



SEP 09 - 11, 2025
CONTAINER
days
CONFERENCE

white duck
The Azure Cloud Experts

WELCOME
CONFERENCE



Build a Platform that Matches YOUR Needs!

ContainerDays 2025



Digital & App Innovation
Azure

Specialist
DevOps with GitHub



Data & AI
Azure

Specialist
Migrate Enterprise Applications
to Microsoft Azure

Who am I?



Nico Meisenzahl
Geschäftsführer | COO



+49 8031 230159-112



nico.meisenzahl@whiteduck.de



@nmeisenzahl



www.linkedin.com/in/nicomeisenzahl

- Cloud Solution Architect
- Azure & Developer Technologies MVP

The pains we're trying to kill

- 2-week wait for a simple database
- Manual security sign-off every release
- 5 different, undocumented deployment paths

➤ **Does this sound familiar?**

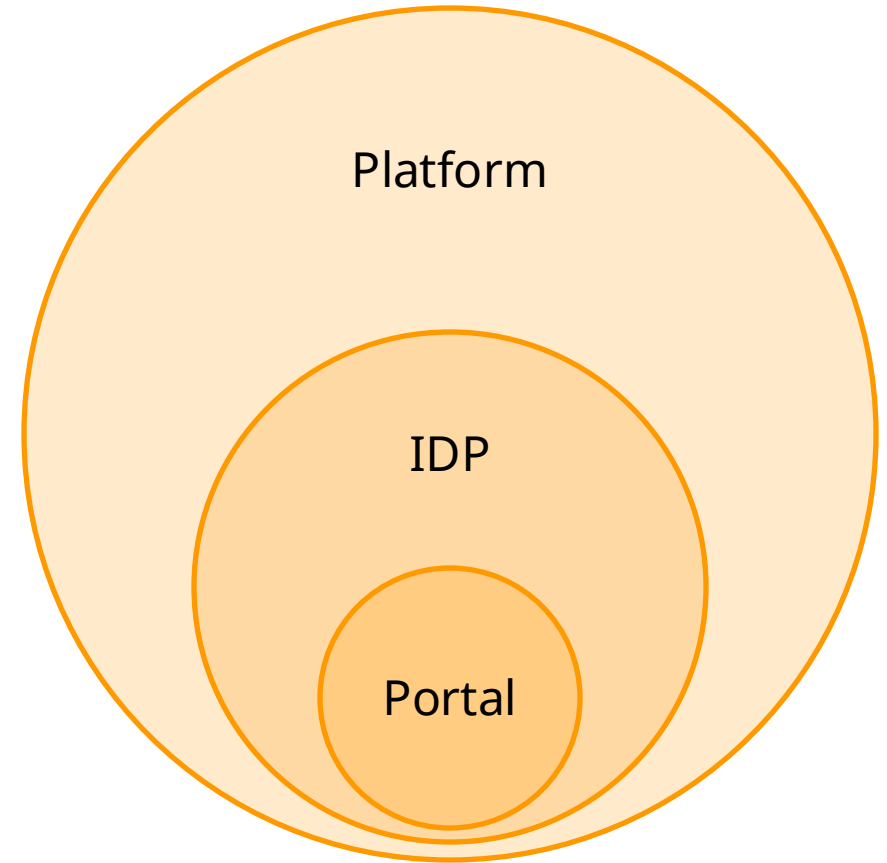
Goals of a platform

- Abstraction of complexity through Golden Path
- Self-Service and great Developer UX
- Built-in security & compliance
- Enhanced team collaboration
- Streamlined cost optimizations and governance
- **Better Dev Experience → Faster Delivery → Happier Humans**







Platform ≠ IDP ≠ Portal

- Platform = the underlying & essential foundation
- Internal Developer Platform (IDP) = a self-service layer to scale
- Portal = optional user interface
- **An IDP depends on a Platform, but a Platform can exist without an IDP**



A Platform can be everything and nothing

- Cloud Governance and some blueprints 
- Terraform Modules & shared CI/CD templates 
- One Kubernetes Cluster with templates and guardrails 
- CAPI-based K8s Clusters with GitOps, Crossplane & Backstage 

Match YOUR needs

- Focus on your requirements, not someone else's
- Start with problems, not tech
- Define your platform by
 - Top 3 frictions your team faces
 - Personas & their "jobs to be done"
- Example: A mid-sized ISV's platform ≠ a 200+ developer platform
 - But they might solve the same problems!

Core capabilities

- Replace human interaction with self-service
 - Foster autonomy & remove bottlenecks
- Golden Path: Paved roads with guardrails baked-in
 - But: Allow exit ramps for flexibility
- Developer UX: Unified AND flexible
 - Meet developers where they live (Portal, CLI, API)



Before you start: Assess Maturity

- First, know your requirements
- Buy vs. Build
 - Unique business needs?
 - Internal skills or resources?
 - Time-to-market?
- Reality check: Skills, budget, governance appetite
- Understand where your team stands today

➤ **CNCF Platform Engineering Maturity Model**

Treat the platform like a product

- Start with a Thin Viable Platform – solve one dedicated issue first
 - Form a dedicated Platform Team with a budget
 - Do not build your platform as “side work” project
 - Publish a transparent roadmap & SLOs
 - Implement feedback loops
 - Office hours, chat channels, issue tracking
 - Define and track KPIs
 - Don't miss the human metric: Developer satisfaction
- **Measure, gather feedback, improve. Repeat.**

Cultural change & adoption – required for success!

- Self-service is key, but requires a cultural change
 - Shift mindset from “open a ticket” to “push a button”
 - More importantly, this requires an organizational cultural shift across the company/IT
 - Freedom WITH responsibility: paved road, not golden cage
 - Open for everyone: transparent docs, shared contributions, and collaboration
- **Culture > Tooling (Culture is the foundation)**

Don'ts & common pitfalls

- Avoid Over-Engineering → Netflix-grade complexity for a 3-team startup?
- Foundation First → A slick UI ≠ success
- “Set and forget” mentality:
no metrics, no iteration → no adoption
- It's NOT a Project → Platforms are long-lived products
- Self-Service ≠ “Golden Cage” → Mandating one true path pushes developers away
- Dependencies kills velocity → Clarify ownership
handshakes across teams





Thank you!