DATA FLOW DIAGRAMS

for Business Analysts

Following information through a system, from traditional analysis to modern privacy questions

COURSE HANDBOOK

A practical guide to modelling information movement and investigating data protection

Northbridge Equipment Hire case study • Worked examples • Exercises • Practical reference

How to use this handbook

This handbook is a complete narrative companion to the Data Flow Diagrams for Business Analysts course. It develops the ideas in a logical sequence, explains why each modelling decision matters, and uses the Northbridge Equipment Hire case study to connect diagrams to real business questions. It is intended to remain useful long after the lessons have finished.

CORE IDEA

A Data Flow Diagram describes where information comes from, what transforms it, where it is retained and where it goes. It is not a flowchart of activities.

The first part develops conventional DFD modelling: scope, external entities, processes, stores, flows, decomposition and quality. The second part applies those same models to personal data, sharing, retention, individual rights and privacy investigations. Throughout, a DFD is treated as a disciplined way to ask questions, not a substitute for speaking to the people who understand the business.

Learning outcomes

- Identify and name the four basic DFD elements and distinguish them from flowchart symbols.
- Establish a system boundary and construct balanced context and Level 0 diagrams.
- Explain logical versus physical information movement and use both views to investigate change.
- Diagnose black holes, miracles, grey holes and other common modelling errors.
- Overlay privacy questions on data flows, stores, recipients and boundary crossings.
- Use DFD evidence to support data minimisation, retention, sharing, rights and ROPA investigations.

Contents

1. Understanding information movement
2. The four elements and their rules
3. Scope, context and decomposition
4. Logical and physical perspectives
5. Reviewing and improving DFDs
6. Applying a privacy lens
7. Sharing, security and individual rights
8. The Northbridge final investigation
9. Practical BA reference

CHAPTER 01

Understanding information movement

What a DFD reveals, and why the technique still matters.

WELCOME

Follow the information, not merely the activity

When an analyst hears that a customer makes a reservation, a warehouse prepares equipment and a payment provider takes a deposit, it is natural to describe the sequence of activities. Yet an equally important set of questions is easily missed. What information does each participant need? Where did it originate? Who changes it? Which copy is authoritative? Where does it eventually come to rest?

A Data Flow Diagram (DFD) makes those questions visible. It represents the system as transformations of information connected by named flows. It deliberately leaves out many details of timing, user interface and technical implementation. That restraint is its strength: it allows a business analyst to investigate information requirements without prematurely choosing software or hardware.

The method emerged from structured systems analysis in the 1970s, when analysts needed to understand complex mainframe systems. The technology has changed almost beyond recognition, but the analytical problem has not. A mobile application may send a request through an API gateway, a cloud service and a third-party platform. Somebody still needs to know exactly what information moves across each boundary.



From the structured-analysis era to modern information systems.

CASE STUDY

Northbridge Equipment Hire

Northbridge rents tools and construction equipment. Its customers register, check availability, reserve equipment, collect and return items, report damage and pay charges. Behind those seemingly simple activities are customer records, equipment details, reservations, rental agreements, payment confirmations and information exchanged with external services.

Our purpose is not to design Northbridge's final system at the outset. We begin with the information entering and leaving a proposed system, then uncover its internal transformations and stores. Later we ask what changes when some of that information is personal data.

KEY POINT

A DFD is an information model. The arrows are named information, not instructions telling the system what to do next.

CHAPTER 02

The four elements and their rules

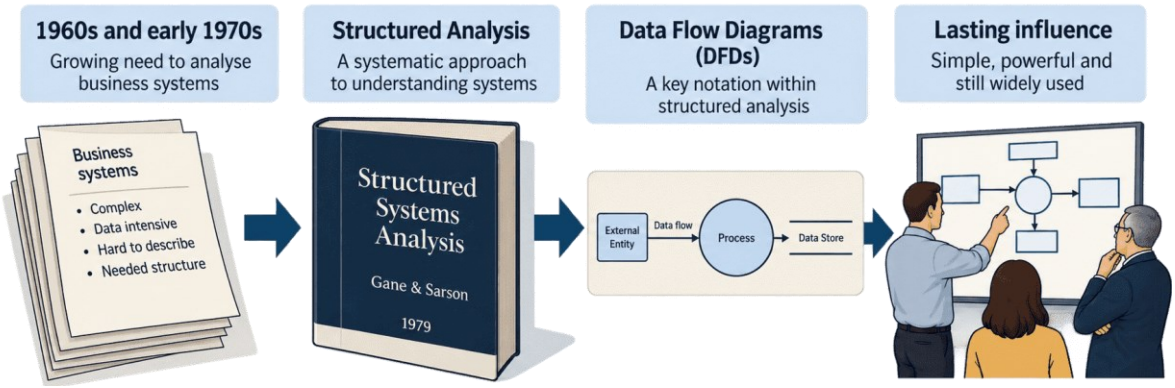
Build a sound model from meaningful sources, transformations, flows and stores.

EXTERNAL ENTITIES

Where the information enters and leaves

An external entity is a person, organisation or other system outside the boundary of the system under study. The entity may supply information, receive information or do both. For Northbridge, a Customer, Payment Provider and Insurer are plausible external entities.

External does not necessarily mean outside the company. A Collection Branch might be part of Northbridge as an organisation yet still be external to a narrowly defined Online Reservation System. The analyst must therefore ask “external to what?” before placing any entity on the diagram.



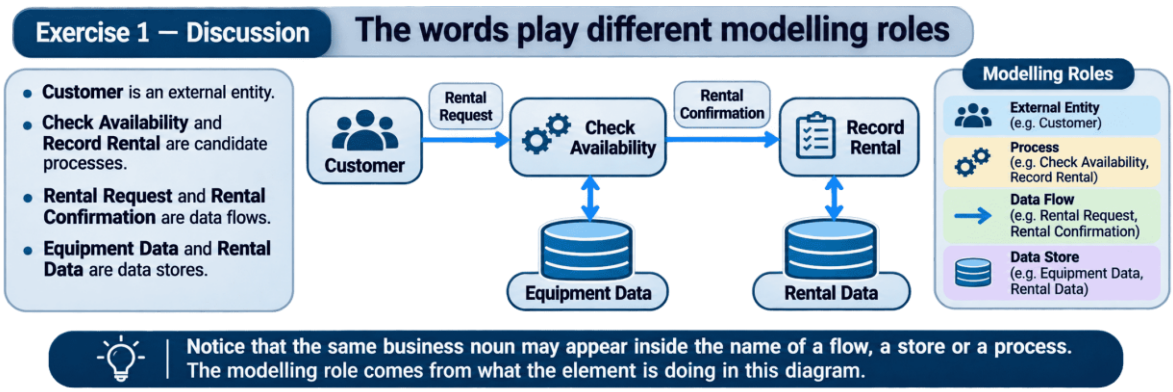
An external entity is defined relative to the modelled system boundary.

PROCESSES

Show a genuine transformation

A process takes information and transforms, validates, combines, derives or records it. Process names are normally active verb phrases such as Check Availability, Create Reservation or Record Rental. A noun such as Reservation does not explain what processing takes place.

Every process should have meaningful inputs and outputs. A process with an input but no output is a black hole; one with an output but no sufficient input is a miracle; and one that produces more information than its inputs could justify is a grey hole. These labels are useful because they turn diagram inspection into a conversation about missing requirements.



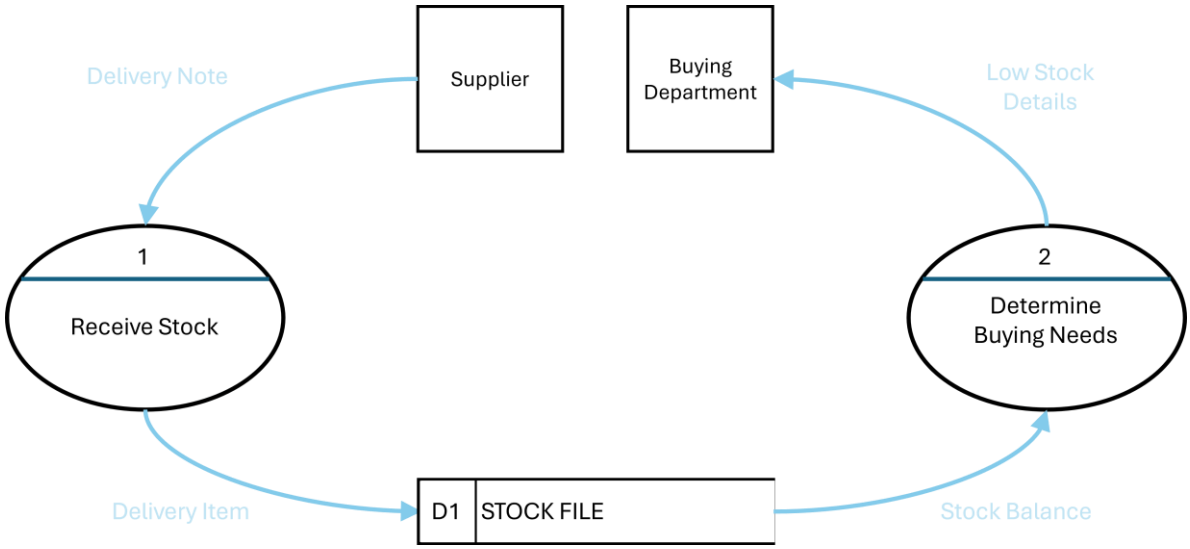
Process logic should be checked for inputs, outputs and derivability.

DATA FLOWS

Name what actually moves

A data flow is information passed between elements. Names should be noun phrases that tell a reader what the arrow contains: Rental Request, Availability Details, Payment Confirmation or Customer Details. An arrow labelled “data” gives no one a useful question to investigate.

Information can be copied to more than one destination and separate flows can be combined when their meaning is clear. But a branching arrow does not represent an if/then decision. A DFD documents the existence and content of information exchanges, not the order in which a program executes.



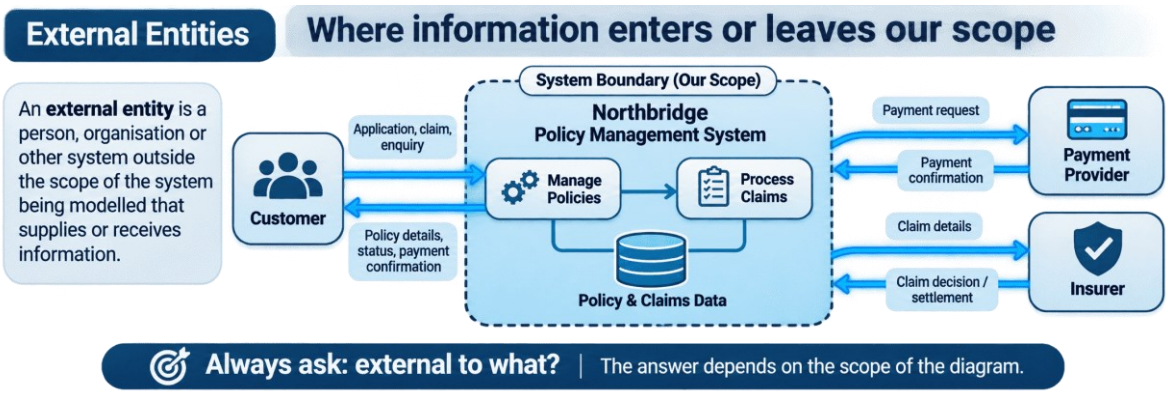
A meaningful flow name describes information rather than a vague message.

DATA STORES

Information retained for later use

A data store is information that remains available beyond a single processing step. Customer Data, Equipment Data and Reservation Data are plausible logical stores. A logical store does not commit the design to a database table, spreadsheet, file, cloud repository or paper folder.

Stores should be named for the information they contain. If a process reads or updates a store, the flows should make that exchange explicit. On a logical DFD an external entity should not be drawn directly accessing a data store: processing within the system must mediate the exchange.

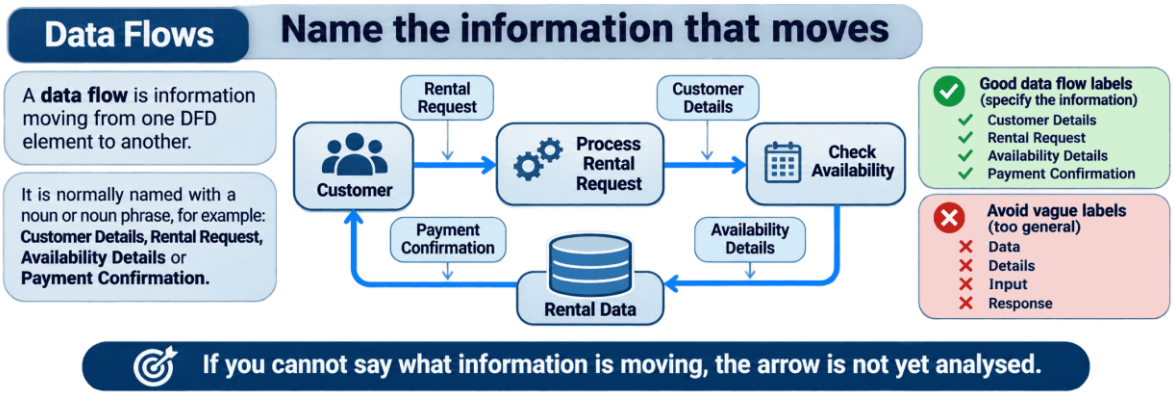


Logical data stores represent retained information, not particular storage technology.

EXERCISE 1

Distinguish the modelling roles

Consider the words Customer, Check Availability, Record Rental, Rental Request, Rental Confirmation, Equipment Data and Rental Data. The first is an external entity; the next two are processes; the two requests or confirmations are flows; and the final two are stores. The same business noun can appear in several labels, but its modelling role depends on what the element represents.



The four element types play different roles in one information model.

CHAPTER 03

Scope, context and decomposition

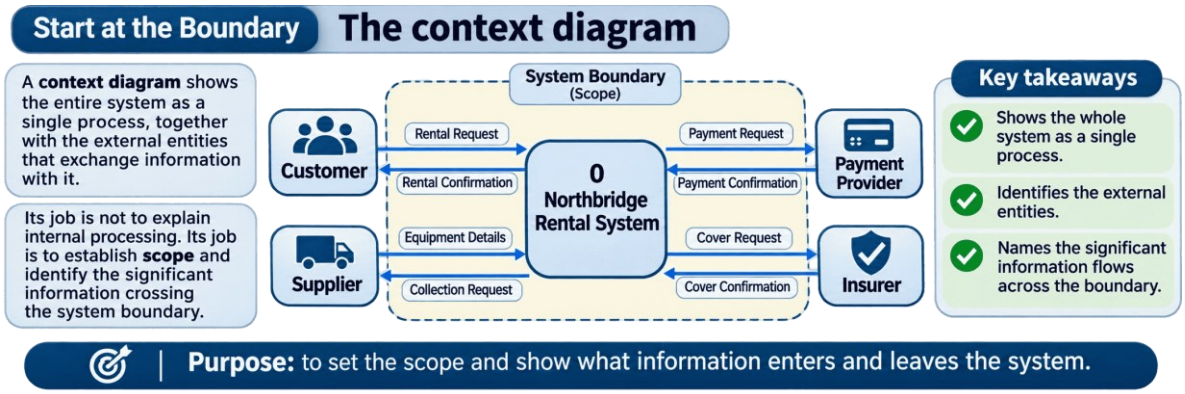
Move from the system boundary to a balanced view of its major processes.

START AT THE BOUNDARY

The context diagram

The context diagram is the highest-level picture of a system. It represents the whole system as a single process and shows the significant external entities exchanging information with it. The central question is not how the system works internally; it is what lies inside its scope and what crosses that boundary.

For Northbridge, a Rental Management System might receive a Rental Request from a Customer, send a Payment Request to a Payment Provider, receive a Payment Confirmation and exchange claim information with an Insurer. The labels must describe real information, and the boundary must be agreed with stakeholders.



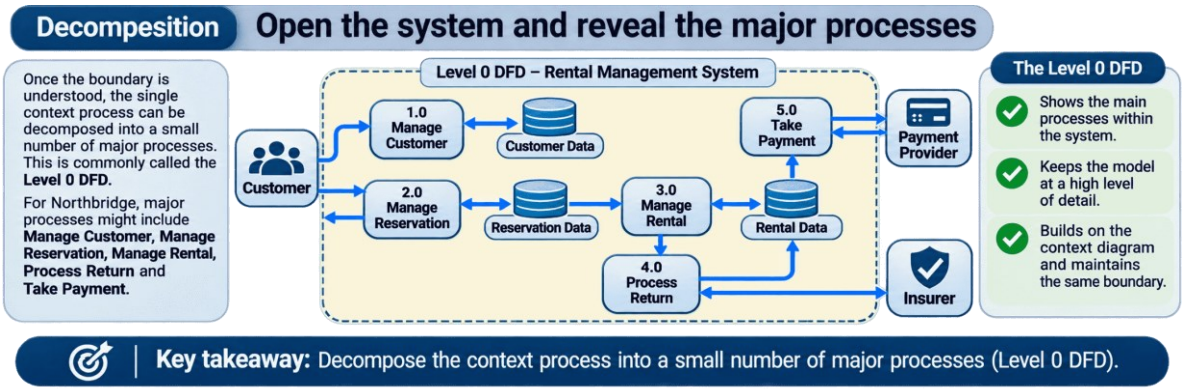
A context diagram makes system scope and external information exchanges explicit.

DECOMPOSITION

Open the system without losing its external obligations

Once the context is understood, the single process can be decomposed into a Level 0 DFD. Northbridge might use Manage Customer, Manage Reservation, Manage Rental, Process Return and Take Payment. The decomposition reveals major transformations and may introduce logical stores such as Customer Data, Equipment Data and Rental Data.

Processes are commonly numbered 1.0, 2.0 and 3.0. When 3.0 is decomposed, its child processes might be numbered 3.1, 3.2 and 3.3. Those numbers identify a position in the hierarchy; they do not prescribe execution order. DFDs are not flowcharts.



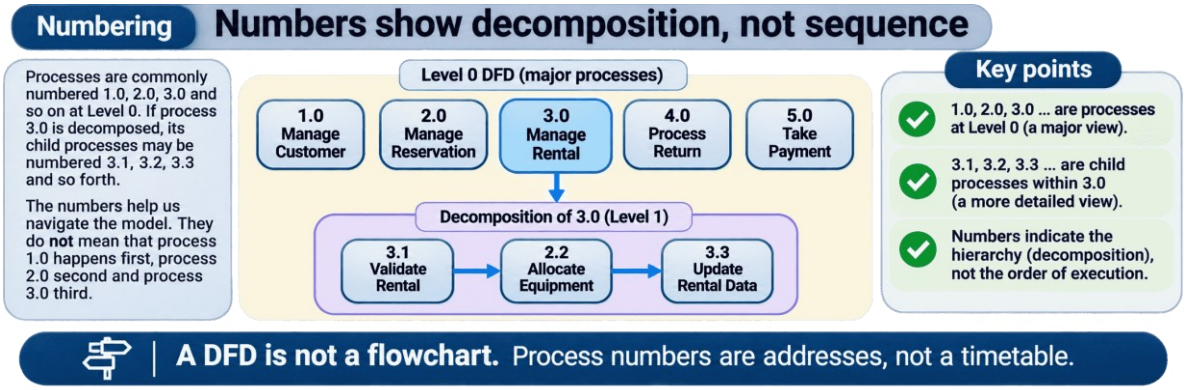
A Level 0 DFD reveals major processes and internal information stores.

BALANCING

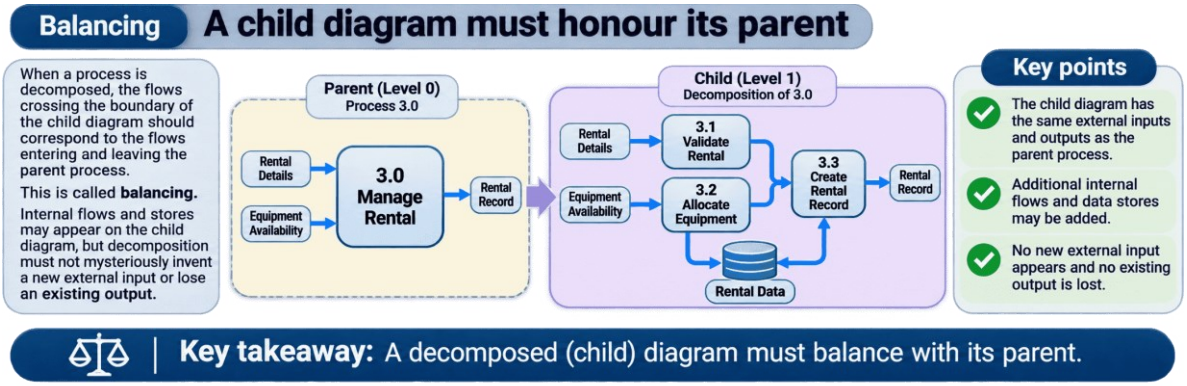
A child diagram must honour its parent

Balancing means that the information crossing a decomposed process boundary must correspond to the information entering and leaving the parent process. A child may reveal additional internal stores and flows, but it should not invent an unexplained external input or silently lose an output.

Imagine the parent process receives Reservation Request and produces Reservation Confirmation. If its child diagram suddenly requires Credit History from an external organisation, the analyst must investigate. Perhaps the parent diagram omitted a genuine exchange, or perhaps the child has exceeded the agreed scope. The discrepancy is valuable evidence, not merely a drawing error.



Process numbers identify decomposition levels, not the sequence of activities.



Balancing checks the external flows of parent and child diagrams.

BA INSIGHT

Do not decompose until every process is microscopic. Stop when the model is detailed enough to answer the requirements questions it was created to address.

CHAPTER 04

Logical and physical perspectives

Separate the enduring information requirement from the technology that implements it.

LOGICAL VERSUS PHYSICAL

Do not freeze today's solution into tomorrow's requirements

A logical DFD describes the information processing required by the business. A physical DFD describes how that processing is or could be implemented: people, applications, files, devices and organisational units. Both views can be valuable, but they answer different questions.

Consider a paper rental form handed from a counter clerk to an office administrator. That is a current physical arrangement. The logical requirement might be simply to capture Rental Details, validate them and make the resulting Rental Record available. A future implementation might use a web application and API, but the underlying information need remains recognisable.

Confusing the physical and logical views causes analysts to preserve avoidable constraints. "The administrator rekeys the form" is a description of current work, not necessarily a requirement for a new system.



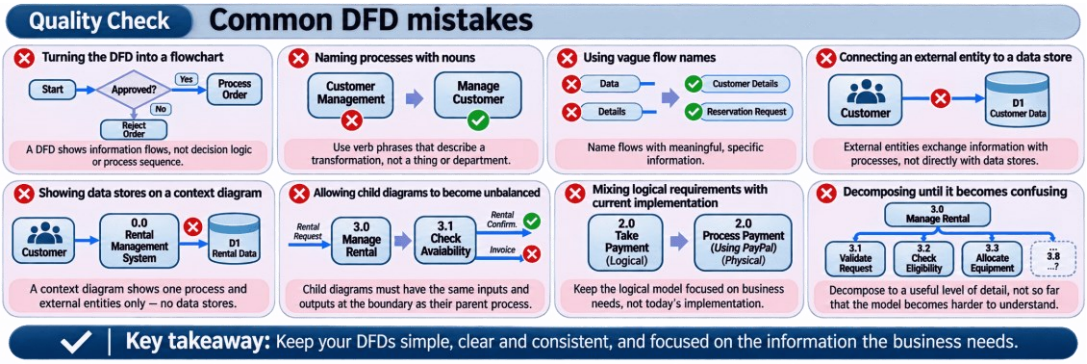
Current physical, logical requirement and future physical are distinct views.

DEFINING THE ARROWS

A diagram is only the beginning

Flow labels should be supported by definitions of their contents, origin, destination and meaning. "Customer Details" might contain name, contact information and a customer reference. Does it also include date of birth, billing history or identity evidence? The answer matters for requirements, integration design and privacy.

When a flow splits, confirm whether each recipient receives the same information or a carefully selected subset. A delivery partner may need a name, delivery address, telephone number and delivery reference, but not the customer's entire account history.



Named flows should be defined, not treated as decorative arrows.

CHAPTER 05

Reviewing and improving DFDs

Use the model to find gaps, ambiguities and misleading assumptions.

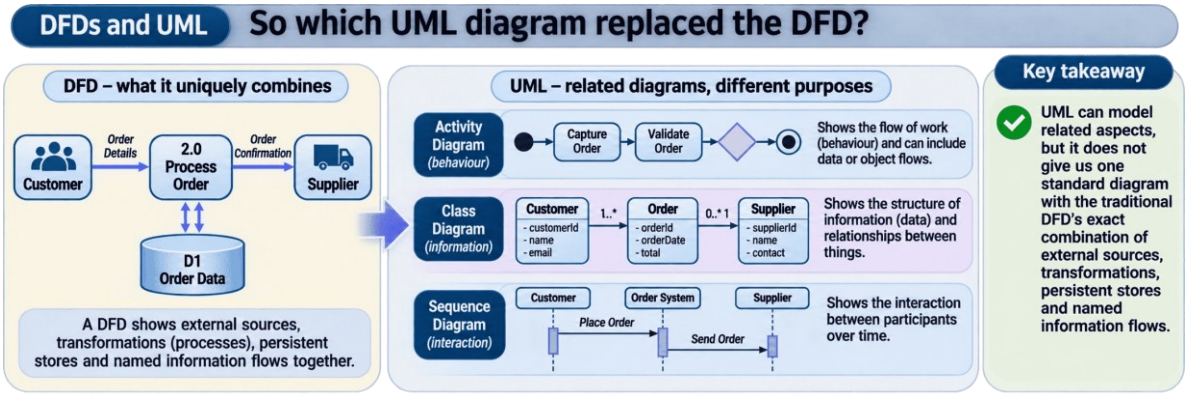
QUALITY CHECKS

Make the diagram explain something true

A DFD is useful only if its elements make business sense. The most common failures are surprisingly basic: processes named as nouns, flows called “data”, stores appearing on a context diagram, direct links from an external entity to a store, and child diagrams that no longer balance with their parents.

Another error is turning the DFD into a procedural flowchart. An arrow from Check Availability to Create Reservation may legitimately carry Availability Details; it should not mean “do this next”. The analyst should be able to read every arrow aloud as a sentence about information.

Equally, a DFD can become too detailed. A forest of tiny processes may conceal the real transformations. Decomposition is valuable only while it improves understanding.



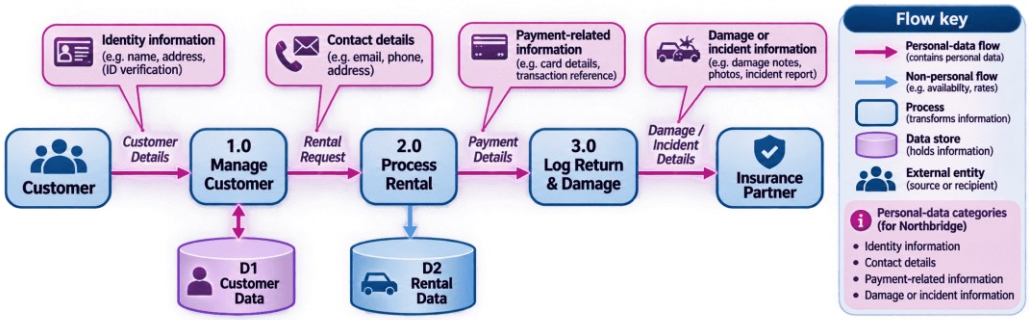
Reviewing a DFD means testing meaning, connections and consistency.

USING OTHER MODELS

DFDs and UML answer overlapping but different questions

No single UML diagram replaces the traditional DFD exactly. An activity diagram can describe behaviour and object movement, a sequence diagram shows interactions over time, and a class diagram models information structure. None offers precisely the traditional DFD combination of external sources, processes, persistent stores and named flows.

The models are complementary. A class diagram may explain the structure of Customer and Reservation, while a DFD explains where Customer Details are obtained, transformed, stored and disclosed. Choose the model that exposes the question you are trying to answer.



CHAPTER 06

Applying a privacy lens

Turn ordinary data-flow models into focused personal-data investigations.

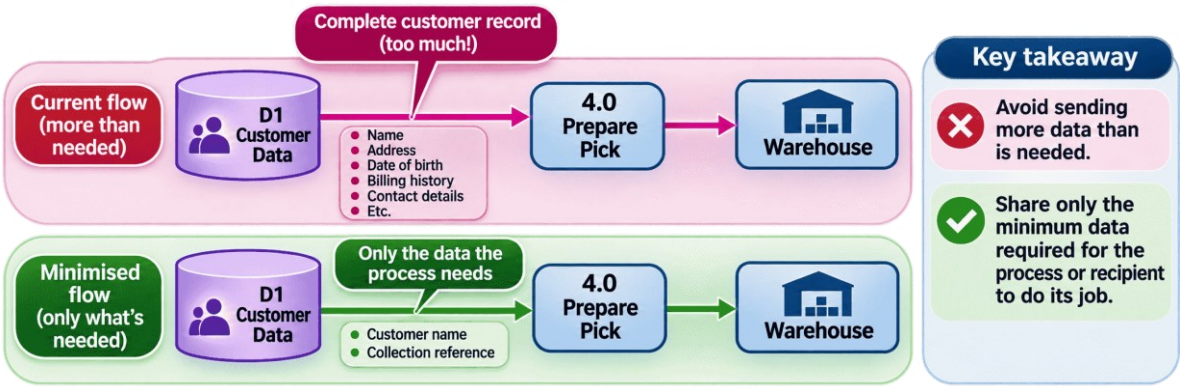
PRIVACY OVERLAY

Use the ordinary DFD as evidence

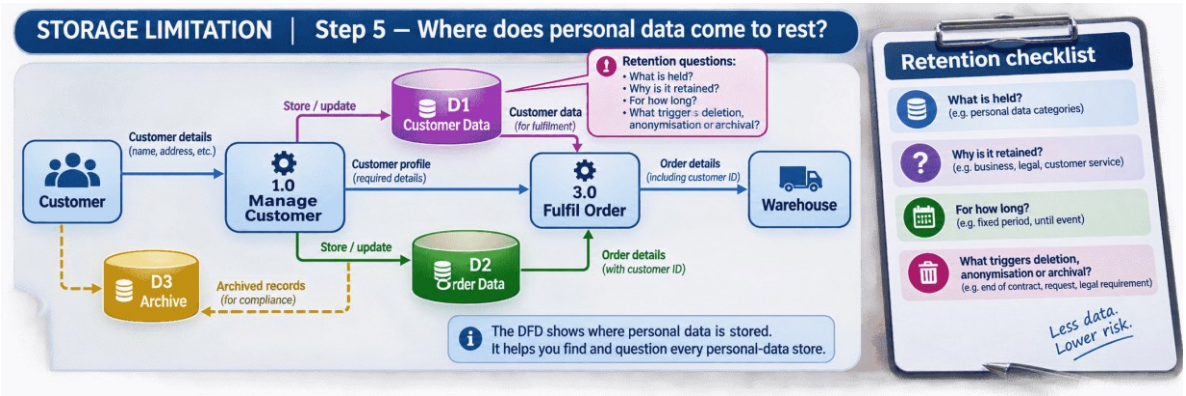
A DFD can become a practical starting point for privacy analysis because it already identifies information sources, transformations, stores, recipients and boundaries. The first step is to mark which flows contain personal data and to identify the actual categories carried. “GDPR” is not a useful flow label; Customer Name, Contact Details, Identity Evidence and Payment-related Information are.

The next question is why each processing activity happens. A reservation process may need contact information to confirm a booking, while an incident investigation may need a different category of information for a different purpose. A map cannot determine the lawful basis by itself; it reveals the processing that requires legal and governance assessment.

For each personal-data flow, ask what data is needed, what is used, who receives it and what happens next. The map can show that a warehouse process receives a complete customer record even though it needs only Customer Name and Collection Reference. That is a concrete data-minimisation question, not a vague appeal to compliance.



Mark personal-data categories on flows rather than labelling an arrow “GDPR”.



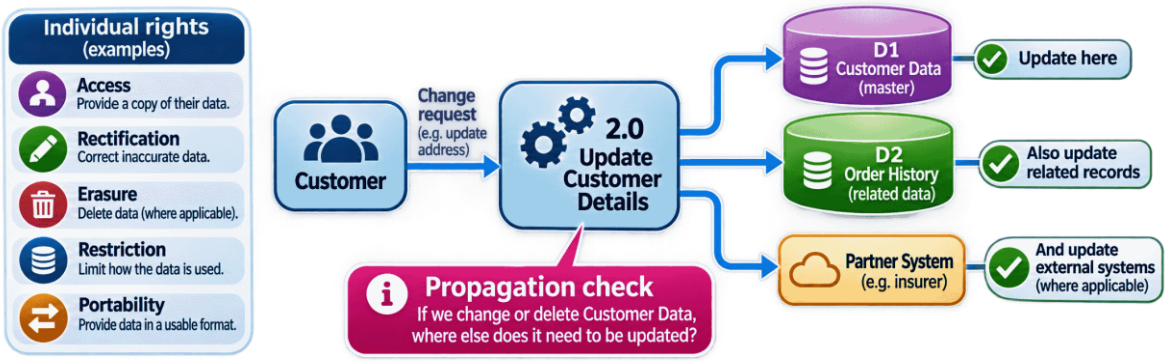
Privacy analysis asks whether every recipient needs all the information it receives.

RETENTION

Where does the personal data come to rest?

Every store containing personal data prompts questions about retention: what is held, for what purpose, for how long and under which deletion, anonymisation or archival rule? A logical DFD may show Customer Data, but a physical investigation may reveal additional exports, caches, backups and spreadsheets.

Those secondary copies matter. Deleting a customer from the primary application does not necessarily remove information from a downloaded report or third-party system. The map helps identify the places that must be investigated, even though it cannot establish retention periods on its own.



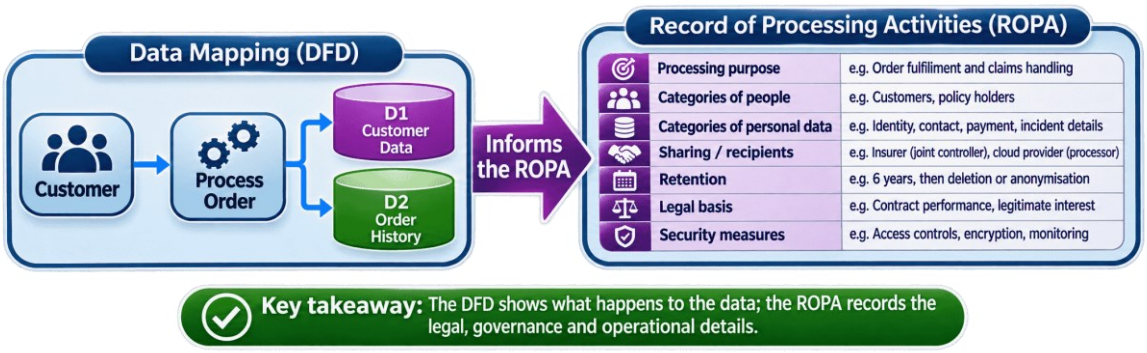
Storage limitation depends on knowing where copies of information are retained.

FROM MAPPING TO GOVERNANCE

Support a Record of Processing Activities

A Record of Processing Activities (ROPA) captures processing purposes, categories of individuals and personal data, recipients, retention and other governance information. The DFD contributes a practical map of the processing landscape; the ROPA records information that the diagram itself does not contain.

The distinction matters. A diagram does not certify compliance. It is a maintained analytical source that can help prevent forgotten processing activities from falling outside the organisation's records.



A maintained DFD can inform the ROPA without replacing it.

CHAPTER 07

Sharing, security and individual rights

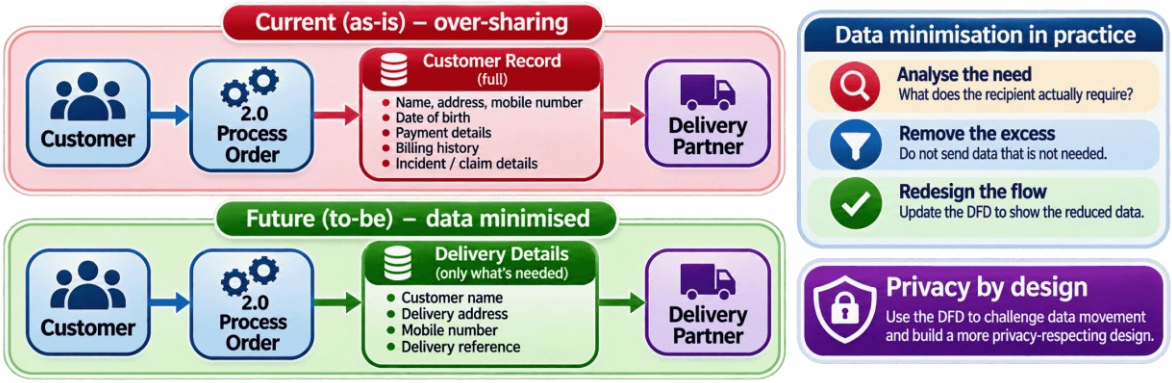
Follow personal data across organisational and national boundaries.

SHARING

Identify recipients and their roles

External entities are especially important when personal data is involved. Northbridge may exchange information with a Payment Provider, Insurer, Delivery Partner or cloud service. Each exchange should be examined for its purpose, the categories disclosed and the recipient's role.

An external organisation may act as a separate controller, a joint controller or a processor operating on instructions. These are not interchangeable labels. They influence which agreements, responsibilities and governance arrangements must be examined. The DFD tells the analyst where to ask; it does not decide the legal classification.



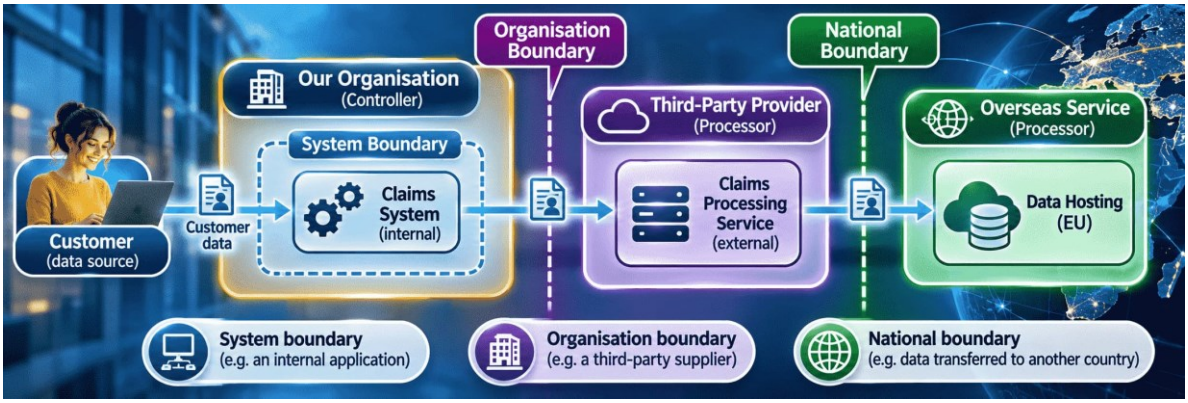
Data sharing requires analysis of the recipient, the purpose and the information disclosed.

BOUNDARIES

System, organisation and country are different boundaries

Information may leave a system without leaving its organisation, leave an organisation without leaving the country, or cross a national border through hosting, support or another recipient. Those are three distinct boundary questions and should not be collapsed into one.

A national transfer can require further legal investigation. The diagram helps locate the exchange and identify the data categories and destination, but the rules applicable to a particular transfer need separate assessment.



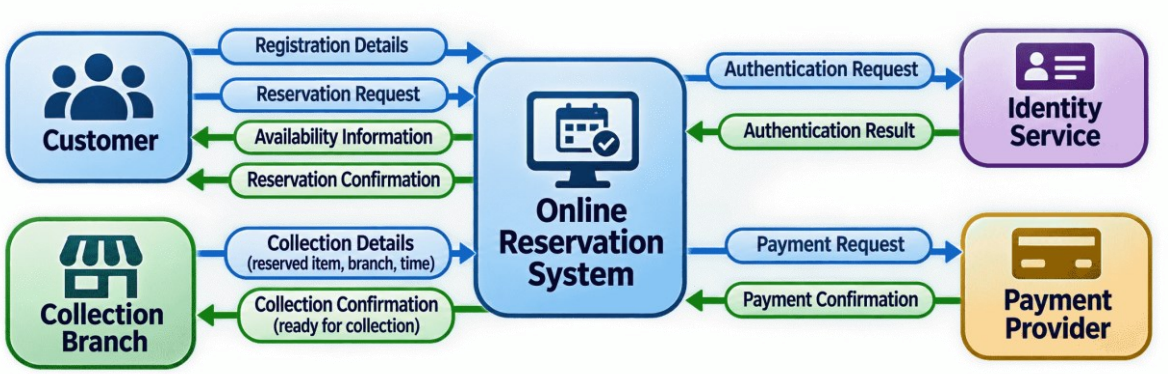
Information can cross system, organisational and national boundaries.

INDIVIDUAL RIGHTS

Can we find and act on the information?

Access, rectification and erasure requests become much harder when data has propagated into downstream systems. If Northbridge corrects Customer Data, what happens to reservation exports, delivery records and external recipients? Does the organisation know where copies are held and who is responsible for acting on them?

The DFD gives a route through that investigation. It shows how information is distributed, allowing analysts to trace possible consequences of correcting or deleting a record. It does not determine whether a particular request must legally be granted.

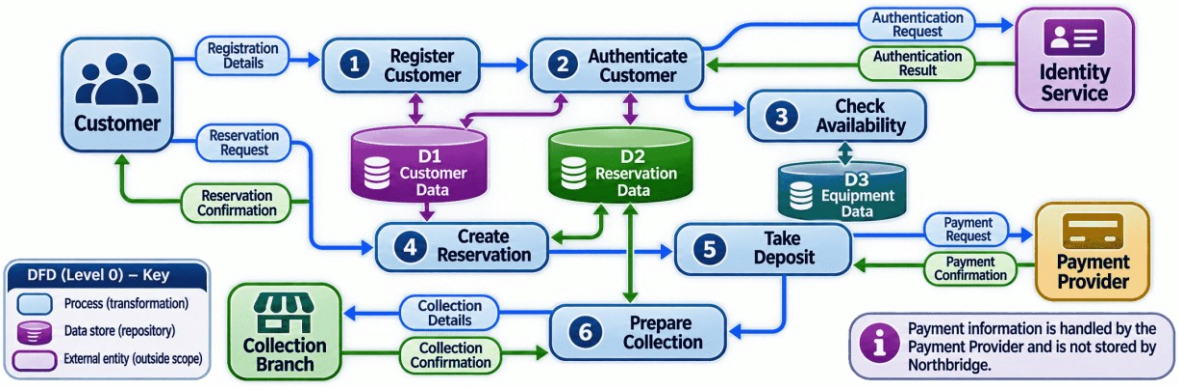


Rights investigations require tracing personal data through downstream copies.

SECURITY AND RISK

Make exposure points visible

Every additional recipient, interface, store and export may introduce another exposure point. The map can support a risk conversation about access controls, logging, encryption, supplier arrangements and higher-risk processing. Those safeguards are not automatically specified by a logical DFD, but the diagram provides a systematic place to begin asking.



Trace processing and recipients before deciding which safeguards to investigate.

CHAPTER 08

The Northbridge final investigation

Bring modelling and privacy analysis together in one realistic investigation.

FINAL INVESTIGATION

Define the Online Reservation System

Imagine that Northbridge is introducing an Online Reservation System. Begin by agreeing its boundary. A Customer supplies Registration Details and Reservation Requests. An Identity Service may receive an Authentication Request and return an Authentication Result. A Payment Provider may receive a Payment Request and return Payment Confirmation. A Collection Branch may receive Collection Details.

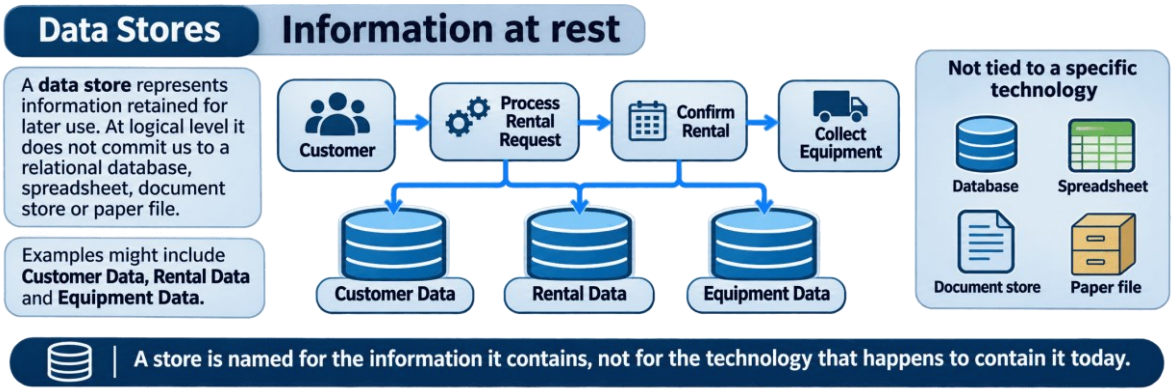
These are candidate flows, not assertions about a completed design. Each one must be confirmed with the relevant stakeholders. In particular, whether a Collection Branch is external depends on whether it lies outside the online system being modelled.

REVEAL THE PROCESSING

A plausible Level 0 decomposition

Inside the boundary, processes might include Register Customer, Authenticate Customer, Check Availability, Create Reservation, Take Deposit and Prepare Collection. Candidate stores include Customer Data, Equipment Data and Reservation Data. The external Payment Provider might retain payment credentials, while Northbridge retains only a payment reference and confirmation.

That arrangement illustrates a deliberate design decision: the system can record the business outcome of a payment without storing every piece of payment information. The analyst should test whether this supports refunds, disputes and reconciliation before treating it as complete.



Northbridge modelling begins by distinguishing the elements and their information exchanges.

APPLY THE PRIVACY QUESTIONS

Challenge the information that crosses each boundary

Now inspect Registration Details. Which personal-data categories does it contain, why are they needed and where are they retained? Examine Authentication Result: is it a simple status, or does it expose unnecessary identity attributes? Examine Collection Details: does the branch need the entire customer record or only the information required to hand over equipment?

Follow each item into the stores and outward to recipients. Ask about retention, onward sharing, overseas access, security and the ability to correct or remove data. The purpose is to uncover design and governance questions while the requirements are still open to improvement.

BA INSIGHT

A useful DFD does not merely confirm the proposed solution. It exposes information movements that stakeholders have not yet justified.

CHAPTER 09

Practical BA reference

A reusable checklist for building, reviewing and maintaining DFDs.

PRACTICAL REFERENCE

A repeatable DFD method

Start with a clear statement of scope. Identify the people and systems that exchange information across that scope. Name each exchange with a meaningful noun phrase and agree its contents. Draw the context diagram before attempting internal detail. Then decompose the central process into a manageable set of major transformations and stores.

Check that every process has defensible inputs and outputs, every store exchange passes through a process, and each child diagram balances with its parent. Separate logical requirements from current implementation. Review the model with the people who know the work and use their disagreements to discover missing requirements.

When personal data is involved, mark the relevant flows and stores. Investigate categories, purposes, lawful basis, minimisation, retention, recipients, transfer boundaries, safeguards, individual rights and the relationship to the ROPA. Keep the map current as systems and processing change.

Questions to ask in a workshop

- What is inside the boundary, and what is outside it?
- What exactly is moving along each arrow?
- What changes in each process, and how is its output derived?
- Where is the information held, and who is responsible for it?
- Does the child diagram preserve all of its parent's external flows?
- Are we documenting a logical requirement or today's physical arrangement?
- Which flows contain personal data, and does each recipient need every item?
- What happens to copies, exports, retention and rights requests?
- Which questions remain unresolved, and who can answer them?

FINAL THOUGHT

The value of a DFD is not the diagram itself. It is the better requirements, design and privacy questions that become possible when information movement is made visible.

Course complete

The same discipline that helped analysts understand mainframe systems can help modern organisations understand APIs, cloud services, data sharing and privacy. Follow the information and keep asking why it moves.