eficode

PLATFORM ENGINEERING BEYOND THE TOOLS: LESSONS LEARNED ACROSS INDUSTRIES

whoami

Stefan Daugaard Poulsen
stefan.poulsen@eficode.com
https://www.linkedin.com/in/cyberzed/

Background:
Developer, Team Lead, CTO, SRE, Staff Engineer, DevOps Consultant

Current:
Solution Architect - DevOps & AI @ Eficode
Co-Organizer @ CNCF Aarhus
Core member @ Cloud Native Platform Engineering
Co-facilitator @ Platform Coffee during KubeCon

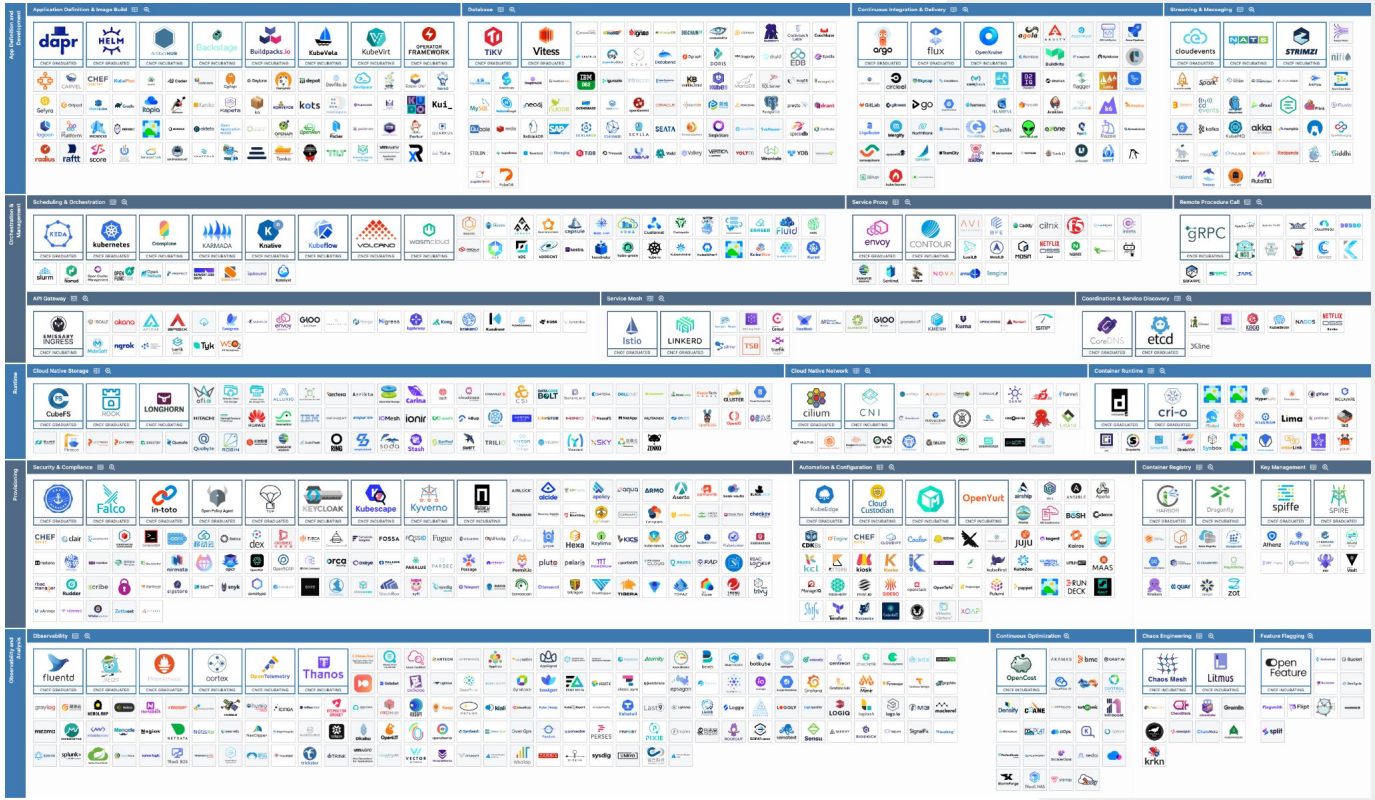
Platform Engineering

...is it the dream or a nightmare?

The promise

...vs. Reality

But why do we fight all of these monsters?



Tales from the crypt

eficode

Mindset is key - it’s less about fancy quotes

“

If you build it, they will come

“

If you build the right thing, they will come

The great dissonance of many organizations

“Be an enabler”

A long history of silos and gatekeeping
- Mindsets takes quite some time to shift
- …and in some cases new people

“We can’t afford slowing down”

The organization needs stability and uptime
- Though it wishes to leverage modern technology
- Business units request something as fancy as AI

This is no different from Agile and DevOps
…it’s mindset and practices and no tool can solve that

eficode

Know your customer

What do your customers need?

Developers rarely wish to know intricate infrastructure details

Most don’t want to know about YAML...especially kubernetes object specs, some don’t even want a CLI

Find collaborators when you need to build a new capability

- Interview them
- Prototype with them
- Ask when they expect
- Tell them what you can automate
    - Good defaults go a long way

It’s more than “just” developers, remember security and governance when creating new things


So tell me what you want
what you really really want

Feedback collection

When not collaborating be very deliberate in gathering your feedback

How are things performing
...can be regular technical metrics, but not always

Is something used so little that it should be deprecated due to unbalanced maintenance?

Gather **structured** feedback

- Surveys
- Interviews
- Etc.

The coffee and cookie principle

...and yes there be stickers!

Paved paths or fenced gardens?

Larger organizations tend to have some high fences around their gardens

- High conflict level
- Distrust among areas

What if we can lower the friction by showing the value?

- Built-in security
- Automatic reports for compliance audits
- Organizational flexibility
- A set of well-known languages

You’ve built it, but do they know?

Start small and easy

Unrealistic goals are often stated, which turns into mandated platform adoption

Adopting a platform is optional! It is up to you to make sure it’s beneficial to be there

The better way is thinking about advertising from day 1

- Give the platform a name
- Give it a logo
- MAKE STICKERS
    - AI can help you out on this though it might need some iterations

If adoption halts, you might need to go out into the community

Hand out candy with the stickers if needed, it seems cheap but it works

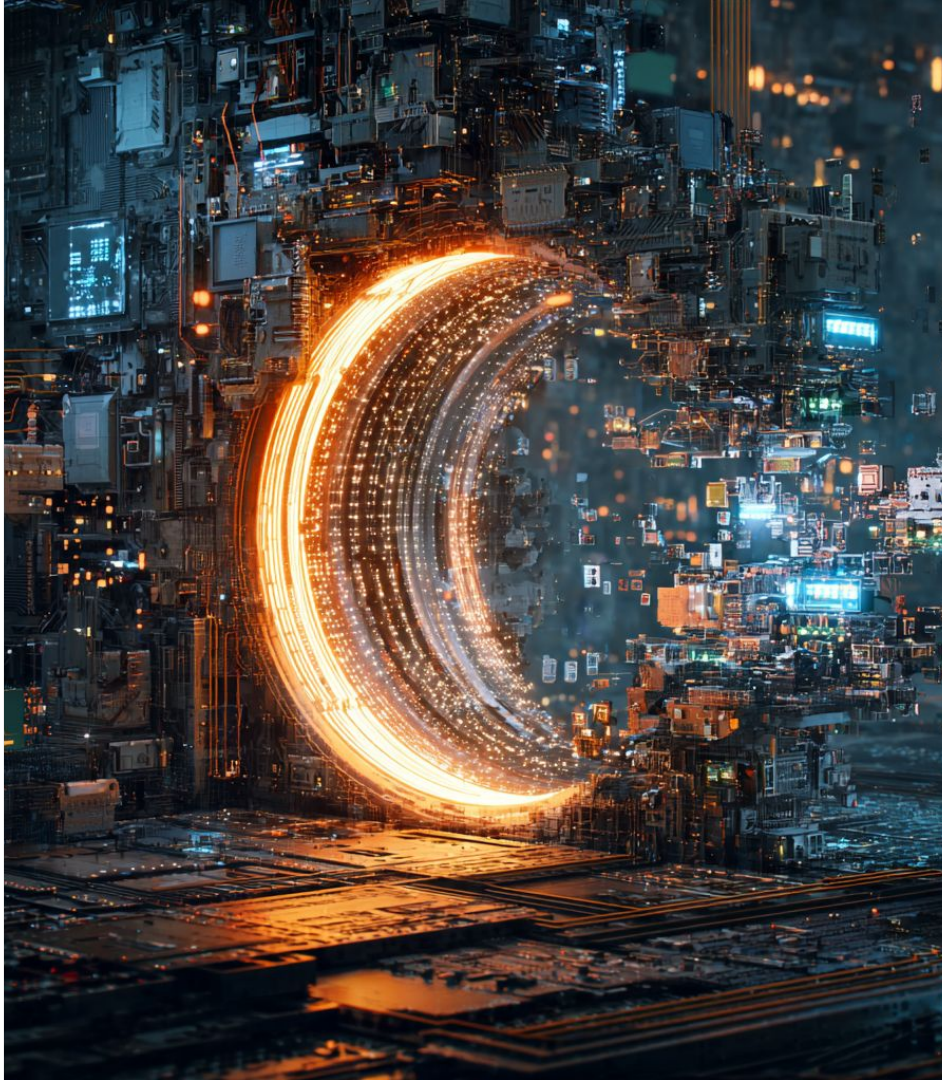
Portals

We need to talk about when a portal is fitting

Having Backstage doesn’t mean you have a platform

A good modern portal can assist you in building a platform

- Feedback collection
- Structure in chaos
- Views for many personas
- Not only for developers



Community

Just as if you were creating any champion programme, you need a good community

How to build your community?

- STICKERS for starters
- Find those who have seen the value in the platform
  - …but do also bring a critical voice
- Make sure your team have a person who can nurture the community
  - You don’t have to be a raging extrovert to talk to other people
- Make them help you tweak and tune the documentation
  - …you do have good documentation right?

The essence of it all

Rome wasn't built in a day

eficode

PLATFORM ENGINEERING

TECHNICAL COMMUNITY GROUP

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

Take it easy

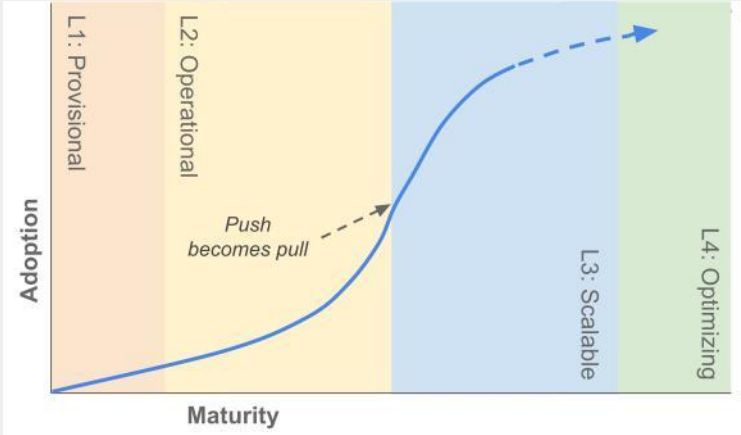
The maturity model is about figuring out where you are

The maturity model is good for planning your next steps to keep things balanced

Trying to jump directly from 1 to 4 will most likely fail

You don’t have to reach level 4, the investment might be higher than the reward

The maturity model is for everyone



Take a peek

https://cloud-native-platform-engineering.github.io/pemm-assessment/

Thank you!

Remember

- It’s your users platform
  - Listen to them
- Progress comes in steps
- Don’t rush it

Stefan Daugaard Poulsen
stefan.poulsen@eficode.com
https://www.linkedin.com/in/cyberzed/