



OPITZ CONSULTING  

---

**PUBLIC**

# PLATTFORM-SALAT ODER PLATTFORM-STRATEGIE? WARUM DAS ZUSAMMENSPIEL ZÄHLT

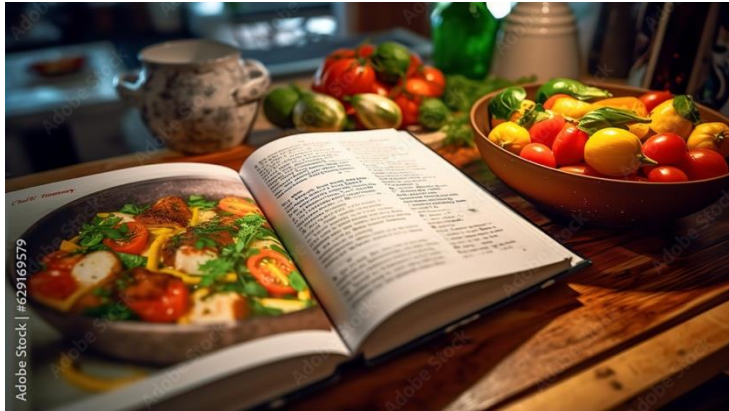
Sven Bernhardt,  
Fabian Hardt

Online, 02.12.2025



Source: Adobe Stock

# WHAT IS NEEDED TO COOK A PERFECT MENU?



**To create a perfectly balanced menu, we need more than just a variety of food and ingredients.**

# WHAT DOES THIS HAVE TO DO WITH PLATFORMS?

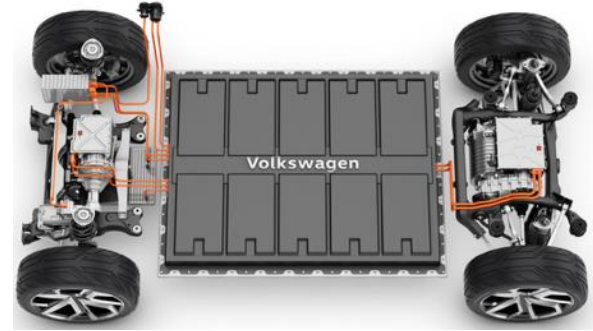


# DIFFERENT INDUSTRIES, SAME PRINCIPLE: A PLATFORM ENABLES OTHERS TO BUILD

## Consumer electronics



## Automotive



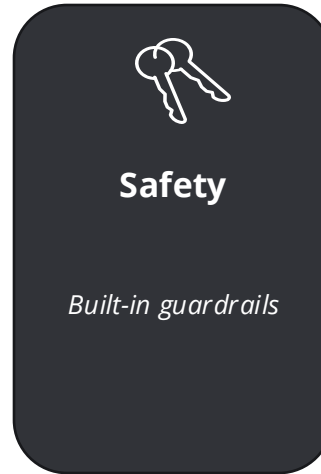
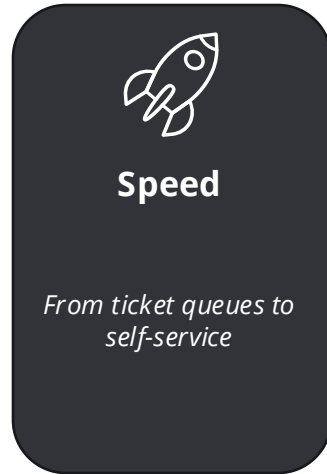
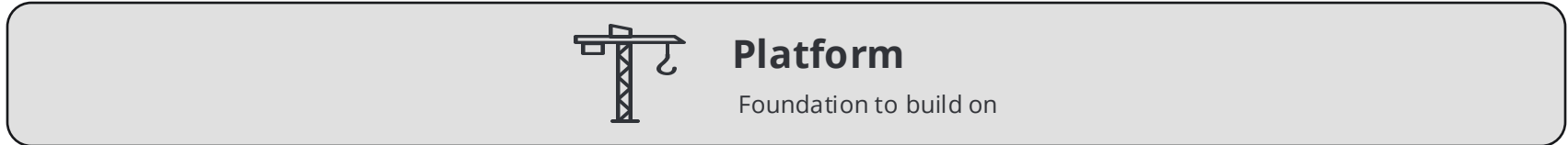
## Business / eCommerce



## IT / Development



# WHAT MAKES A PLATFORM A PRODUCT?



# WHY PLATFORMS ARE ESSENTIAL TODAY

- Foundation of Modern IT
  - Platforms are the foundation of modern IT architectures
  - drive digital transformation
- The Challenge of Platform Chaos
  - A multitude of platforms leads to complex,
  - Difficult-to-integrate landscapes within companies
- Strategic Platform Integration
  - Strategies help create synergies
  - Efficiently integrate AI and new technologies
- Securing Competitive Advantages
  - Integrated platform approach increases innovation
  - Secures long-term competitive advantages



**A platform is a product  
and a foundation to build on.**

# REFERENCE BACK TO OUR MENU



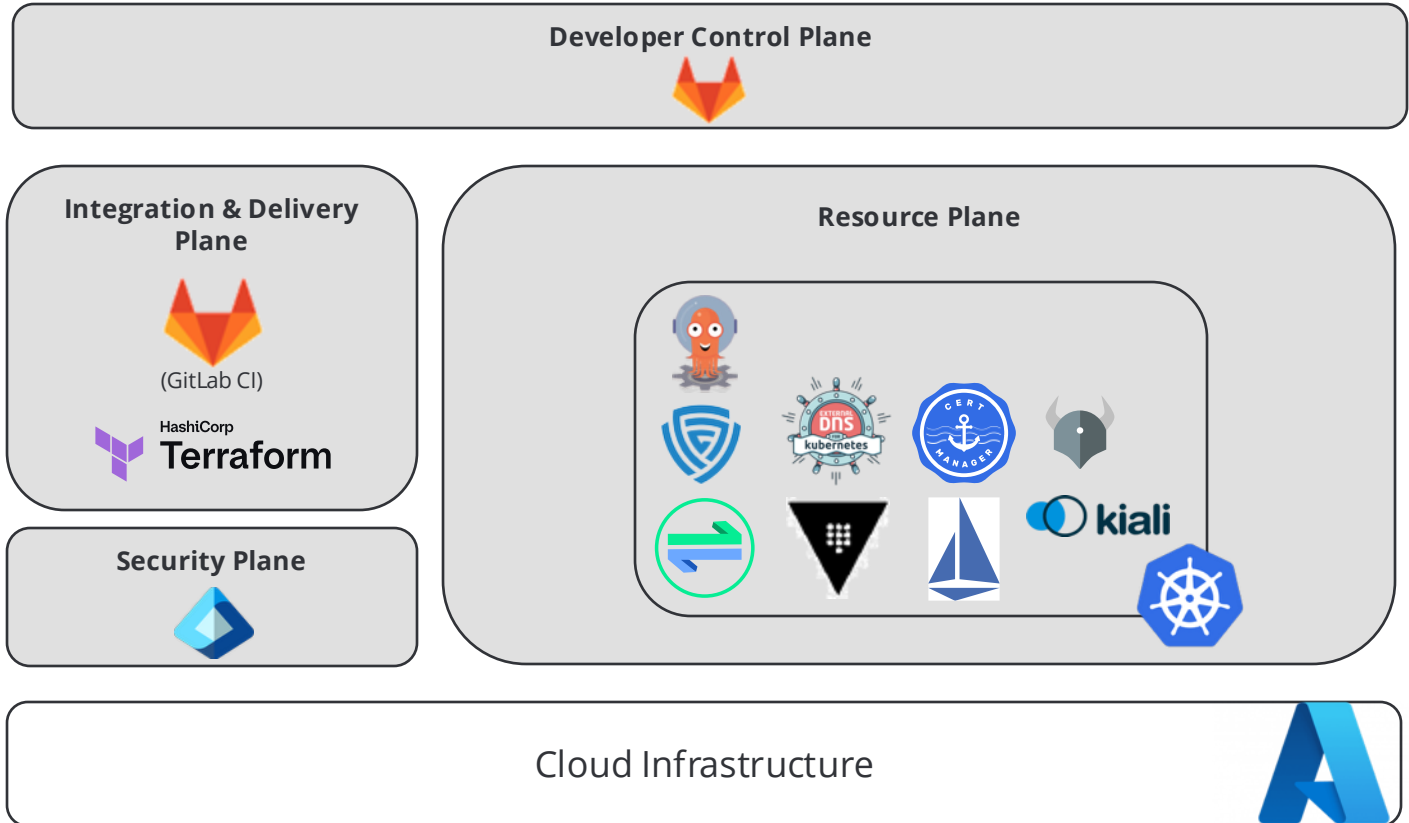
- Speed
  - Experience (through a chef)
  - Coordination (through a chef)
- Reusable Services
  - Good kitchen utensils
  - Great ingredients
- Safety
  - Respecting kitchen rules (dictated by the chef)
  - Safe kitchen equipment
- Standardization
  - Recipes
  - Kitchen processes

# BAD EXAMPLE FOR A MODERN DATA PLATFORM

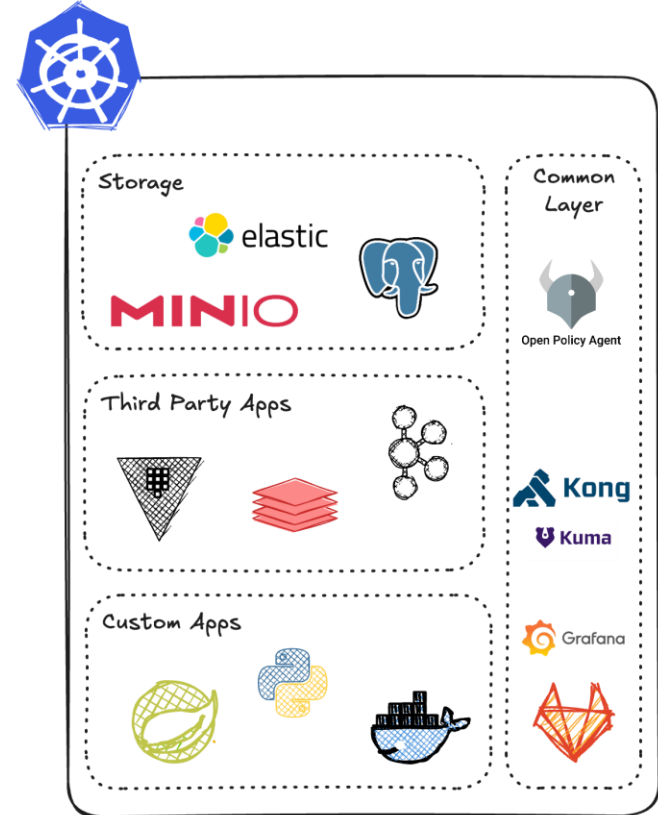


**Developer Team**

What steps are necessary to deploy a data integration with dbt?



# PLATFORMS ARE FREQUENTLY DIFFERENTIATED BY THE KIND OF APPLICATION



# HAVING MULTIPLE PLATFORMS IS LIKE COOKING EACH COURSE OF A MENU IN DIFFERENT KITCHENS



# WHAT IS ESSENTIAL?

- Organizational structures needed
  - New platform team with experts
  - High customer focus
- Cooperation between the various departments must be encouraged
  - Software
  - KI
  - Data
- Feeling that everyone is working towards a product

## 4 fundamental topologies

- Stream-aligned team
- Enabling team
- Complicated Subsystem team
- Platform team



**Platform projects mostly fail due to cultural and process reasons; technology is not the problem here.**

# A COMMON SET OF INGREDIENTS AND UTENSILS IS NECESSARY TO CREATE COORDINATED MENUS



# A MODERN KITCHEN NEEDS A DESIGN THAT ALLOW FOR



# INGREDIENTS AND STANDARDS ARE ESSENTIAL FOR DELIVERING HIGH-QUALITY, SOVEREIGN PRODUCTS

- Open-source technology is a good foundation
  - No direct dependency on a vendor
  - Not necessarily costs for licenses and support
  - Even Open-Source projects can change their licenses (Elasticsearch, Terraform, ...)
- Use of common APIs / Open standards is highly recommended
  - Connect your components using standardized APIs
  - Guarantees interoperability and flexibility
  - Ensures extensibility



**CLOUD NATIVE  
LANDSCAPE**

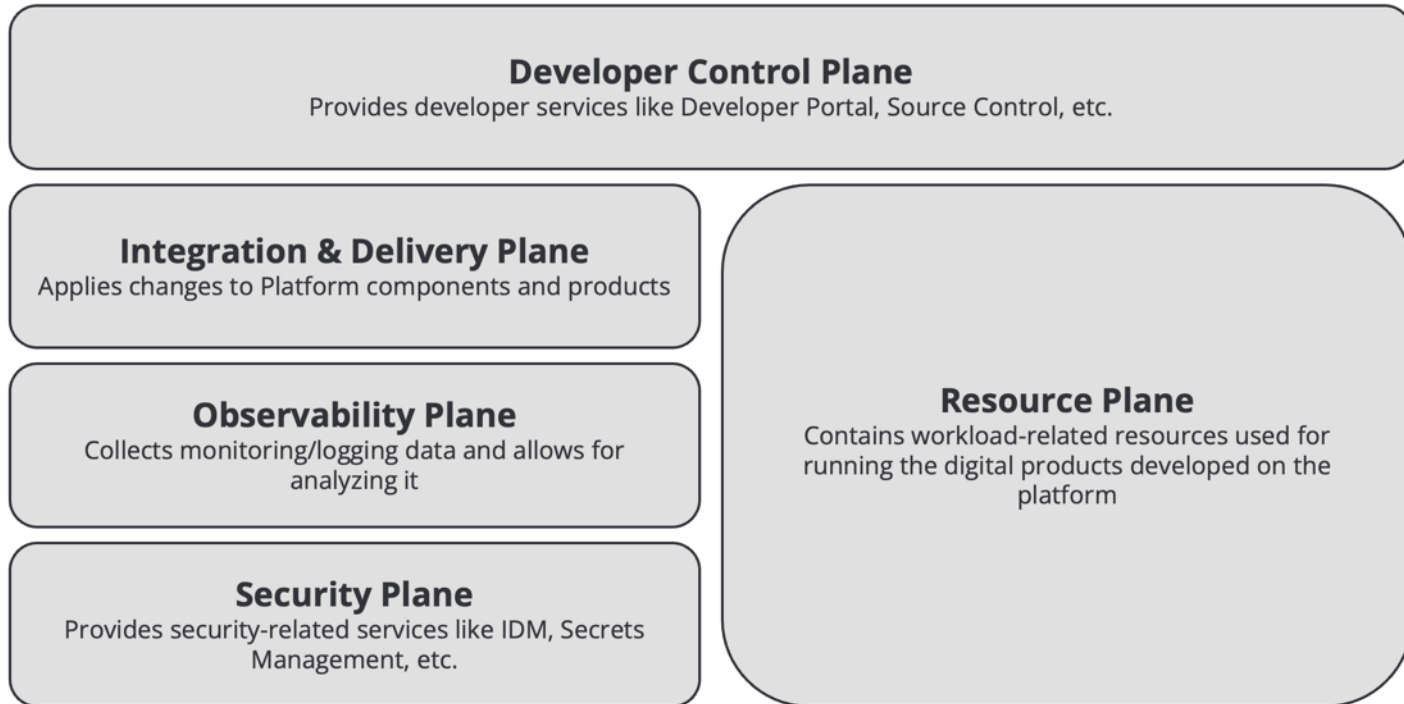


# REFERENCE BACK TO OUR MENU



- Every dish needs a good foundation
  - Basic services of a platform as fundamental ingredients
  - Security layer
- Chef decides where to shop: at the supermarket, at local weekly markets, or from their own garden (Sovereignty)
  - Know what's included in the dishes on the menu
  - Ensure that everything is compatible
- Spices and sauces determine the character
  - Data platform
  - Software platform
  - AI platform

# THINK FROM A CAPABILITY PERSPECTIVE, AND THE NEEDS YOUR TEAMS HAVE



Sources: [Platform tooling landscape](#), [Platform reference architectures](#)

# CREATE A BASIC PLATFORM TO PROVIDE SHARED TOOLS, APIS, AND PROCESSES



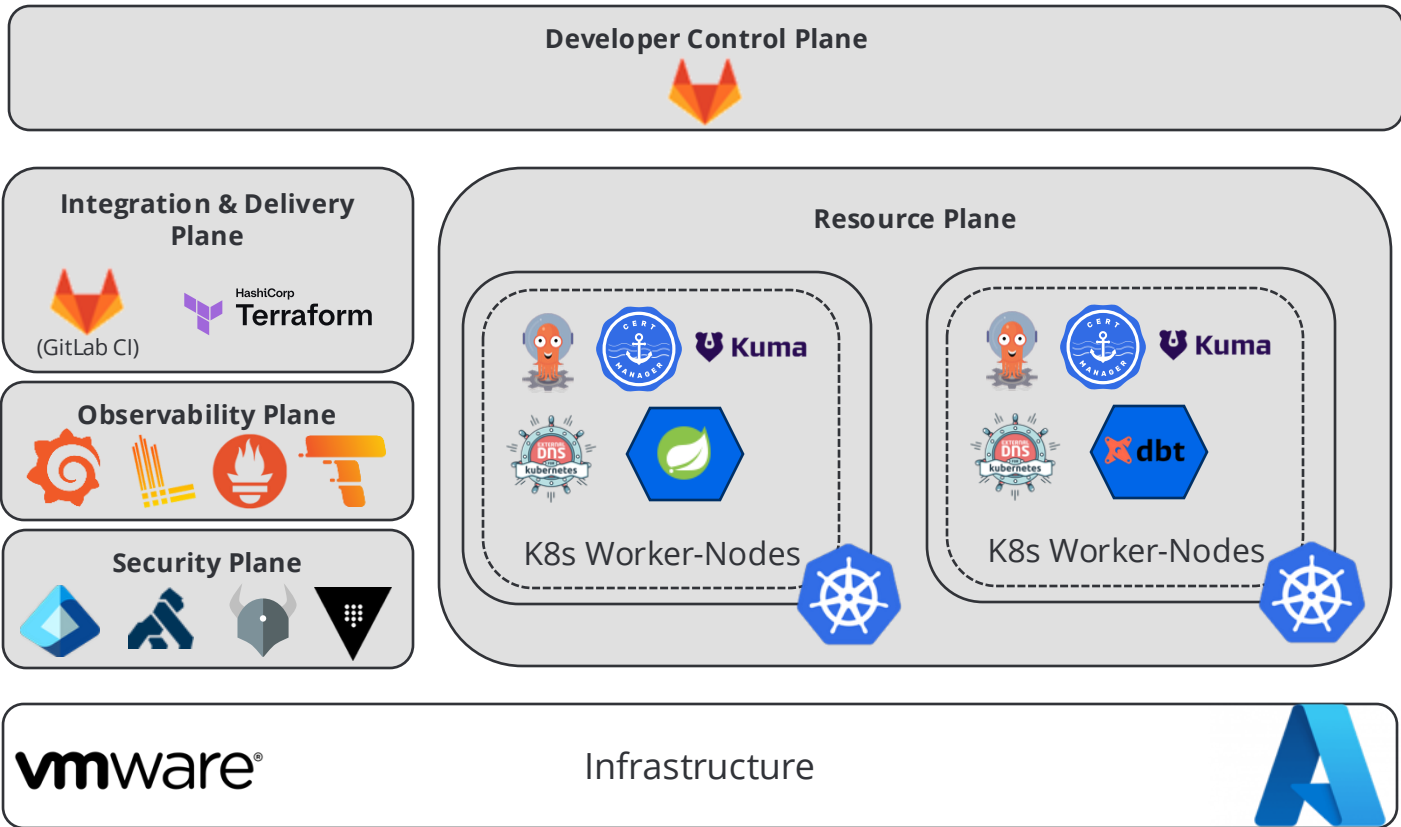
## Developer Team

It's easy to get up and running with new Java applications!



## Data Engineering Team

To create a new data integration using DBT, we can use the app template for the respective app.



**Start with the essentials, then build the rest – that's the power of a Minimum Viable Platform.**

# AT SOME POINT, IT MIGHT BECOME NECESSARY TO EXPAND THE OFFERINGS AND DEVELOP ALTERNATIVE MENUS

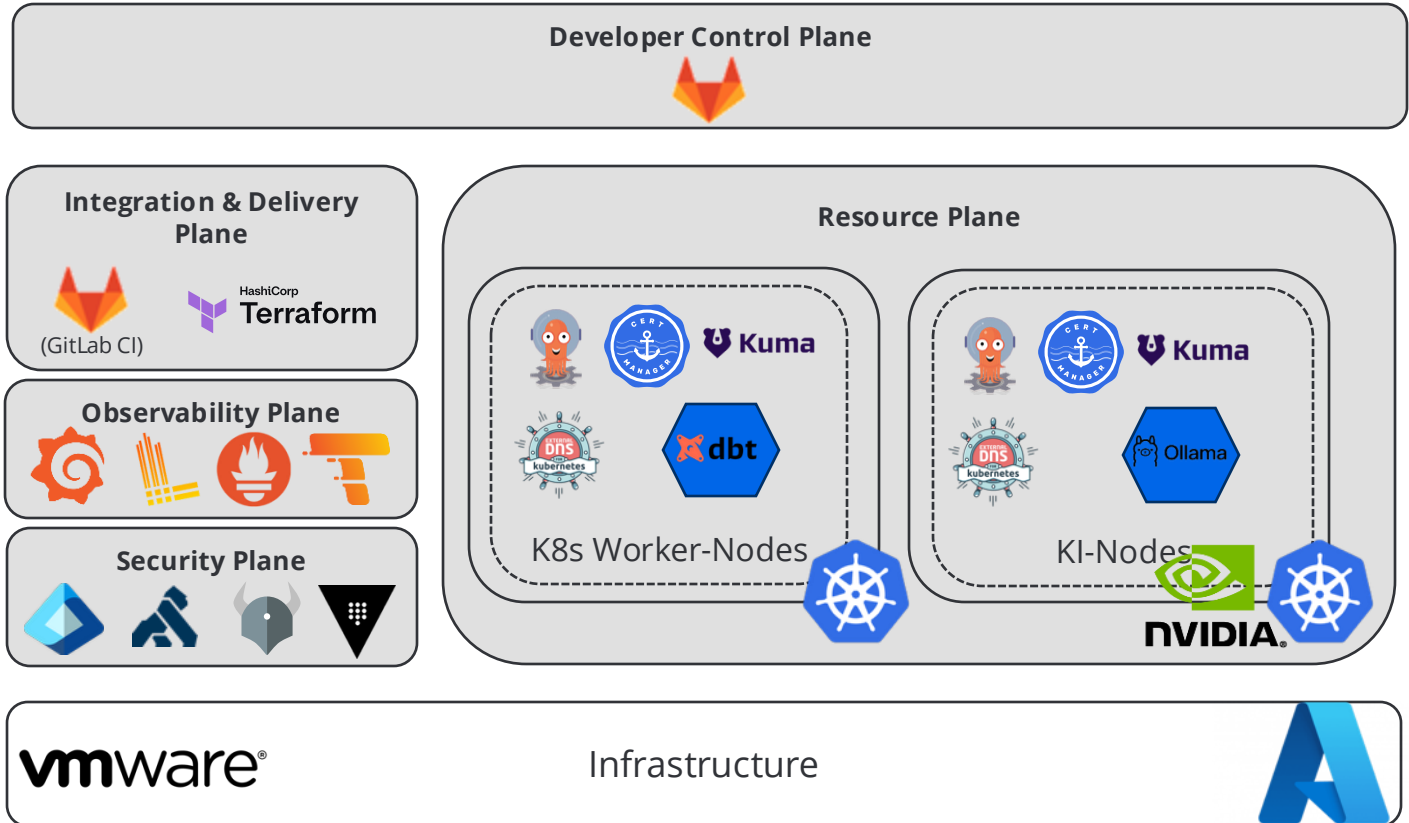


# EXAMPLE FOR A MODERN AI PLATFORM?



Developer Team

How can I easily deploy a new LLM?



# CARE ABOUT THE DESIGN TO ENSURE PLATFORM VALUE!



Source: [Luca Galante, LinkedIn](#)

# SUMMARY

- **Open Source** can be an essential building block towards sovereignty
- **Open Standards** (OCI, OpenAPI, CRDs, Otel, OIDC, ...)
- Treat your **platform as a product** with its own lifecycle
- Organizational change (**Platform team**, etc.)
- Make use of synergies (**multi tenancy**)
  - Cost control/optimization (FinOps)
  - Software, AI, Data, API, ...
- And there it is, the menu of a real Michelin-starred chef 🍷



**To create a perfectly balanced menu, we need more than just a variety of food and ingredients.**

**THE MOST IMPORTANT THING IS THAT EVERYONE ENJOYS THE MENU AND HAVE A GOOD TIME!**



Adobe Stock | #663354858

# KURZE ABFRAGE...

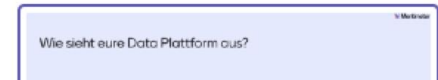


## Menti

TDWI Hot-Topic



## Choose a slide to present



# KONTAKTIEREN SIE UNS...



OPITZ CONSULTING  
**PUBLIC**



**Sven Bernhardt**  
Senior Solution Architect

[Sven.bernhardt@opitz-consulting.com](mailto:Sven.bernhardt@opitz-consulting.com)

<https://www.linkedin.com/in/sven-bernhardt-0570b823/>



**Fabian Hardt**  
Solution Architect

[Fabian.hardt@opitz-consulting.com](mailto:Fabian.hardt@opitz-consulting.com)

[www.linkedin.com/in/fabian-hardt](https://www.linkedin.com/in/fabian-hardt)



[www.opitz-consulting.com/public](http://www.opitz-consulting.com/public)





OPITZ CONSULTING  

---

**PUBLIC**

# PLATTFORM-SALAT ODER PLATTFORM-STRATEGIE? WARUM DAS ZUSAMMENSPIEL ZÄHLT

Sven Bernhardt,  
Fabian Hardt

Gummersbach, 02.12.2025



Source: Adobe Stock

01

PLATFORMS / PLATFORM  
ENGINEERING

02

KONKRETISIERUNG SW /  
ANALYTICS

03

MODULARISIERUNG /  
STANDARDISIERUNG /  
SOVERÄNITÄT

04

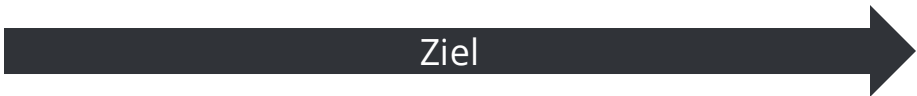
KI-  
ANKNÜPFUNGSPUNKTE

05

FAZIT (AUF ORGANISATORISCHE  
HERAUSFORDERUNGEN  
EINGEHEN)

# IS PLATFORM ENGINEERING MORE THAN IDP AND DEVEX?

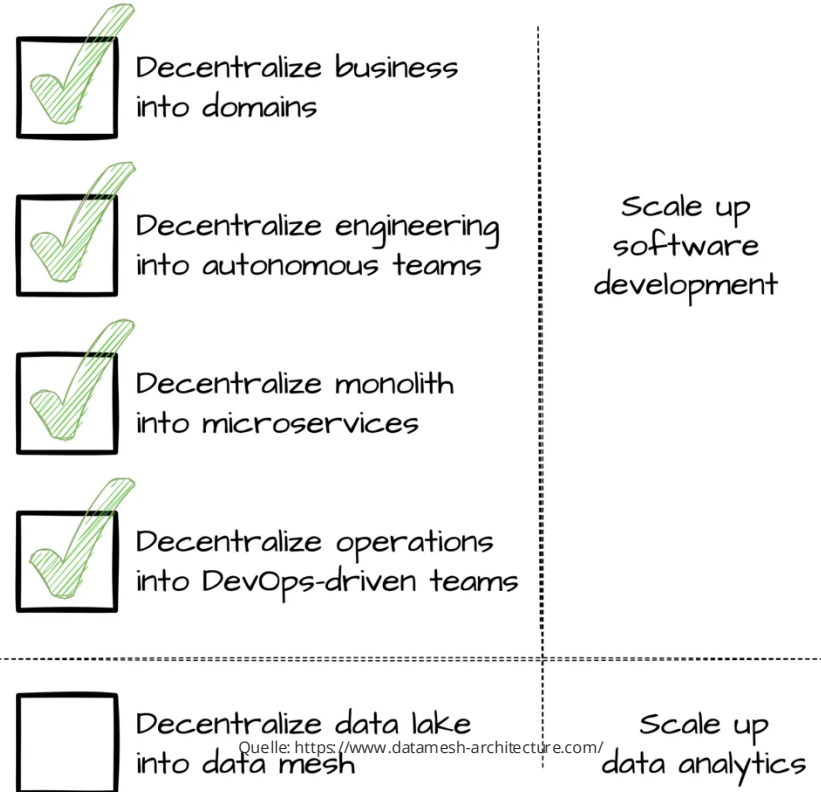
- **Platform Engineering:** Focus on infrastructure, automation & governance
- **DevEx:** Focus on user-friendliness, efficiency & developer happiness
- Platform Engineering provides the foundation – DevEx optimizes its use
- **Common Goal:** Efficient, scalable, and secure software delivery



Plattform Engineering	DevEx
Technical basis: Automation, infrastructure, security	Focus on user-friendliness and smooth workflows
Creates Standards & Self-Service-Platforms	Optimized tools & processes for developers

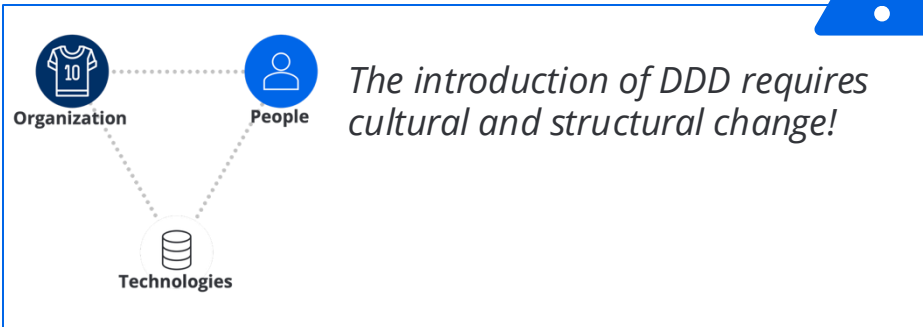
# COMMON CURRENT SITUATION

- Central data teams are too slow – they can't answer all analytical questions quickly enough.
  - Bottleneck
  - Lack of autonomy
- What's wasting time on?
  - Technical: Fix ETL jobs – by changing the source code
  - Remaining time: Understanding the business data – building new pipelines



# DATA MESH TREND / BUZZWORD

- Many approaches from Data Mesh are very good
  - Very close to DDD in software
  - Strong decentralization approaches
  - Loose coupling of systems



- Buzzword
  - It is often not very specific
  - Many talk about it, but haven't implemented it technically
  - How can decentralization be implemented without cognitive overload?
- Conclusion
  - Platform Engineering, and ultimately a good platform can help

# WHAT IS ESSENTIAL?

- Organizational structures needed
  - New platform team with experts
  - High customer focus
- Cooperation between the various departments must be encouraged
  - Software
  - KI
  - Data
- Feeling that everyone is working towards a product

## 4 fundamental topologies

- Stream-aligned team
- Enabling team
- Complicated Subsystem team
- Platform team

Team Topologies



Zum Glück ist das Leck nicht auf unserer Seite.



# SOVEREIGNTY

- Open Source technology can be a good foundation
  - No direct dependency to a vendor
  - Not necessarily costs for licenses and support
  - Even Open Source projects can change their licenses (Elasticsearch, Terraform, ...)
  
- Middleware / abstraction layer recommended
  - Connect your components through APIs
  - Makes it easier to change single components
  - Last resort: Build your own fork of a tool
    - Very expensive and time consuming



# TODAY, WE OFTEN FIND HETEROGENEOUS PLATFORMS

- Plattform salat darstellen (Eine Seite Software-Plattform, eine Seite Analytics-/Datenplattform)

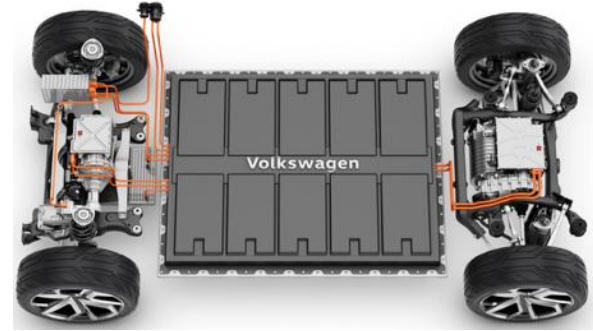
**A platform is a product  
and a foundation to build on.**

# DIFFERENT INDUSTRIES, SAME PRINCIPLE: A PLATFORM ENABLES OTHERS TO BUILD

## Consumer electronics



## Automotive



## Business / eCommerce



## IT / Development

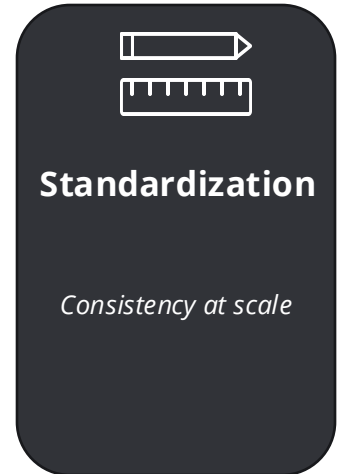
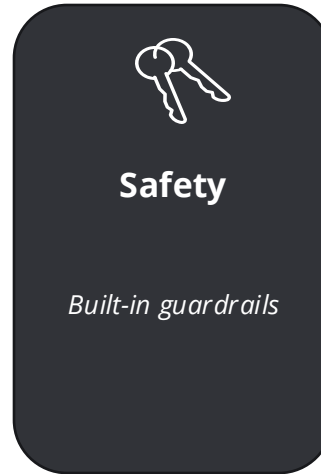
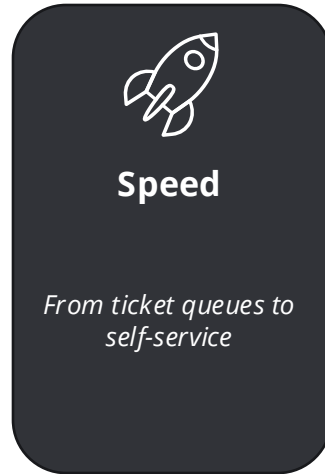
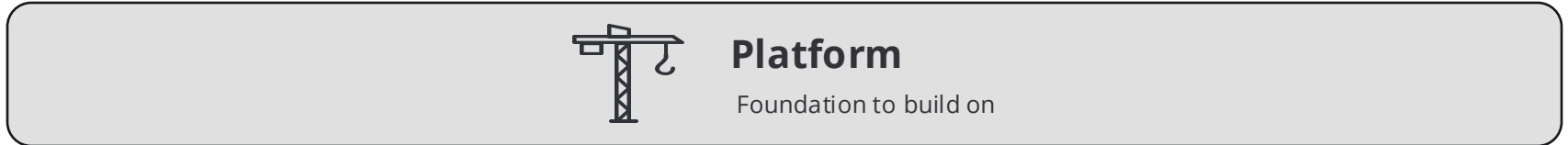


# PLATFORM DEFINITION



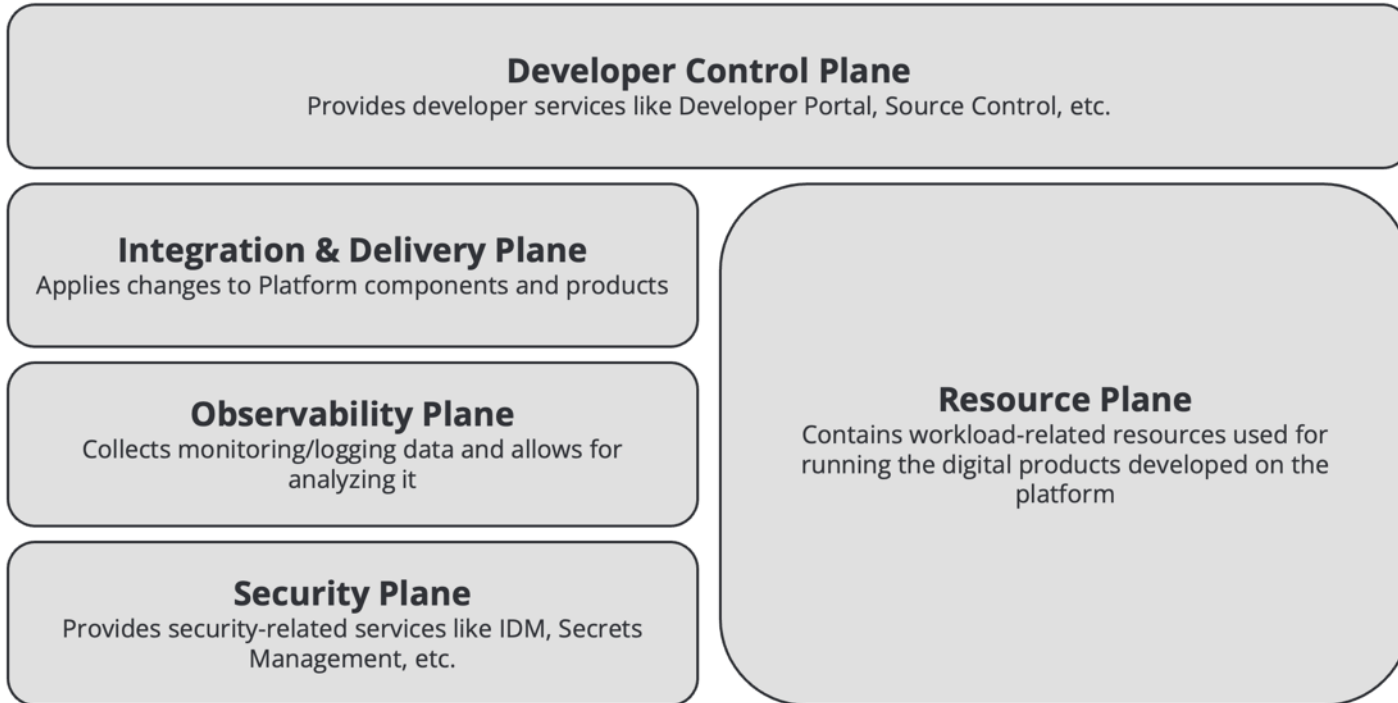
*A platform is a **product** that provides **developers** with reusable services, tools, and workflows – making their work easier, faster, and safer. It enforces **standardization** and enables **governance**, ensuring **consistency**, **compliance**, and **efficiency** across the organization.*

# WHAT MAKES A PLATFORM A PRODUCT?



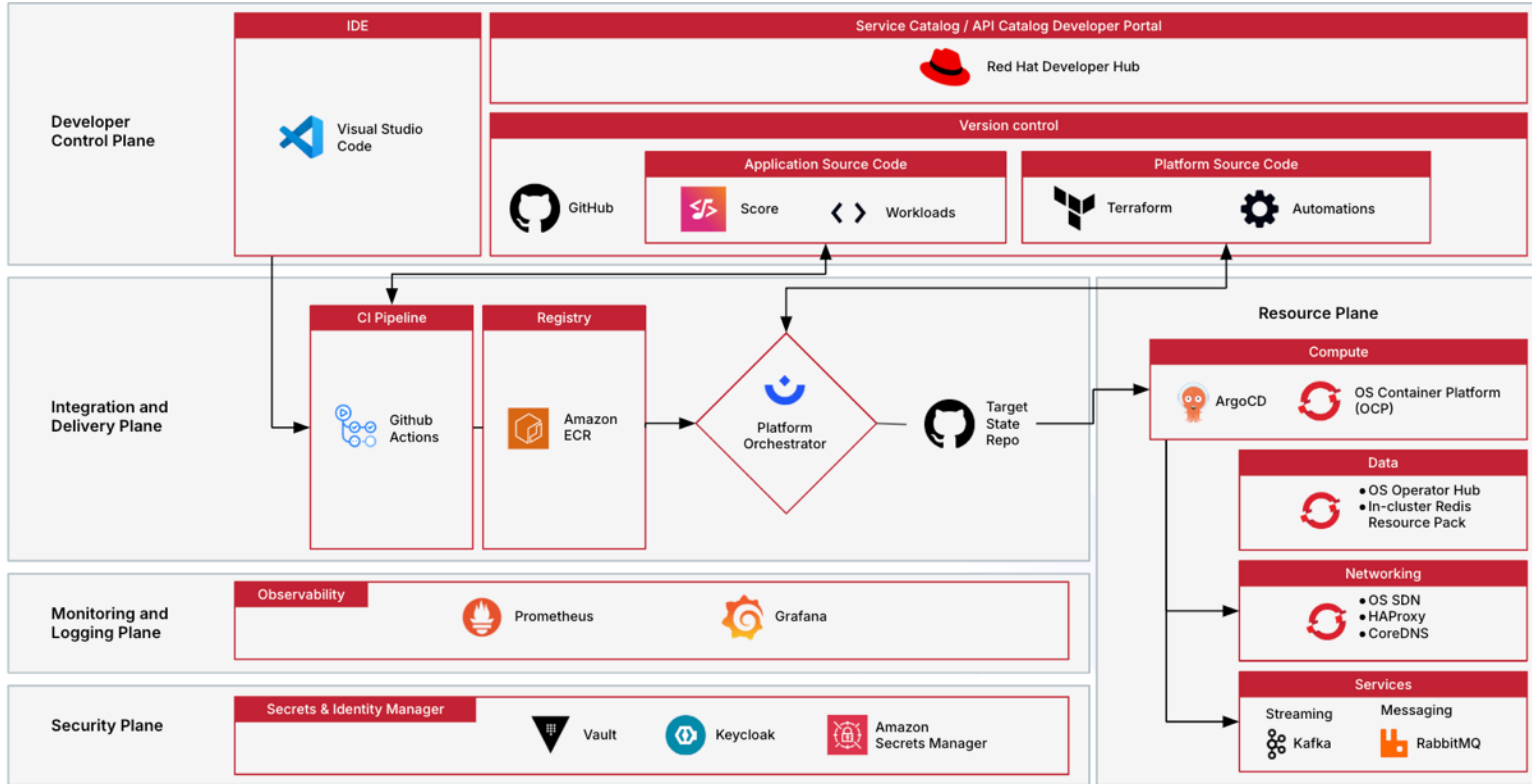
**A platform is not an end in itself –  
it's the foundation for something bigger.**

# BASE IDP BUILDING BLOCKS



Sources: [Platform tooling landscape](#), [Platform reference architectures](#)

# EXAMPLE IDP ARCHITECTURE ON OPENSHIFT



Source: [platformengineering.org](https://platformengineering.org)

**A platform is a product for developers  
that must provide value to an  
organization.**

A photograph of a stack of light-colored wooden blocks. The stack is tall and narrow, with several blocks at the top appearing to be in the process of falling or having just fallen, creating a sense of instability and failure. The background is a plain, light gray.

## **REASONS WHY PLATFORM PROJECTS FAIL**

Source: Adobe Stock



# MISSING PLATFORM OWNERSHIP



Source: Adobe Stock

# INSUFFICIENT OR MISSING BUDGET



Source: Adobe Stock

# DON'T TALK, HEAR OR LISTEN: BUILD A PLATFORM ON BEST GUESS



Source: Adobe Stock

# UNNECESSARY COMPLEXITY



Source: Adobe Stock

# CUSTOMER EXAMPLE: DATA PLATFORM



**Developer Team**

What steps are necessary to deploy a data integration with dbt?

Developer Control Plane



Integration & Delivery Plane



(GitLab CI)




HashiCorp  
**Terraform**

Security Plane



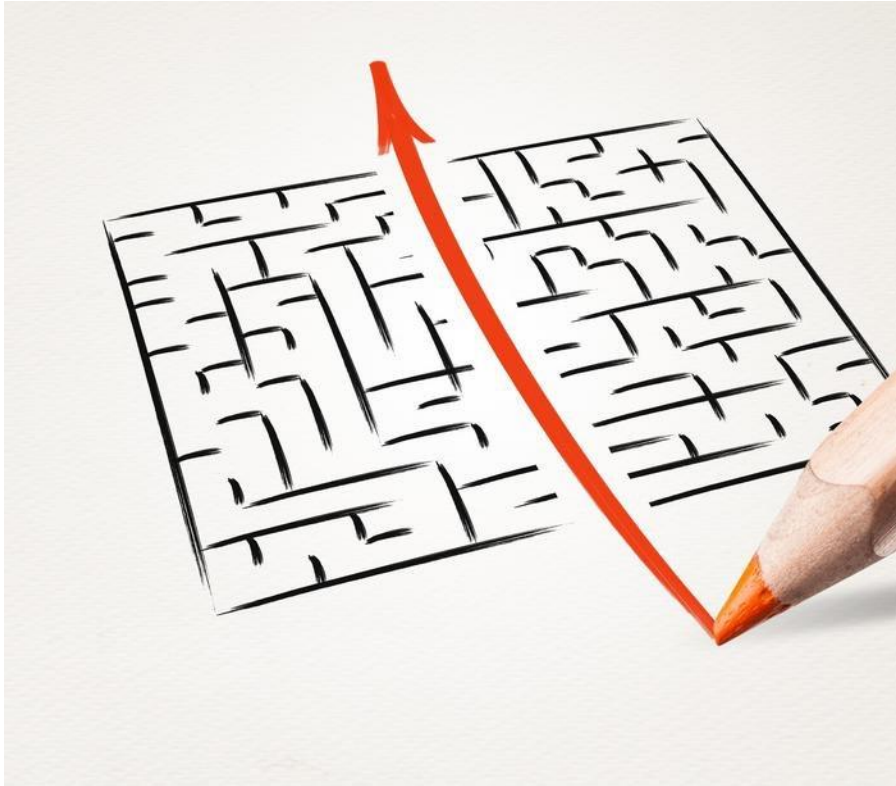
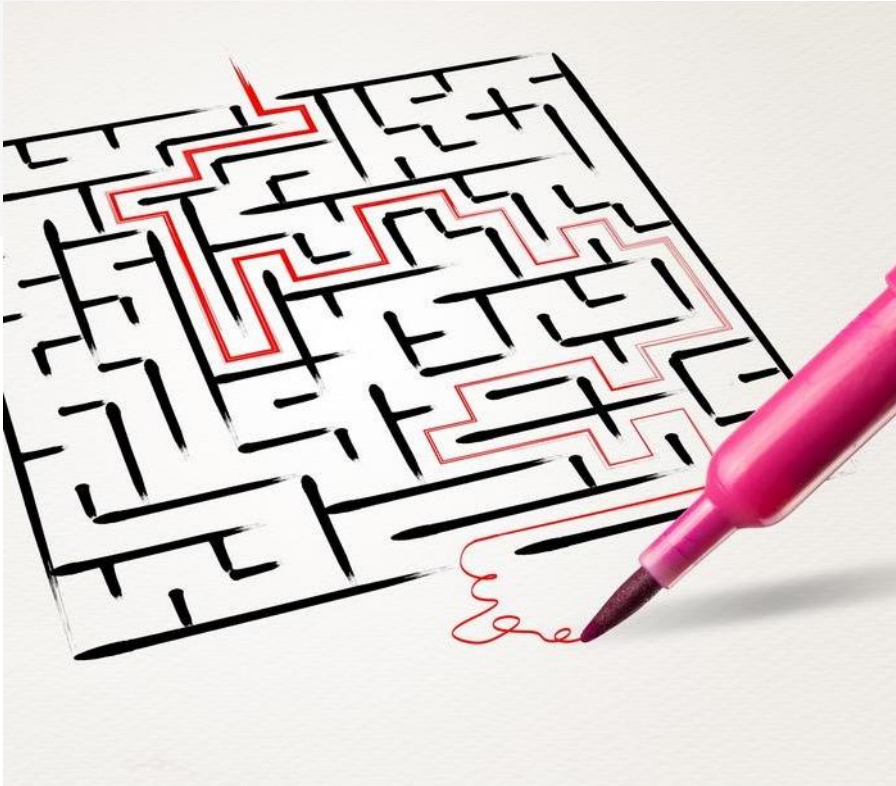
Resource Plane



Cloud Infrastructure



# MISSING GOLDEN PATHS



# GOLDEN PATH: EXPECTATIONS VS. COMMON REALITY



Source: Gregor Hohpe, Platform Strategy

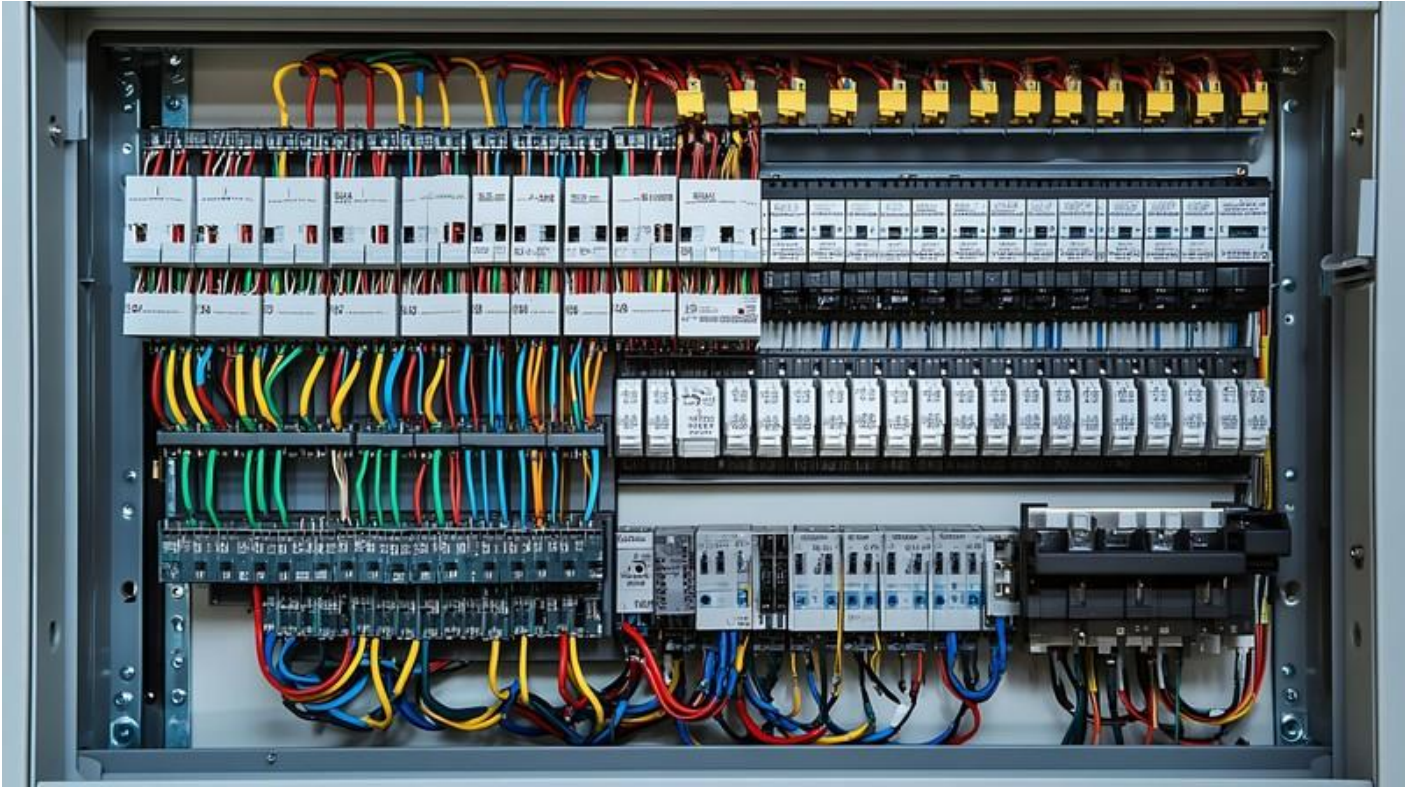
# FAILED PRODUCTIFICATION / TICKET OPS



# FRONTEND-FIRST PLATFORM DESIGN



# INSUFFICIENT ABSTRACTION



Source: Adobe Stock

# CUSTOMER EXAMPLE: INTERNAL DEVELOPER PLATFORM



## Senior Management

How can I optimize the platform in terms of costs?

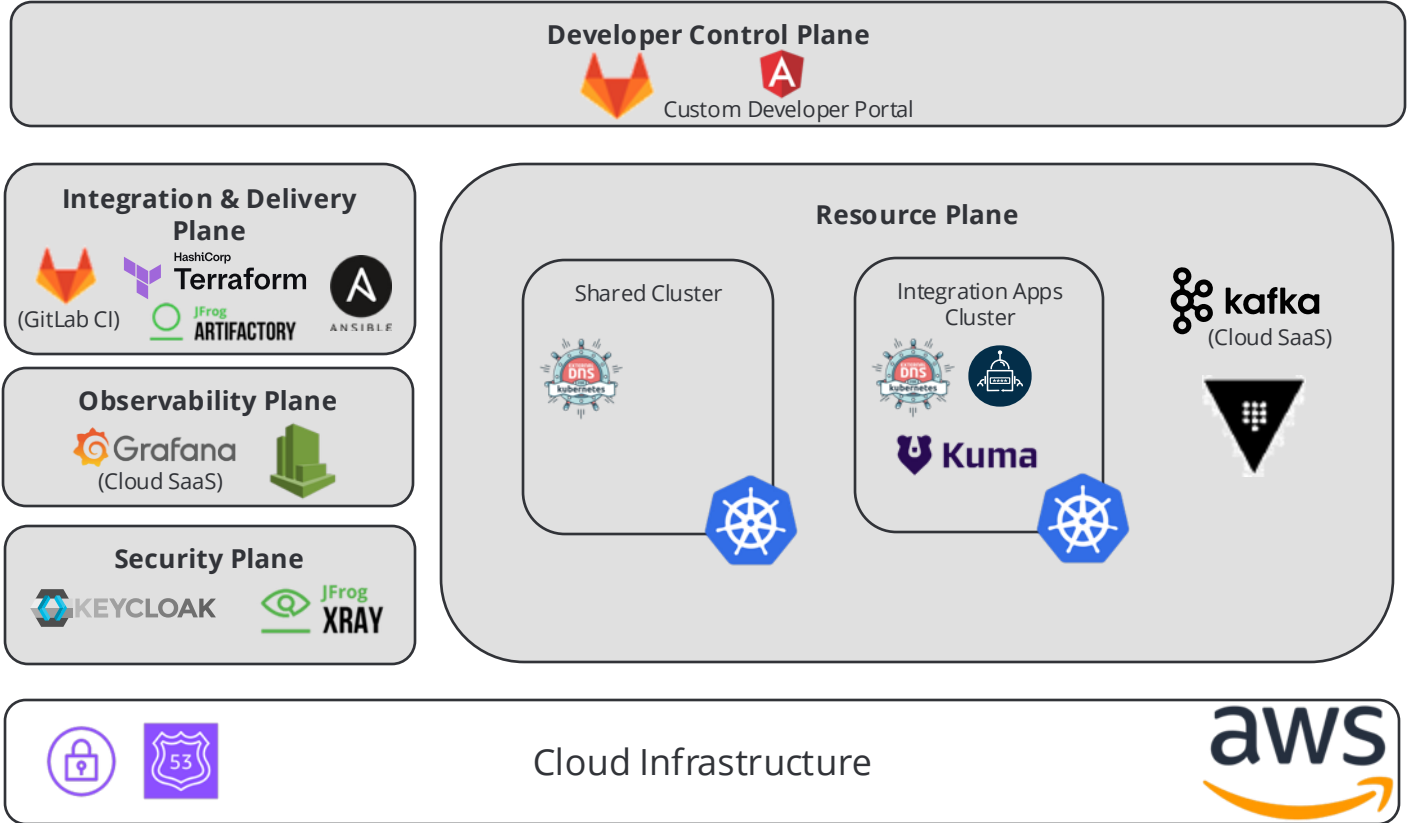
What is the ROI of the platform?



## Developer Team

How can I monitor my app in the shared cluster?

Why am I restricted from registering a privileged GitLab runner for integration testing?



# NO CONSISTENT OBSERVABILITY APPROACH



Source: Adobe Stock

# NO COST CONTROL / MISSING FINOPS APPROACH



**Platform projects mostly fail due to cultural and process reasons; technology is not the problem here.**

# STOP PROCRASTINATING: KEEP IT SIMPLE, STUPID!

- Keep in mind: **Platform = Product**
- Create an architecture vision
- Start with a single, simple use case and evolve from there
- Set up the Product marketing
- Identify your internal advocates
- Use the **Minimum Viable Platform** approach



Source: Adobe Stock

# WHAT MAKES A MINIMUM VIABLE PLATFORM?

- Representative
- Repeatable
- Iterative
- Lightweight
- Value-driven







Source: Adobe Stock

**Start with the essentials, then build the rest – that's the power of a Minimum Viable Platform.**

# EMBRACE THE NEEDED ORGANIZATIONAL CHANGE

## 4 fundamental topologies

-  Stream-aligned team
-  Enabling team
-  Complicated Subsystem team
-  Platform team



Source: "Team Topologies", Matthew Skelton & Manuel Pais

**The Platform team needs to treat the Developer platform as a product and the developers as customers.**

# CUSTOMER EXAMPLE: API PLATFORM



Security Dept

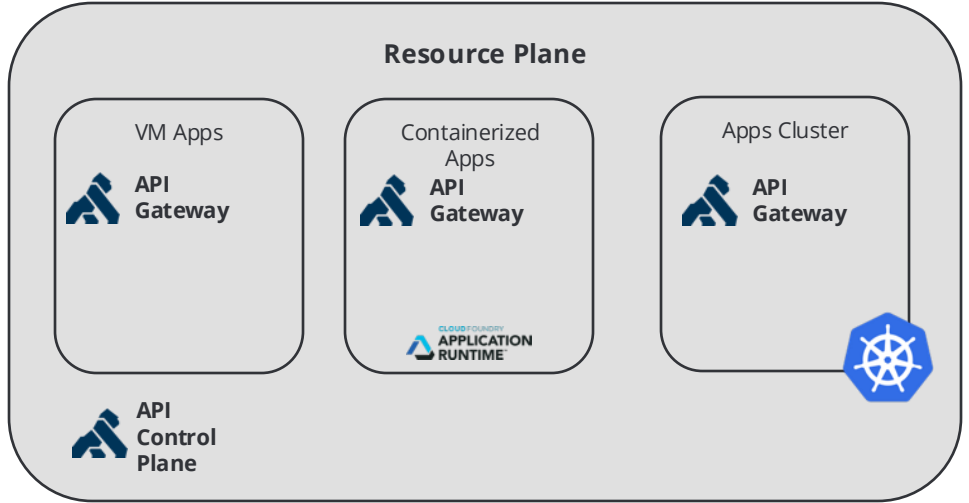
We can centrally declare & enforce security policies!



Developer Team

My app is automatically exposed and secured by the API Gateway in my target environment!

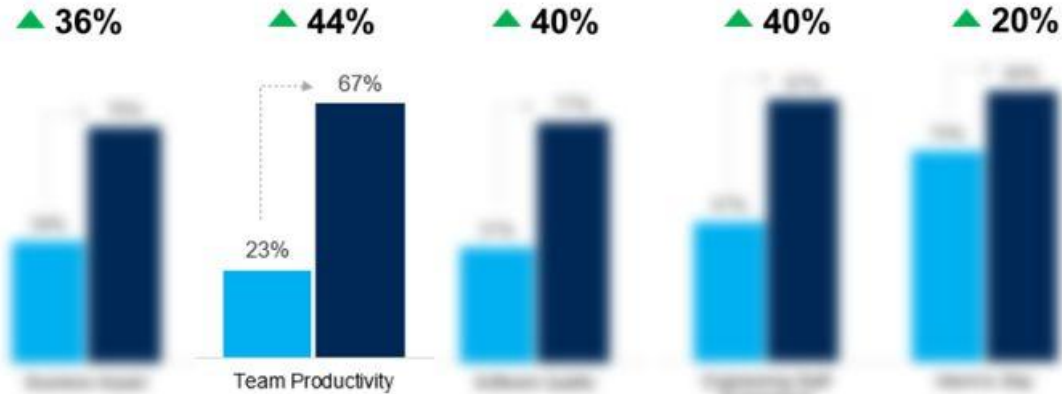
We don't need specific new skills!



# PLATFORM ENGINEERING HELPS TO FURTHER DEVEX

## Impact of a High-Quality Developer Experience

- Software Engineering Staff with a High-Quality Developer Experience
- All Other Software Engineering Staff



n= 5,112 Software Engineering Staff from 51 Organizations

Source: Gartner Developer Experience Assessment Survey

Note: A High-Quality Developer Experience is where Software Development Team Members have an average satisfaction rating of "satisfied" or "extremely satisfied", on average, across all 21 attributes of the developer experience.

© 2025 Gartner, Inc. and/or its affiliates. All rights reserved.

Gartner

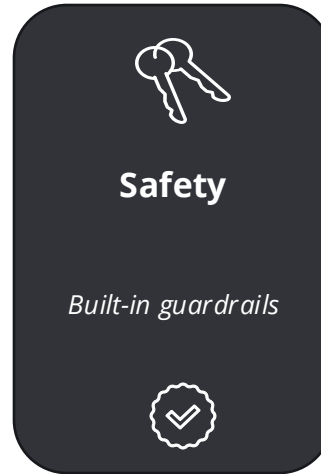
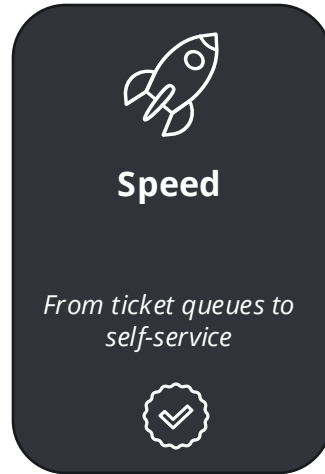
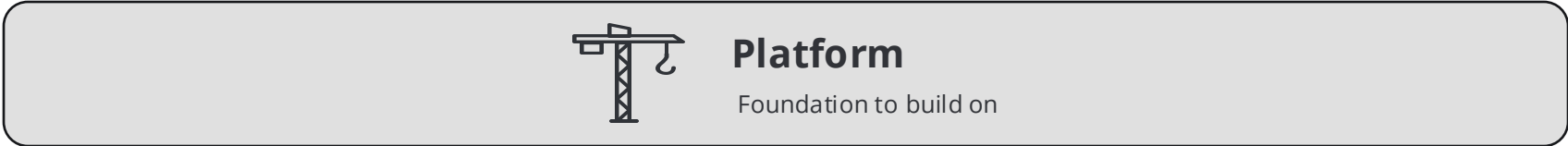
# CARE ABOUT THE DESIGN TO ENSURE PLATFORM VALUE!



Source: [Luca Galante, LinkedIn](#)

**Platform Engineering is not a self-purpose  
– it's a competitive advantage when done  
right.**

# DEVELOPER PLATFORM AS A FOUNDATION FOR SOMETHING BIGGER





Source: Adobe Stock



**A platform is a product for developers  
with speed, safety, and standardization  
built in.**