

PlatformOps- Platform Services Landscape

Purpose & Benefits

Rapid mapping of Services/Platforms/Domains or else. Do not get stuck on what they are called or linkage that can come after, this is to understand 'at high level' what exists and who uses it

- Supports a 'Services' lens, allowing to identify services by named 'platforms' or not.
- Enables better conversation between infra, platform, product, and business teams
- Help identify invisible dependencies and unmanaged responsibilities
- Complements ITSM, DevOps, platform engineering & team topologies thinking

Preparation

Audience: Identify who should be involved — likely, Enterprise Architects, Platform leads, product managers, infra, and business teams etc.

Who can facilitate using this: Service Lead, Change lead, technical agile coach, tech lead, Service Lead, Product manager. (this is not an exhaustive list, providing people with an understanding of services in use is in the room, the session can be run)

Materials: Print the canvas or use a digital whiteboard (e.g., Miro, MURAL, or Microsoft Whiteboard).

Facilitate a Workshop

Structure the session around the 10 service areas in the canvas.

Suggested Time	Activity
10 min	Intro & Objectives - Confirm layers to use, if unsure leave them all and discuss each.
10 min per layer	1. What do we have that may fall into each of these layers? 2. When existent, who provides it? 3. For whom? Who consumes it? 4. Is it classed as a platform? (maybe by name) or not.
	Discuss amongst attendees' findings- ensure shared understanding. Use following questions as guidance:

	1. Which of what we mapped appears to have the largest number of consumers.? 2. Which of what we mapped appears to have the largest number of dependencies? 3. Do we collect feedback for this service or have an indication of how it is understood/perceived? - Useful? Of value? Priority? Appropriate? Good? Slow? – note down.
--	---

Layers on canvas - definitions

Service layers and definitions below. **Note:** these are the ones added to the canvas and more (you can replace what is there or add, according to your context).

Layer	Service Type
Layer = Experience/Customer-facing services (In the canvas)	Experience Services touchpoints that deliver value directly (internal or external).IE: Customer/employee portals, mobile apps, public website.
Layer = Domain specific – Business capability services (In the canvas)	Services that own a clear business outcome, within a domain. Examples: Payments, Claims.
Layer = Shared services (in the canvas)	Shared by many but not necessarily indispensable to all, this may be classed as optional. Or be offered in different ways. Examples: CI/CD, observability, automation.
Layer = Foundational (in the canvas)	The foundational components/services ‘always present’ that the rest cannot do without, Characteristics: often used by many, indispensable.Examples: compute, networking, baseline security.
Builder (Dev) Enablement Services (not in canvas, can be added)	Things that help builders build and ship – For example, workflows, Orchestrators and tools that speed safe delivery for builders.

...	...
-----	-----

Tips for facilitators & Key considerations

- Encourage cross-functional collaboration, even if maps exist, this is a fresh view with a different lens.
- Not all service types may be relevant, adapt canvas to your context.
- Teams or functions behind identified services may be external, note them down too.

Next

Once this exercise has been completed, you may want to look for areas of opportunity for improvement. For that, you can use the *Platform Flow Quick Evaluation*, which is also shared separately.