



Platform Engineering XXL

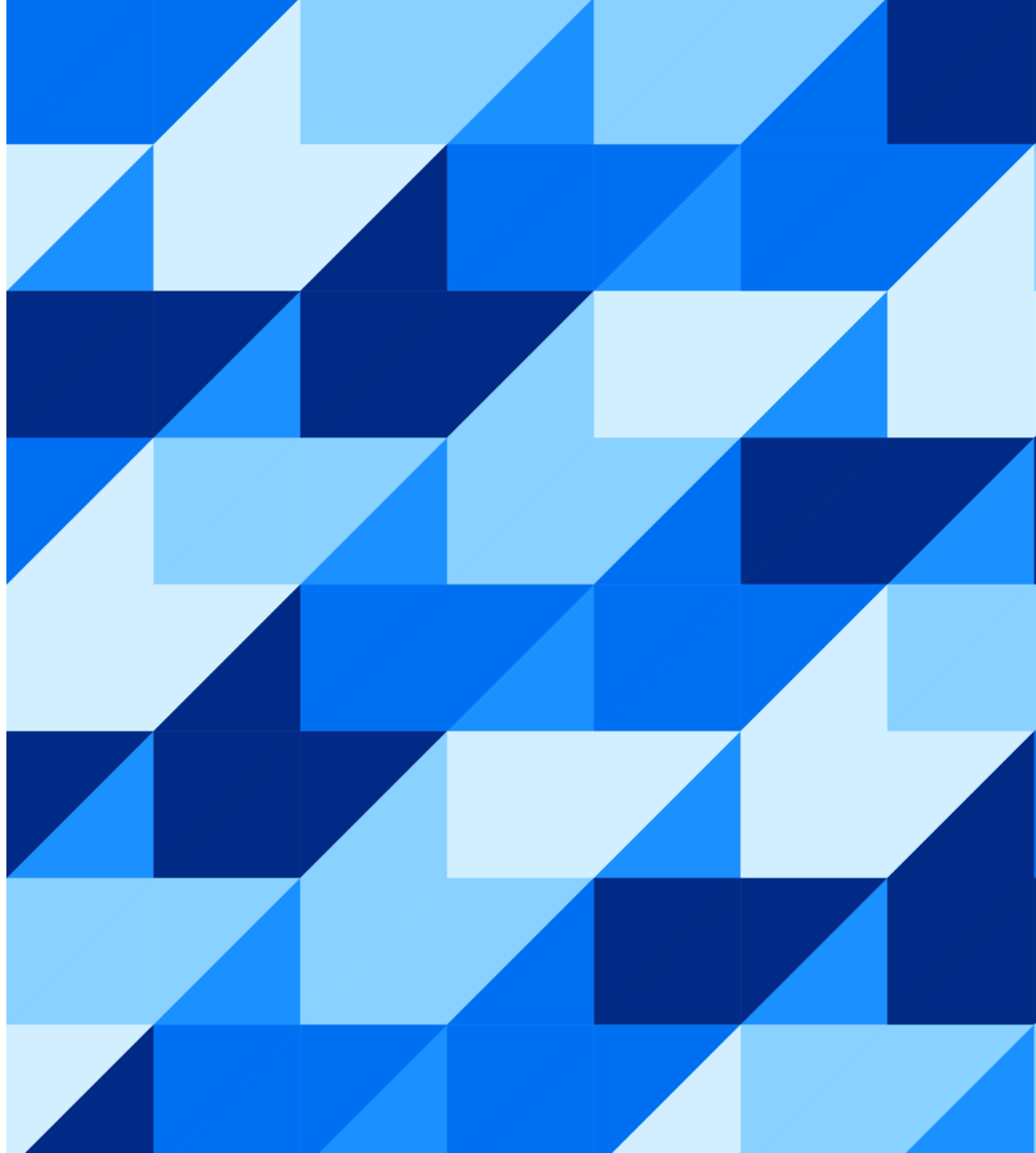
oder:

**Wie wir eine CI/CD Plattform für
30.000+ Entwickler:innen bauen
und betreiben**

Dirk Lehmann (he/him), SAP
November 15, 2023

Public

>> Continuous
Lifecycle >>



About Me



With SAP since 2001



Implemented the first daily delivery process at SAP (2014)



Current job: Solution Manager for internal development platform



SAP Contact person for german language user group (DSAG) for the topic “DevOps”



Co-Organizer DevOpsDays Zurich (16.-17.04. 2024 – Winterthur, CH)
<https://devopsdays.ch>



Conference speaker
<https://doergn.github.io/>



1 Wife, 2 Kids



@doergn@mastodon.social

Public



SAP Facts



Founded 1972



Enterprise software solutions

- Tailored for 26 industries
- Offered in 180 countries (in 30+ languages)



105.328 employees worldwide (30.06.2023)

- 36.100 (34.2%) employees in Research and Development (30.06.2023)



Programming languages

- ABAP (Advanced Business Application Programming)
 - Own proprietary language and technology stack in (mainly) core ERP suite
- Non-ABAP
 - Java ~30%
 - JavaScript ~15%



Delivery channels

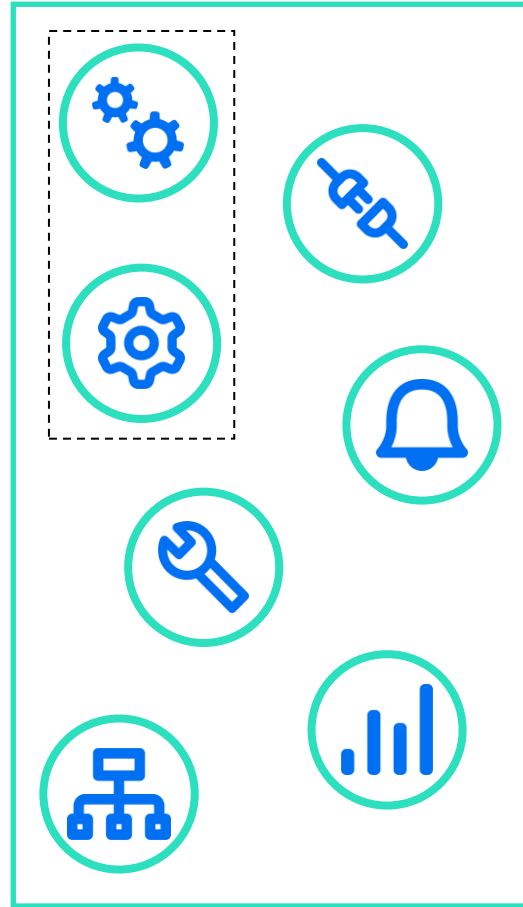
- On premise, Cloud, Hybrid, Mobile
- Multi-Cloud Strategy (Hyperscalers + own 52 Data Centers)

```
1 CLASS zbp_generate_bookings_xxx DEFINITION
2   PUBLIC
3   FINAL
4   CREATE PUBLIC .
5
6   PUBLIC SECTION.
7     INTERFACES if_oo_adt_classrun.
8   PROTECTED SECTION.
9   PRIVATE SECTION.
10  ENDCLASS.
11
12
13 CLASS zbp_generate_bookings_xxx IMPLEMENTATION.
14
15   METHOD if_oo_adt_classrun~main.
16     DATA:it_bookings TYPE TABLE OF ztbooking_xxx.
17
18     *   read current timestamp
19     GET TIME STAMP FIELD DATA(zv_tsl).
20     *   fill internal table (itab)
21     it_bookings = VALUE #(
22       ( booking = '1' customername = 'Buchholm' numberofp
23         country = 'Germany' dateofbooking = '20180213125959'
24       ( booking = '2' customername = 'Jeremias' numberofp
25         country = 'USA' dateofbooking = '20180313125959' da
26     ).
27
28     *   Delete the possible entries in the database table - in c
29     DELETE FROM ztbooking_xxx.
30     *   insert the new table entries
31     INSERT ztbooking_xxx FROM TABLE @it_bookings.
32
33     *   check the result
34     SELECT * FROM ztbooking_xxx INTO TABLE @it_bookings.
35     out->write( sy-dbcnt ).
36     out->write( 'data inserted successfully!' ).
37
38   ENDMETHOD.
39
40  ENDCLASS.
```

The Situation of a Development Team

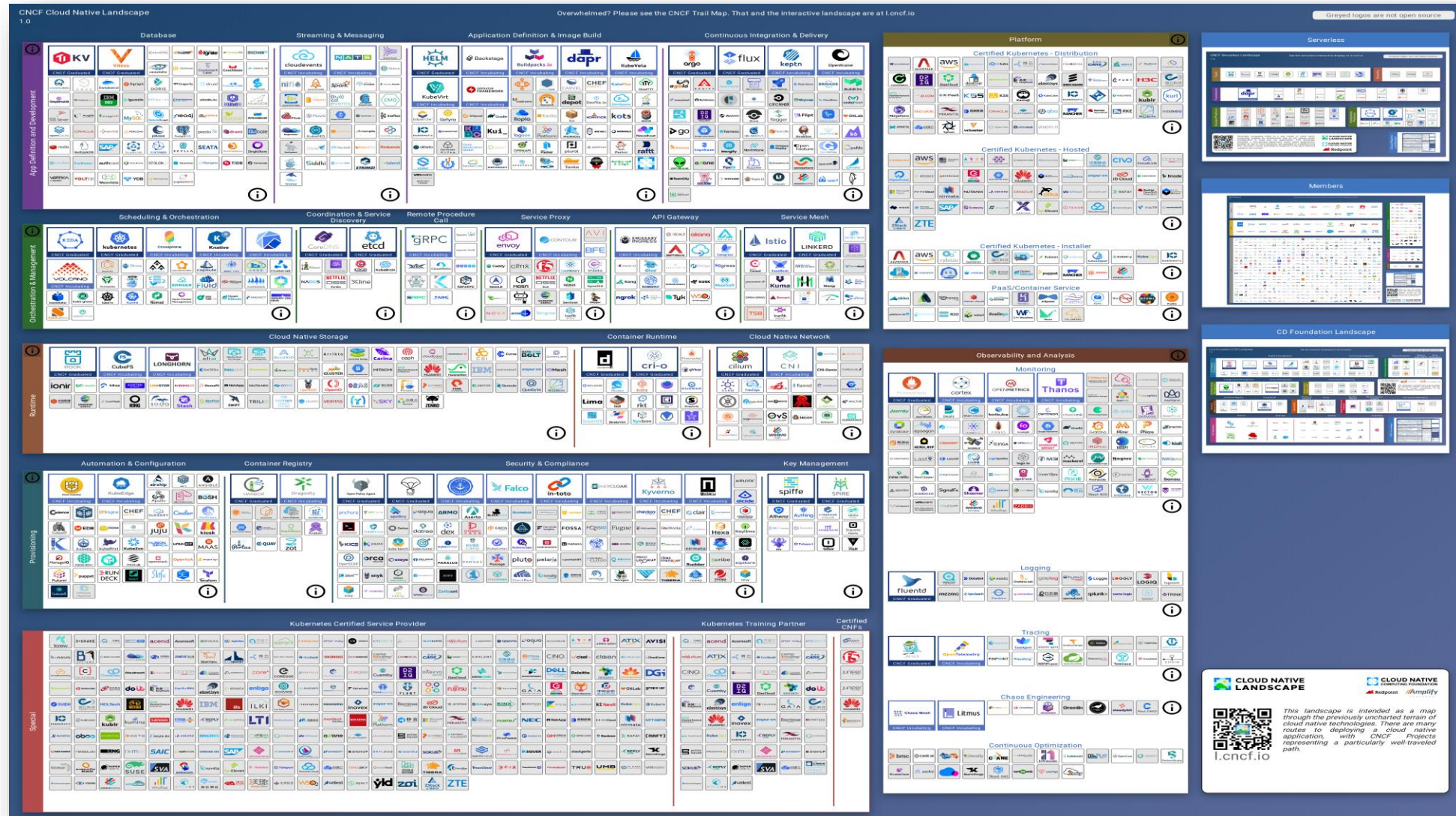


Development
Team



Tools

Normal Tools. Just Innocent Tools.

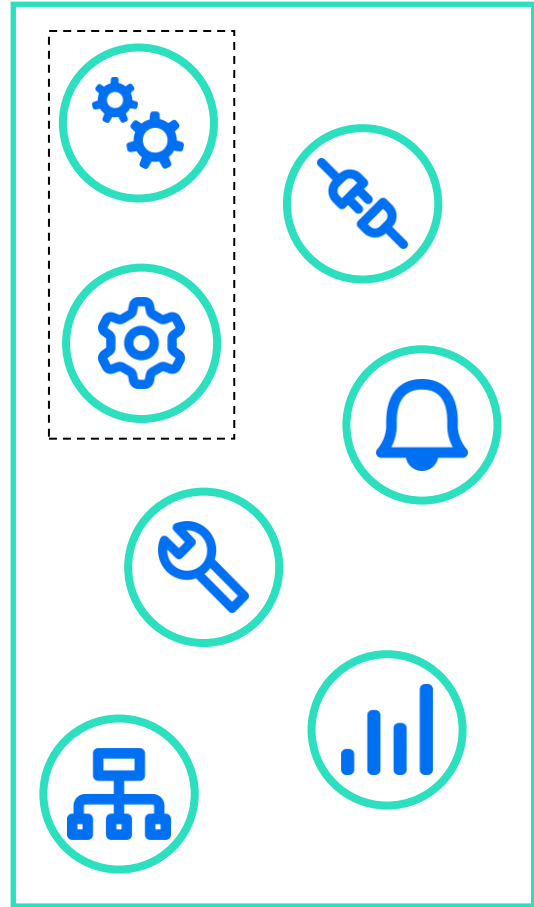


Source: Cloud Native Computing Foundation, [CNCF Landscape](https://l.cncf.io), Oct. 2023

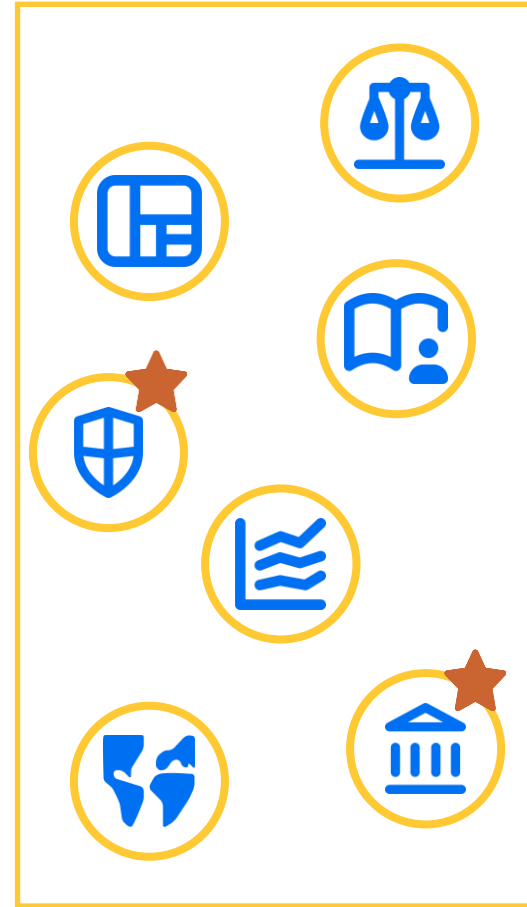
The Situation of a Development Team



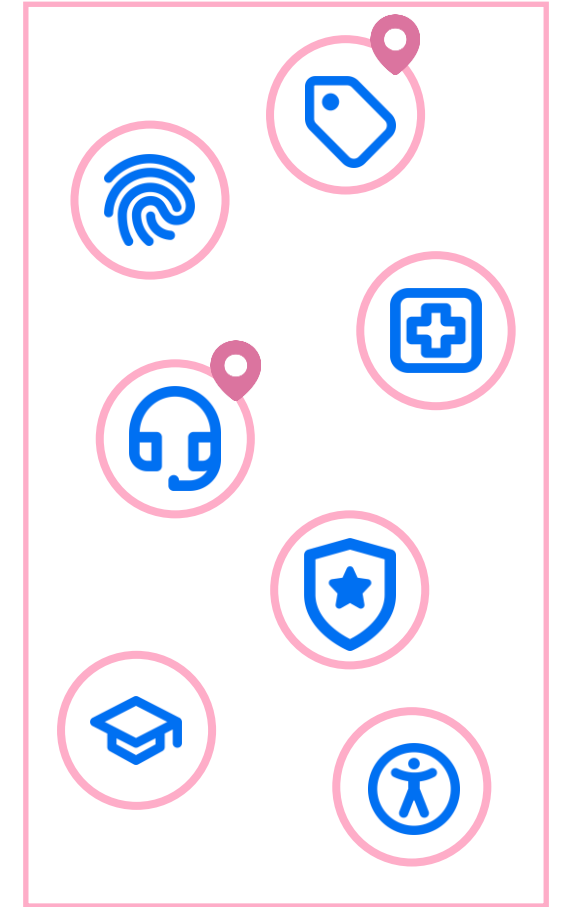
Development
Team



Tools



Processes

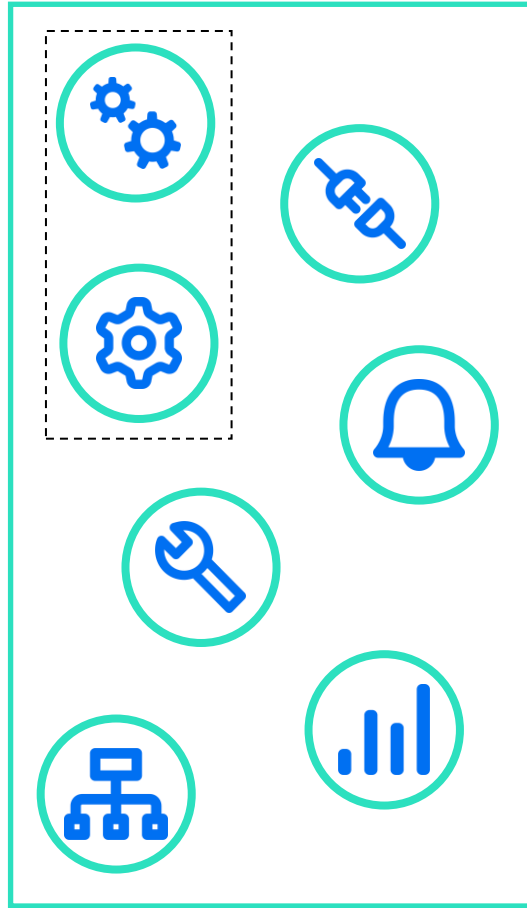


Guidelines

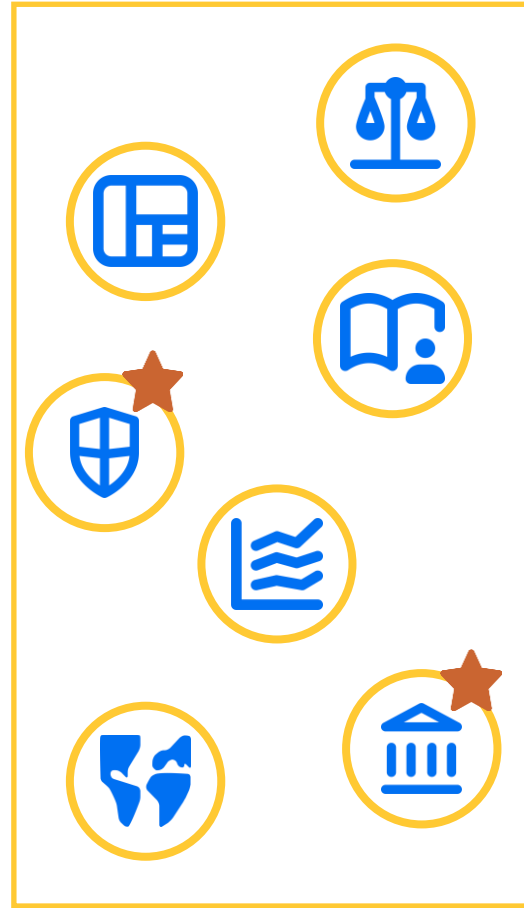
The Situation of a Development Team



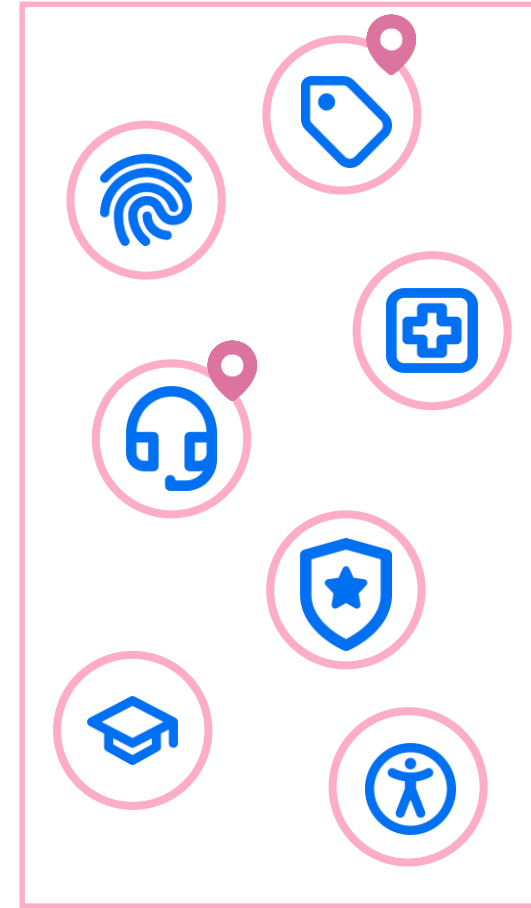
Development Team



Tools

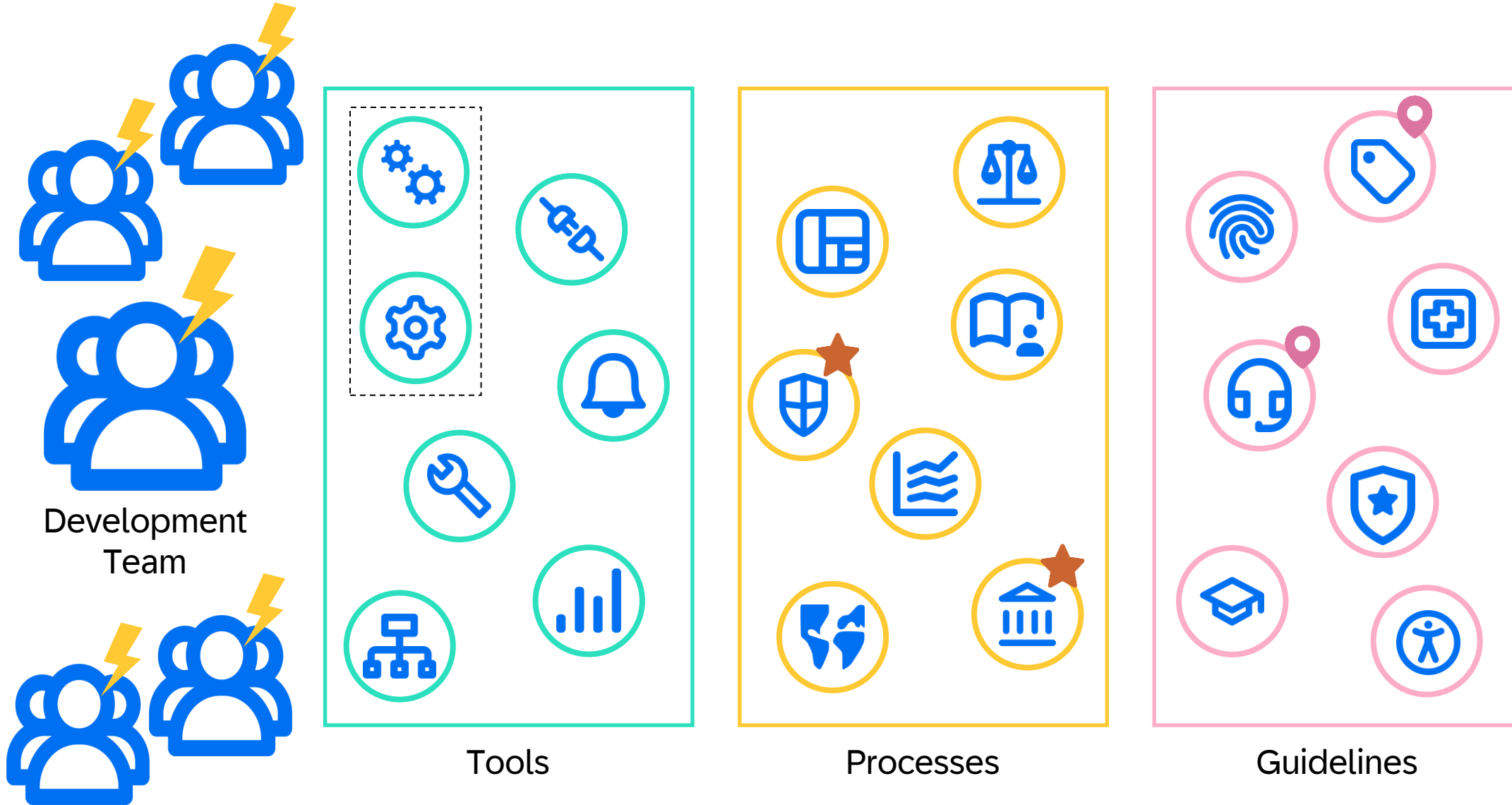


Processes



Guidelines

The Situation of a Development Team



“<...> many organizations find themselves with outdated, poorly governed, and unmanageable toolchains as they scale DevOps initiatives.”

“Developers face challenges and time constraints while maintaining these complex, stand-alone tool siloes, building fragility and technical debt that the [infrastructure and operations] leader has to deal with”

Source: Gartner Research - [“Beware the DevOps Toolchain Debt Collector”](#), April 2022

Platform Engineering



Platform Engineering – A Manifestation of DevOps

Technology Radar

An opinionated guide to today's technology landscape

“The adoption of cloud and DevOps, while increasing the productivity of teams who can now move more quickly with reduced dependency on centralized operations teams and infrastructure, also has constrained teams who lack the skills to self-manage a full application and operations stack. Some organizations have tackled this challenge by creating **platform engineering product teams**. These teams operate an internal platform which **enables delivery teams to self-service deploy and operate systems with reduced lead time and stack complexity**. The emphasis here is on API-driven self-service and supporting tools, with delivery teams still responsible for supporting what they deploy onto the platform.”

- [Thoughtworks Technology Radar 2017](#)

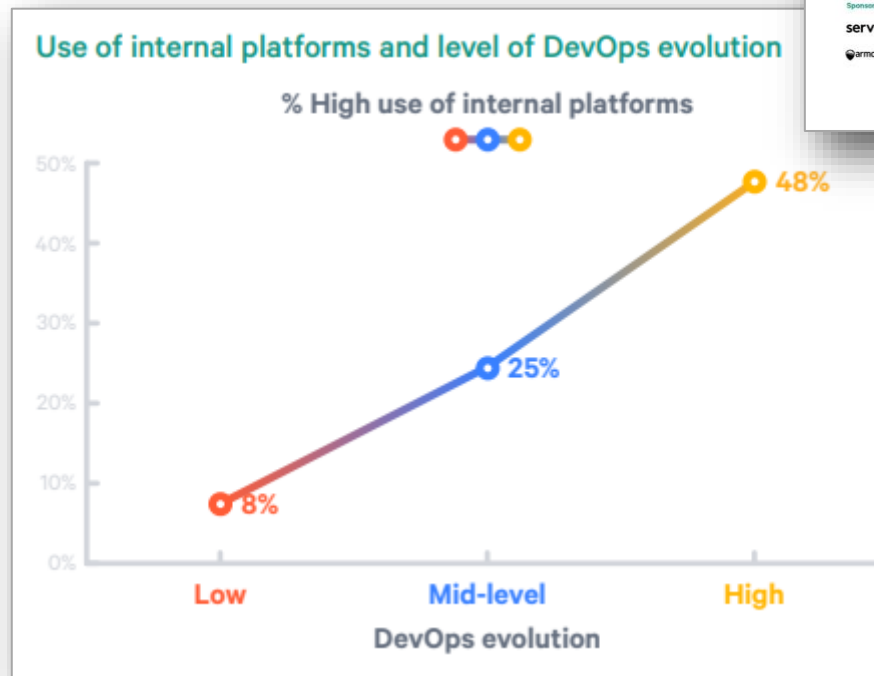
“**A digital platform** is a foundation of **self-service APIs, tools, services, knowledge and support** which are arranged as a **compelling internal product**. Autonomous delivery teams **can make use** of the platform to deliver product features at a higher pace, with reduced co-ordination.”

- [Evan Bottcher, Thoughtworks, 2018](#)

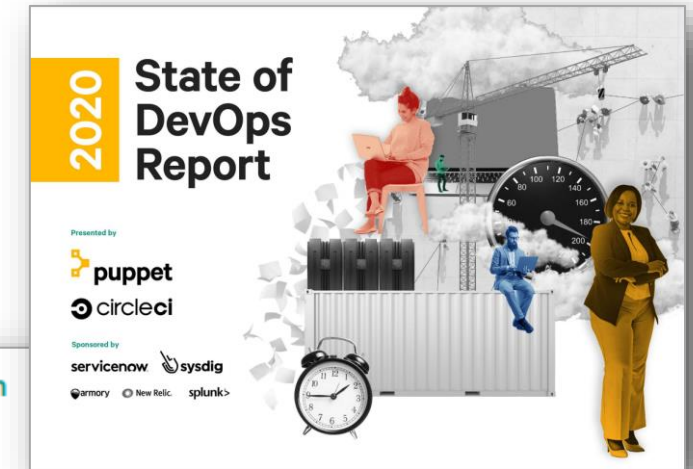
Platform Engineering – Takes Off

“**This year’s DevOps survey has shown** two areas of structural change that can yield excellent results: a platform approach to software delivery and applying DevOps principles to change management. When **organizations successfully establish a platform model for enabling application development**, or significantly improve their change management effectiveness, **they achieve the goal that DevOps initiatives aim at: faster and easier delivery of better quality, more secure software.**”

Source: [Puppet’s State of DevOps report](#), 2020



Source: [Puppet’s State of DevOps report](#), 2020



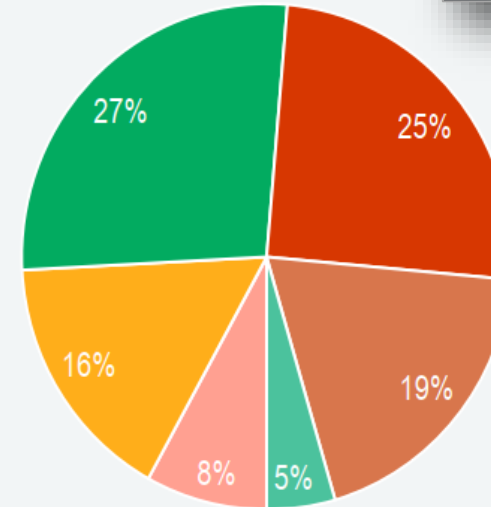
Platform Engineering – State of the Art?

“Organizations that have adopted platform engineering are **experiencing significant improvements across standardization and process**, helping them **to better realize the benefits of DevOps.**”

Source: [Puppet's State of DevOps report](#), 2023

How long has there been a platform team at your organization?

- Less than 1 year
- 1 to 2 years
- 2 to 3 years
- 3 to 5 years
- 5+ years
- Don't know



Source: [Puppet's State of DevOps report](#), 2023



Minimal Viable Platform

Getting Started

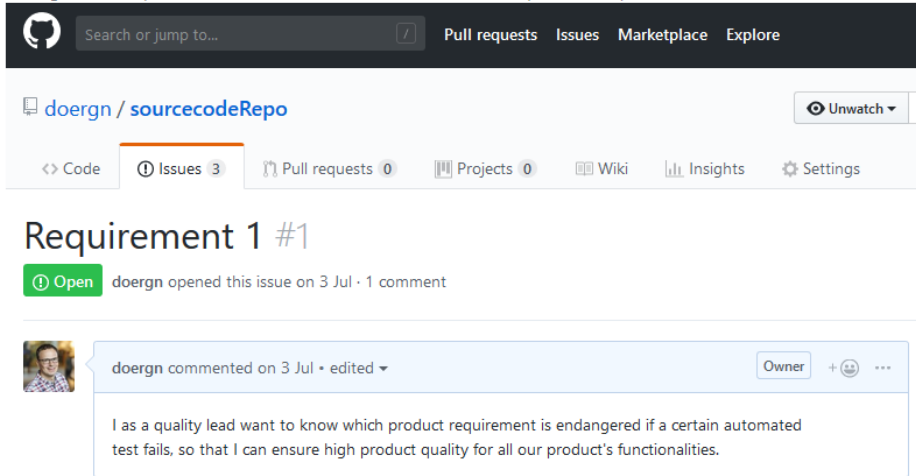
Dirk Lehmann edited this page on Aug 21, 2018 · 39 revisions

Getting started with CTM [↗](#)

On this page you can find some examples, which should give you a quick start into CTM.
If you want to deepen your CTM knowledge, please see our [CTM Guidebook](#) on more configuration options.

Example conditions [↗](#)

- In all given examples below we use [GitHub](#) issues to define our product requirements:



[CTM Guidebook](#) which other tools are supported by CTM

Requirement 1 #1

Open doergn opened this issue on 3 Jul · 1 comment

doergn commented on 3 Jul · edited [↗](#) Owner [+](#) [👤](#) [...](#)

I as a quality lead want to know which product requirement is endangered if a certain automated test fails, so that I can ensure high product quality for all our product's functionalities.

See our

CTM Guidebook

- [Requirements](#)
- [Tracing Methods](#)
 - [2.1 Source Code Annotations](#)
 - [2.2 Requirement Mapping File](#)
- [Types of Test Results](#)
- [Configuration File](#)
- [Delivery File Mapping](#)
- [Command Line Parameters](#)
- [Overall Traceability Report](#)
- [Delivery Traceability Report](#)
- [Traceability Repository](#)

Quick-Navigation [↗](#)

Getting Started

- [1. Requirement mapping in the source code](#)
- [2. Requirement mapping via one dedicated file](#)
- [3. Continuous trace of automated test results per product requirement](#)
- [4. Working with releases](#)

CTM Guidebook

- [1. Requirements](#)
- [2. Tracing Methods](#)
 - [2.1 Source Code Annotations](#)
 - [2.2 Requirement Mapping File](#)
- [3. Types of Test Results](#)
- [4. Configuration File](#)
- [5. Delivery File Mapping](#)
- [6. Command Line Parameters](#)
- [7. Overall Traceability Report](#)
- [8. Delivery Traceability Report](#)
- [9. Traceability Repository](#)

Clone this wiki locally

<https://github.com/SAP/quality-con> [📄](#)

- !** All CTM command line calls are given here as `ctm <arguments>`.
In case you used `go get github.com/SAP/quality-continuous-traceability-monitor` to obtain your CTM instance, please use `quality-continuous-traceability-monitor <arguments>` to execute CTM.
Mac and Linux: You could also create a symbolic link as `ctm` to `quality-continuous-traceability-monitor` to call CTM by it's abbreviation (`ln -s $GOPATH/bin/quality-continuous-traceability-monitor $GOPATH/bin/ctm`).

Internal Developer Platform (IDP)



Internal

Targeting enterprise's internal users only



Self service

Users access capabilities autonomously



User experience

Central Access point to documentation, support, tools, ...



Tool catalog

Comprehensive tools as part of aligned toolchains



Paved Roads (Golden Paths)

Drive standardization and leverage best practices



Compliance

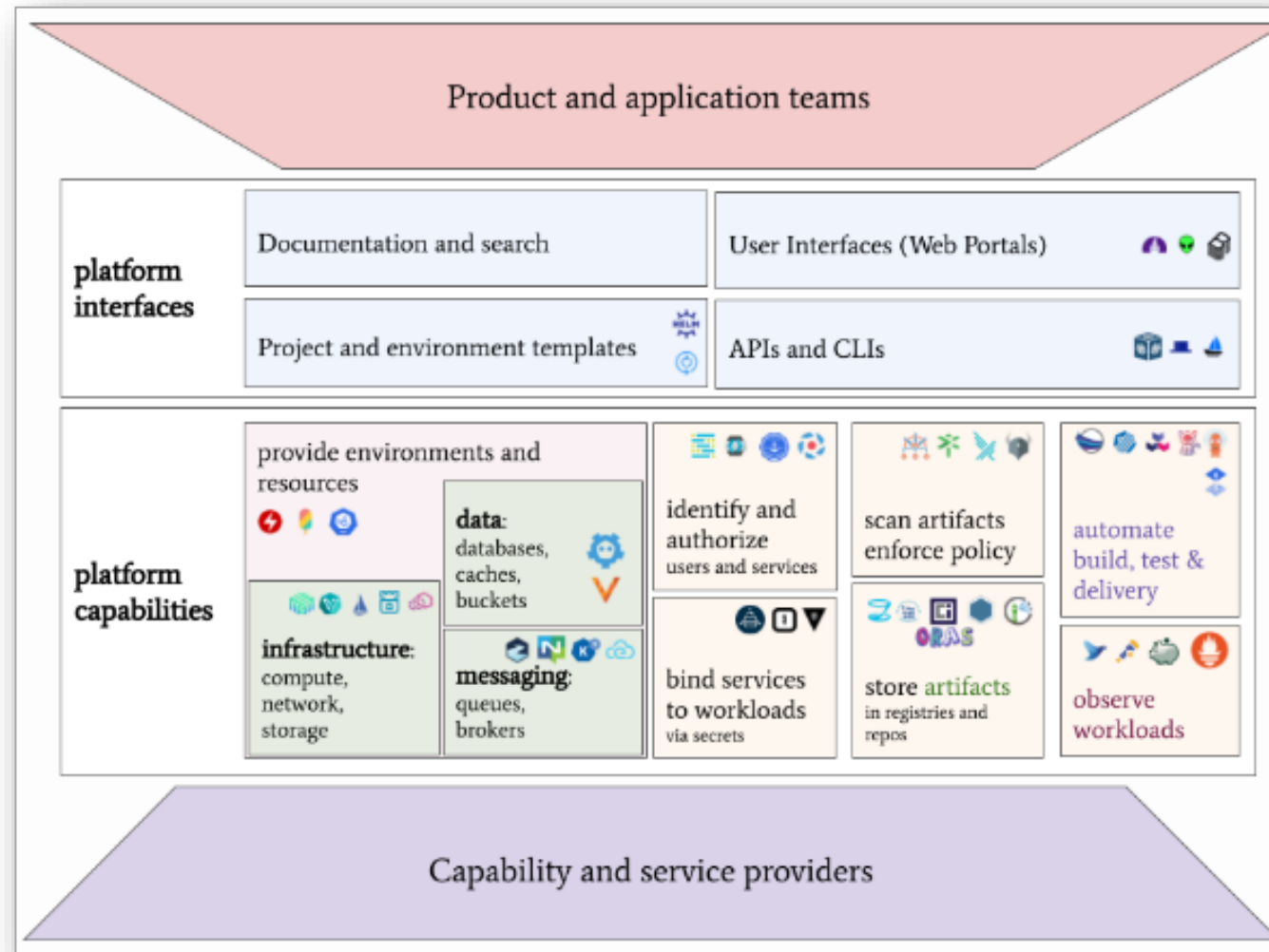
Ensures rules and standards of the organization by default



APIs

Capabilities are composable (and optional)

Platform Capabilities



Source: [Cloud Native Computing Foundation Platforms White Paper](#), March 2023

How to Build a Internal Developer Platform



Treat the platform as a product

- Provides a “Holistic user experience”
- Provides support and great documentation



Strong collaboration with stream-aligned teams

- Designed with users (internal teams) in mind with focus on UX and Developer Experience
- Measure customer satisfaction



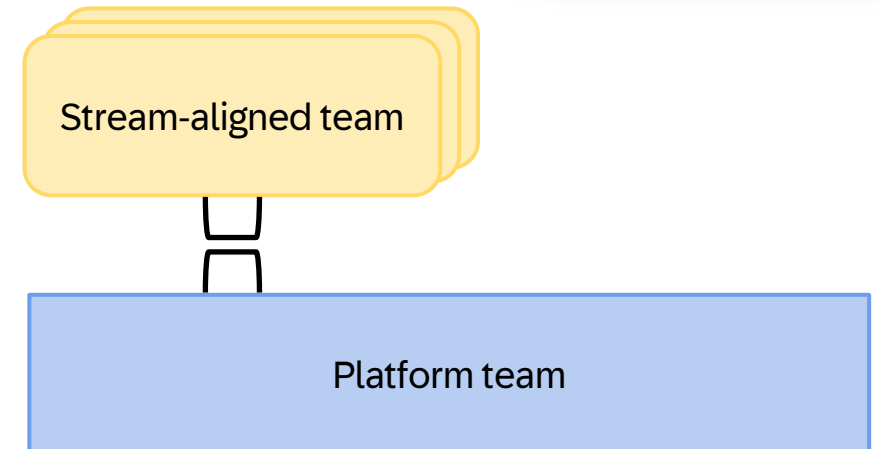
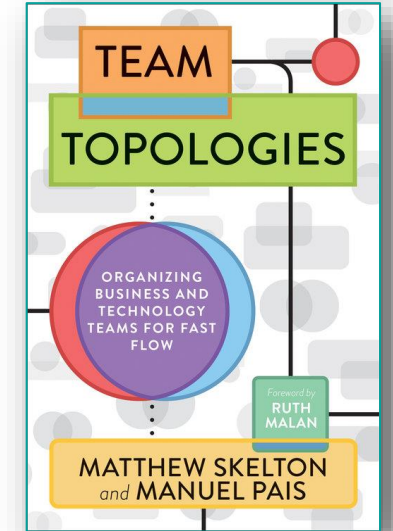
Focus on common uses cases

- Target ~80% of all Stream-aligned teams



Optional to use

- Platform lock-in decreases value
- Platform options increase value



Benefits of Internal Developer Platforms



Emphasize differentiators

Platform capabilities as commodity



Harmonization of environments

Paved Roads provide clear guidance based on best practices



Increased velocity

High degree of automation



Reduce complexity

Integrated capabilities accessible by central access point



Cut costs

Reduce duplication of work



Speed up onboarding

Easy access to various capabilities



Higher developer satisfaction

Reduce team cognitive load



Compliance by default

Corporate rules and governance centrally managed

DevOps, SRE, Platform Engineering – How Does It Blend?

Descriptions:

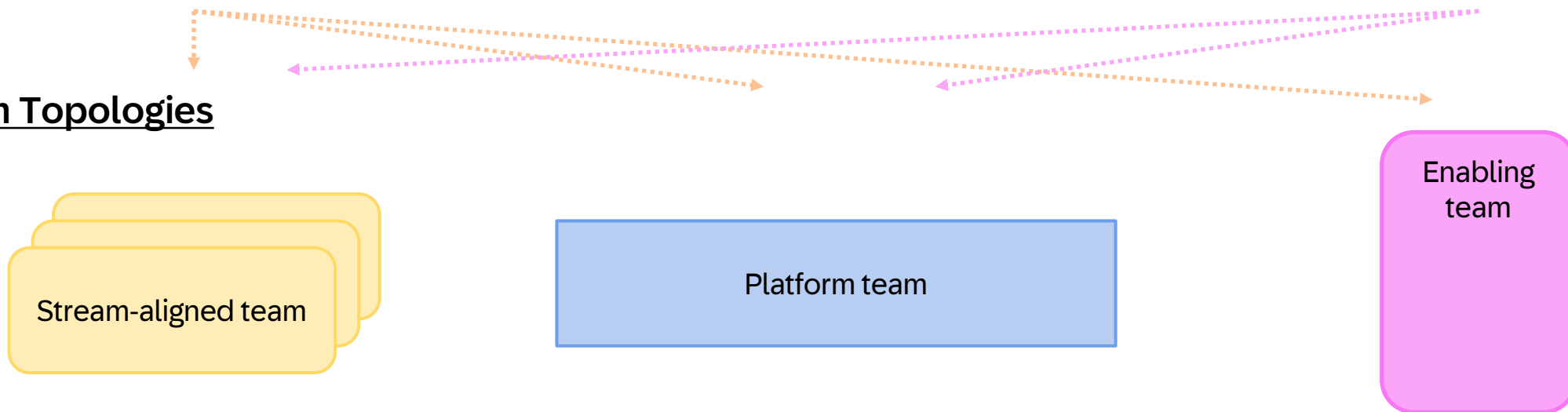
DevOps: A culture where people, regardless of title or background, work together to imagine, develop, deploy and operate a system.

Platform Engineering: Reduce team cognitive load of application teams by eliminating toil and duplicate work.

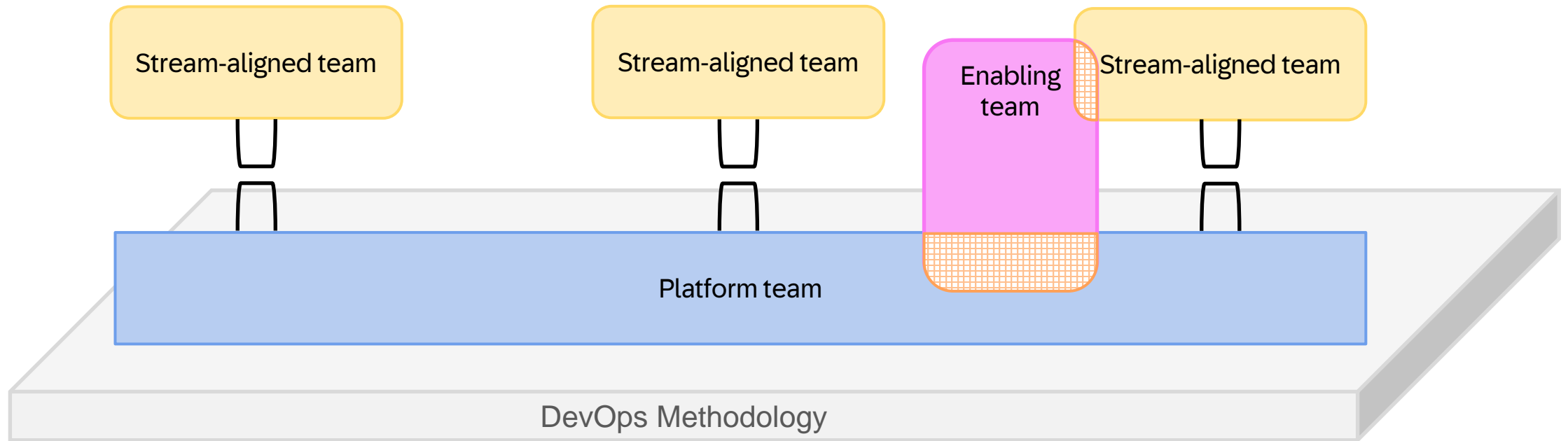
Site Reliability Engineering (SRE): Help teams to increase reliability and scalability of their products.

Ken Mugrage - Thoughtworks

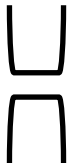
Team Topologies



DevOps, SRE, Platform Engineering – How Does It Blend?



Team Interaction Modes

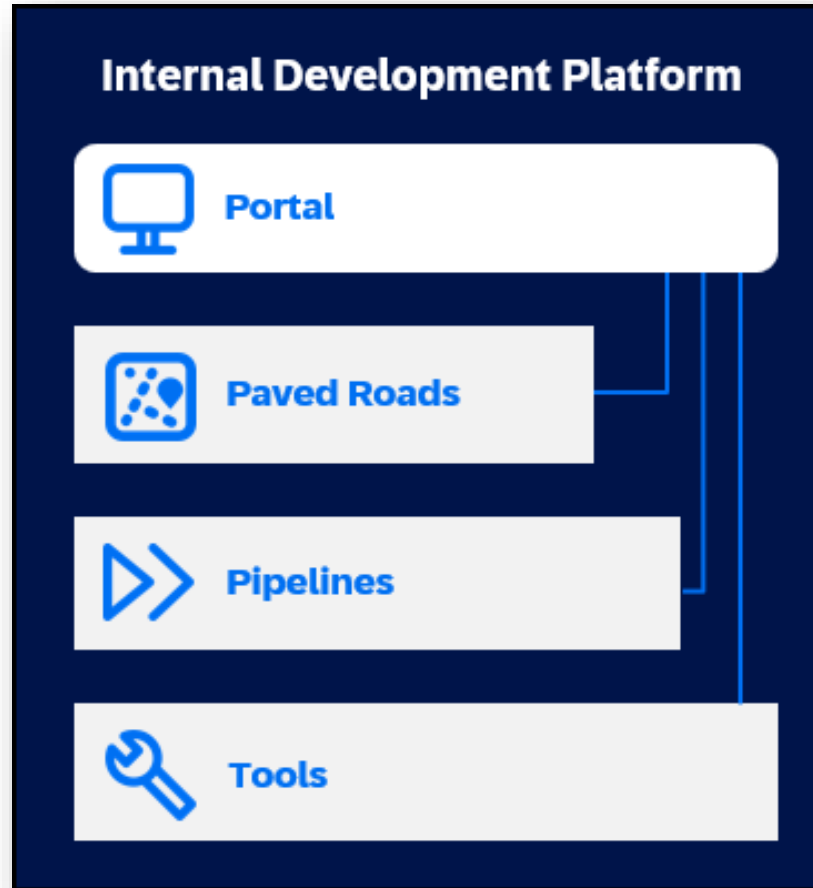
 = X-as-a-Service

 = Facilitating

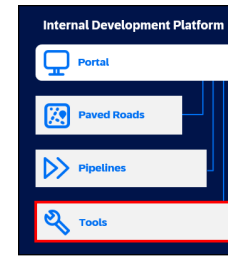
SAP's Internal Development Platform



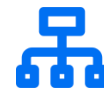
SAP's Internal Development Platform



Tools



Tool catalog
Self-service inventory



Consolidate tools
One organization



Harmonize support channels
One support channel



Integrate tools
Provide open APIs and align data models



Adjust SLAs
Communicate and measure SLAs



User management
One authorization and access management



Improve documentation
Structure documentation and terminology



Services
Replace individual tools with services

Catalog

All



Artemis 

Artemis is a standalone application enabling Scrum Teams to easily set up their SAP Jira project. Via Artemis new SAP Jira projects can get created, either using the SAP default configuration or using a reference project as template. SAP Jira projects can be staffed, using single users, distribution...

Category:
Manage Platform
Services



Deploy with Confidence 

Deploy with Confidence (DwC) is a combination of cloud-native engineering principles, a feature-driven development process, and the right toolset to develop, deploy, and deliver

Portal



Self service

Users access capabilities autonomously



Central access point

Entry point for all platform capabilities



Tool catalog

Self-service inventory



Support

Context enriched support



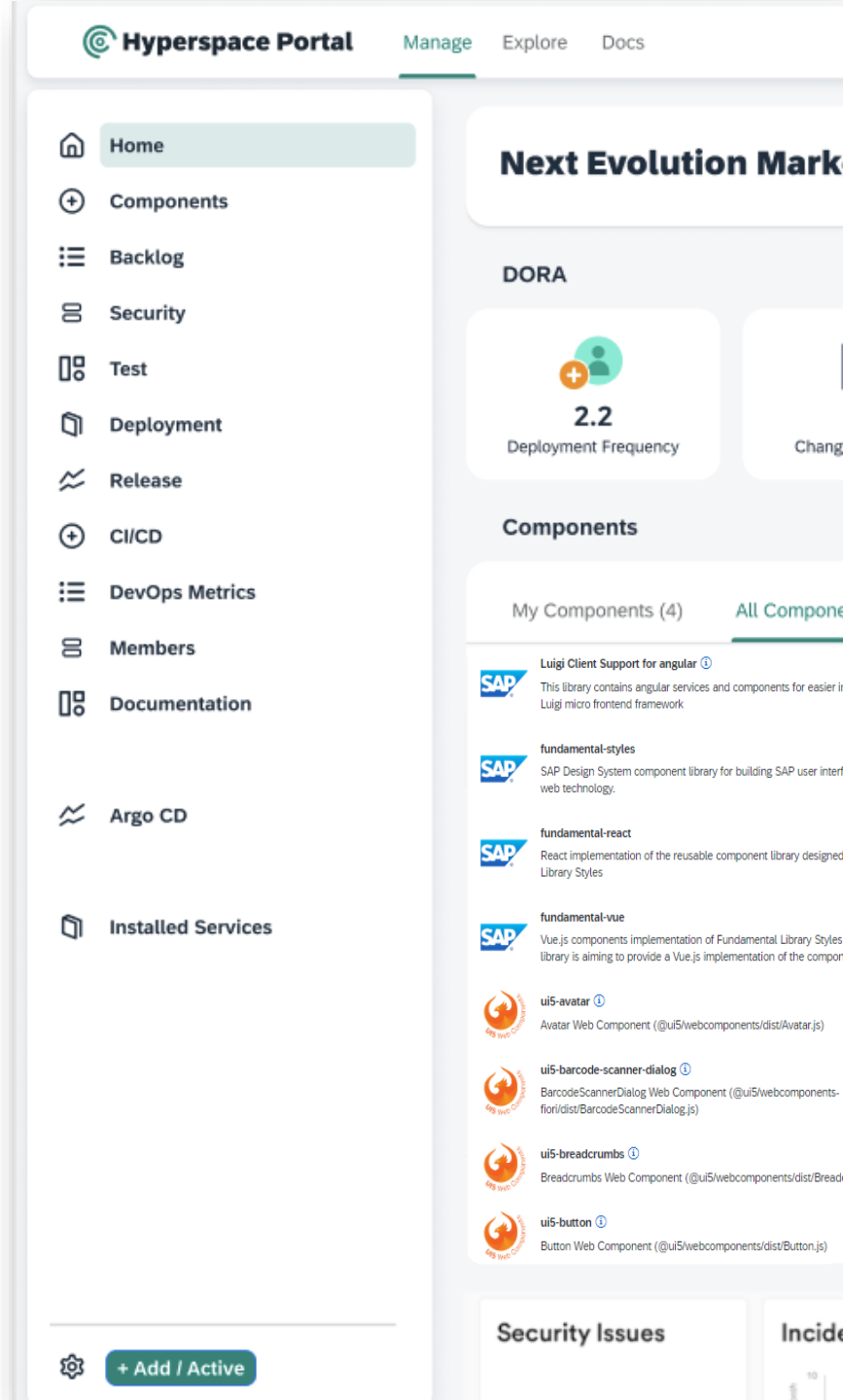
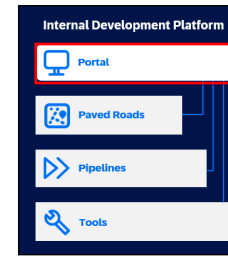
Documentation

Access to documentation

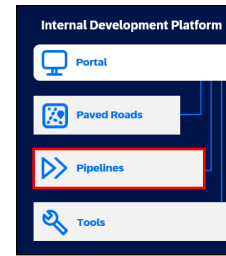


Extensible

Open for contributions



Pipelines



Integrate tools

Provide open APIs and align data models



Grow on demand

Extend pipeline following team needs



Fast set up

Quick access to CI/CD pipelines



Extensible

Open for contributions



Compliance by default

Set up according to corporate rules and governance



APIs

Library for easy tool access

RECOMMENDED

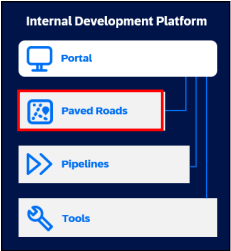


Set up a pipeline

Get a new end to end
Hyperspace pipeline which is
compliant by default.

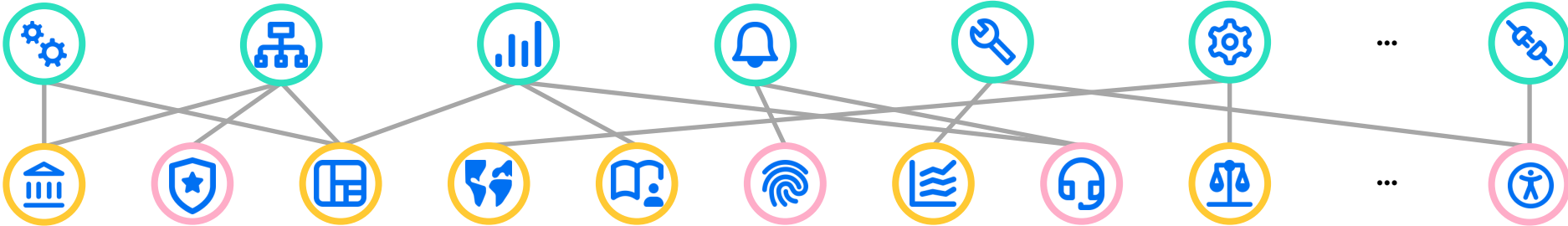
Set up Pipeline

Paved Roads

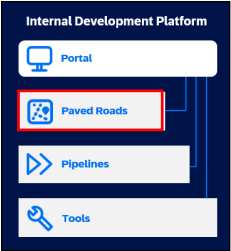


Tools

Processes & Guidelines



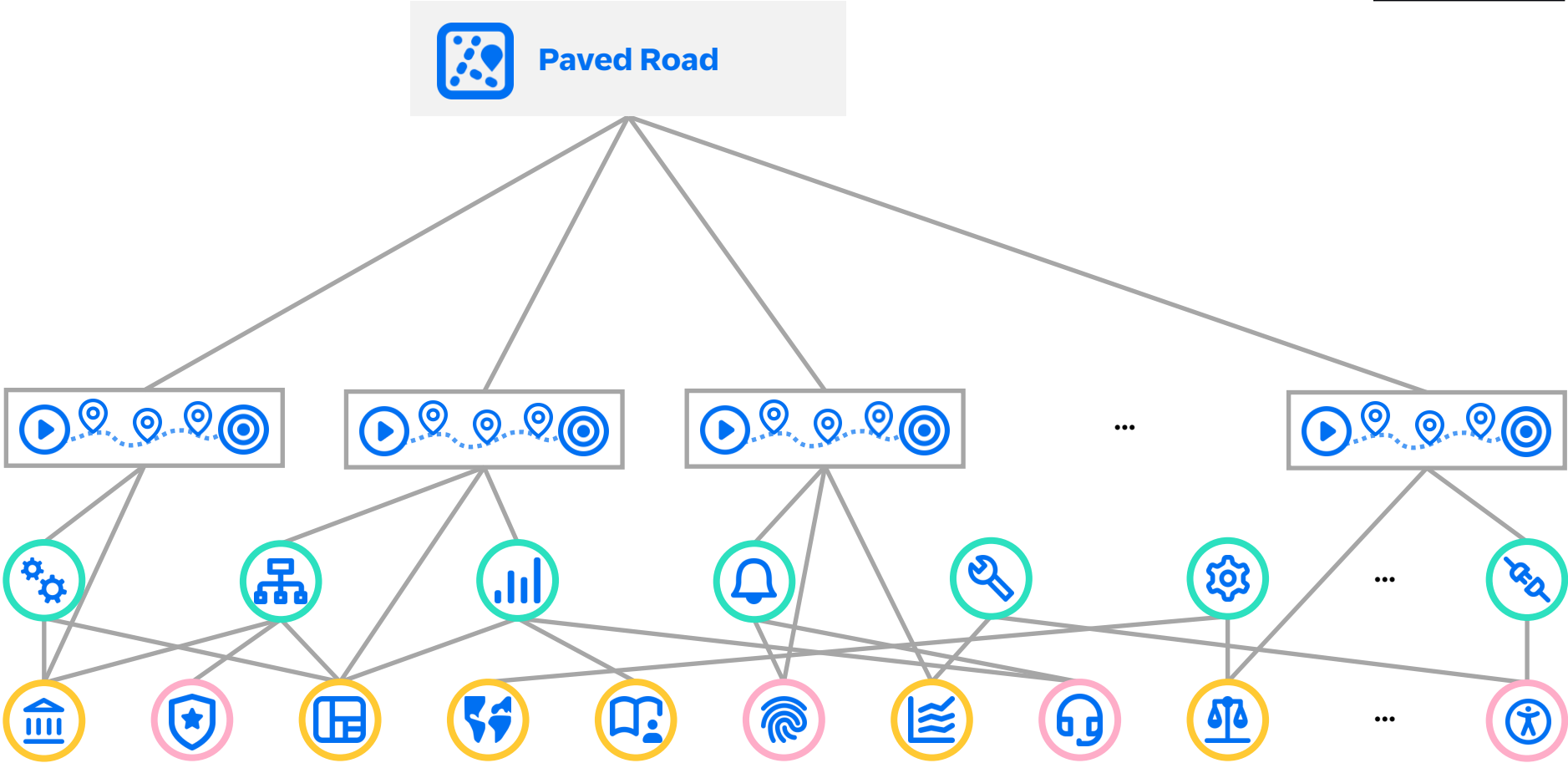
Paved Roads



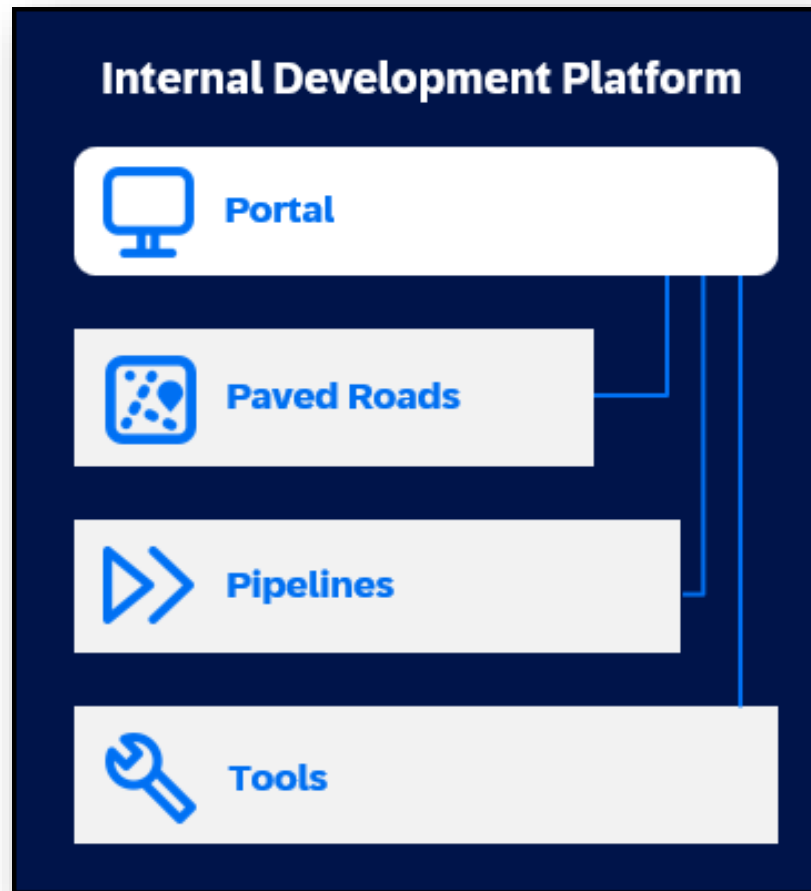
Development
Procedures

Tools

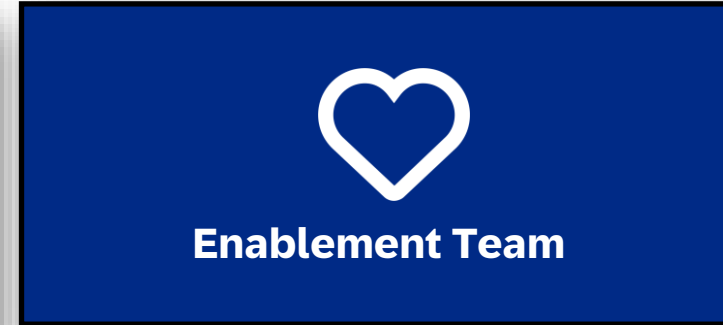
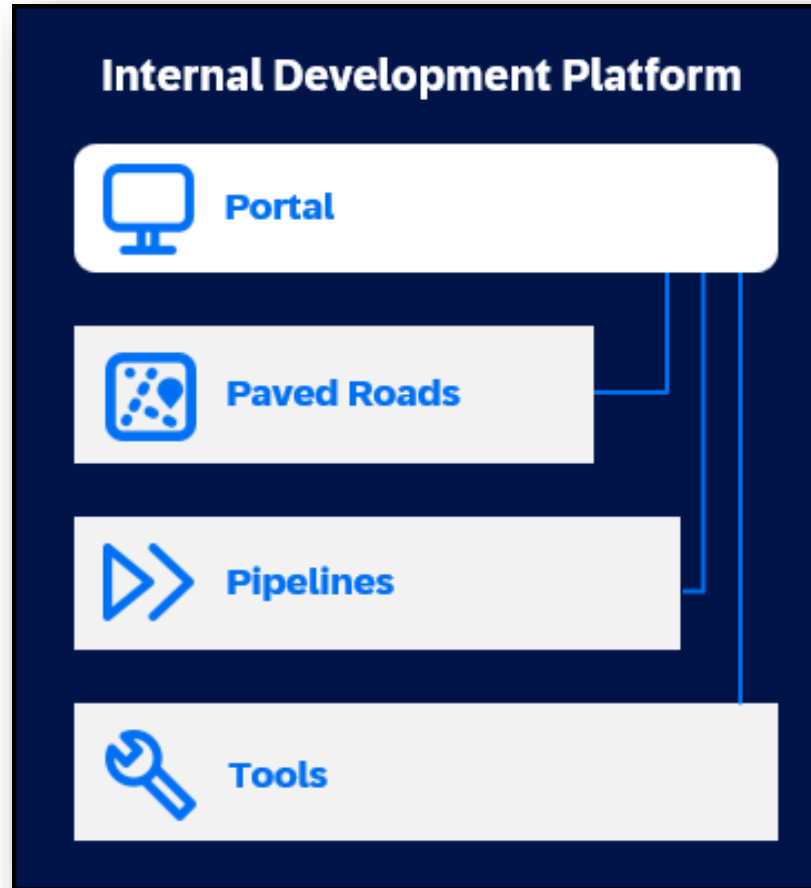
Processes & Guidelines



Development Process Framework



Enablement Team



Criticism on Platform Engineering

Technology Radar

An opinionated guide to today's technology landscape

“The adoption of cloud and DevOps, while increasing the productivity of teams who can now move more quickly with reduced dependency on centralized operations teams and infrastructure, also has constrained teams who lack the skills to self-manage a full application and operations stack. Some organizations have tackled this challenge by creating platform engineering product teams. These teams operate an internal platform which enables delivery teams to self-service deploy and operate systems with reduced lead time and stack complexity. The emphasis here is on API-driven self-service and supporting tools, with delivery teams still responsible for supporting what they deploy onto the platform.

Organizations that consider establishing such a platform team should be very cautious not to accidentally create a separate DevOps team, nor should they simply relabel their existing hosting and operations structure as a platform.”

- [Thoughtworks Technology Radar 2017](#)



- One size (does not) fit all
- Stream aligned teams loose (to much) control
- “Eierlegende Wollmilchsau”

Learnings from SAP's Internal Development Platform (so far)



Put responsibility into one organization

Distributed setting causes lot of (unnecessary) friction



Paved Roads / Golden Paths are your friend

Help to get systemic view



It's a marathon

It takes times to change status quo



KPI reporting is difficult

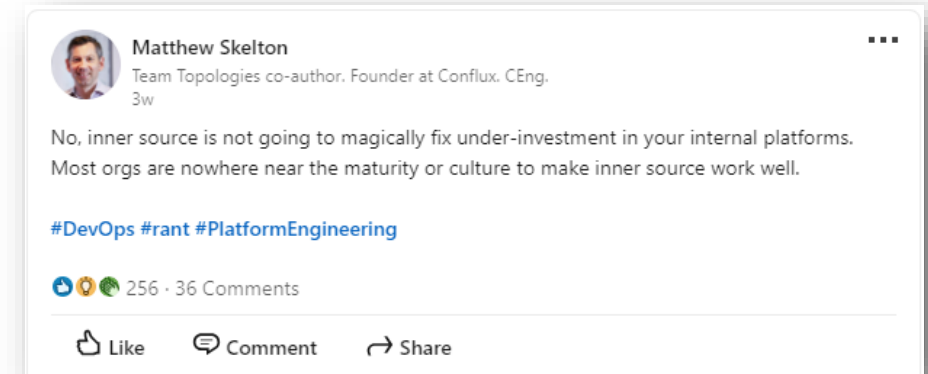
Which KPI indicates platform success?



Requires proper staffing

Dedicated people to work on the platform.

Inner source won't solve the problem



Source: [LinkedIn post by Matthew Skelton](#), Sep. 2023



Trust

Reliability and ability to execute is key

Learnings from SAP's Internal Development Platform (so far)



Share a vision

For customers and
platform team members

Vision of Hyperspace – Engineer POV

Scene 01



Timur gets disrupted by an urgent incident. He was actually finishing his day. Now he has to do a nightshift.

Scene 02



The day after, Timur meets his friend Paul for an after-work drink. Timur has to vent his frustration.

Scene 03



Paul explains what his life looks like since he uses Hyperspace. He can track all his changes and gets notified immediately if something went wrong.

Scene 04



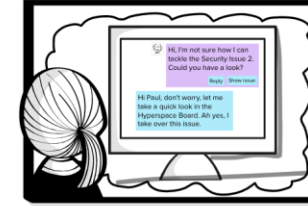
Paul has an overview of all tasks related to this change sorted by relevance. He even gets recommendations how to tackle them.

Scene 05



In case Paul doesn't know how to resolve an issue, he can easily request help from an Expert.

Scene 06



The security expert Ema gets notified. And takes over Paul's issue.

Scene 07



Paul explains how his life changed to the good. Timur gains hope and can't wait to discuss Hyperspace with his teammates.

Thank you.

Contact information:

Dirk Lehmann (he/him)
dirk.lehmann@sap.com

