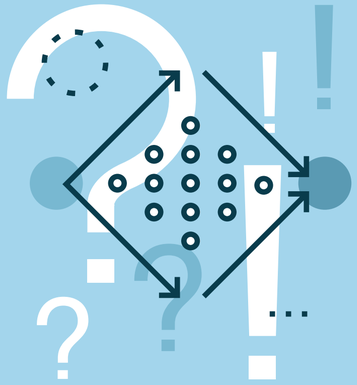
Creative thinking involves rounds of idea generation and refinement

Let's see how many ideas we can come up with

Which ideas are doable and worth trying?

Divergent Thinking

Convergent Thinking



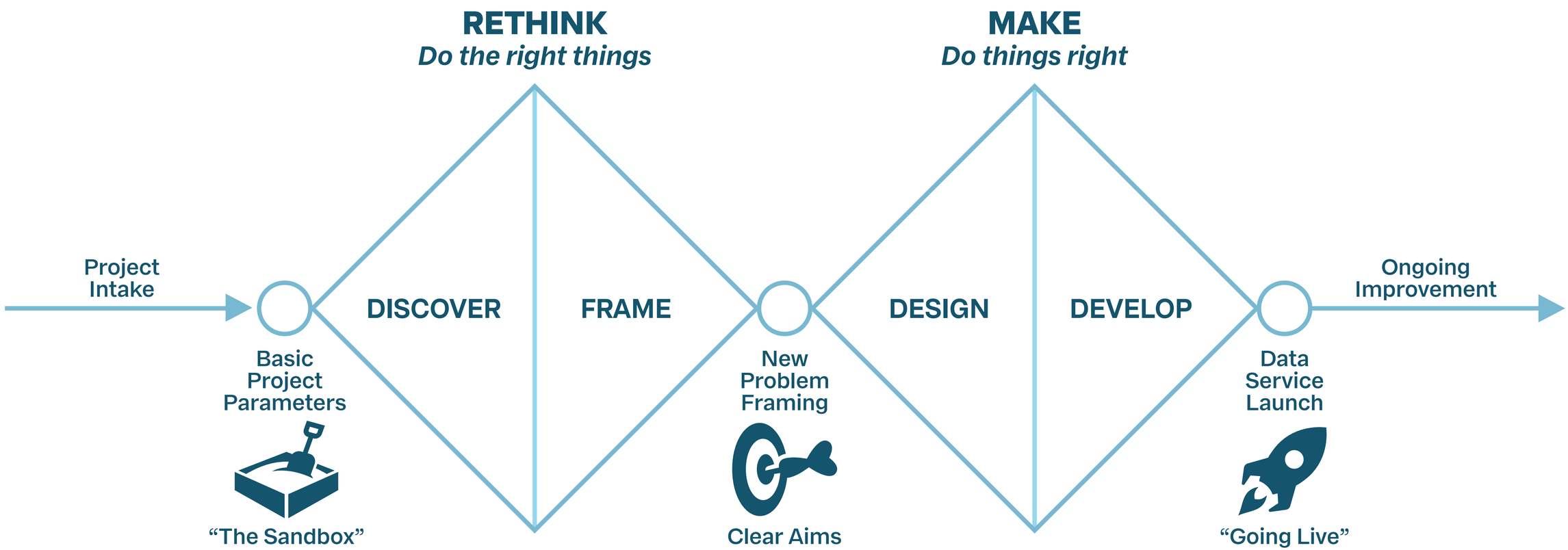
DIAMONDS OF DELIBERATION

Exploring a problem space involves different thinking styles.

- **Divergent thinking.** Possibilities are generated by suspending judgement and considering a wide variety of ideas with an open mind. New ideas are thought up. Hunches are sussed out. Interesting ideas are sought from unfamiliar sources.
- **Convergent thinking.** Good ideas are selected and synthesized by applying scrutiny and drawing connections. Promising ideas are refined and built upon. Less practical ones are reformulated or set aside.

DOUBLE-DIAMOND PROCESS

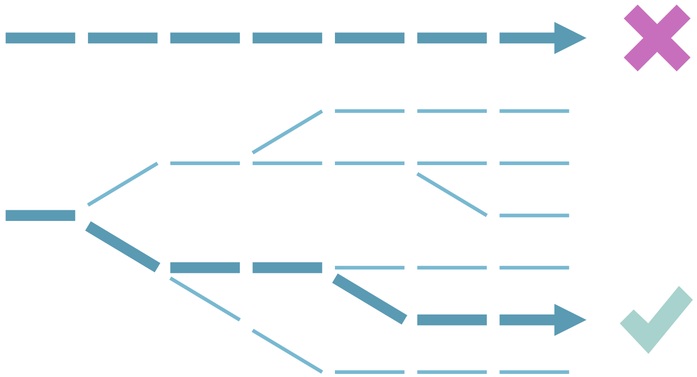
The project proceeds according to successive rounds of divergent and convergent thinking



FAST AND FLEXIBLE MAKING, NOT PAT RECIPES

Projects inevitably fail to meet expectations when they are over-planned, full of dependencies, and inflexible.

APPROPRIATE PLAYS NOT **SET FORMULAS**

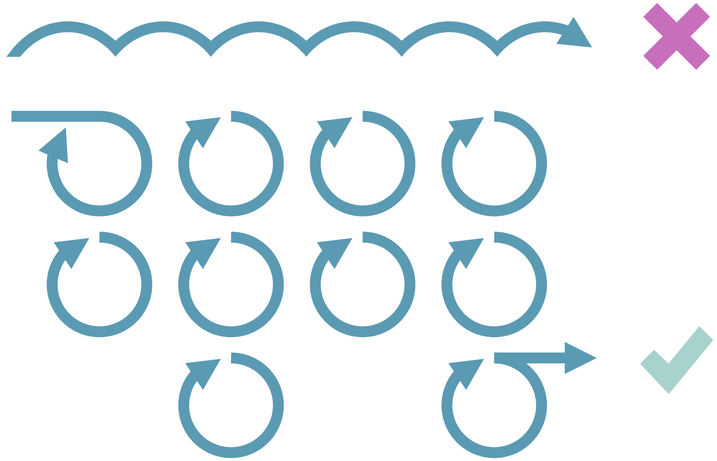


Design processes are not pat formulas with everything planned out in advance. Methods are applied flexibly as the team explores the problems with humility. Instead of imagining a linear sequence like a recipe, think of it more like a tree in which ***playbook plays*** are selected depending on their relevance. That is a large part of what "designing the design process" means.

MANAGEABLE "SPRINTS" NOT **TIGHTLY COUPLED ENTANGLEMENTS**

Major projects stages are not implemented as large entangled initiatives with tightly coupled dependencies. Instead, stages are broken down into groups of tasks small enough to be implemented quick ***sprints***. Likewise, discussions go around in circles discussing topics that are too big and amorphous to resolve. Big topics are broken down into manageable conversations.

ITERATIONS NOT **ONCE-AND-FOR-ALL PUSHES**



Risk is managed by getting workable versions of features and subsystems up and running quickly. Improvement then comes from ***iteration***. If a proposal is not viable, that is discovered early and abandoned, rather than leaving it until it is too late to change. Moreover, much of a service's innovation comes from iterative improvement as ideas are put to the test.

ON-BOARDING HAND-OFF

Everyone involved in project inception comes together in a “warm hand-off” to negotiate the project’s “sandbox” (initial thinking and basic parameters). Care is taken not to pre-empt exploration of the problem space by presuming too much and pre-maturely jumping to solutions.

CORE TEAM ASSEMBLY

Full-time members of the core team are recruited. A few highly specialized roles may likely require less of a time commitment. The intent is to involve everyone in early exploration of the problem, even those who will be mostly be working in late-stage development tasks.

ORIENTATION

The Data Service Team process will be unfamiliar to many within the design circle. Participatory design, service design, and rapid development methods are introduced. The risk is that would-be collaborators revert to unproductive work patterns that undermine the project’s chances of success.

STOCK TAKING

The project begins by delving into the subject area. Issues and controversies are identified. Relevant documents are compiled and reviewed. Important scientific details may need to be understood. Surveillance program history and legacy systems are mapped out. Everyone gains a common understanding.

CLIENT AWARENESS

The direct- and down-stream clients of the service are identified. That is partly to recruit clients as co-creators. It is also to get an initial sense of the user-base of the data, including the various types of clients. Some prospective clients will be under-served. Others will be important strategically.

CLIENT INVOLVEMENT

Designers are lead astray by preconceptions about clients. Real clients are engaged to understand their world. A few are invited into the design circle as genuine co-creators. Care is taken to respect their time by mostly involving them in the decisions that matter to them the most.

USAGE SCENARIOS

The clients' context of data use is explored. Data ultimately has to be integrated into the workflows and lifestyles of clients. Usage scenarios are mapped out in detail to better understand what the data is for. The risk is that data which does not cater to real-world tasks will be ignored.

THE END-GAME

Both direct- and downstream clients have their own goals and agendas. If the service is to be "client-centred," it has to help fulfil those goals. However, there are also large public-health policy goals to serve. All these competing goals are explored and brought into alignment.

FIELD WORK

Bringing clients into a small team will not going to reflect the full diversity of the client base. Thus, the team goes to the clients. Research is conducted. Ethnography picks up on the subtler features of the context of use, including implicit understandings and socio-technical dynamics.

CLIENT PROFILING

Client research is made practical for design and development work. For example, clients can be summarized into personas, or concise profiles that describe actual clients and their circumstances. Decisions are made with these profiles in mind rather than vague, self-serving stereotypes.

CLIENT JOURNEYS

Any service involves a sequence of steps, including discovering the service in the first place. It is crucial to get a sense of the mindsets and mental models of clients as they perform tasks using the data. “Pain points” (difficulties and frustrations) are identified.

REFRAMING

Issues are reinterpreted given the research and insights that have been gathered. The problem is reformulated to address underlying root causes (not just surface symptoms), plus clients’ needs and challenges. Often the initial service request will be revised significantly.

FORESIGHT

Whatever challenge the team is tackling is a moving target. Ongoing and emerging trends within the subject area are identified. Future scenarios are speculated about. Emerging technologies are discussed. The aim is to create solutions that do not go prematurely obsolete but adapt as the challenge evolves.

SYNTHESIS

The initial understanding of the problem is revisited and a full problem statement is developed. The statement brings together various threads of discussion to put forward a vivid picture of what is wrong with the current state of affairs so that the path forward will be clearer.

IDEATION

The team brainstorms ideas for data service that address the problem. The aim is to generate as many ideas as possible by temporarily setting aside critical judgement. Additional research may be conducted to follow-up on hunches and see what others are doing.

CONSTRAINTS

Ideas become viable by reworking them given various logistical requirements and constraints. Constraints are not an excuse not to do something, they are an opportunity to think creatively about the most elegant solution. A *design critique* is an evaluation based on reasoned criteria and trade-offs.

HUMAN FACTORS ANALYSIS

There is no such things as an “average” or “normal” user. Human diversity is factored into service proposals. Affordances and accommodations help make services better complement the way people actually think and move. The desired service experience is spelled out from the clients’ perspective.

SERVICE CHANNELS

A data service will stream data through one or more “channels.” The service will be developed using mediums that clients already use or can be encouraged to adopt. Ideally, services are *omni-channel*, meaning that they cater to all the media that clients prefer to use.

MOCK-UPS

The vagary of words can make proposals ambiguous. Quickly sketching out options and assembling models can add clarity. The resulting mock-ups can also be used to sell an idea to sceptical stakeholders and clients by showing the art-of-the-possible in a tangible way.

PROTOTYPES

Once the most promising proposal has been selected, the mock-up is turned into a prototype with some functionality. How functional and realistic depends on the tools available and the time constraints. New insights will emerge. Testing with clients can begin early with simulations of the service.

USER EXPERIENCE (UX)

As the prototypes and builds advance, the specifics of the user interface are refined. Visual storytelling techniques are added to the data displays to make the data more meaningful. Aspects of the service that lead to frustration and confusion are reworked. The service is made more intuitive.

ARCHITECTURE

Back-end systems will need to be in place for the data flows to work. Often, there will be a need to develop back-end subsystems to enable the specific data service under development. Platform developers are engaged early and often to make sure the service will function as intended.

OPENNESS & SECURITY

Not every potential use of the data can be anticipated. Open data streams allow others to experiment with the data and develop alternative services. The service will also have to be made *inherently secure*, which is why security experts should be consulted before development tools are selected.

SHARING & REUSE

In order to make the most efficient use of resources, developers will not reinvent the wheel with every module and sub-routine. Open-source modules will be used. Newly created modules will be uploaded to repositories for others to use. Project lessons will be shared.

ALPHA BUILD

The rapid-development process is highly iterative, with the project broken down into smaller chunks that can be worked on in short bursts of effort (“sprints”) according to Agile principles. The aim is to get a “click-able” version up and running as soon as possible.

FIELD TESTING

Some client testing happens in the develop-ment space. Yet there is often no substitute for testing on-site, within the actual setting of use under realistic circumstances. Some contextual obstacles may require major reworking of the service. Some stress testing of systems should also happen.

BETA BUILD

The beta build is a fully working version of the service. There may be non-vital features to add. There will also be plenty of bugs to squash. The service can be released to a larger group of testers and stakeholders. Major modification may have to wait to future releases.

OFF-BOARDING HAND-OFF

Once a beta build is nearing release, the Data Service Team will start to disband. Full control of the project is handed back to the original program. Throughout the process, the intent is to build capability. A developer may become part of the team for a longer spell to work on ongoing development.

LAUNCH

The product is up and running for clients to use. A larger cross-section of clients will reveal more bugs and edge cases. Updates follow quickly to keep the service in working order. Some outreach and client education are expected, ideally in advance of the launch to generate “buzz” for the service.

CONTINUAL IMPROVEMENT

Working with “live” code is a different development model that does not allow for major disruptions. A-B testing may allow developers to test different versions of the service on clients in real time. The service evolves as new features are added and improvements are made.

SUMMARY OF ACTIVITIES

ON-BOARDING HAND-OFF
Sandbox stipulated and project started

CORE TEAM ASSEMBLY
Recruitment of full-time team members

ORIENTATION
Briefings on design and technical issues

STOCK-TAKING
Sorting out current state-of-affairs

CLIENT AWARENESS
Mapping the client base and identifying clients

CLIENT INVOLVEMENT
Inviting clients into design circle as co-creators

USAGE SCENARIOS
Understanding the worlds of clients

THE END-GAME
Understanding public health policy priorities

FIELD WORK
Research to fill in the gaps in knowledge

CLIENT PROFILING
Sorting out clients and promoting inclusion

CLIENT JOURNEYS
Understanding clients' stories at the task level

REFRAMING
Rethinking how data can be used to further goals

FORESIGHT
Thinking about how the subject area is evolving

SYNTHESIS
Developing a concise problematique with clear aims

IDEATION
Brainstorming service ideas

CONSTRAINTS
Understanding limits and design trade-offs

HUMAN-FACTORS
How human diversity is accommodated

SERVICE CHANNELS
Selecting the media and technologies

MOCK-UPS
Making promising ideas more tangible

PROTOTYPES
Test ideas with functional models

USER EXPERIENCE (UX)
Evaluate prototypes from clients' perspective

ARCHITECTURE
Make needed changes to back-end platforms

OPENNESS & SECURITY
Set access policies and safeguards

SHARING & RE-USE
Use, extend, and share software libraries

ALPHA BUILD
Early working version of service

FIELD TESTING
Test with actual clients in realistic setting

BETA BUILD
Fully working version with work to be done

OFF-BOARDING HAND-OFF
Transfer service to program maintaining it

LAUNCH
Allow clients to use and make minor adjustments

CONTINUAL IMPROVEMENT
Iterate service based on usage and feedback

4 HOW ARE SERVICES BUILT **AS A SYSTEM?**

DATA SUPPLY CHAINS

Data flows between the measurement source and the service's end-user can be thought of as supply chains, much like those in industrial production between raw material extraction and the finished product arriving in the stores. A variety of processing steps happen en route.

DATA ARCHITECTURE

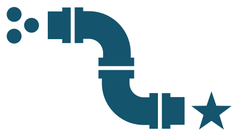
The networked computing and communication technologies made available to data services is the **infrastructure**. The overall way those technologies are configured and interact with services is the **data architecture**. Public sector service design is often constrained by legacy systems. Even so, each data service project is an opportunity to make improvements to existing infrastructure.

THREE AXES OF DATA INFRASTRUCTURE

The technological infrastructure of data services has three major dimensions worth considering

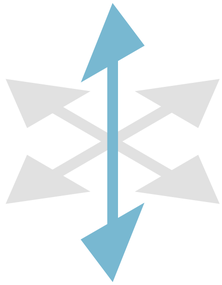
Every service inherits legacy systems. Even so, each project is a chance to improve infra-structure, adding capabilities and allowing others to reuse our hard work.

1. END TO END



DATA SUPPLY CHAINS

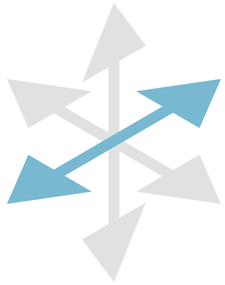
From end to end (data collection to data use), the raw data is processed to get the right data to the right people at the right times



2. TOP TO BOTTOM

TECHNOLOGY STACK

From server to user inter-face, modular and reusable software components are deployed in a way that is cloud-service agnostic, highly secure, and easily updatable



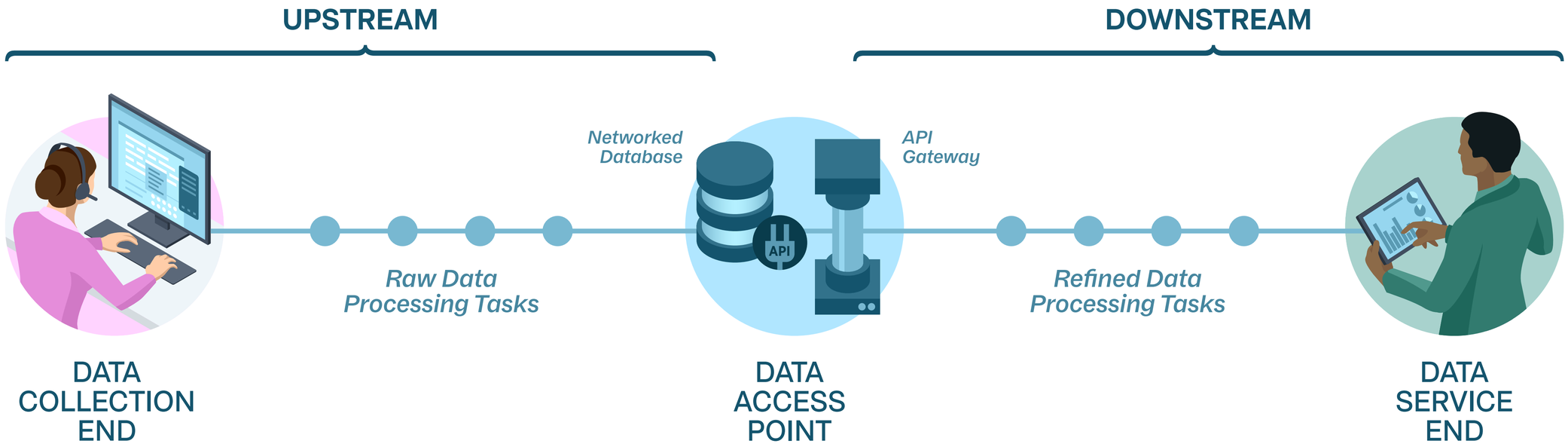
3. ACROSS OPERATIONS

SERVICE ECOSYSTEM

Reapplying service infra-structure to other policy areas and challenges, while making software available to the larger community so that every-one can advance

END-TO-END SUPPLY CHAIN

Look at the whole data supply chain to rework broken processes and explore opportunities



Data may be collected first-hand. Or it may be collected by a trusted third party, such as a specialized non-governmental organization. Or it may be aggregated from various sources, such as provincial governments. In any case, the raw data will be processed. For example, quality control tasks will be performed, such as code cleaning. The aim is to ***productize the data*** to make it easy for others to access and use.

The midpoint is where the data is stored for ready access. At minimum, a database is placed in the cloud. With a ***federated data architecture***, an API gateway provides a single point of access for several databases so that separate pipelines do not have to be built for each one. In a centralized legacy configuration, multiple data sources may be in a data warehouse, which adds more constraints to development.

When data is combined and displayed for the user, several processing tasks are involved. If the service is providing basic data access to analysts, many of the ***processing tasks*** will be manual. Even so, what the user can and cannot do with the data will be automatically controlled. If the service involves providing data to a dashboard or end-user application, much of the processing will be automated behind the scenes.

DATA COLLECTION

Even though public data takes many forms, the goal is to streamline intake


SAMPLES


SENSOR DATA


CREDENTIALS


FORMS


RECORDS


CASE REPORTS


FEEDS


APP DATA


SYNTHETIC DATA

TABLE FILES


Digitalization of physical media into a machine read-able format adds usable data


Point-of-input correction, autocomplete, and interface cues reduce errors at source


Code cleaning removes errors and format problems for additional quality control


No manual reentry steps reduce errors and delays as data is relayed


Digital tracking of samples and cases adds efficiencies and improved oversight


Automation of processing reduces delays and frees up time for higher-value tasks


Safe ingestion of legacy document formats prevents security breaches

SECURE INTAKE

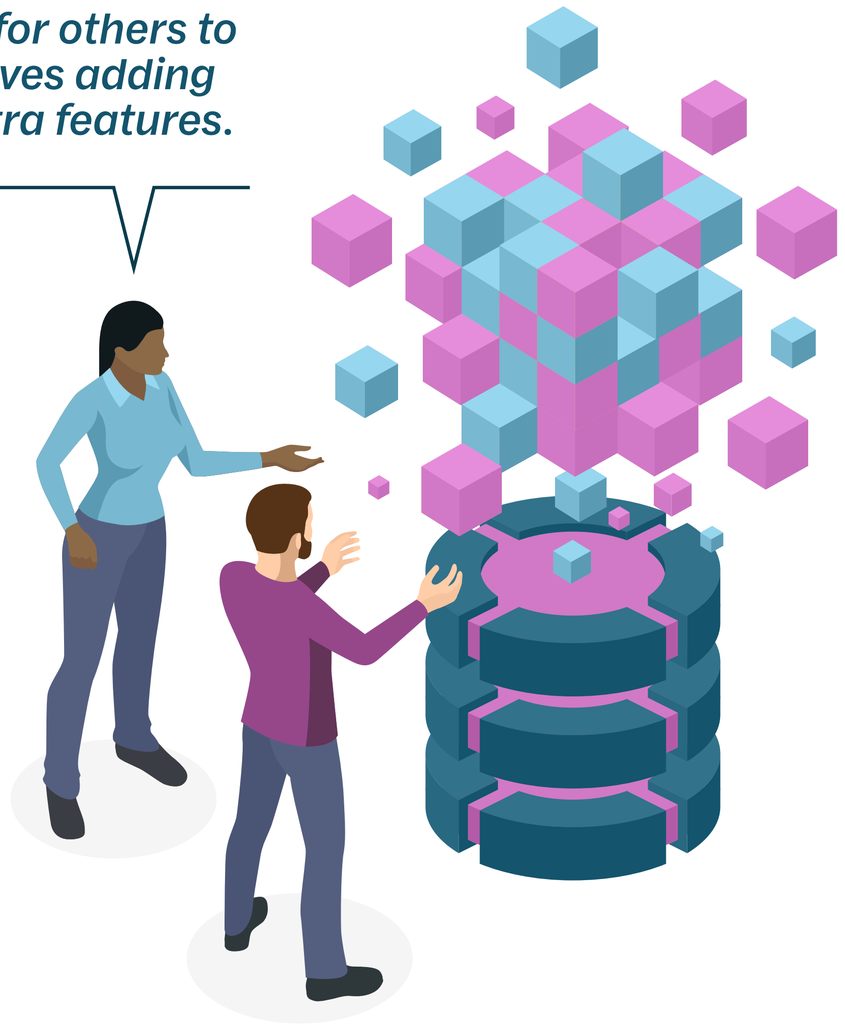
Data ingestion from legacy document formats (e.g., emailed spreadsheets) is inherently insecure. If adding APIs to upstream sources is not an option, a safe EPL process is used, meaning:

- data is **Extracted** from documents before the file enters the system;
- data is **Transformed** into a useful form with metadata preserved, suspicious code is removed in an *input validation* process; and,
- data is **Loaded** into encrypted pipelines for safe transport.

DATA AS A PRODUCT

Data owners retain decentralized control in exchange for making their data easier to discover, access, and use

Treating data like a product for others to use involves adding a few extra features.



Databases are **in the cloud** so that the latest data are always accessible for others to use

Access is through **self-serve data streams** using APIs that apps can easily tap into

Data are **self describing** via meta-data so that meanings and methods are obvious at the point of access

Data are managed according to **FAIR principles**, making them Findable, Accessible, Interoperable, and Reusable

Flow-control code enforces policies, such as access restrictions, terms of use, and provenance assurance

FEDERATED DATA

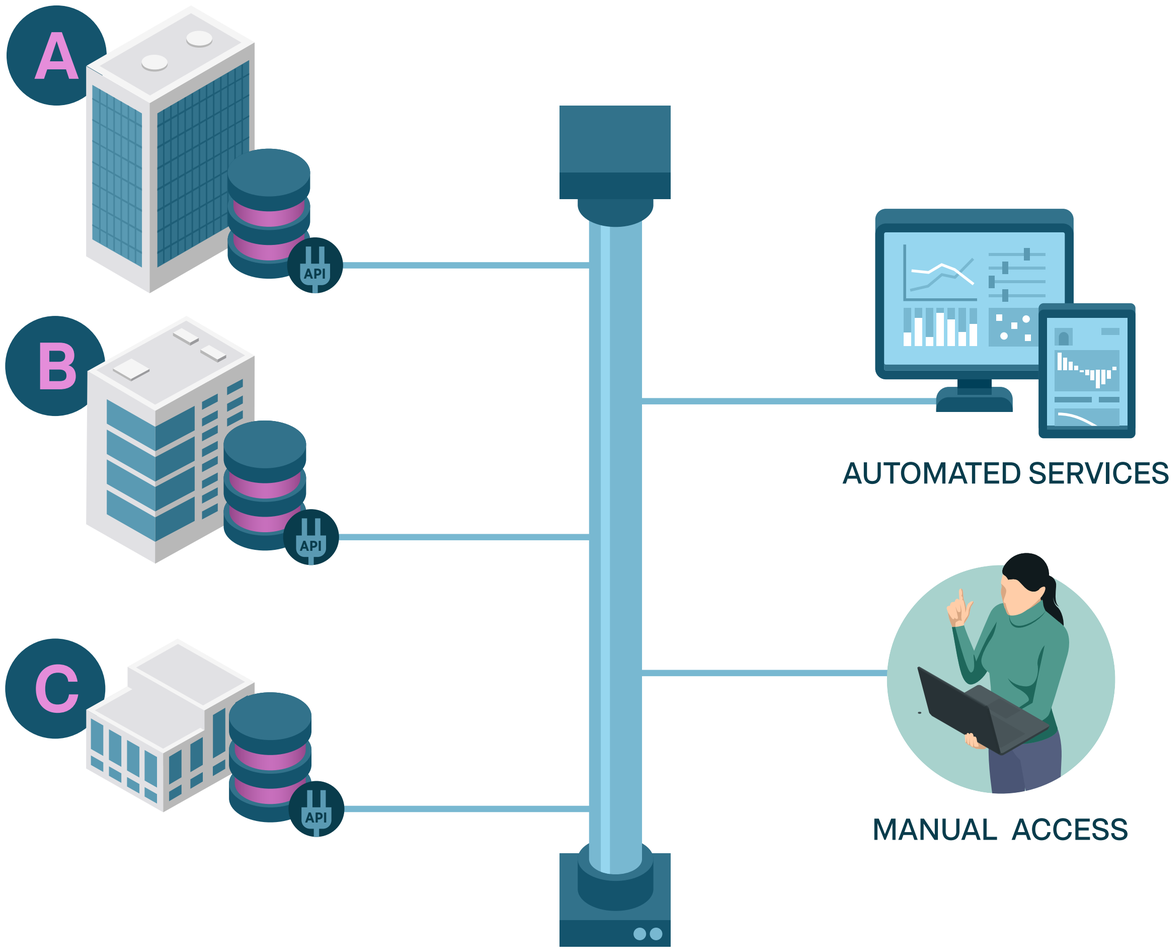
Data owners retain control of their data but make streams available to services through a common gateway

LOCAL CONTROL

Those closest to the subject-matter being measured, the data owners, develop their own policies. Access and use remains under their control.

PRODUCTIZED DATA

Data is prepared for use elsewhere by making it self-serve and self-describing through APIs. Basic interoperability is preserved according to FAIR principles. Flow-control code is used to enforce policies.



INTEROPERABILITY

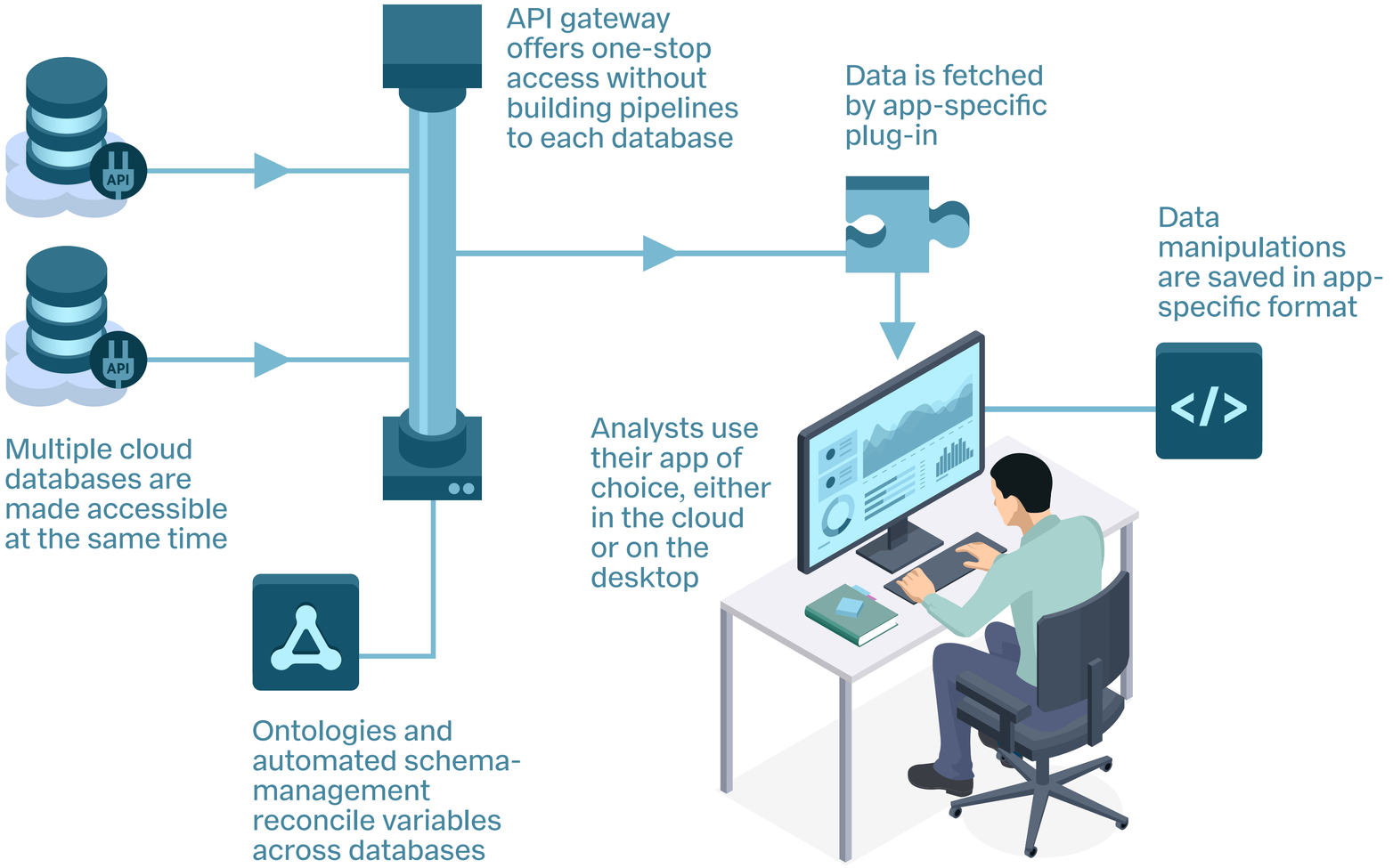
Policies necessary to preserve inter-operability governed cooperatively by data owners and gateway administrators. Semantic ontologies are openly published and help manage schema between databases.

NO DATASET TRANSFER/POOLING

Datasets stay where they are as a “single source of truth.” Individual variables are accessed as streams of data and are always up-to-date. No copies of the data-set are transferred. There is no pooling of datasets, as with data warehouses or data lakes. The only operational data is temporarily stored by micro-services.

DATA ACCESS BY ANALYSTS

Advanced users can analyze up-to-date data with their preferred apps with a few tweaks to their workflows

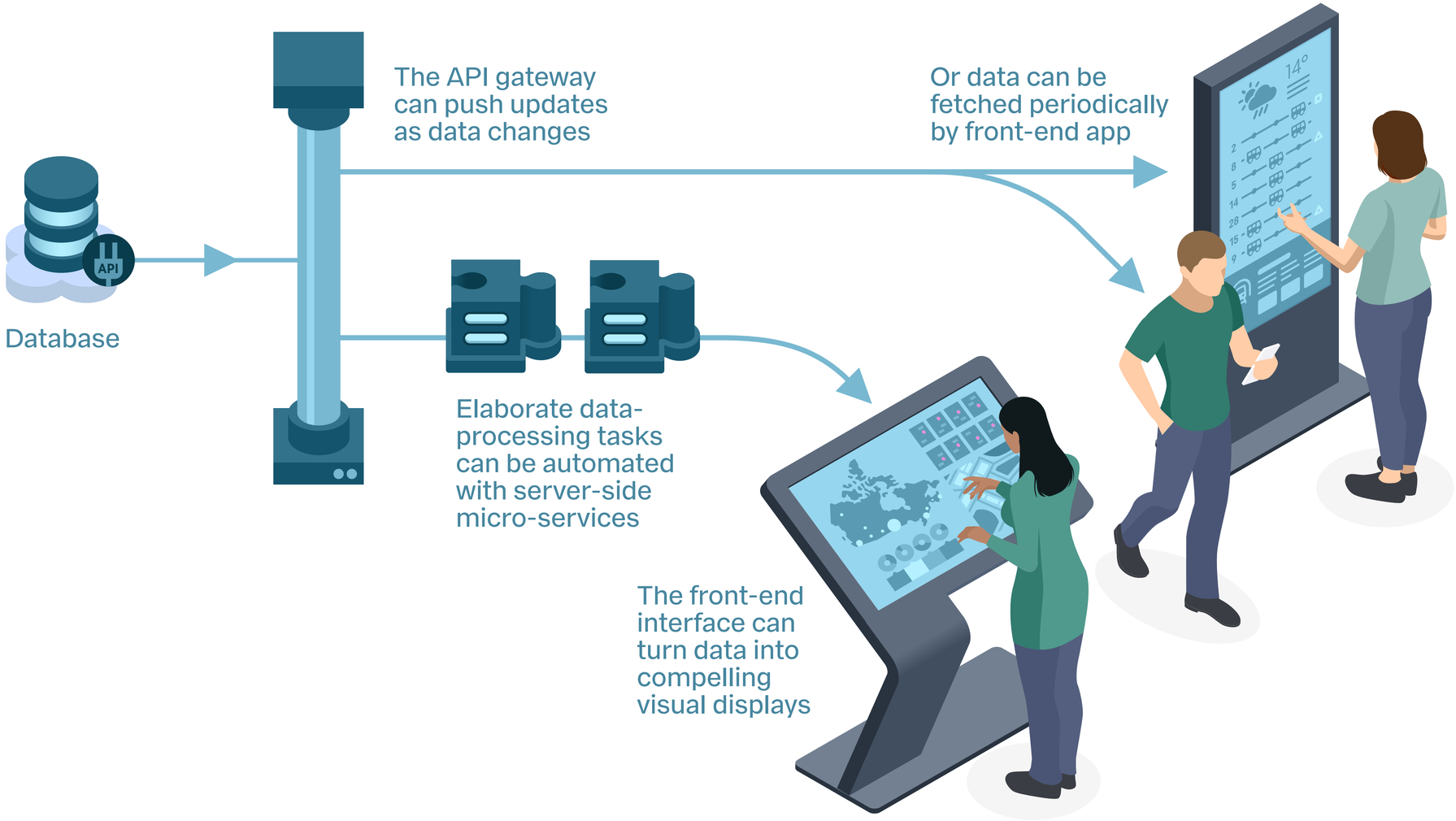


There is no copying and sharing of data files. Analysts have no direct database access. Instead, the latest data is fetched through an app-specific plug-in. Multiple databases can be accessed and combined, with the reconciliation work done automatically for maximum interoperability.

Professional applications have their own syntax (such as Python, R, or SAS language) or tools (such as Tableau Prep Builder) for saving data manipulations. Each time the data is refreshed, the manipulations can be reapplied, keeping everything current.

REAL-TIME DATA DISPLAYS

If data is automatically fed into the database, data displays can be updated immediately for the user

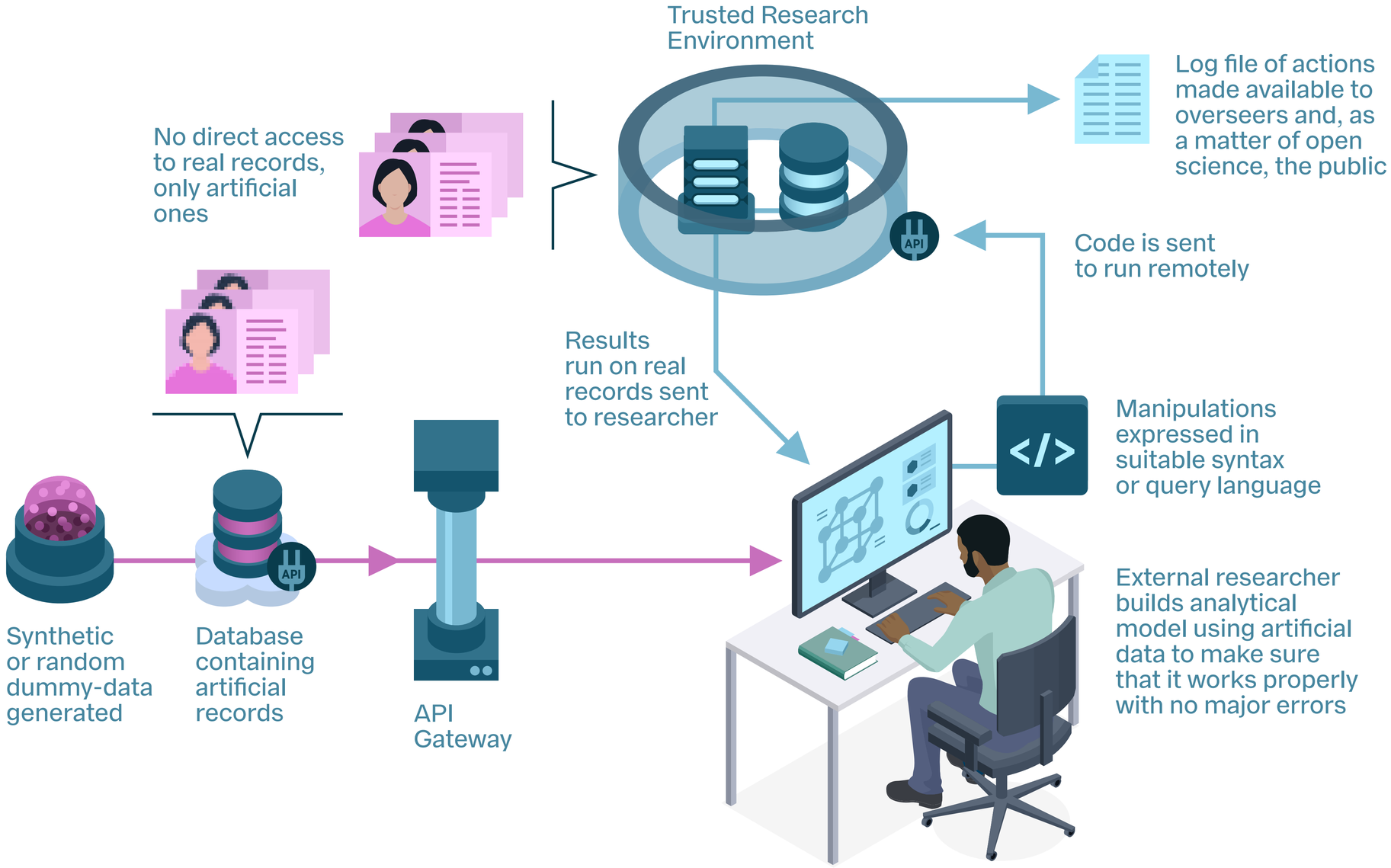


Much of the work of displaying data is done by front-end interface applications running on the user's device. The data is also fetched by a front-end module, such as Relay if the interface is built using the React framework.

If data processing requires a lot of computer horsepower, that will have to take place using apps on the cloud server. For example, a string of micro-services may perform several artificial intelligence operations, such as translate and summarize text.

EXTENDED ACCESS

Outside researchers can work with real data without any risk to citizen privacy

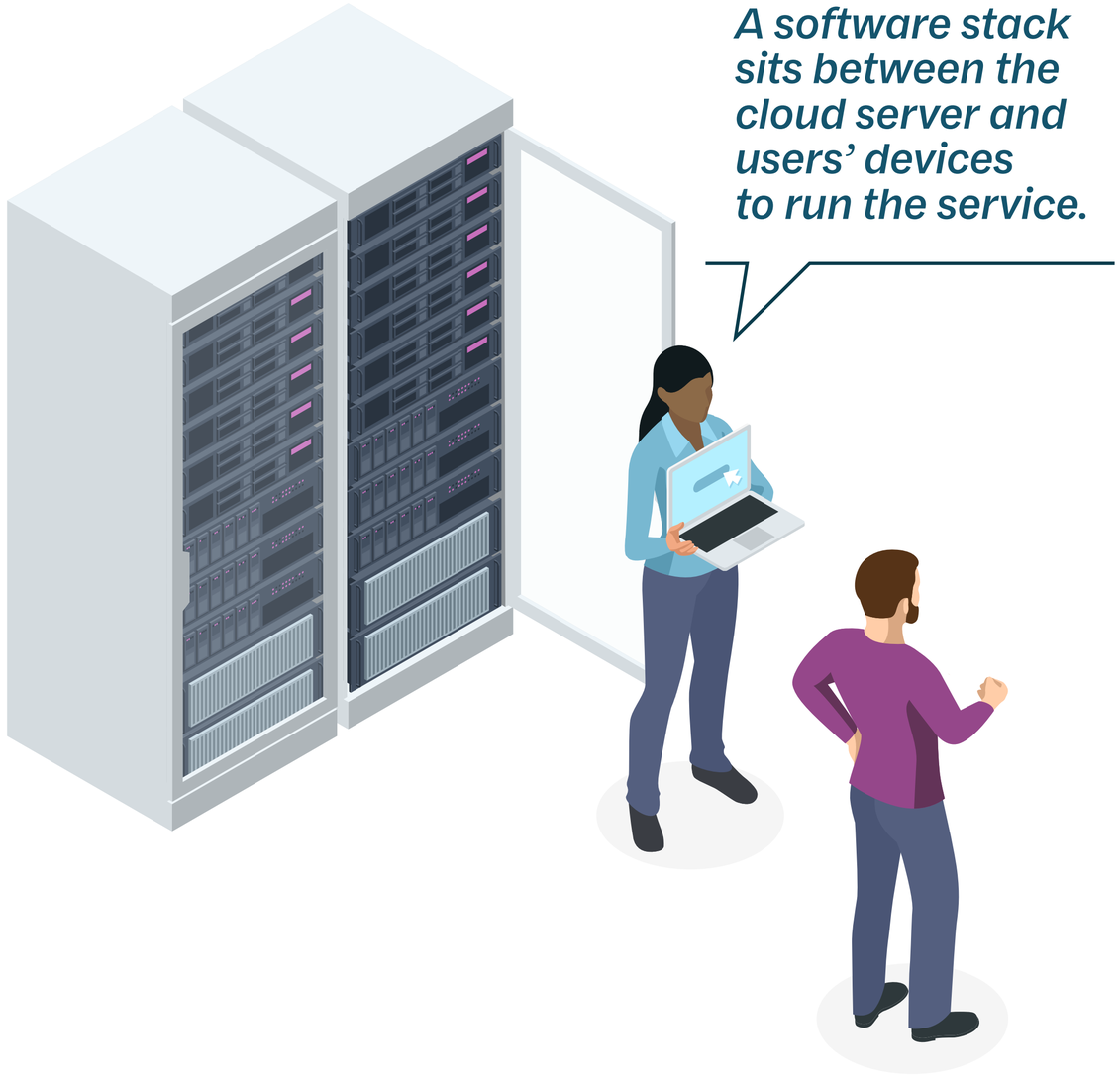


An API gateway can offer fine grained permissions to data controlled by data owners. With zero-trust security measures, access can even be tied to particular tasks, devices, and settings. For outside actors, a Trusted Research Environment (TRE) adds further assurances.

For example, Oxford University's OpenSAFELY system offers access to medical records. Instead of redacting real records or working only with synthetic data, real records can be worked with in a way that does not put privacy at risk.

SOFTWARE STACK

Application architectures will vary. The trend is away from all-in-one monoliths and towards flexible modularity.



A software stack sits between the cloud server and users' devices to run the service.

USER INTERFACE

The way users interact with the data service is made helpful, intuitive, and accessible to different users and devices.

CLIENT-SIDE APP MODULES

Various tasks may be run by app modules on the user's device, making calls to micro-services or the database.

MICRO-SERVICES

Server-side micro-services can be used for compute-intensive tasks or strung together, often with help from coordination services, such as a service registry or event manager.

DATABASE

A networked database with an API allows any-time access to data by services.

CLOUD CONTAINER

Containers (such as Docker and Kuber-netes) are self-contained spaces in which apps run so that services can scale easily and are not tied to one cloud provider.

SOFTWARE FACTORY

Generating open-source, re-usable software assets allows new data services to get up and running quickly

Software is made available in a public repository for reuse; more consistency, less rework.

A unified **design language** maintains a common look-and-feel automatically by sharing design tokens.

Software modules and sub-systems are **reused** across projects to minimize duplicated effort.

New services are created **faster and cheaper** with less need to demonstrate policy compliance across projects.

Everything stays current by **swapping out modules** with newer versions without the need to refactor whole systems.

Extra **quality control** is added to code as a larger community of developers squash bugs and security vulnerabilities.