

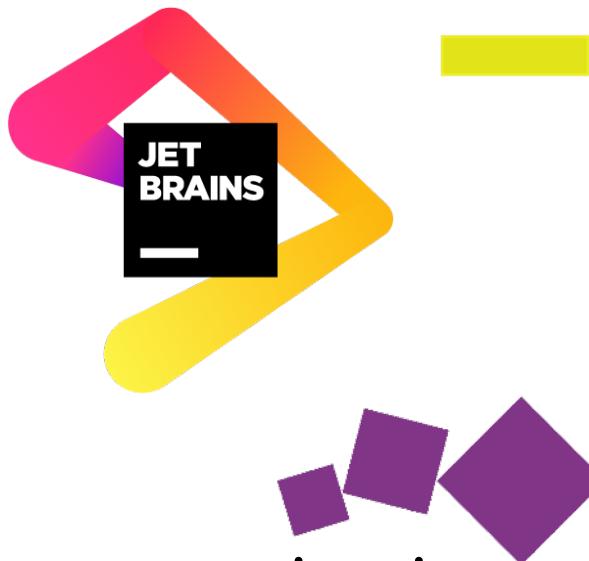


A journey to the Internal Developer Platform





Sponsor & Org



Who I am



Leonardo Romanato

Global Head of DevOps Platform & Services @ Yoox Net-A-Porter



leonardo.romanato@ynap.com



<https://www.linkedin.com/in/lromanato/>



[@F3ar873](https://twitter.com/F3ar873)

When I am not working on the platform...





NET-A-PORTER

MR PORTER

THE OUTNET

YOOX

ONLINE FLAGSHIP STORES



YOOX NET-A-PORTER in numbers

• 180+
COUNTRIES

• 5.3m
HIGH-SPENDING
CUSTOMERS

• >1.2bn
VISITS

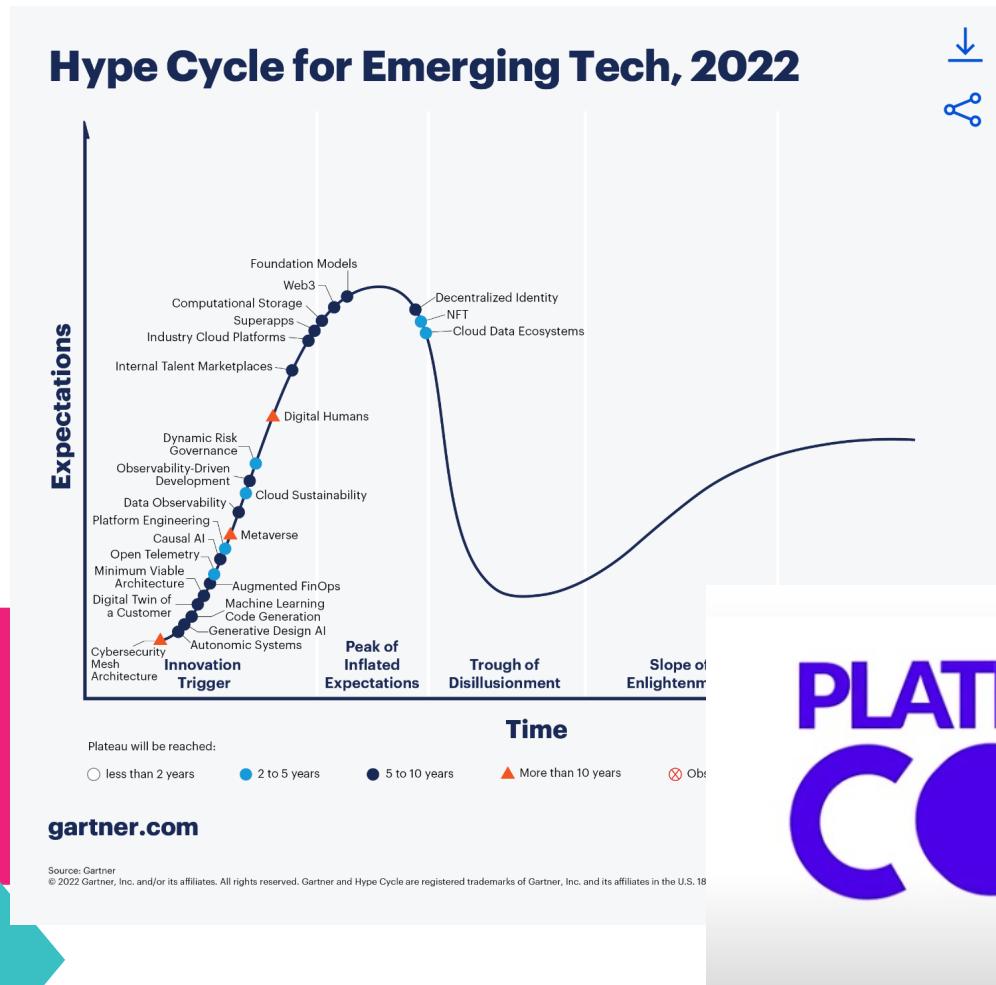
• 8m
• MOBILE ORDERS

10
LOCAL OFFICES

9
DISTRIBUTION CENTRES

7
DIGITAL PRODUCTION
CENTRES

The Rise of Platform Engineering



PLATFORM
CON²²

| Year | 2000 | 2005 | 2010 | 2015 | 2020 |
|---------------------------------------|-------------------------|--|---|---|--|
| Tech developers are exposed to | IDE, CVS, deploy portal | IDE, Mercurial, Jenkins, [PXE, Bash, Puppet] | IDE, Git, Heroku CLI, Heroku UI, New Relic UI | IDE, Git, Docker Hub, Jenkins+plugins, AWS Console, bash, Terraform, Chef CLI, Heroku UI, | IDE, Git, Docker Compose, K8s, Terraform |
| Developer responsibility | Code | Code, ship [limited run] | Code, run | Code, ship, run | Full lifecycle (code, ship, run) +++ |
| Infra / fabric | In-house tin | In-house / cloud | Heroku / CF | Cloud | K8s, IaC |
| App architecture | Monolith | Monolith / SOA | Monolith | Microservices | Microservices +++ |

Cognitive load

What is cognitive load?

In psychology, cognitive load is the mental effort needed to execute a task.

When the cognitive load is too high, it makes not possible to deal with a topic or issue because it's more complex than what you (or your team) are able to manage.

If information doesn't fit with the way you think or feel, you may become **angry at having your feelings or beliefs challenged**.

Where it started...

Everything began with domain-specific siloed team...

- Continuos Integration – later adding Continuos Deployment
 - Historical team (...probably back to 2010, if not before...)
- Application Delivery
 - Date back to 2016
- Monitoring team
 - Date back to 2017

Where it started...

Everything began with domain-specific siloed team...

- Continuos Integration – later adding Continuos Deployment
 - Historical team (...probably back to 2010, if not before...)
- Application Delivery
 - Date back to 2016
- Monitoring team
 - Date back to 2017

!!!

Central Management
No self-service capabilities

Rise of the automation...

In 2018, in parallel with a strong push toward cloud migration, a new team – Automation – has been created in order to leverage automation products as a Centre of Excellence across other teams.

The team was leveraging technologies such as Terraform, Ansible, Docker, Kubernetes and AWS.

In the meanwhile there was a consolidation of a central DevOps team which was operating on top of the capabilities offered by CI/CD, Monitoring, Application Delivery and Automation.

Creation of the platform....

CI/CD, Application Delivery, Monitoring and Automation team has been brought together to form the DevOps Platform area having the DevOps team as the main customer to serve.



Bottleneck!

Requests from DevOps team were flowing.

Development teams were exposed more and more to the cloud, acquiring skills and generating requests on their own, bypassing the DevOps team.

Teams were not able to cope anymore with requests following the former centralized model.

Automation team to enable self service capabilities was not able to cope with requests to enable “shift left”.



And then... CHAOS!

Platform teams started to develop their own automations but in a siloed way.

DevOps in order to cope with requests started to build automation and products on their own.

Development teams as well started to build capabilities on their own to cope with the increasing demand.



Time to review the model...

In order to bring back order and governance teams have been re-organized:

- New structure for the DevOps Platform area
- Team relying on cloud-based architecture have been transformed to E2E teams embedding DevOps
- DevOps / SRE working on products have been moved to the DevOps Platform area
- Products across the company have been assessed and rationalized

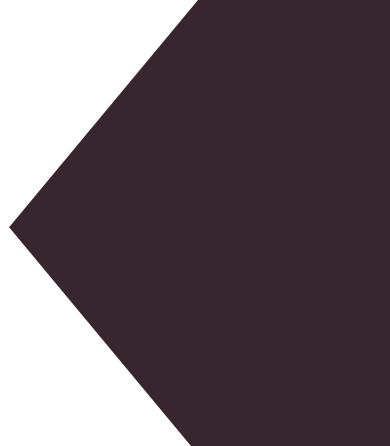




The New DevOps Platform Era

DevOps Platform Northstars

- Find the balance between **meeting needs across delivery teams** and **minimizing the complexity and cost of a central toolchain**. Focus on reduce the cognitive load and improve developer experience.
- Platform **co-created with delivery teams**, treating them as our internal customers, relying on a collaborative and transparent approach when selectin tools and defining the platform roadmap
- Continually assess the platform to ensure it stills meet delivery team's needs, and to retire redundant or end-of-life tools, striving for tools' rationalization
- **Implement platform metrics** (Net Promoter Score, % of adoption, % decrease in feature cycle time, toolchain cost, ...)
- Actively markets the platform to drive adoption; do not mandate the platform



SUBJECT MATTER EXPERTS
CO-CREATION
INTERNAL DEVELOPMENT PLATFORM

SUBJECT MATTER EXPERTS
CO-CREATION
MINIMIZE COMPLEXITY

PLATFORM METRICS
MINIMIZE COMPLEXITY

INTERNAL DEVELOPMENT PLATFORM
EMPOWER DELIVERY TEAMS

PRODUCT APPROACH

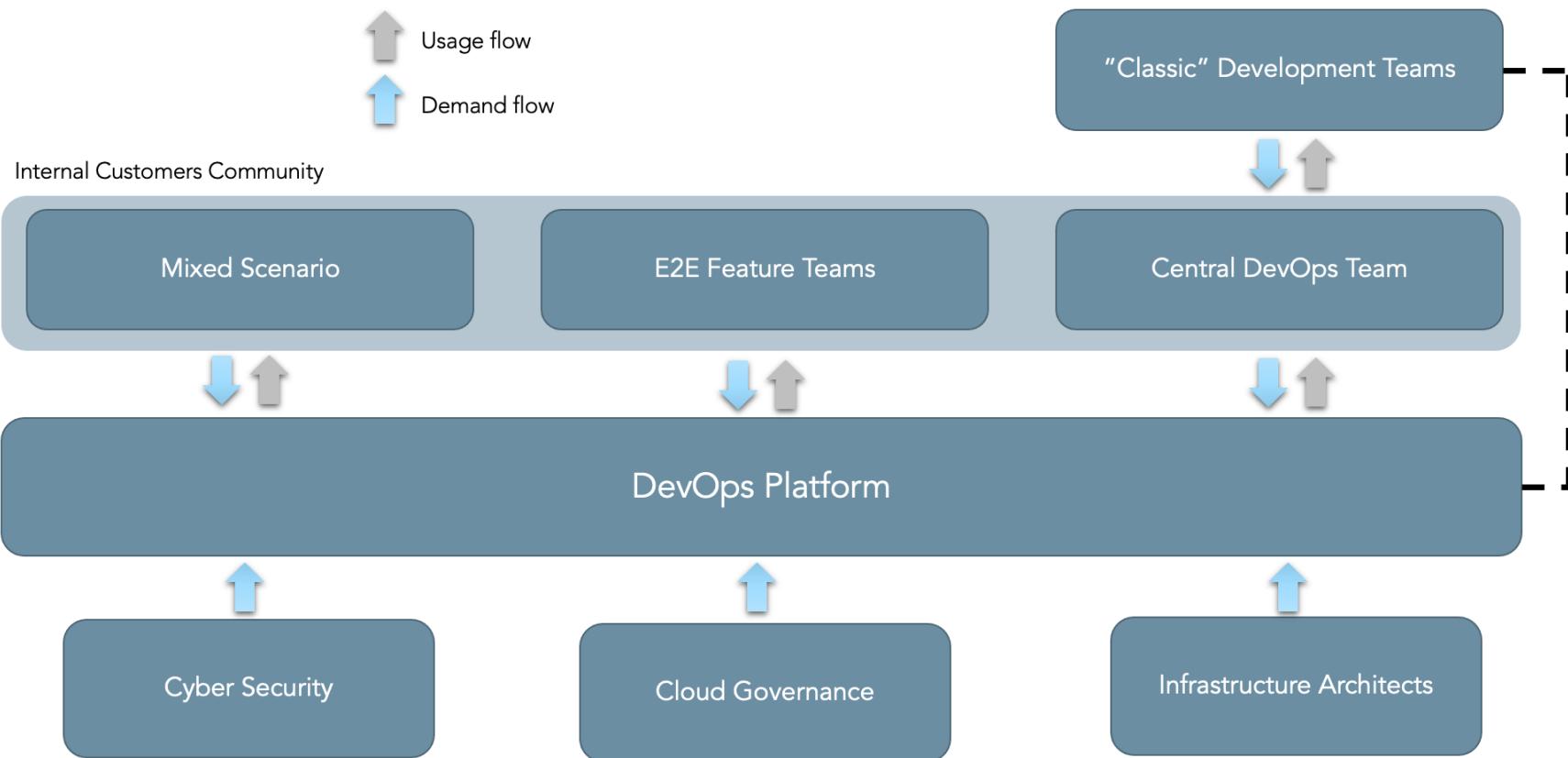
TRAINING & RESOURCES
TRAINING & RESOURCES

DevOps Platform Northstars

- Mixed approach between Open Source Software and proprietary, trying to avoid internal development
- If proprietary, try to rely on SaaS offering
- Provide tools and services to **allow teams to take responsibility for what they use and what they build**
- Embedded Security, Architectural and Governance guidelines
- Rely on a composable approach, where each service can be used independently
- Manage the platform as a product
 - Common guidelines
 - Unified Way of Working
 - Consistent way in how we interact with our internal customers
 - Long-lived and stable product team tasked with both build and run



DevOps Platform Organizational Model



Platform Numbers

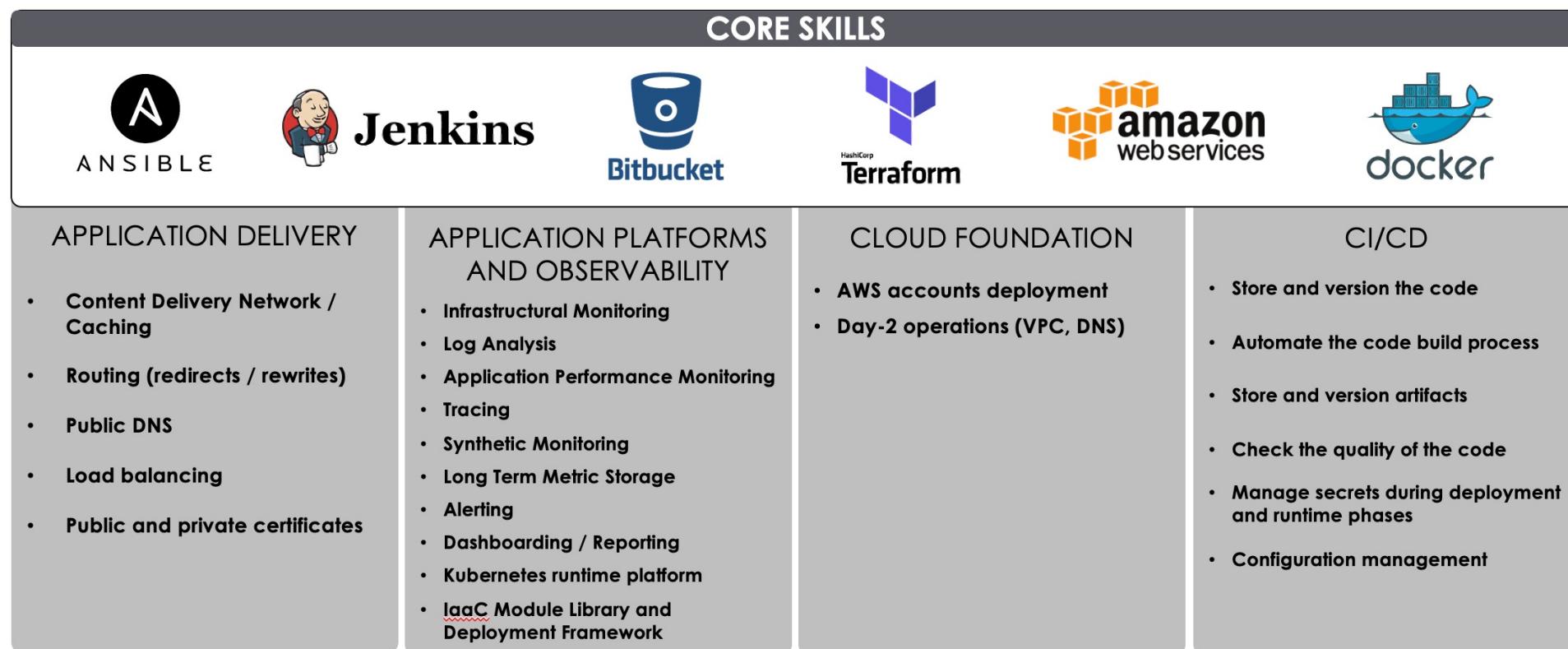
- Customer Community: around 600 users
- Supporting multiple languages (Java, .NET, Scala, NodeJS, React, ...) and infrastructure (VM, container, serverless, Kubernetes)
- 250 BitBucket projects and 9'000 pipelines
- 120 Kibana spaces, 8TB of logs
- 220 AWS accounts
- 25.000 secrets served monthly
- 40 new certificates monthly
- 30'000 monitored hosts
-

DevOps Platform Organization

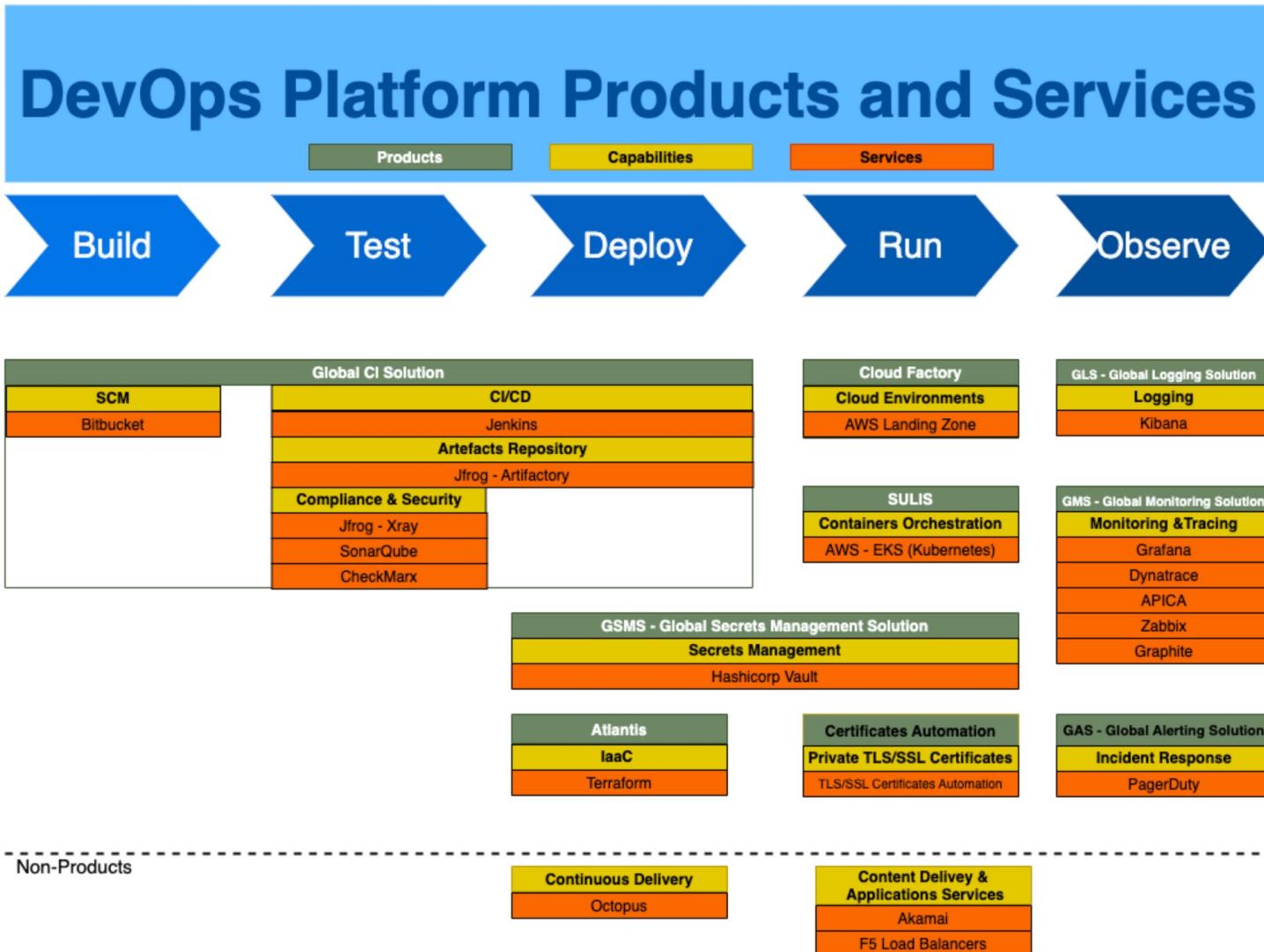
- Domain-based teams to take care of the different services, leveraging on their specific skills, while having a common “core” set of competencies

4 Teams 1 Platform

Everything you need to deliver and operate your applications

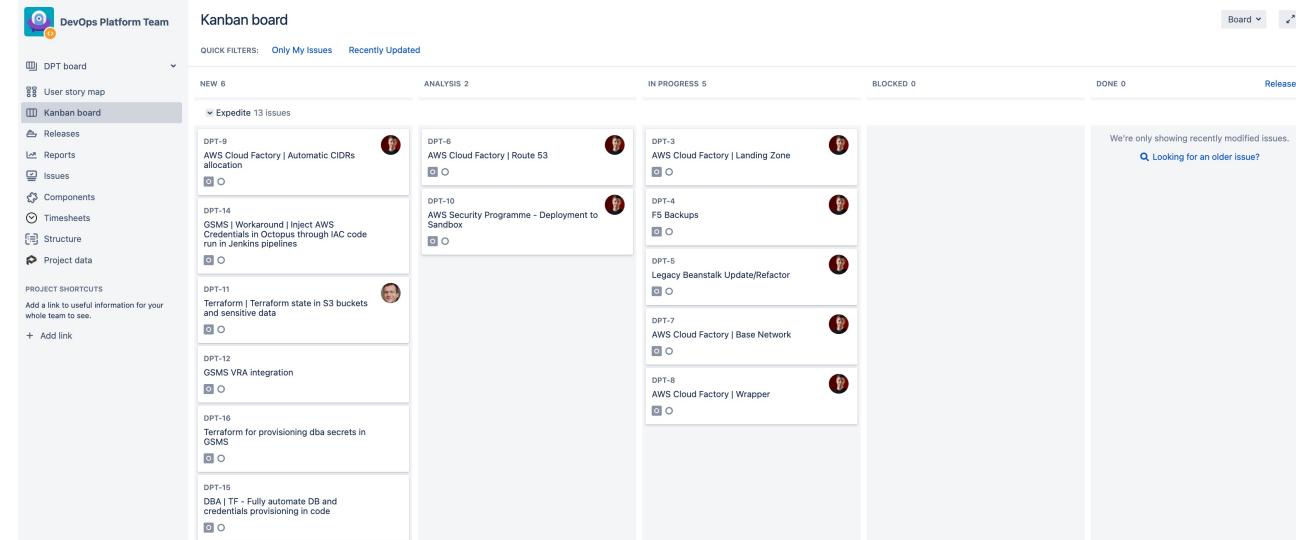


DevOps Platform Portfolio



Centralized Demand Management

- Where to track all requests for new work on DevOps Platform product, either external (new requests from our internal customer, eg: new capabilities, new integration, etc...) or internal (new requests from within the platform itself, eg: product upgrade, new capabilities, new integration, etc...)
- A single place where to have all the requests and the current work in progress, to have full visibility and to be able to prioritize, leveraging all platform resources
- For external request, having a clear understanding of the requestor and build on top of their commitment to analyze, build and test, and eventually socialize and scale what has been done



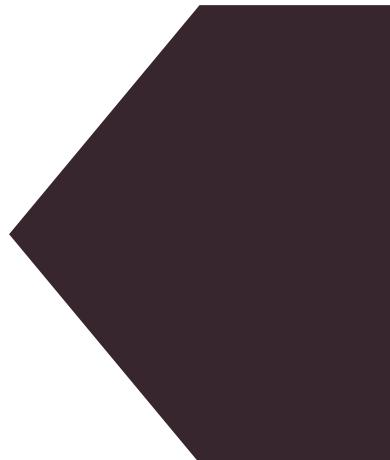
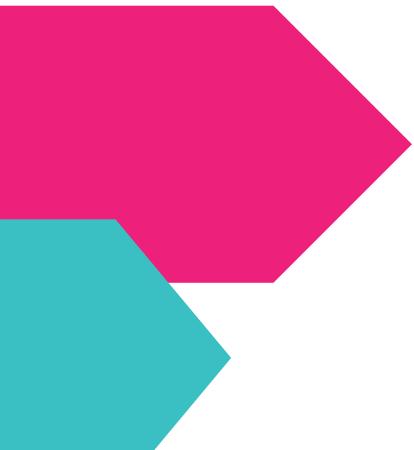
Developer Portal (for documentation)

- A place where to have all the documentation about the platform products and services to be used by our internal customer
- The goal is to offer to our customers a single repository where to find all the information about our product and services, in a coherent way across all products

The screenshot shows the 'DevOps Platform Home' page in a Confluence space. The left sidebar includes links for 'Pages' (Blog, Calendars), 'SPACE SHORTCUTS' (Certificate Management & Automation, Global CI Solution, Cloud Factory, Global Secrets Management Solution - GSMS), and a 'PAGE TREE'. The main content area features a large diagram titled 'DevOps Platform Products and Services' with five main stages: Build, Test, Deploy, Run, and Observe. Each stage is represented by a blue arrow pointing right. Below the stages, there are several colored boxes representing different products and services, such as SCM (Bitbucket), Global CI Solution (Jenkins, Artifacts Repository, Jfrog - Artifactory, Compliance & Security, Jfrog - Xray, SonarCube, CheckMarx), Cloud Factory (Cloud Environments, AWS Landing Zone), GSMS - Global Secrets Management Solution (Secrets Management, Hashicorp Vault), and various monitoring and logging tools like Grafana, Dynatrace, APICe, Zabbix, Graphite, and PageDuty.

Did we reach the heaven?

Our team with the new platform
looking at the rest of the world...



Challenges

Communication

Make sure people are aware of changes and updates in the platform.

Make sure we align the platform strategy to the tech strategy which is used to fulfill the business strategy (No.. the demand management process is not enough).

Multi-channel approach: we have Slack, Confluence, Workplace, E-Mail (Distribution List).

Challenges

Migrations

All new projects start relying on the strategic products part of the platform, following golden path and standardized blueprints.

However we have still a lot of applications relying on “non-strategic” / “legacy” products and services and implementing non-standard pattern. It is not easy to migrate them to the strategic products without slowing down development and DevOps teams.

This forces us to maintain both the strategic and the legacy products.

Challenges

Lack of analytics and visibility

Products have been build and given to the customers without implementing data collection and analytics capabilities.

Hard and time-consuming to obtain usage data.

Lack of visibility when need to make change.

Lack of confidence when need to make change since we are blind on what has been implemented on top of what we provide (fear of breaking something).

Challenges

Product Manager role

We don't have a product manager role.

Responsibilities (customers interaction, collecting requests, prioritization, backlog refinement, feedback collection) are currently shared between Team Managers and Project Managers.

Product Manager is a full-time job.

Challenges

Self-Service Portal

Different ways to interact with the platform:

- Automation triggered by JIRA
- GitOps
- ServiceNow
- Manual processes (through JIRA ticket)

Where is not manual, behind is usually Jenkins or Lambda.

Challenges

FinOps

Consolidate platform costs (license cost – either fixed or consumption based – + operational costs) and allocate those across customers teams and products.

Currently implemented just for some services:

- AWS
- Dynatrace

Challenges

Full spectrum of capabilities

The platform does not provide the full spectrum of capabilities required by a development to build and run an application:

Some capabilities sits in a different area:

- DBA
- On-Prem automation sits in a different area
 - Even if we provide Terraform provider and modules for VRA
- Cyber Security (Code Security, IaaC Security, SBOM, Secrets Management, ...)
 - Even if some capabilities are already integrated
- Integration Services (ESB, Message Queuing, API Gateway, ...)
- Functional and performance testing

PlatformOps

Moving away from Jenkins and experimenting PlatformOps / Platform Orchestrator Tool:

- Krateo
- Humanitec
- Mia-Platform
- Cycloid
- ...a lot more are borning

As a platform, we want to provide an integration layer through PlatformOps / Platform Orchestrator Tool, so we allow external team to integrate capabilities but still providing a single point of entry for developers.

Useful Resources

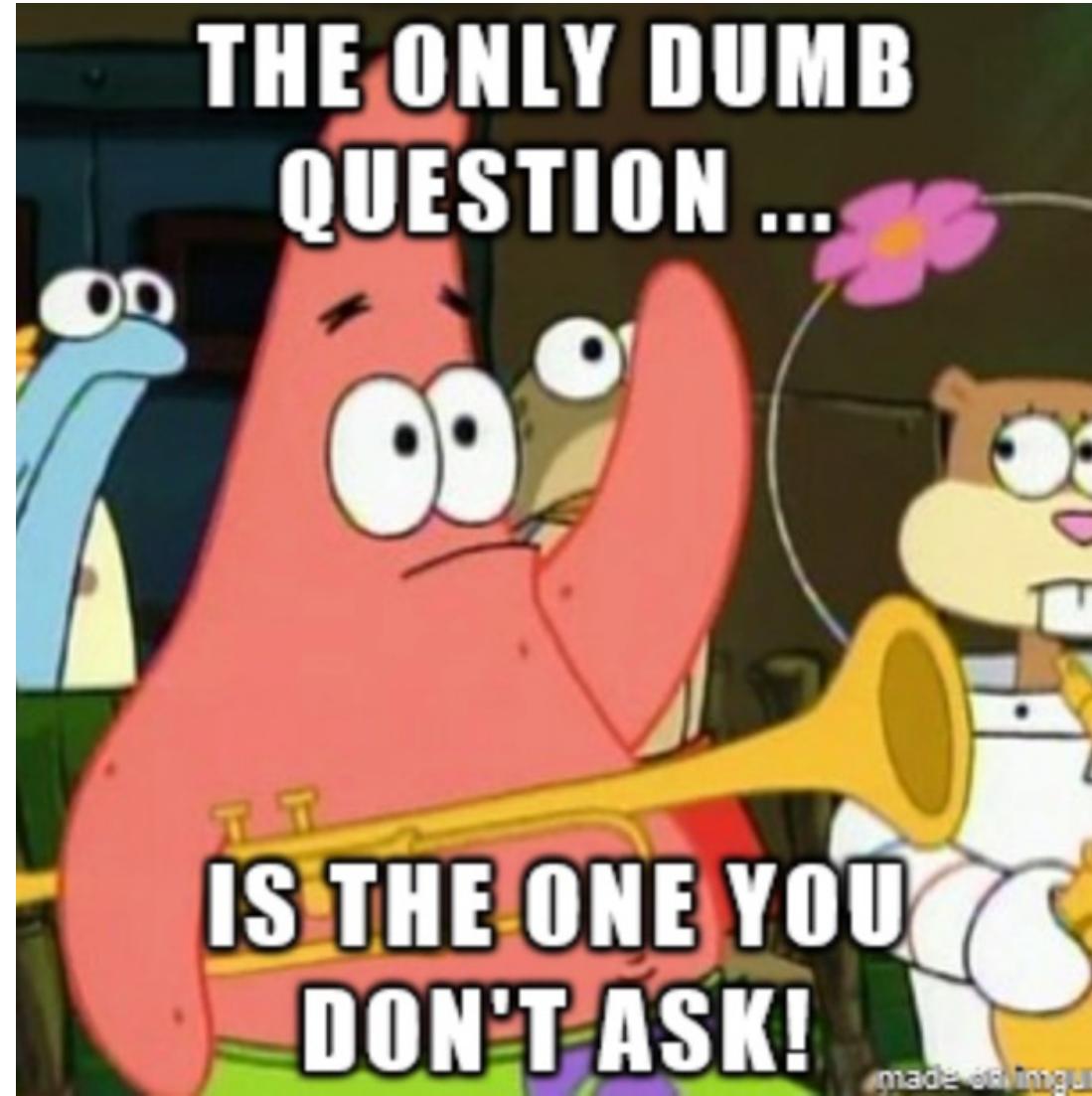
- PlatformCon - <https://platformcon.com/>
- Platform Engineering – Slack Community
- Internal Developer Platform - <https://internaldeveloperplatform.org/>
- Team Topologies (Book) - <https://teamtopologies.com/>
- Humanitec blog - <https://humanitec.com/blog>
- Architectural Decision Records - <https://adr.github.io/>

Useful Resources - BONUS

The whole story became also a song!

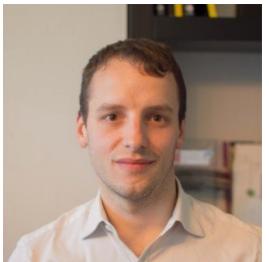


https://www.youtube.com/watch?v=yDcaRkIX7q4&ab_channel=ForrestBrazeal





Grazie!!!



 leonardo.romanato@ynap.com

 <https://www.linkedin.com/in/lromanato/>

 [@F3ar873](https://twitter.com/F3ar873)

