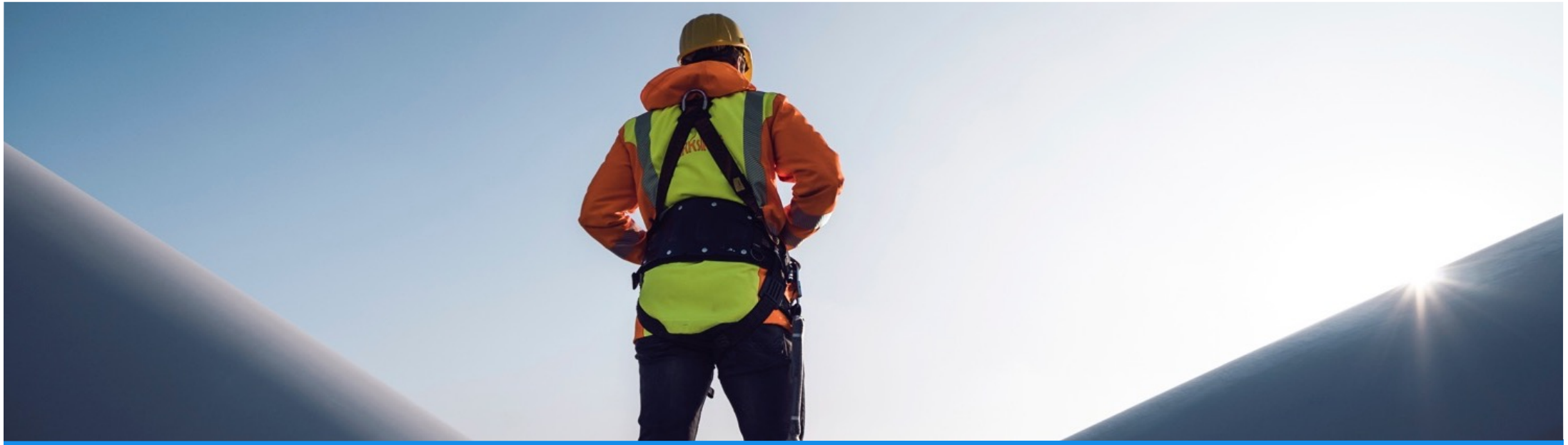


Helping a platform team focused on engineering to switch to a product-based approach

Stéphane Di Cesare, Platform Engineer & Architect

Enterprise Tech Leadership Summit 2024

DKB
Das kann Bank

What is DKB

Not all banks are the same – and many banks don't even know what they stand for. But we do.

We invest in the things that matter: renewable energies, affordable housing, day-care centres, schools, hospitals. We support civic participation and are partners of the domestic agricultural sector.

Founded in 1990
Wholly owned subsidiary of BayernLB

5,3 million customers
Local authorities, companies, retail clients

EUR 121 billion balance sheet total
Among the top 20 banks in Germany

5000 employees

EUR 89.3 billion credit volume

Wir sind #geldverbesserer

100 % #geldverbesserer

Speaker presentation

Stéphane Di Cesare

DKB Standard Operations Platform, Platform Experience

Started at DKB in April 2023

Background in technology consulting, infrastructure automation, sales engineering, technical support, QA and software integration

Aspiring platform-as-product specialist

Member of CNCF Working Group Platforms within TAG App Delivery

Interested in linguistics and languages

DKB
Das kann Bank

Where we started

Standard Operations Platform
formed from different infrastructure-related teams

core: container platform using Crossplane

Vision: standard platform, used as product

"focus on Engineering"
- workloads run!
- communication has Engineering focus

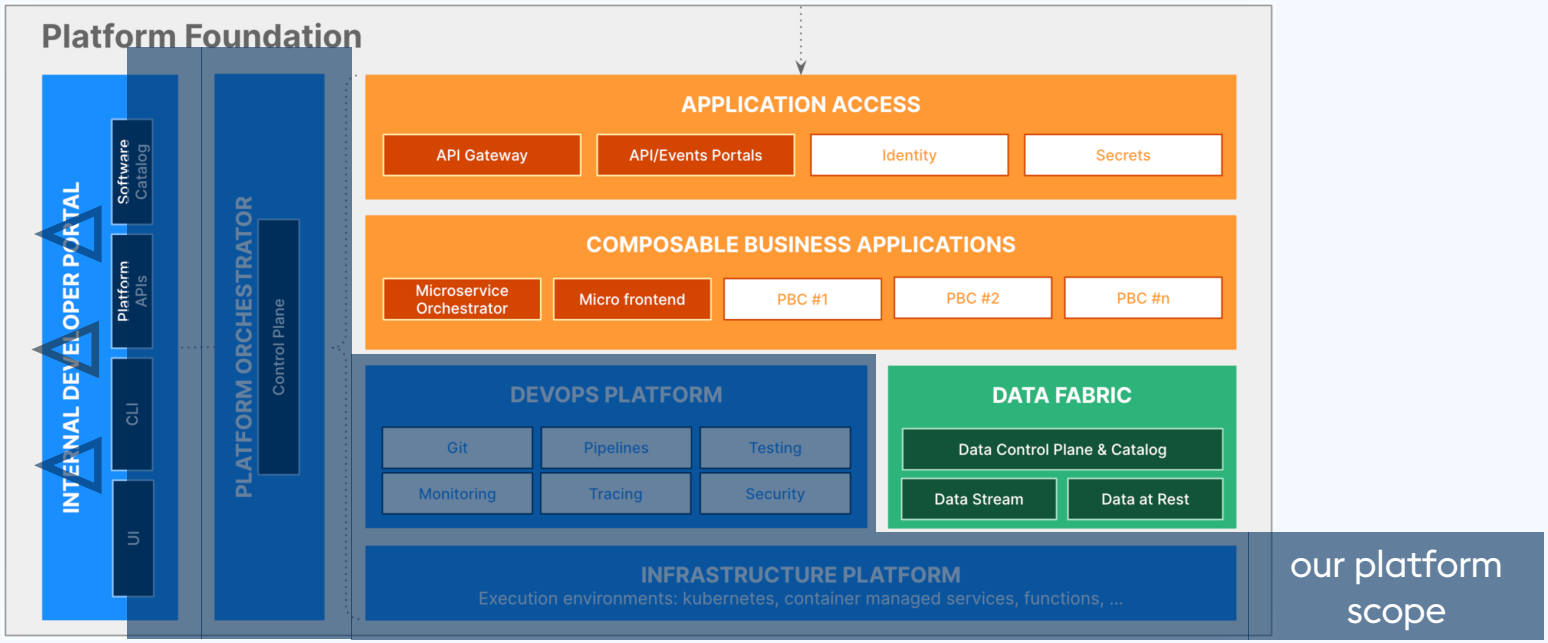


diagram courtesy of Giulio Roggero, mia Platform

Challenges and goals

How can we show the value of the platform?

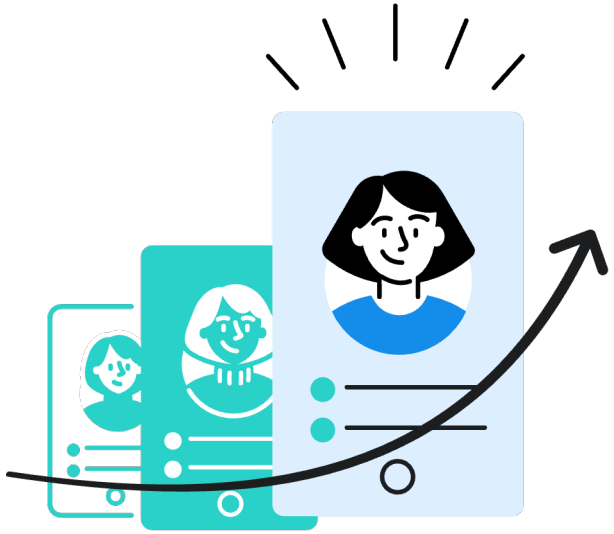
How can we improve the platform?

Decrease cognitive load for product teams

- Make complexity transparent (infra / security / compliance etc) with the platform

Provide standard processes and tools

Improve interface with users and the business



DKB
Das kann Bank

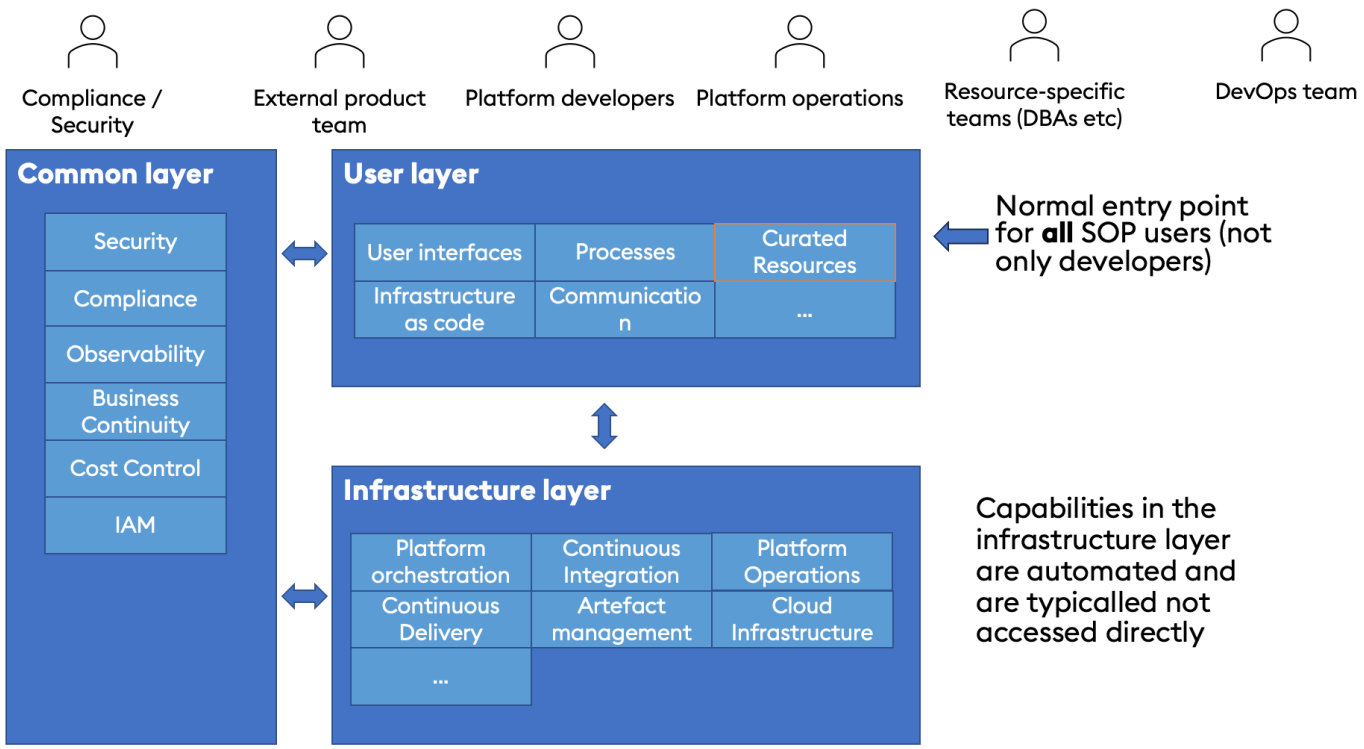
Important topics in our platform journey

1 Defining the platform
2 Clarifying platform maturity
3 Focus on product
4 Communication
5 Documentation and information

DKB
Das kann Bank

Defining the platform

Introduction of a platform architecture



other reference platform architectures:
Humanitec: https://humanitec.com/reference-architectures
Gartner: https://www.gartner.com/en/articles/what-is-platform-engineering
(see also a variation: https://www.syntasso.io/post/platform-engineering-orchestrating-applications-platforms-and-infrastructure)

DKB
Das kann Bank

Defining the platform

How to detail capabilities

Use cases corresponding to the capability are detailed (including actor and value)

Solution

Documentation (implementation and user documentation)

Responsibility

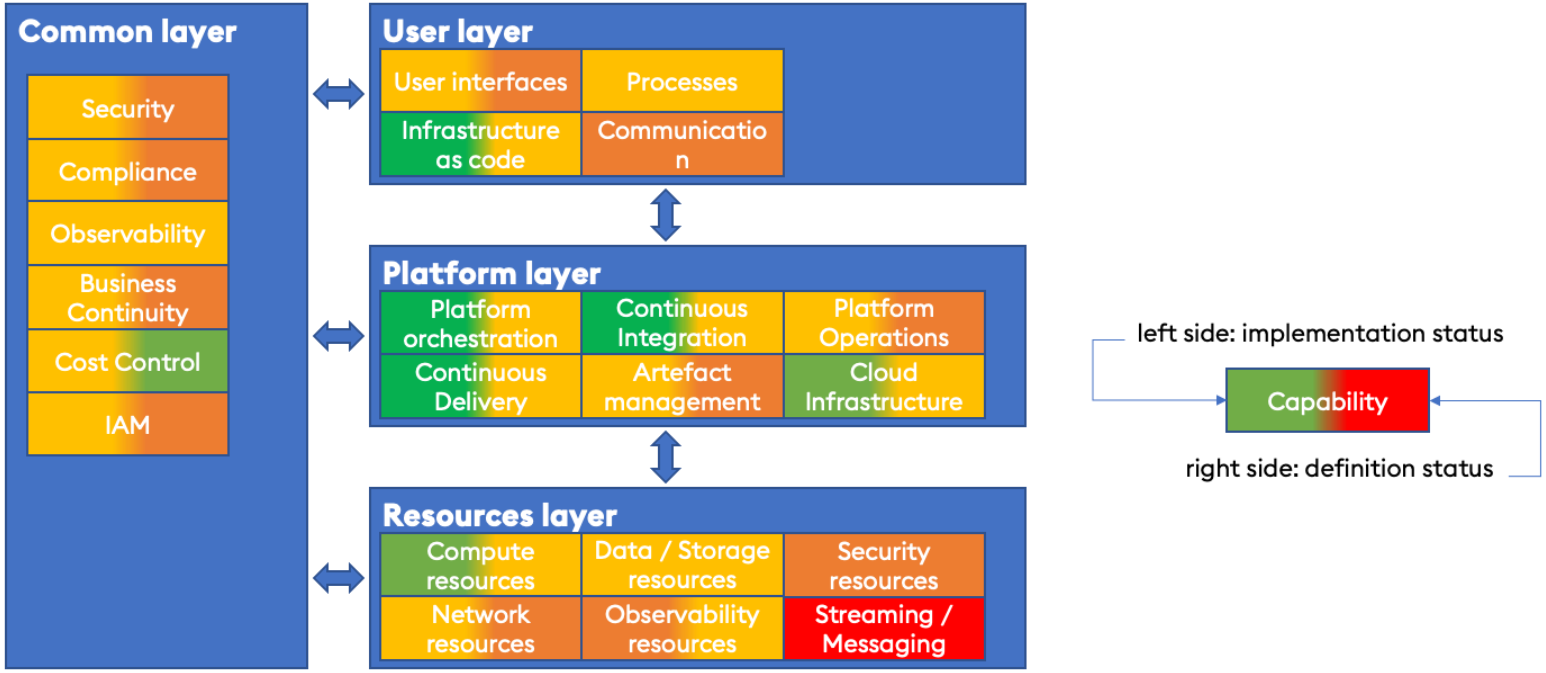
Status (in backlog, beta, productive, ...)

Reporter	Use Case (action + actor)	Value	Comments	*Solution*	*Status*	*Documentation (solution + usage)*	*Responsibility*
John Doe	developer wants to view the total cost of resources related to its application	enable cost visibility					
Jane Doe	cloud engineer wants to be able to push a GitOps cluster configuration change with a maximum of one MR requiring approval	being able to act quickly in case of an emergency					

Defining the platform

Platform heatmap

Make the status of definition and implementation visible



DKB
Das kann Bank

Defining the platform

Continuous improvement

The platform definition is evolving and must be kept up to date

Input from user needs / requirements. Developers are not the only users!

Challenges

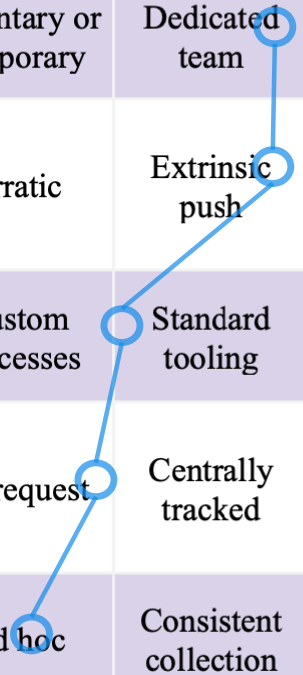
Many people are not used to work at this kind of abstraction level

Updating use cases should be included in the user/stakeholder workflow

Clarifying platform maturity

Where we started from

Aspect		Provisional	Operational	Scalable	Optimizing
Investment	*How are staff and funds allocated to platform capabilities?*	Voluntary or temporary	Dedicated team	As product	Enabled ecosystem
Adoption	*Why and how do users discover and use internal platforms and platform capabilities?*	Erratic	Extrinsic push	Intrinsic pull	Participatory
Interfaces	*How do users interact with and consume platform capabilities?*	Custom processes	Standard tooling	Self-service solutions	Integrated services
Operations	*How are platforms and their capabilities planned, prioritized, developed and maintained?*	By request	Centrally tracked	Centrally enabled	Managed services
Measurement	*What is the process for gathering and incorporating feedback and learning?*	Ad hoc	Consistent collection	Insights	Quantitative and qualitative



standalone platform team
intention to work as product
value not concretely expressed

leadership committed to platform
adoption not always related to value

standard tooling and self-service for many areas
documentation not centralized

no central view of capabilities

measurement tooling available
no central agreement on measurement

from CNCF Platform Maturity Model https://tag-app-delivery.cncf.io/whitepapers/platform-eng-maturity-model/

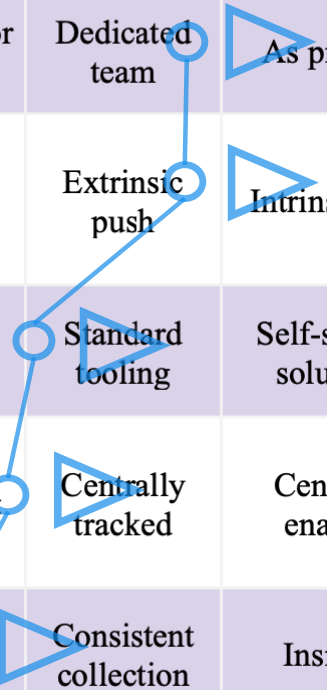
Nicki Watt's talk at Platform Engineering Day https://www.youtube.com/watch?v=MiYn60VWtJk

DKB
Das kann Bank

Clarifying platform maturity

Where we are going

Aspect		Provisional	Operational	Scalable	Optimizing	
Investment	*How are staff and funds allocated to platform capabilities?*	Voluntary or temporary	Dedicated team	As product	Enabled ecosystem	make value clear
Adoption	*Why and how do users discover and use internal platforms and platform capabilities?*	Erratic	Extrinsic push	Intrinsic pull	Participatory	communicate value
Interfaces	*How do users interact with and consume platform capabilities?*	Custom processes	Standard tooling	Self-service solutions	Integrated services	central view of documentation self-service tooling
Operations	*How are platforms and their capabilities planned, prioritized, developed and maintained?*	By request	Centrally tracked	Centrally enabled	Managed services	central view of capabilities
Measurement	*What is the process for gathering and incorporating feedback and learning?*	Ad hoc	Consistent collection	Insights	Quantitative and qualitative	impact of capabilities is measured



from CNCF Platform Maturity Model https://tag-app-delivery.cncf.io/whitepapers/platform-eng-maturity-model/

Nicki Watt's talk at Platform Engineering Day https://www.youtube.com/watch?v=MiYn60VWtJk

DKB
Das kann Bank

Focus on product

What we did

What does product focus mean?

- common agreement on product view

What is our product? (or what are our products – do we have several?)

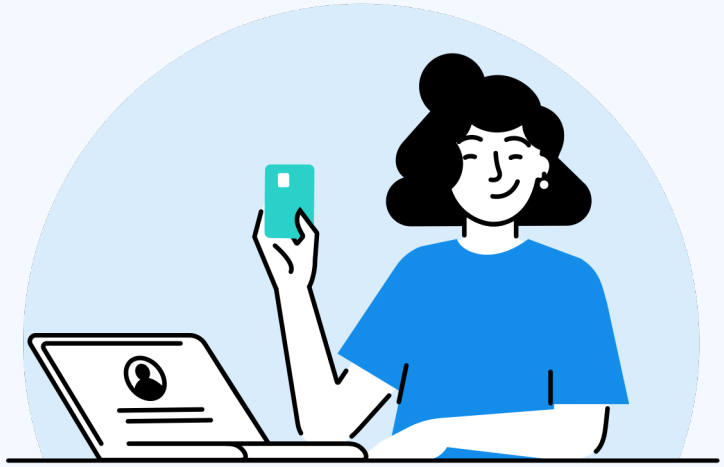
Platform as product – our interpretation

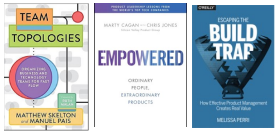
the platform team has the final say on the platform features

the product(s) is/are clearly identifiable for the platform users

product features are streamlined and well defined

product usage is streamlined and well defined

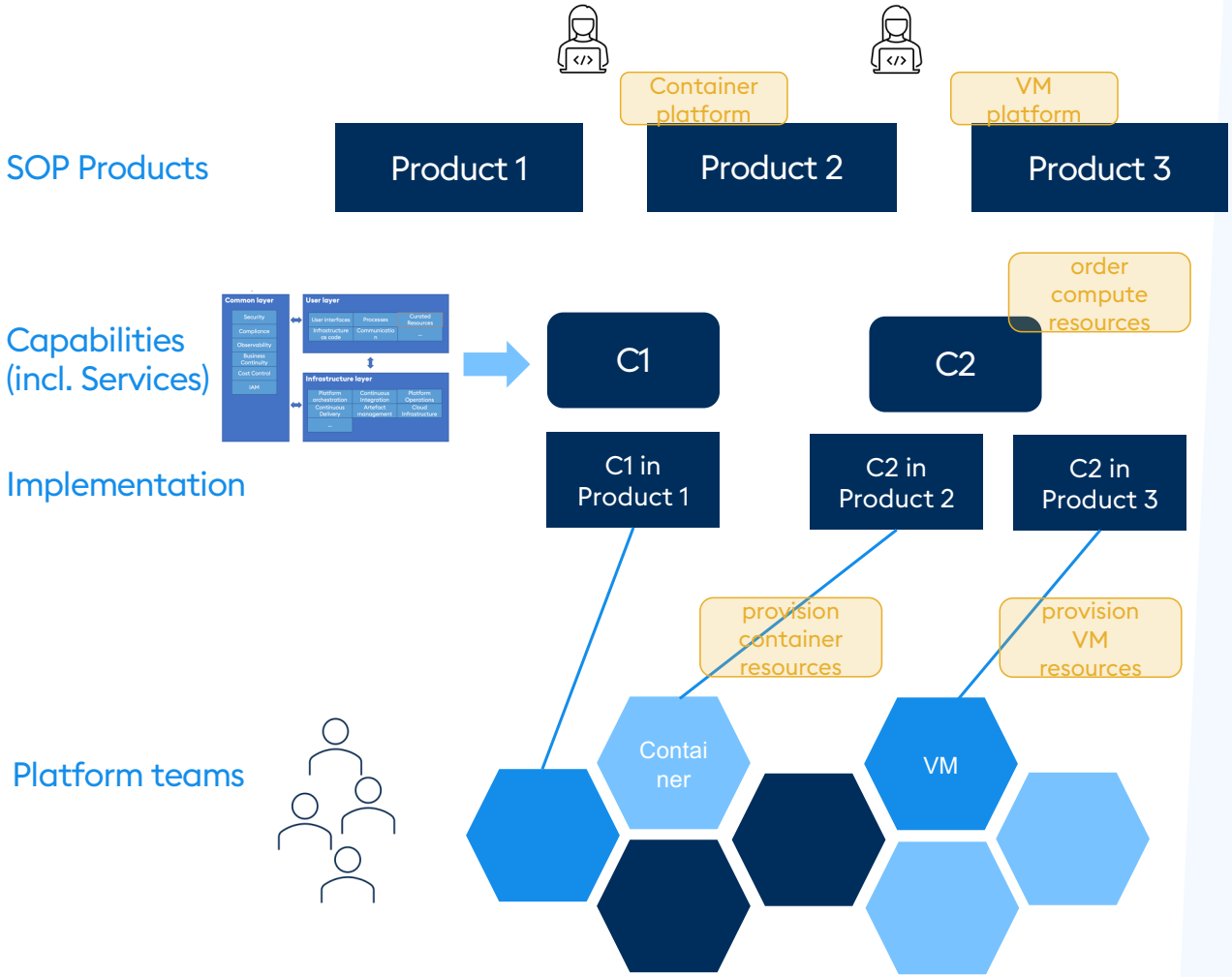




references:
Marty Cagan – Empowered + Transformed
Matthew Skelton & Manuel Pais – Team Topologies
Melissa Perri – Escaping the Build Trap

Focus on product

Organizing platform teams between different products



Value

- users can find information or contact someone without needing internal SOP knowledge
- we can take decisions faster at the product or at the implementation level
- we can organize ourselves flexibly “under the hood” without user impact

Focus on product

Some challenges

Responsibility for product management activities (delivery / strategy / connection) is disconnected

In practice, Scrum product owner role tends to focus on delivery

"team encapsulation" often leads to teams being disconnected from the business view

external teams try to force the implementation of a specific requirement

try to centralize the responsibility for product management activities (possibly with delegation)

try to involve team members in activities with product view, for example use case workshops

ensure that decision stays at the product team level
external stakeholders must provide a background for their requirements, and cannot force a specific implementation

Communication

Common language

Concepts should be defined in the context of the platform (especially overloaded concepts like “service”, “product”, “platform”, “architecture”, ...)

Agreement is more important than “getting it right”

One way to avoid confusion: name the context when the concept is used (“platform service” instead of “service”)

Communication

Channels

Communication is a feature of the platform

Communication should be streamlined as much as possible (but cover all use cases)

Communication is not only incidents or change requests!

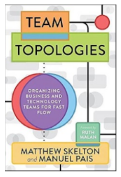
Some communication use cases

quick questions without guarantee of answer

“platform consulting”

communication from the platform (outages, new features, ...)

emergency communication (critical incident, security emergency etc)



references:
Sumeet Gayathri Moghe – The Async-First Playbook
Matthew Skelton & Manuel Pais – Team Topologies

Documentation and information

Documentation is part of the product, with different persona and use cases

Different persona might need different media

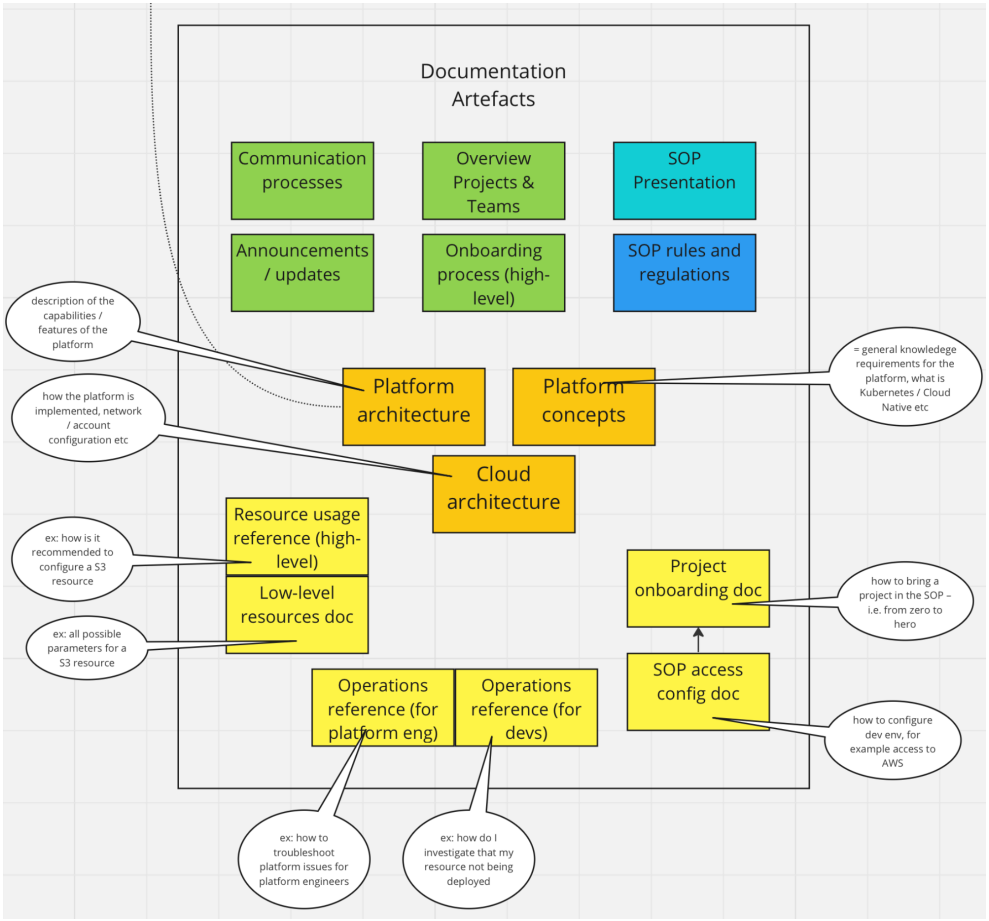
Avoid shadow documentation: focus on keeping the overview

Patterns and anti-patterns

anti-pattern: “it’s obvious”

anti-pattern: “you should already know this”

psychological safety is important!



DKB
Das kann Bank

The future...

... and help we are looking for

Give us feedback about this presentation!

If you also are a platform builder, exchange best practices/information

What are best practices to document the platform scope?

How can we improve participation of all stakeholders in architecture and documentation?

How can we help the team to keep a product mindset?

I am happy to answer questions on the conference Slack: https://devopsenterprise.slack.com channel #discussions

Contact me on LinkedIn: https://linkedin.com/in/sdicesare

We are looking for cloud, network and database specialists! https://jobs.dkb.de

Software Architect: **„Thanks to the SOP components, state of the art infrastructure is available to us without major implementation work“**

Controlling: **„SOP helps us mapping infrastructure costs to actual product teams“**

Application Product Owner: **„Knowledge about application-specific configuration artefacts is now stored under version control, and therefore more visible for the product team“**