ELCA

Platform Engineering

Building a portal for Developers with Backstage

Gaël Gothuey

gael.gothuey@elca.ch

$ whoami

gaelgoth
gothuey.dev

- `Gaël Gothuey`
- `DevOps Engineer` / `Platform Engineer` at ELCA
- `Backstage` Contributor
- `Homelab/Home` server enthusiast
- `Azure` expert
- Street photographer

What this presentation about ?

1. `Platform` vs `DevOps` vs `SRE`
2. Developer Experience `DevEx`
3. `Backstage` Introduction
4. `Backstage` Demo
5. Key Takeaways
6. Q&A

Platform vs DevOps vs SRE

Understanding the distinct engineering roles

Platform Engineer

Building tools for engineering teams

Focus: Internal Tooling, Infrastructure
Clients: Developers, Engineering Teams
Example: Database setup from dev to prod

DevOps Engineer

Improving deployment velocity

Focus: CI/CD Pipelines, Automation
Clients: Developers and SREs
Example: User login dashboard monitoring

SRE

Hosting reliable production products

Focus: Reliability, On-call Support
Clients: Business Stakeholders
Example: 99.9% database uptime

`Platform Engineering` discipline

Focused on the development of self-service toolchains, services, and processes

"discipline of designing and building toolchains and workflows that enable **self-service capabilities for software engineering** organizations in the cloud native era" - (platformengineering.org)

"it allows companies to quickly develop and deploy applications while **maintaining quality standards** "

"With DevOps, the market was more interested in shifting responsibility left and empowering individual developers to oversee the entire software life cycle. Now, as sprawl and shadow IT emerge, the pendulum seems to be shifting back toward a bit **more centralization and governance with platform engineering** " - (devops.com)

"By 2026, 80% of software engineering organizations will establish **platform teams as internal providers of reusable services** , components and tools for application delivery." - (hashicorp.com)

Platform Engineering objectives

Enabling developer autonomy in complex systems

Why Platform Engineering?

- Increasing complexity of cloud-native environments
- Developer cognitive load reduction
- Accelerating development cycles
- Consistent governance and security

Our Priorities

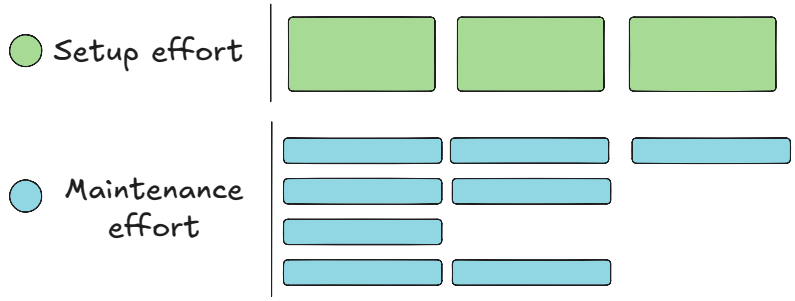
Keep project creativity and accountability: Dictating specific application architecture or hindering team autonomy

Top priority: Self-service developer experience with guardrails

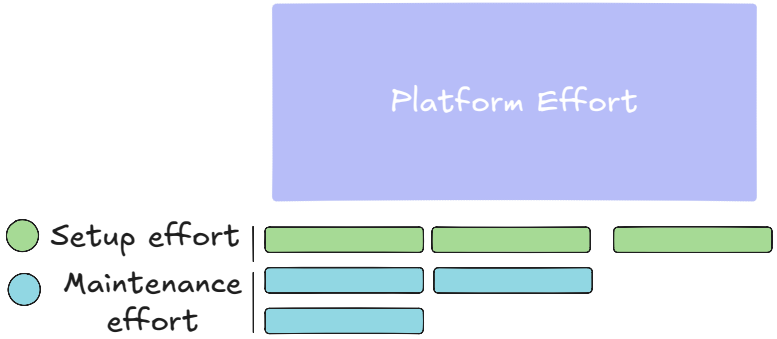
Key metric: Time to production for new services

Project vs Platform

What's the difference with a classical project?

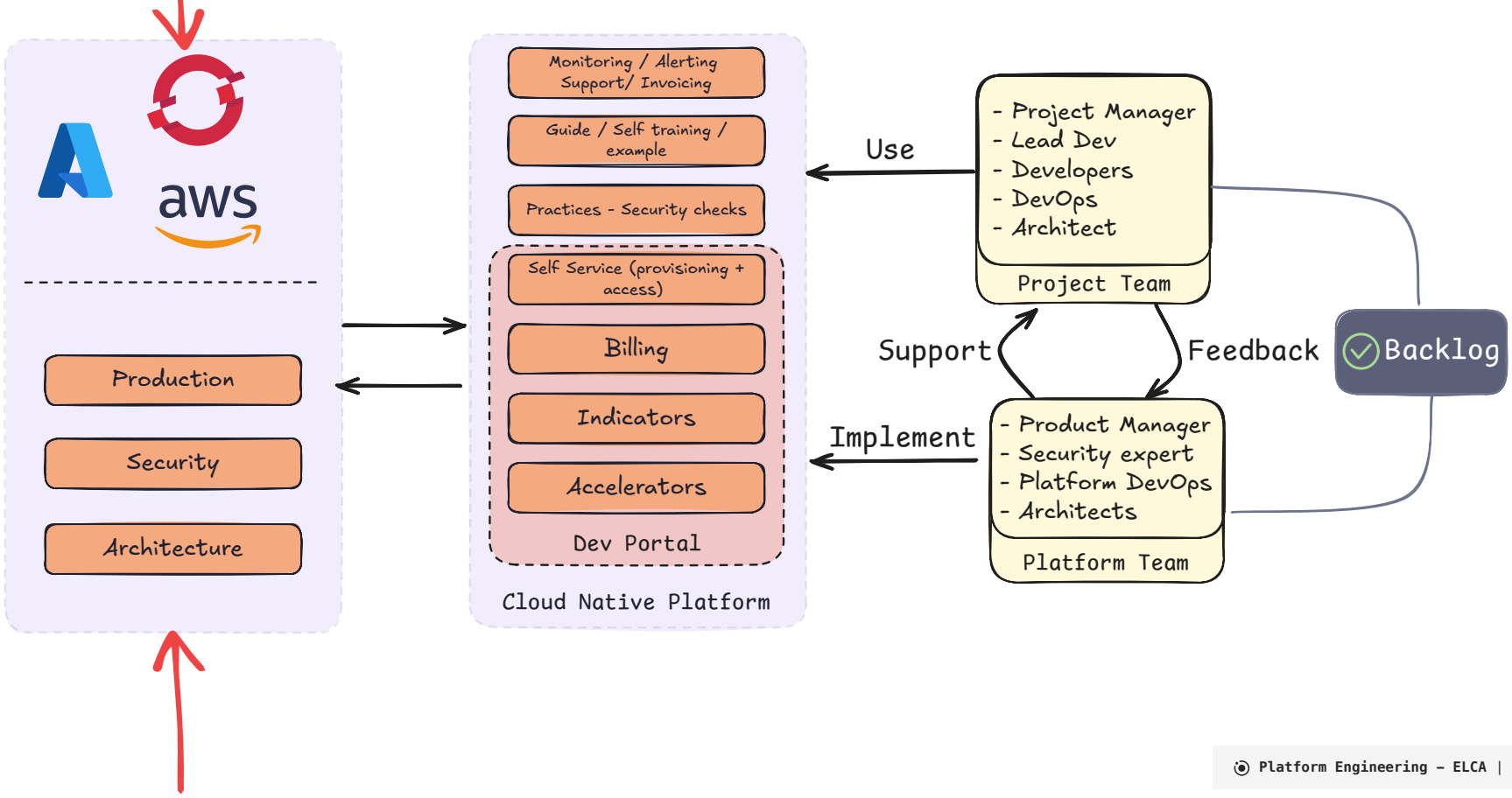


vs



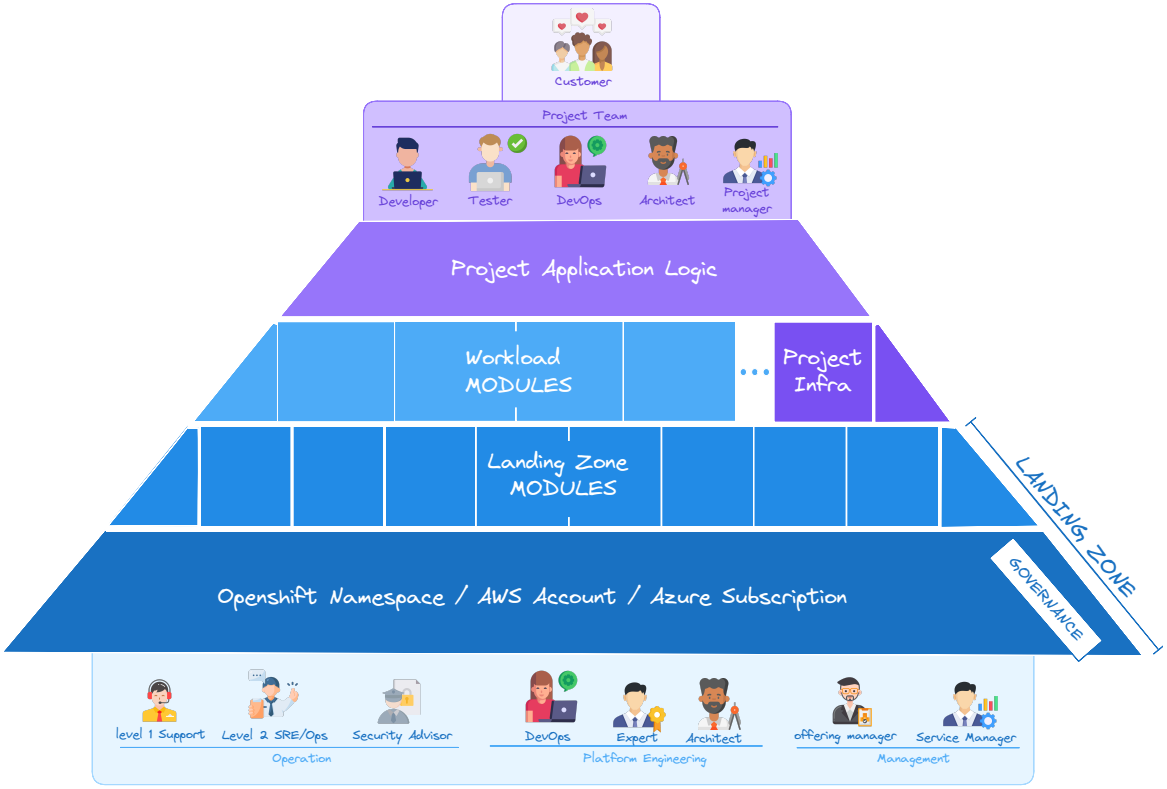
Platform as product

Building ecosystems that scale



Teams organization

Defined purpose and responsibilities



Developer Experience DevEx

Empowering developers with the right tools

Developers have access to a comprehensive ecosystem of tools to enhance productivity

Developer Experience DevEx

Where tools work better together

ELCA

Self Service
Management
Documentation
Golden Samples
Catalog
SW Templates
Announcements
Services Status
Search
Administration
Settings

Jenkins

Jenkins / *create-instance*

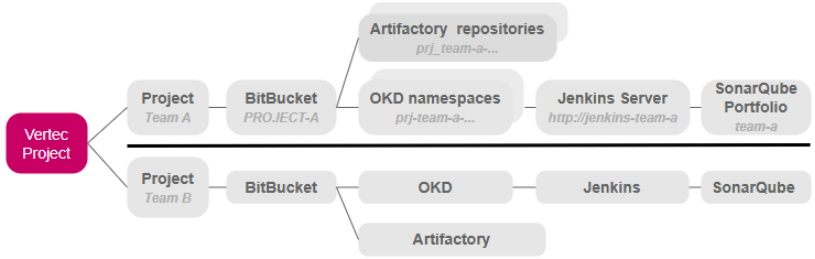
Setup Jenkins for your project

1. Select your ERP Project
   ERP Project *
   80143 - CI/CD DevOps team
   Internal project Customer project
   CONTINUE
2. Select a project team
3. Instances details
4. Summary

Step 1 : ERP Project Number

Each service is linked to an ERP project to allow:

- Only Project Managers to instantiate service and manage members.
- Implement Auto-Archival on Ended/Closed project.



Only projects where you are manager or deputy are listed.

Learn more on ERP project requirements →

Backstage

One hub for developer information and tools

* IDP: Internal Developer Platform

Timeline:

- **Developed at Spotify** to centralize developer tools and documentation
- **Open-sourced in 2020** to benefit the broader community
- **Joined CNCF Incubator in 2022**, growing as an industry standard

Community:

- **60+ open-source plugins** in the Plugin Marketplace
- **100** publicly listed adopters
- **15k+ stars** on GitHub

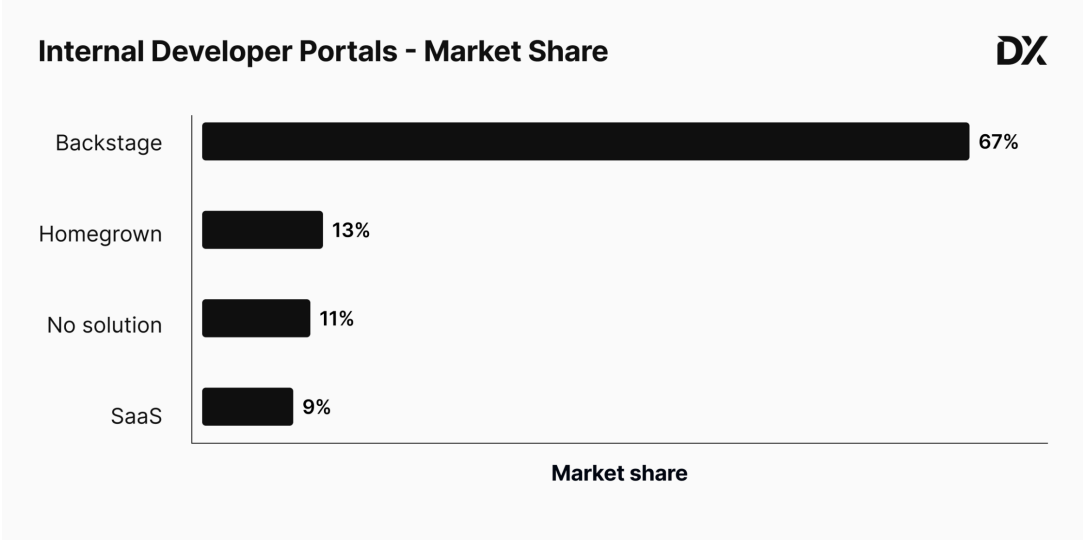
The IDP state

Backstage

One hub for developer information and tools

* IDP: Internal Developer Platform

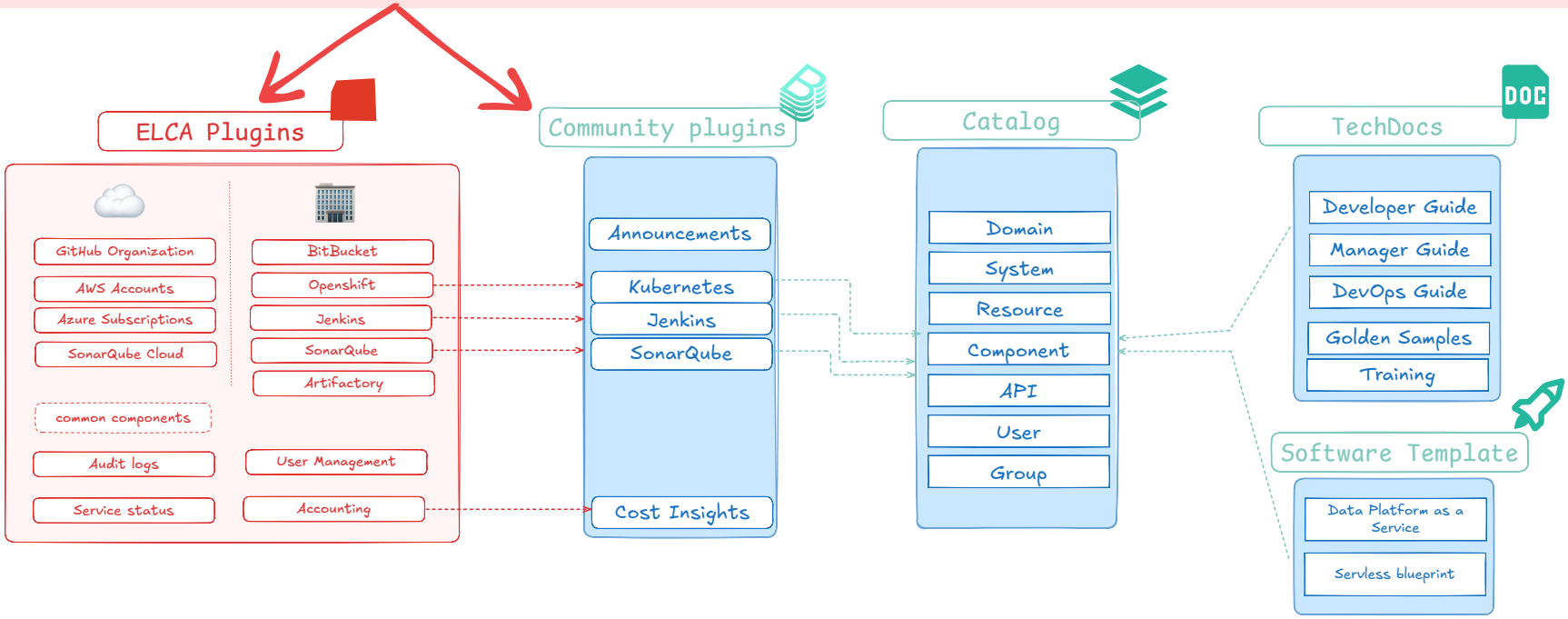
DX surveyed 180 companies to understand how they were approaching internal developer portals (IDPs), their result:



* Published on Mar 19, 2025 - Ref: https://newsletter.getdx.com/p/backstage-and-the-developer-portal-market

and ELCA

Our Backstage model



Self-Servicing

Async Processes – The Agnostic Solution

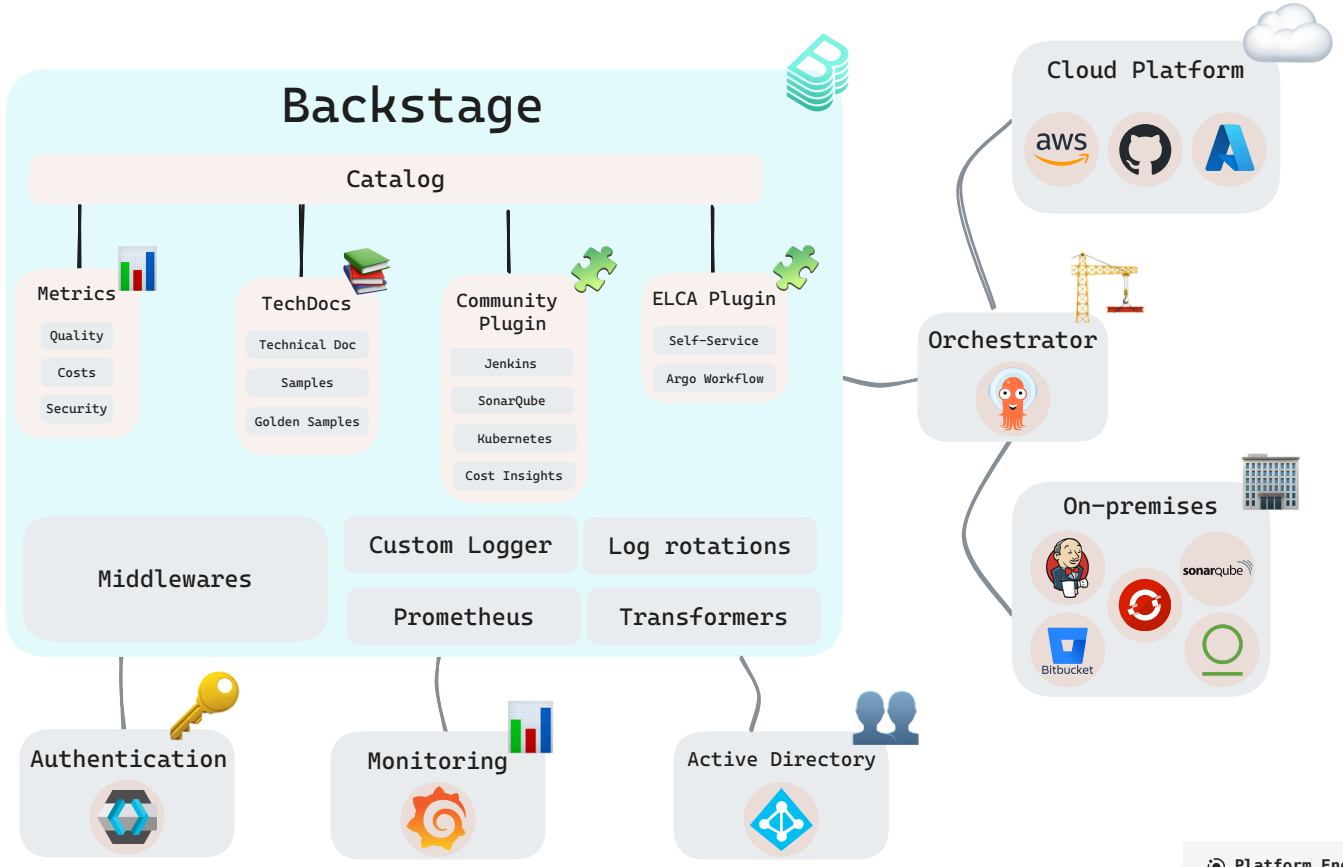
Argo Workflows:

- **Kubernetes-native** workflow engine
- **Handles long-running tasks**
- **Schedules jobs** (state imports, health checks, cleanup)
- **Runs any container-based language** (Python, Open Tofu, bash, Powershell)
- **Web UI for visualization**

Not to be confused with Argo CD

- **Argo Workflows** automates workflows in Kubernetes
- **Argo CD** is a GitOps deployment tool

Big picture


Backstage
Catalog
Metrics
Quality
Costs
Security
TechDocs
Technical Doc
Samples
Golden Samples
Community Plugin
Jenkins
SonarQube
Kubernetes
Cost Insights
ELCA Plugin
Self-Service
Argo Workflow
Middlewares
Custom Logger
Log rotations
Prometheus
Transformers
Authentication
Monitoring
Active Directory
Cloud Platform
aws
Orchestrator
On-premises
sonarqube
Bitbucket

Backstage Demo

Platform Engineering Demo

- ✓ Service Service
- ✓ Cost Insights
- ✓ Documentation
- ✓ Golden samples
- ✓ Catalog
- ✓ Admin view

🚧 Challenges

- Enhancing isibility into monitoring and alerting across the platform
- Facilitating access to short-lived sandbox and test environments (e.g., AI/ML testing, POCs).
- Need skills, the Platform Engineering Team handles everything (*Build*, *Run*, *Support*)
- Steep learning curve for adopting Backstage

🌟 Achievements

- Backstage serves as the source of truth for ~3555 services
- Achieved 95% transition from VMs to containers for on-premises workloads
- 90% of projects have adopted CI/CD pipelines

Key Takeaways

←

Thanks for listening! 👋

Gaël Gothuey

gael.gothuey@elca.ch
/gael-gothuey
/gaelgoth

- https://medium.com/elca-it
- https://platformengineering.org