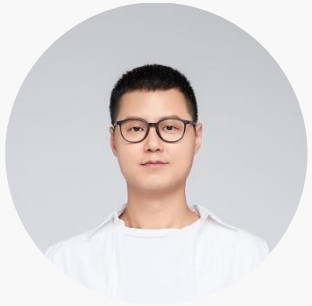
TOP1

主办方 msup

冉冉升起的 平台工程
真的能带来 10x 效能提升吗？

杨振涛 vivo互联网研发总监

讲师简介



杨振涛
Chris (Gentle) Yang
— e/acc

PECommunity.cn
平台工程社区发起人
vivo 互联网研发总监

当前专注于：

研发组织管理
工程师体验与工程文化建设
企业开源治理 OSPO

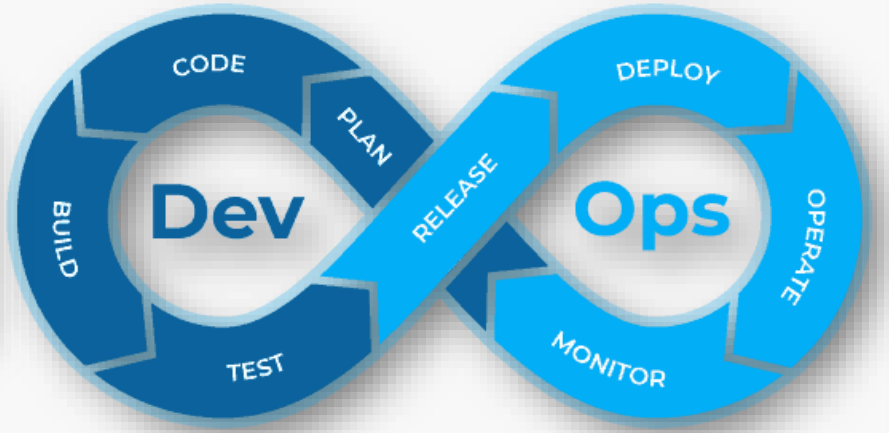
- 从 Jenkins 时代起关注和实践 CI/CD 与 DevOps，到云原生时代关注平台工程等最新实践，在软件生产力的道路上持续耕耘，并提出 **EngEx** 即“工程师体验”概念。
- 期望继续探索 AIGC 时代的软件生产力以及工程师技能要求与开发者体验。

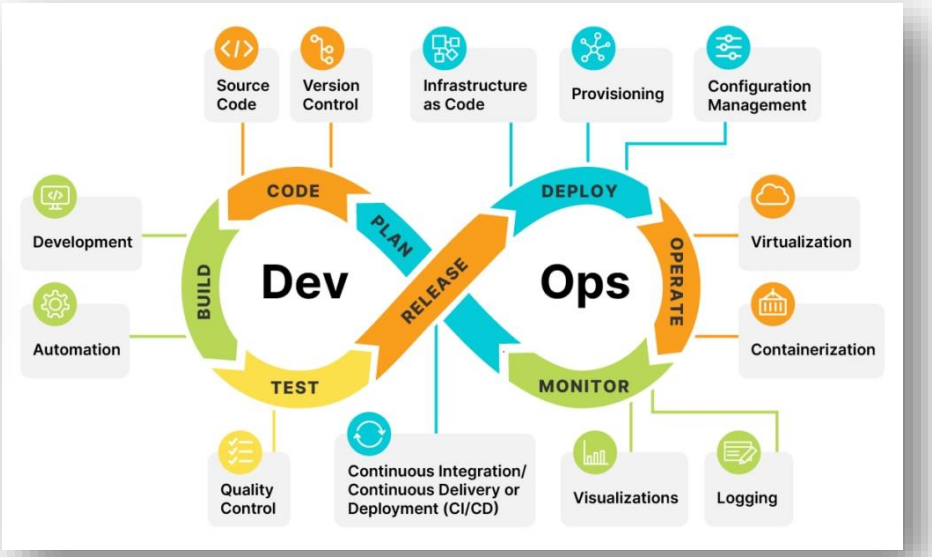
- @CNCF app delivery TAG
- @TODO Group
- TED Translator/Reviewer

内容目录

软件是怎么发布的?

SHELL





~ 2004
Hudson

~2006
AWS

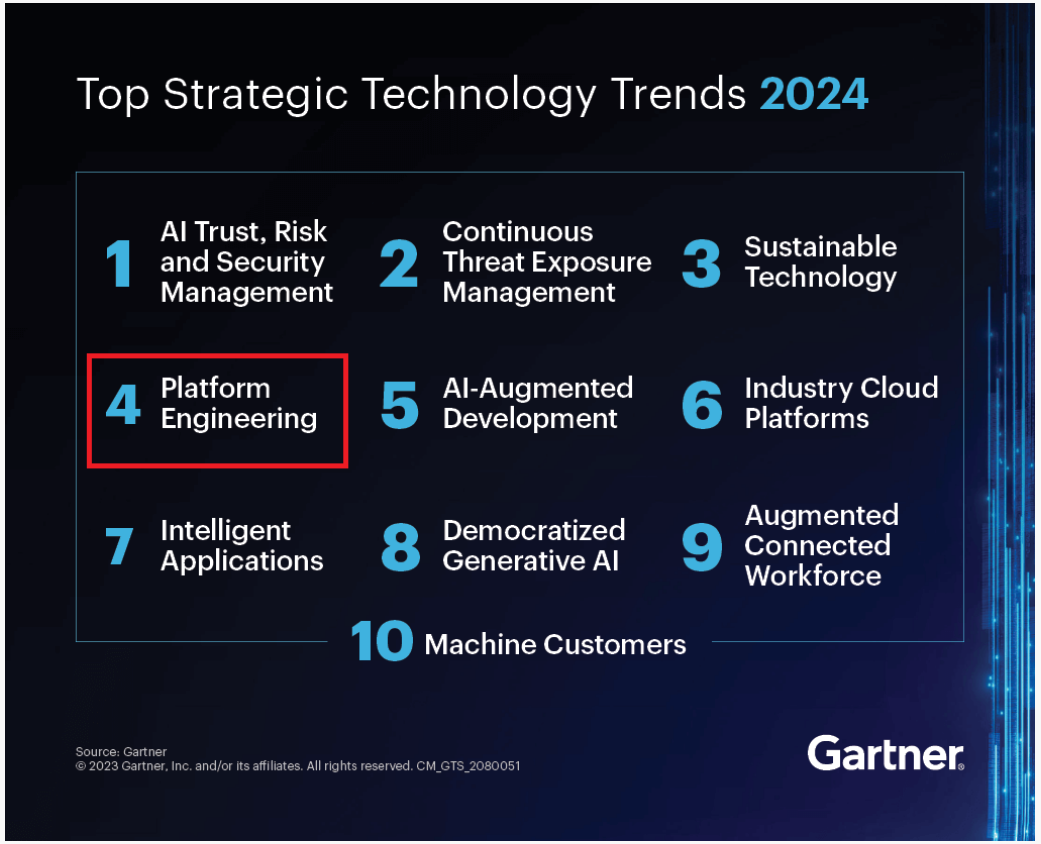
~2010
DevOps

~ 2015-2019
CNCF

~2017
GitOps

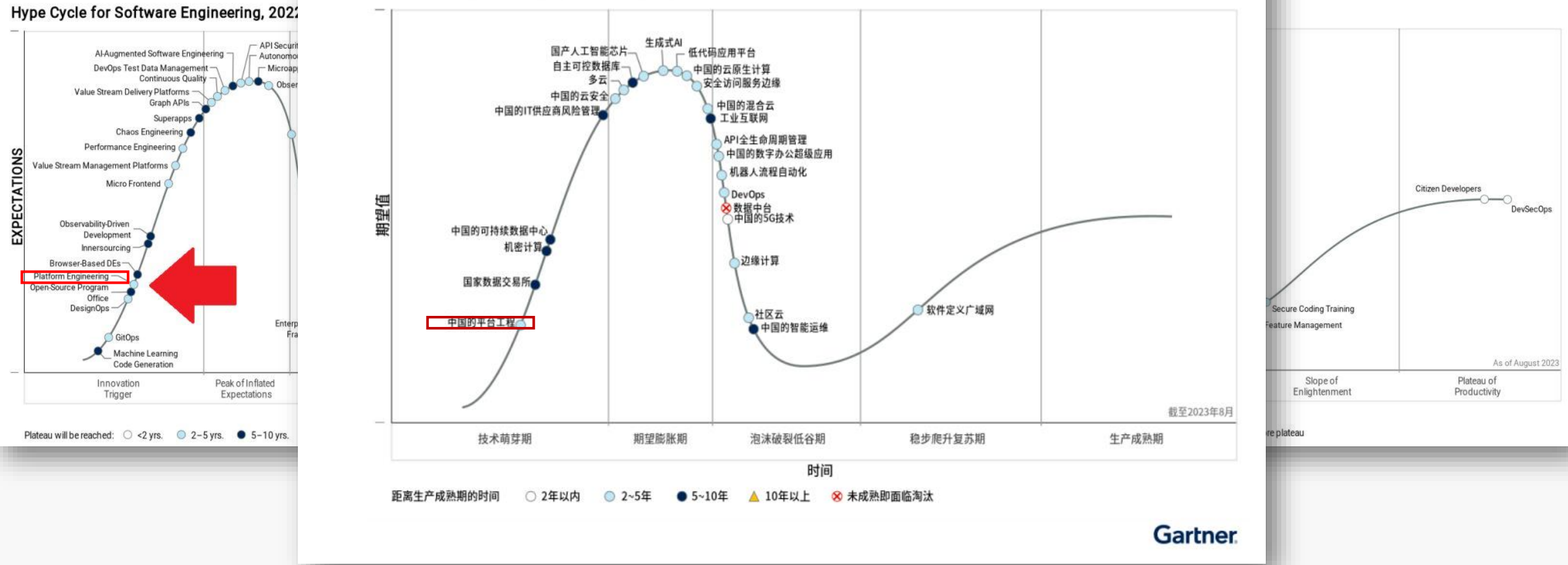
~2022
Platform
Engineering

https://github.com/PECommunity/awesome-platform-engineering



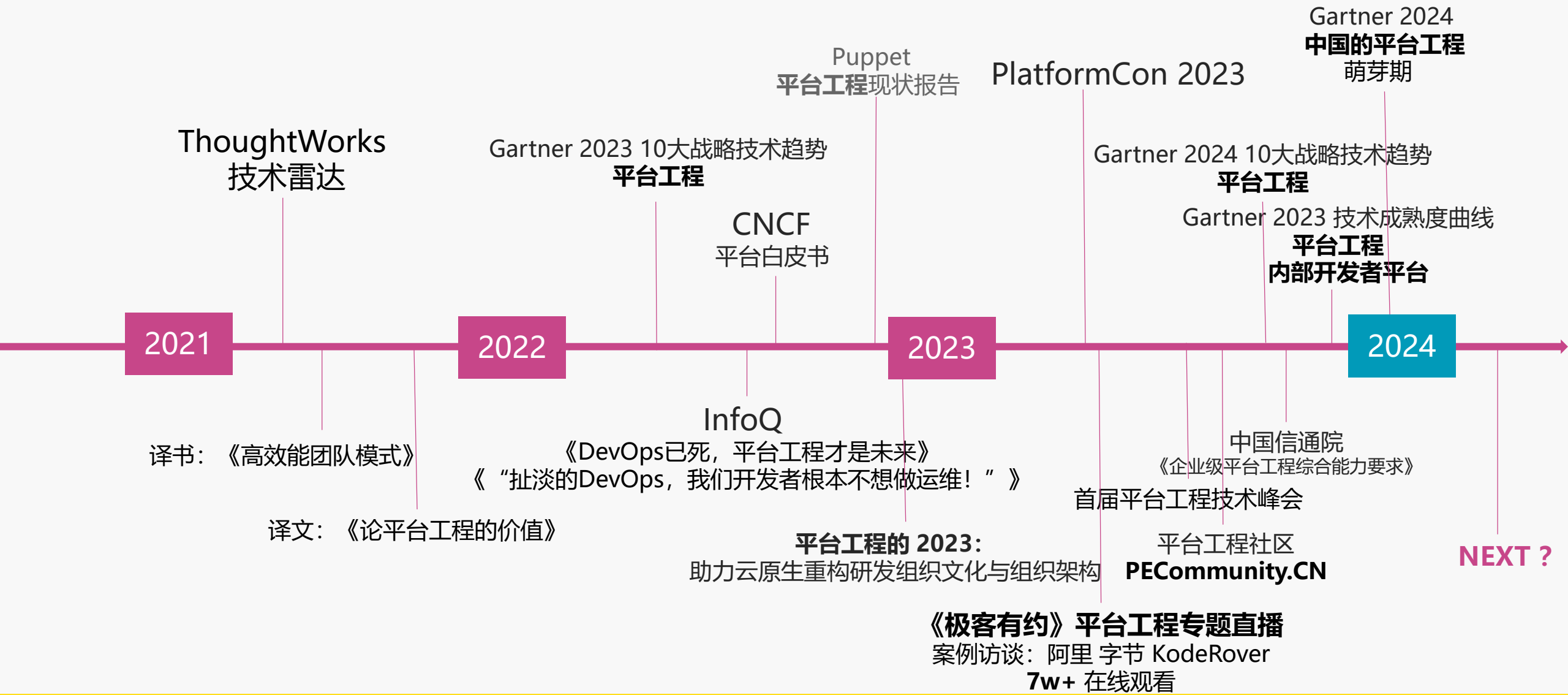
https://www.gartner.com/en/articles/gartner-top-10-strategic-technology-trends-for-2024
https://www.gartner.com/en/articles/gartner-top-10-strategic-technology-trends-for-2023

图1：2023年中国信息与通信技术成熟度曲线



https://mp.weixin.qq.com/s/zxw3UFb0oy3vKTou2ZaTJg

01 中国的平台工程故事



平台工程
Platform Engineering

PECommunity.cn

WHAT

- 平台工程是一门**设计、构建、维护和改进软件开发工具链和工作流程**的学科，可在云原生时代通过全面一致的工具和流程为软件工程组织提供自助服务功能。
- 平台工程提供了以“**内部开发者平台**”为代表的集成产品，涵盖了应用程序全生命周期的操作需求。

HOW

- 内部开发者平台/门户 IDP
- 平台即产品 PaaP
- 自服务 Self-service
- IaC & DCM & GitOps

WHY

- 软件系统复杂度
  - 业务复杂度
  - 技术复杂度
  - 架构复杂度
  - 运行态复杂度
- 开发人员认知负荷
- 规模化 DevOps 难题

Measure

- 软件生产力
  - 业务开发团队的交付速率与质量
  - 以DORA为代表的各种模型和框架
- 开发者体验
  - DevEx
- 平台自身
  - 平台能力 & 平台成熟度

Glossary

- Platform 平台
- Platform engineering 平台工程
- Platform consumer 平台客户
- Internal developer platform 内部开发者平台 IDP
- Platform as a Production 平台即产品 PaaP
- Golden Paths 黄金路径
- Thinnest viable platform 最薄可用平台 TVP
- Team Topologies 团队拓扑
- Cognitive Load 认知负荷
- Architecture Decision Record 架构决策日志 ADR
- Open Decision Framework 开放决策框架 ODF
- Jobs to be done 焦糖布丁理论/待办任务 JTBD
- Flywheel effect 飞轮效应
- Value Stream Mapping 价值流图 VSM
- Second system effect 第二系统效应
- KPI
- OKR
- SLO/SLI/SLA
- MVP
- POC
- NPS/CSAT/CES
- DevOps
- SRE

https://github.com/PECommunity/platform-engineering-glossary

02 平台工程火热背后的问题与挑战

对现状不满

规模化DevOps的复杂度和低成功率
各种“左移”为何移不动?

开发者面临越来越高的认知负荷
CNCF全景图有超1000+卡片

“谁开发，谁运维”？做不到!
系统控制权争夺战 Dev&Ops摩擦

有更高要求

研发团队之间如何协作
才能更少摩擦、更多丝滑，
从而获得研发效能和开发者体验的双赢

平台及平台团队期望
更高的用户满意度和NPS

软件系统的控制权：提供开发者控制面

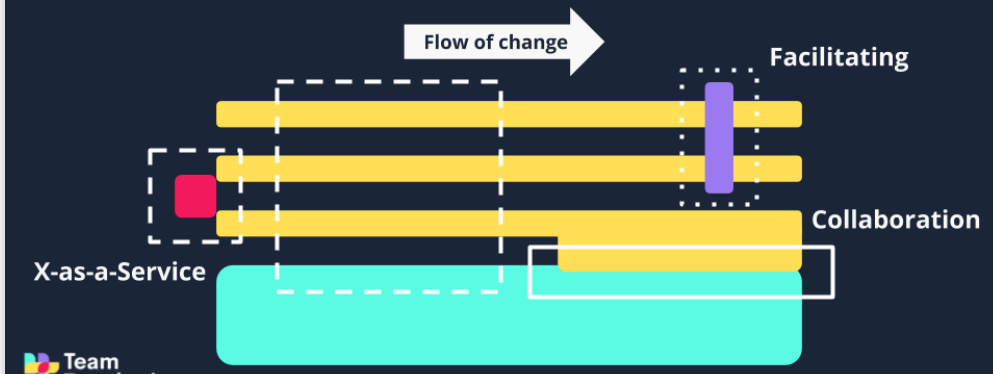
破题思路

4 fundamental topologies

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

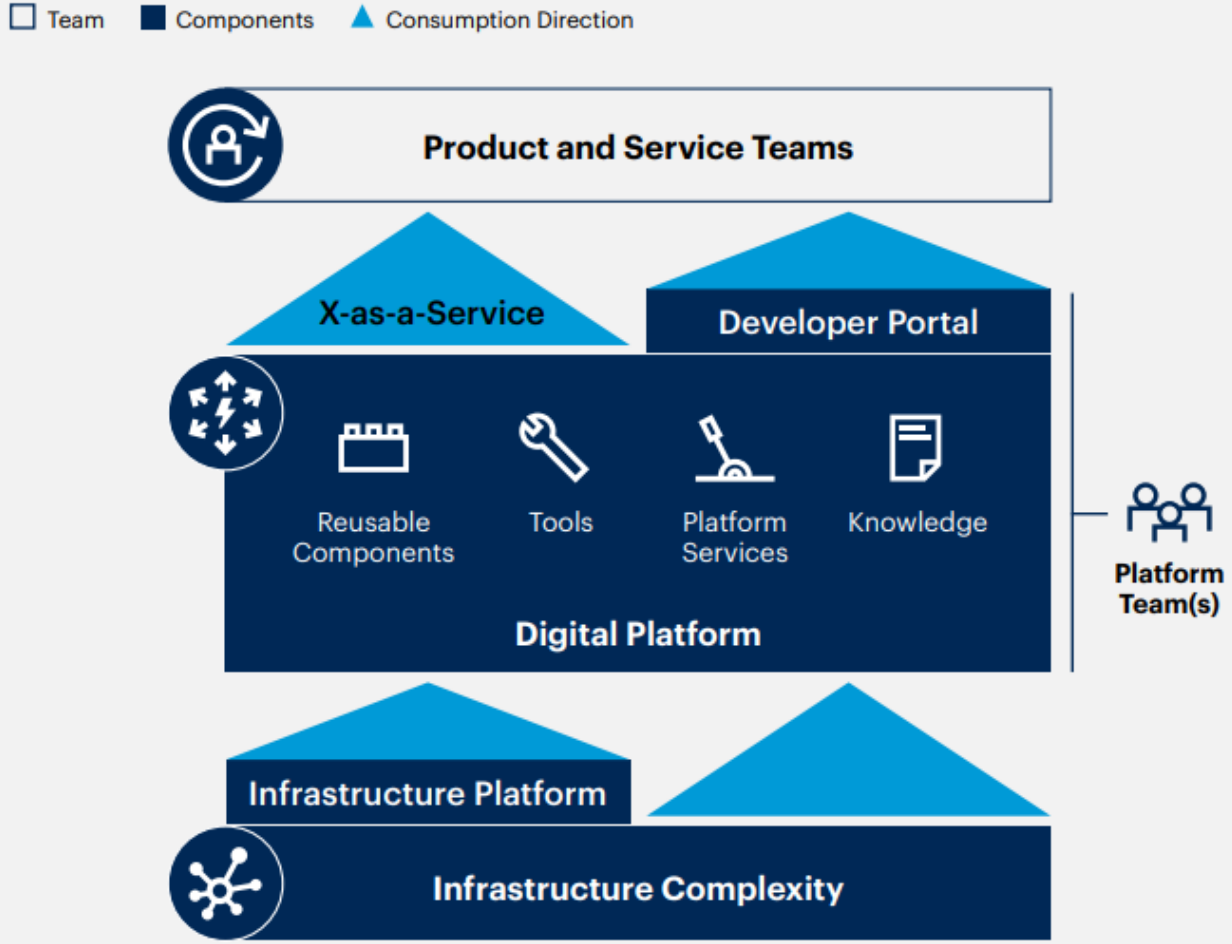
Team Topologies

3 core interaction modes



Team Topologies

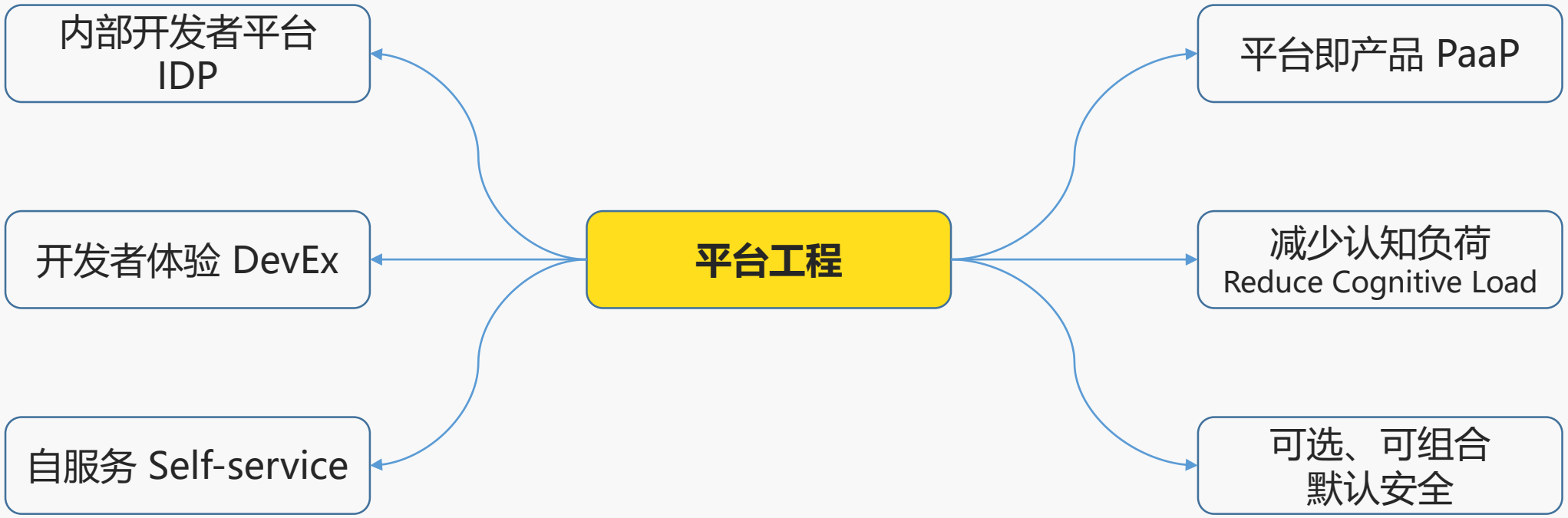
Diagram of Platform Engineering



Source: Gartner

03 平台工程的新思路与关键实践

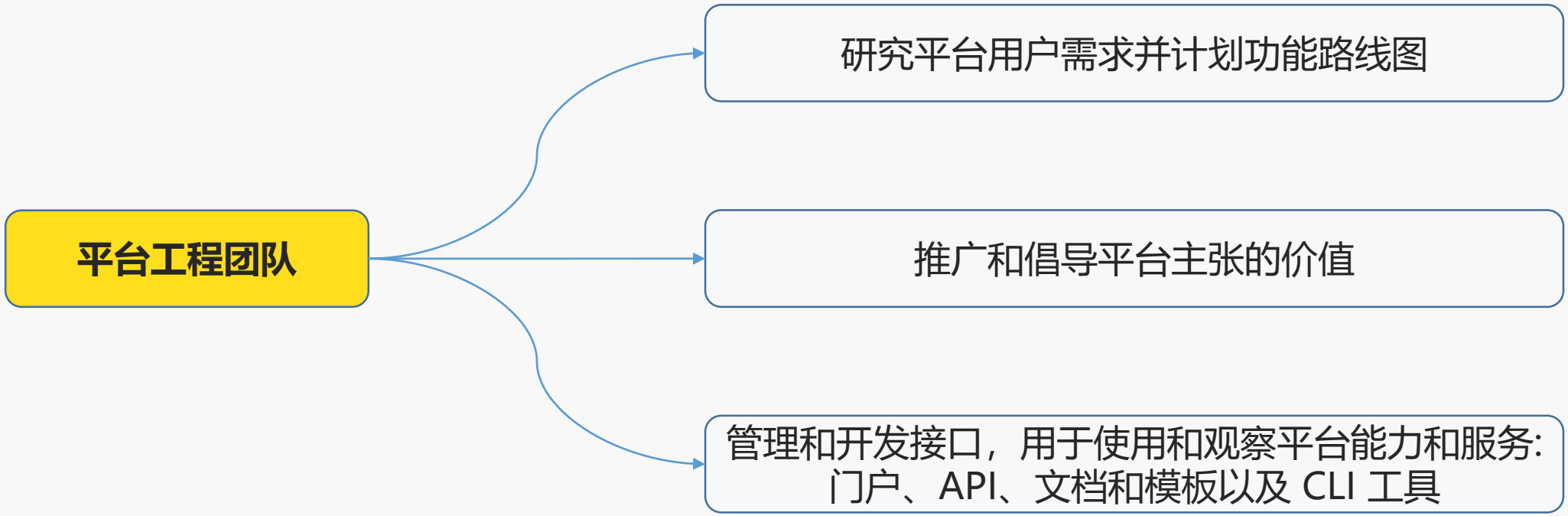
- CNCF app delivery TAG – Platform WG 平台白皮书：***平台工程的关键属性***



https://tag-app-delivery.cncf.io/zh

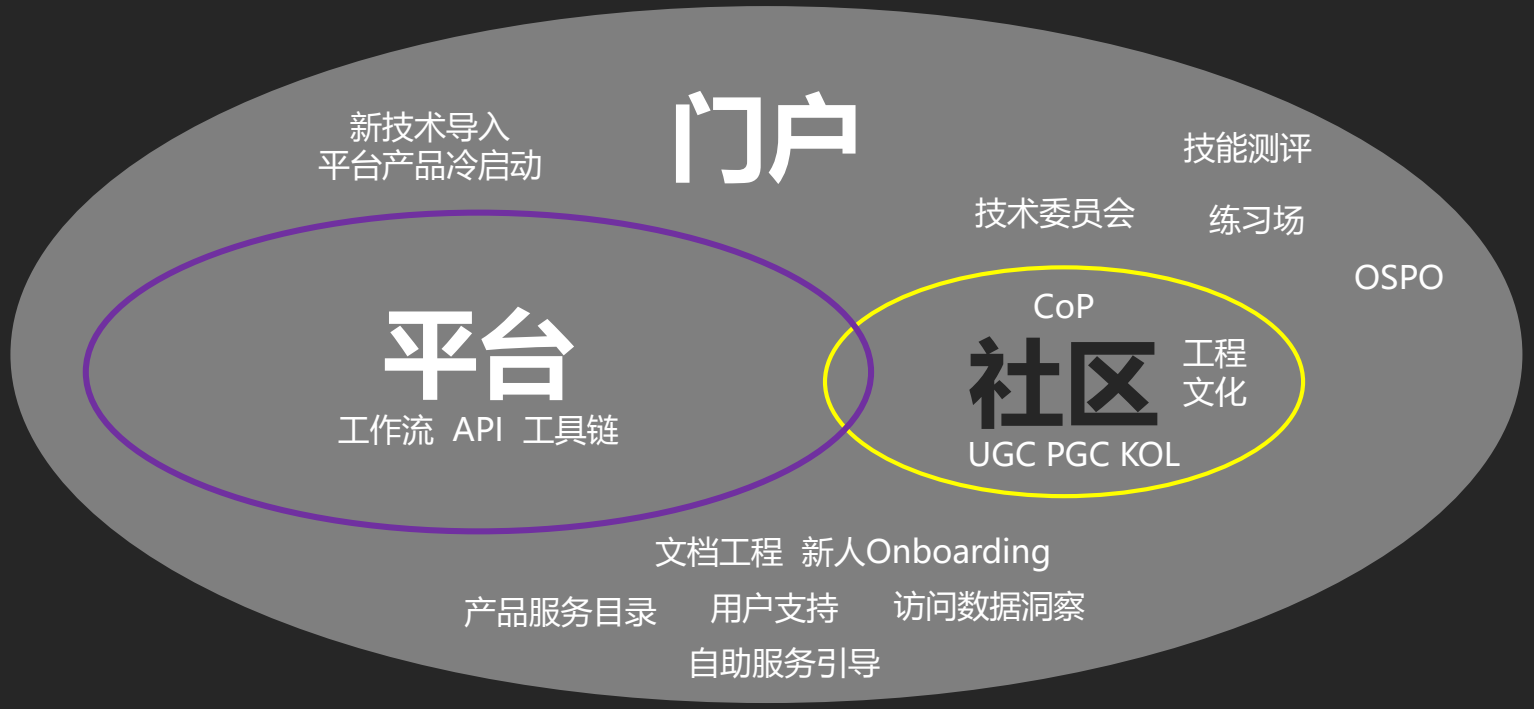
关键实践

- CNCF app delivery TAG – Platform WG 平台白皮书：***平台工程团队的核心职责***



https://tag-app-delivery.cncf.io/zh

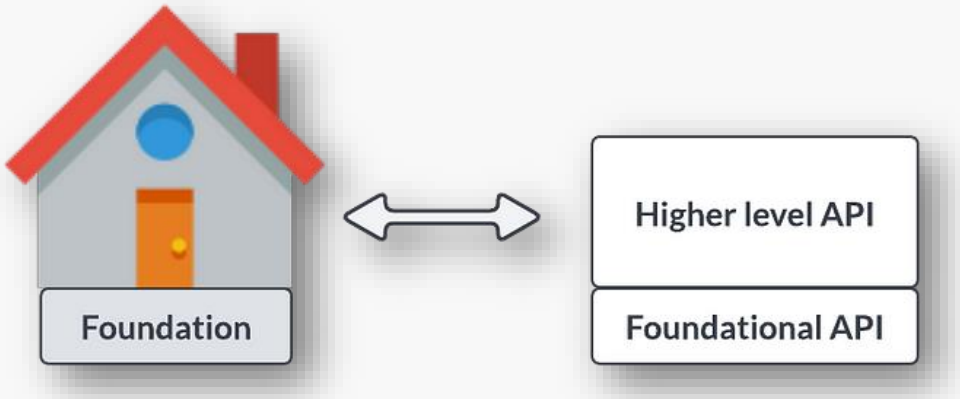
IDP 内部开发者　门户？平台？社区？



关键实践 TVP & PaaP

平台的存在是为了满足平台用户需求

你的平台团队有平台产品经理角色吗?



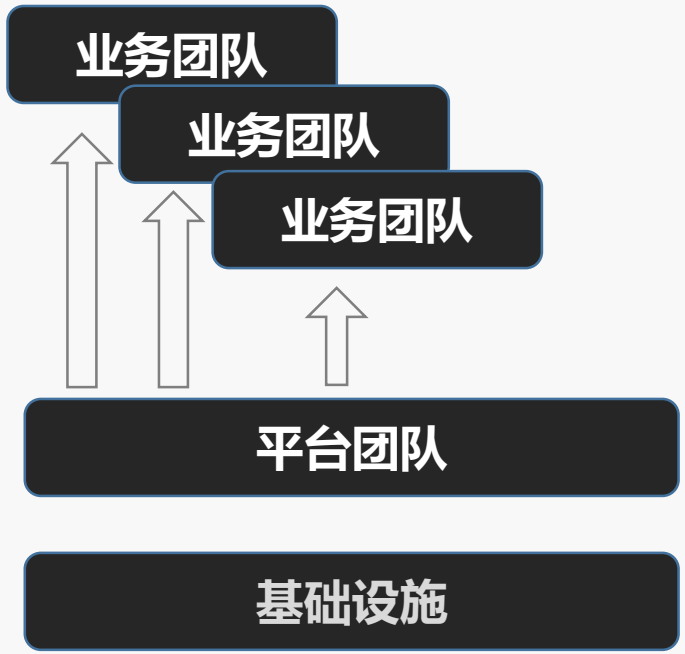
An integrated technology model centers on products and platforms, which differ in important ways.

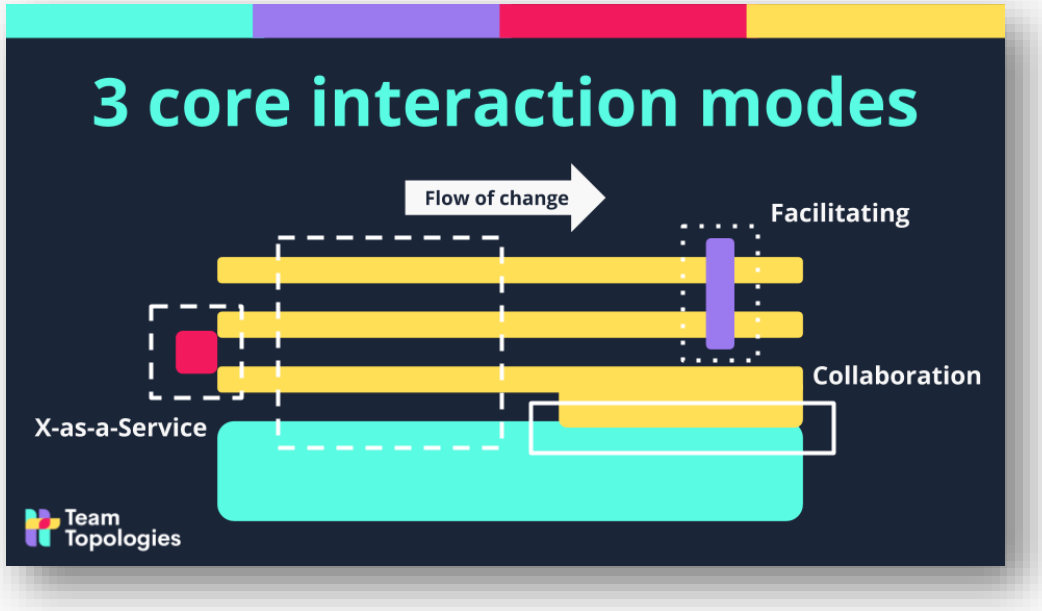
	Products	Platforms
Purpose	Create business value by enhancing end users' experiences	Provide capabilities to products and the enterprise
Primary users	Customers and employees	Digital-product developers, along with functional employees who use platforms directly
Responsible personnel	Business-minded teams of tech specialists, designers, product managers, and functional employees	Technology-minded teams of digital and IT specialists
Pace of innovation	Rapid: upgrades happen as quickly as possible to keep up with users' needs	Variable: changes to support products and modernize underlying systems are made as priorities dictate
Examples	B2C: website/online search B2B: order configuration	B2C: inventory management B2B: pricing

McKinsey & Company

https://towardsdatascience.com/what-i-learned-building-platforms-at-stitch-fix-fc5e0ec72c86

关键实践 组织架构与协作模式





https://towardsdatascience.com/what-i-learned-building-platforms-at-stitch-fix-fc5e0ec72c86

关键实践 DevEx 与 度量

- DORA Metrics [*]
- Flow Metrics [*]
- Space Framework [*]
- Service Level Indicators [*]
- Platform Costs (FinOps)
- Platform’ s security posture [*]

- DX25 is a new measurement framework developed by DX’s research team — which includes the creators of DORA and SPACE.

https://dora.dev/quickcheck

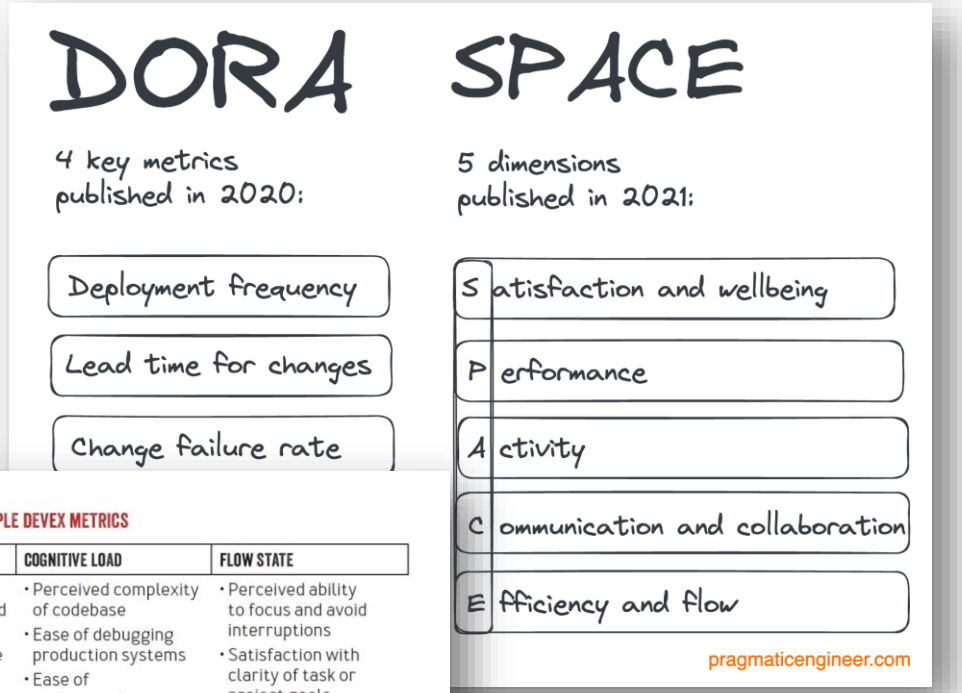
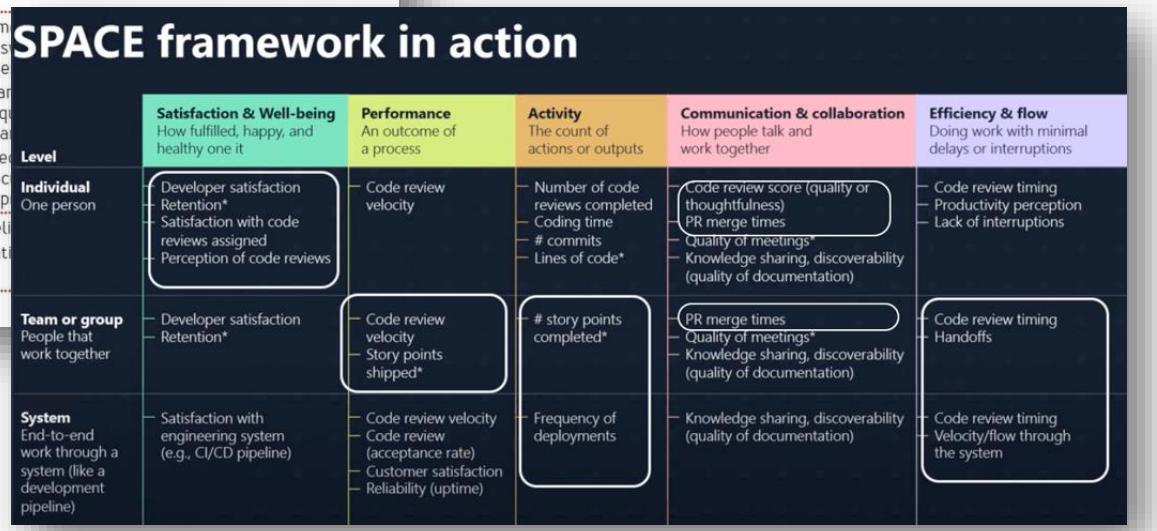


TABLE 1: **EXAMPLE DEVEX METRICS**

	FEEDBACK LOOPS	COGNITIVE LOAD	FLOW STATE
PERCEPTIONS *Human attitudes and opinions*	• Satisfaction with automated test speed and output • Satisfaction with time it takes to validate a local change • Satisfaction with time it takes to deploy a change to production	• Perceived complexity of codebase • Ease of debugging production systems • Ease of understanding documentation	• Perceived ability to focus and avoid interruptions • Satisfaction with clarity of task or project goals • Perceived disruptiveness of being on-call
WORKFLOWS *System and process behaviors*	• Time it takes to generate CI results • Code review turnaround time • Deployment lead time (time it takes to get a change released to production)	• Tim… ans… que… • Mar… req… cha… • Fre… doc… imp…	
KPIS *North star metrics*	• Overall perceived ease of deli… • Employee engagement or sati… • Perceived productivity		



用户满意 -> 组织满意 -> 自我满意

Apache DevLake™

DORA

Velocity

Deployment Frequency

Number of successful deployments to production, how rapidly are your team releasing to users?

Deploying via smaller, more frequent releases makes updates easier to understand, manage, and fix.

Lead Time for Changes

How long does it take from commit to the code running in production?

Lower lead time to successfully implement changes indicates a more agile and responsive team and process.

Stability

Change Failure Rate

How often are your deployments causing a failure in production?

A low CFR indicates that changes are infrequently impacting the stability, availability, and user experience of the software.

Median Time to Restore

How long does it take the team to properly recover from a failure once it is identified?

A low MTTR indicates that the team is effective at resolving significant issues in production, resulting in a better end-user experience.

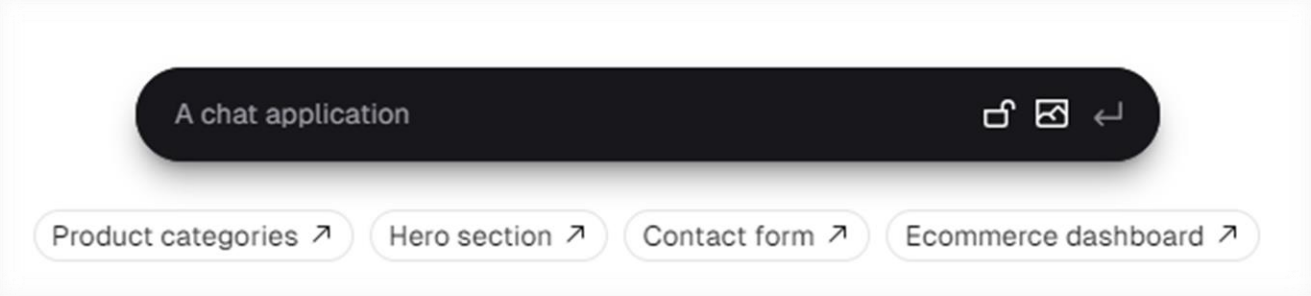
关键实践 DevEx 与 度量

Engineering Metrics Benchmarks: What Makes Elite Teams?

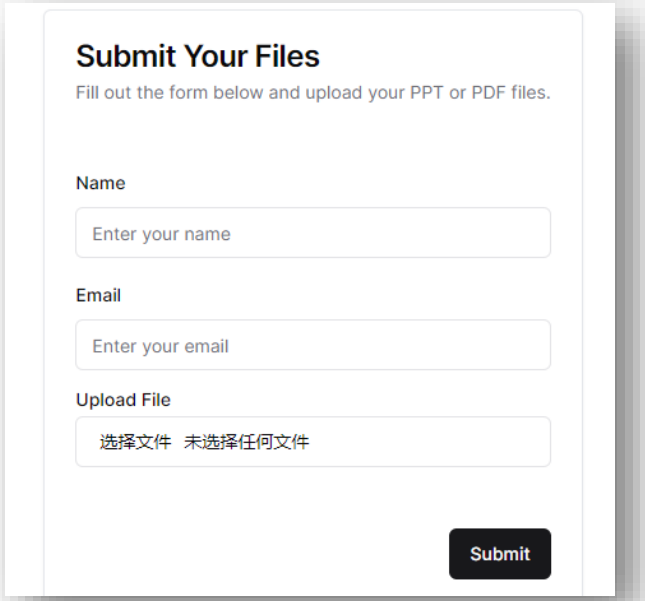
Category	Metric	Elite	Good	Fair	Needs Improvement
Efficiency	Merge Frequency (per dev/week)	> 2	2 - 1.5	1.5 - 1	< 1
	Coding Time (hours)	< 0.5	0.5 - 2.5	2.5 - 24	> 24
	PR Pickup Time (hours)	< 1	1 - 3	3 - 14	> 14
	PR Review Time (hours)	< 0.5	0.5 - 3	3 - 18	> 18
	Deploy Time (hours)	< 3	3 - 69	69 - 197	> 197
DORA	Cycle Time (hours)	< 19	19 - 66	66 - 218	> 218
	Deployment Frequency (per service)	> 1/day	> 2/week	1 - 2/week	< 1/week
	Change Failure Rate (%)	< 1%	1% - 8%	8% - 39%	> 39%
	MTTR (hours)	< 7	7 - 9	9 -10	> 10
Quality and Predictability	PR Size (code changes)	< 98	98 - 148	148 - 218	> 218
	Rework Rate (%)	< 2	2% - 5%	5% - 7%	> 7%
	Refactor Rate (%)	< 9%	9% - 15%	15% - 21%	> 21%
	Planning Accuracy (per sprint)	> 85%	85% - 60%	60% - 40%	< 40%
	Capacity Accuracy (per sprint)	Ideal Range 85% - 115%	Under Commit above 130%	Potential Under Commit 116% - 130%	Potential Over Commit 70% - 84%

https://linearb.io/blog/engineering-metrics-benchmarks-what-makes-elite-teams

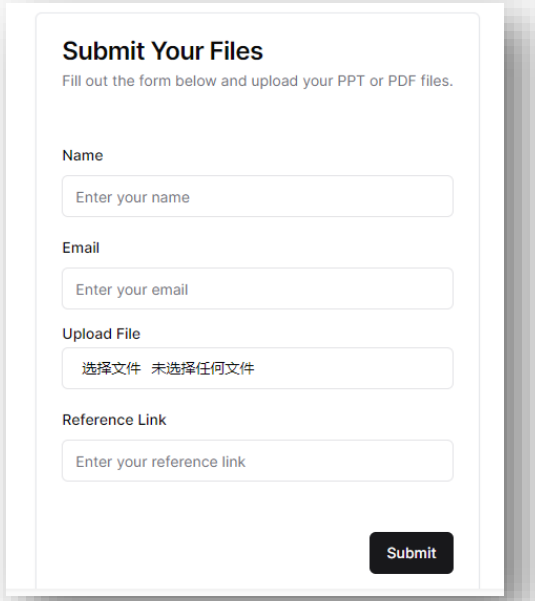
04 平台工程案例0: AIGC - v0.dev by Vercel



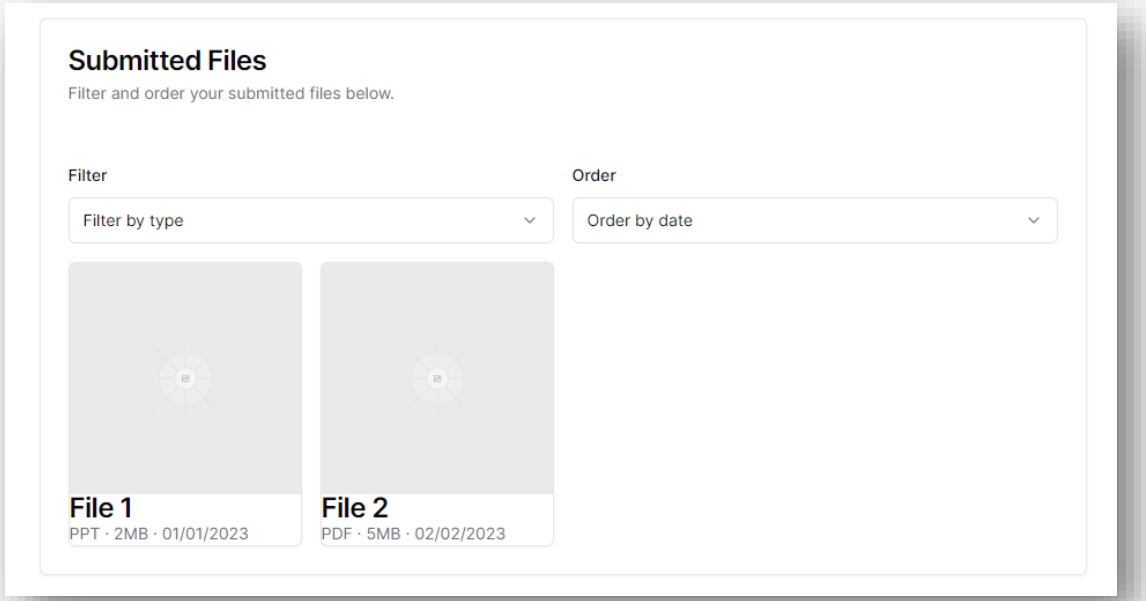
v0: submit PPT/PDF form



v1: add attribute : reference link



v2: display submitted PPT/PDF files in a grid layout with filter & order



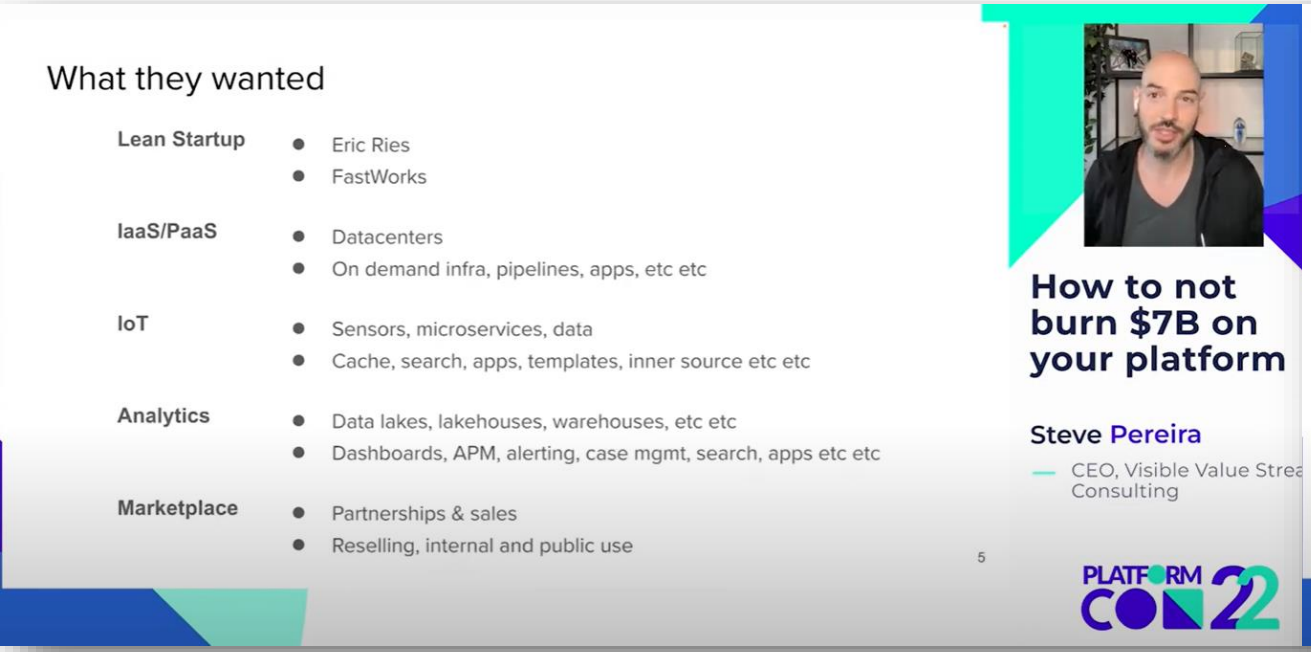
案例1：通用电气

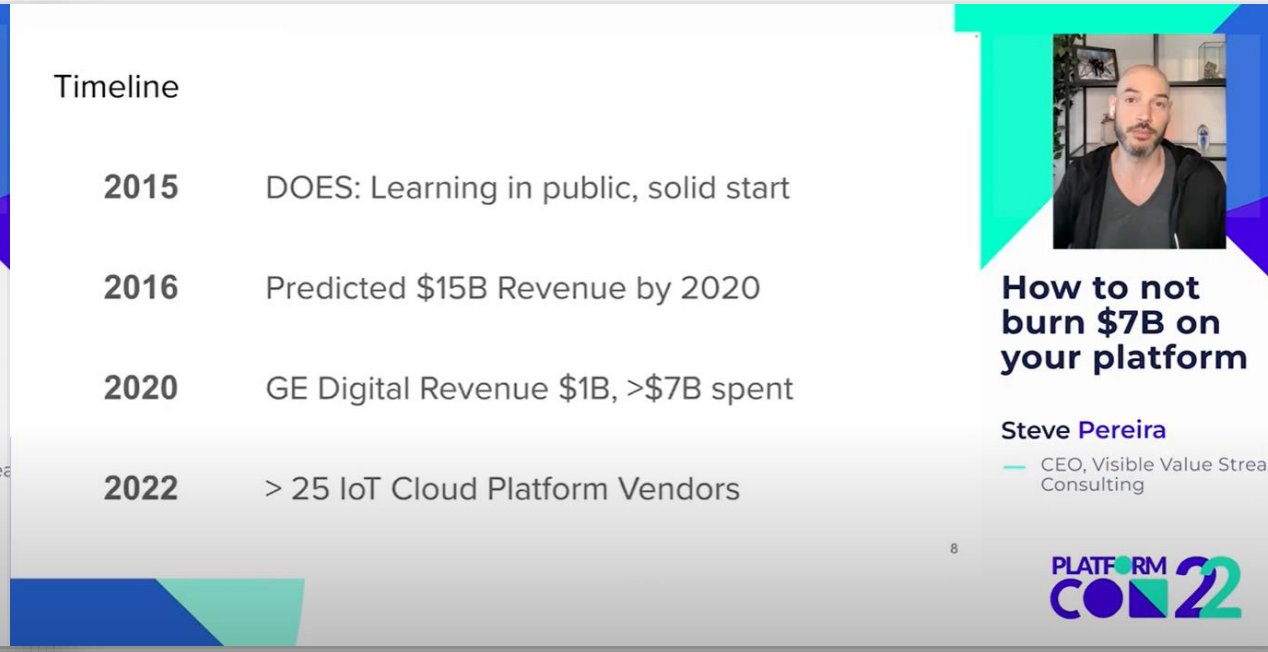
一个价值 70 亿美元的教训！如何避免平台工程变成“大灾难”？

https://www.infoq.cn/article/kfKwKQiGwv0Zbht0fkLW

PlatformCon 2022 : Steve Pereira 分享了**通用电气**如何在类似平台项目上浪费了 70 亿美元的案例。

https://www.youtube.com/watch?v=T4xVQqvPRZs





案例2：Salesforce等

平台工程的失败模式及如何避免，来自一线的宝贵经验

https://www.infoq.cn/article/oAygLlEZHJGmJf7l6vIr

“以下是我在为Salesforce 和其他公司创建 IDP 过程中得到的一些收获”

——Orgspace 的联合创始人兼首席执行官

DONTs

- 先把平台建好，他们一定会用的
- 这是唯一正确的路径
- 讨人喜欢的平台
- 搭积木式的架构方式
- “瑞士奶酪”平台
- 致命的成本旋涡

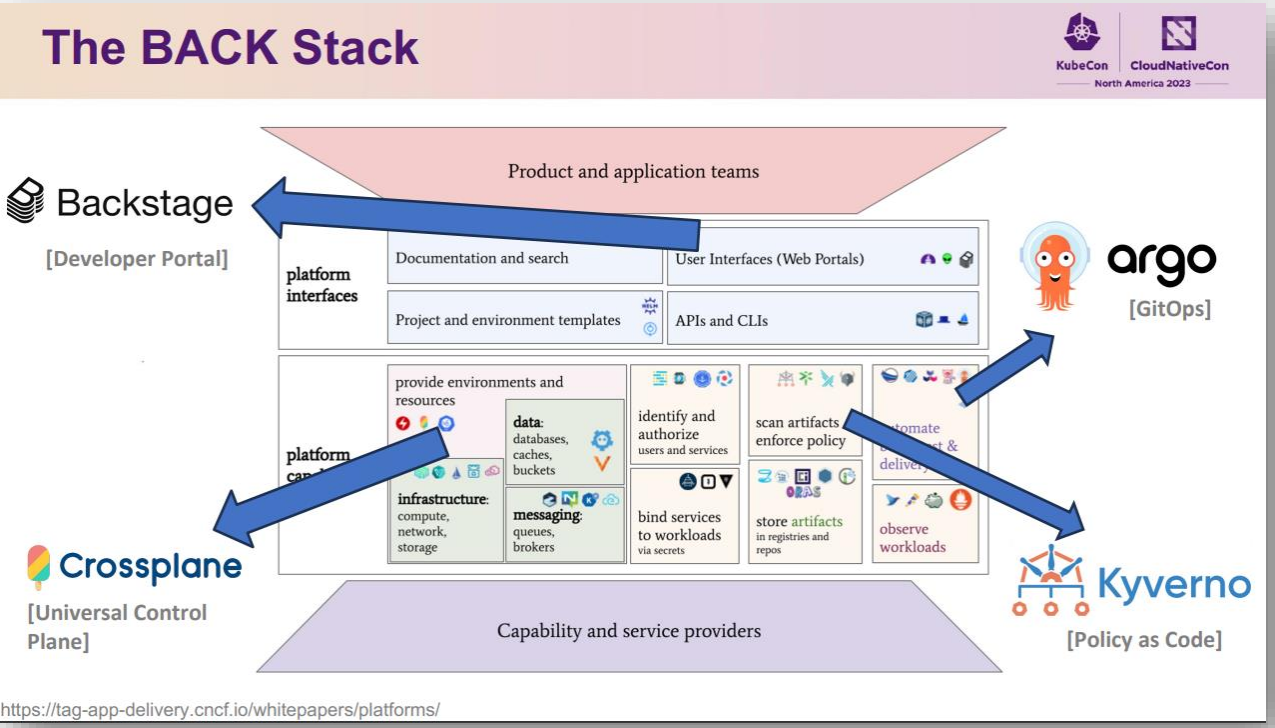
DOs

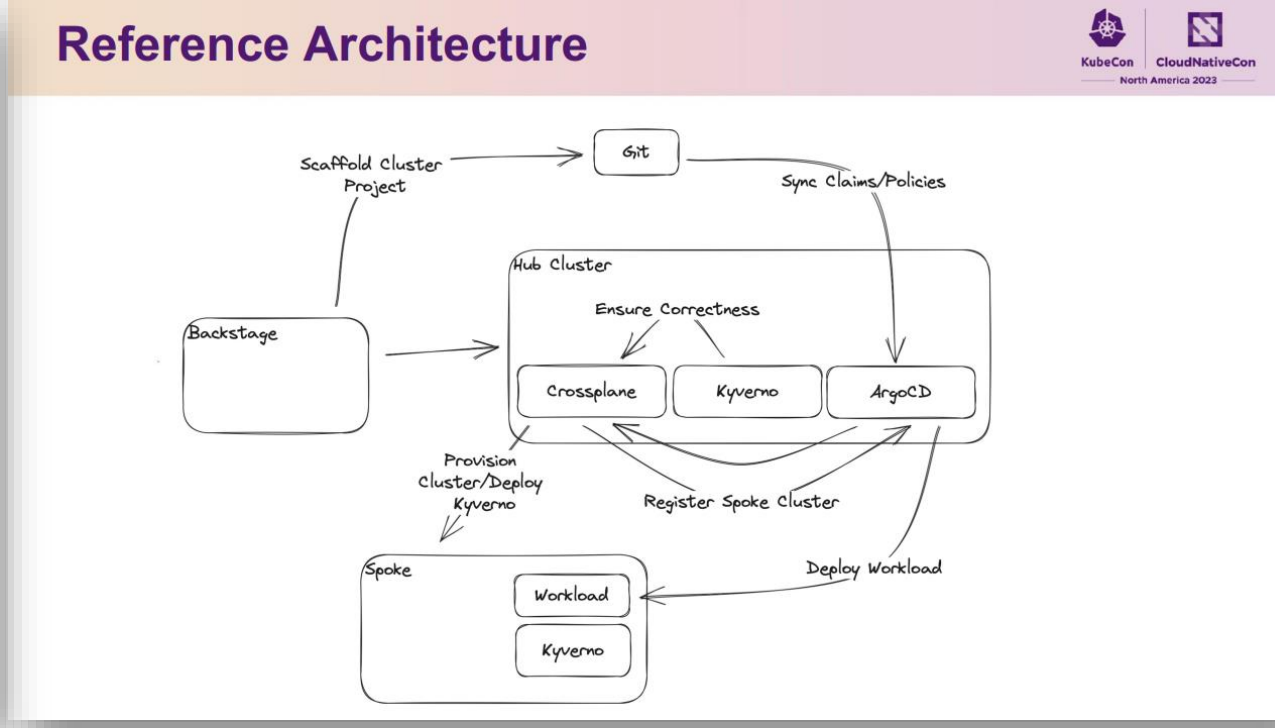
- 从产品经理的角度审视平台。
- 推销你的平台，但不要过度吹嘘。
- 将你的平台视为产品，并确定你的主要客户和利益相关方。
- 接受你不能重新创造 Heroku 或 AWS 的事实，除非你有数亿美元可以花。
- 了解并迭代 MVP，它将帮助您赢得下一轮投资。

案例3：BACK Stack

有可能会发展为CNCF生态
平台工程标准技术栈

- BACK Stack: Backstage & Argo & Crossplane & Kyverno

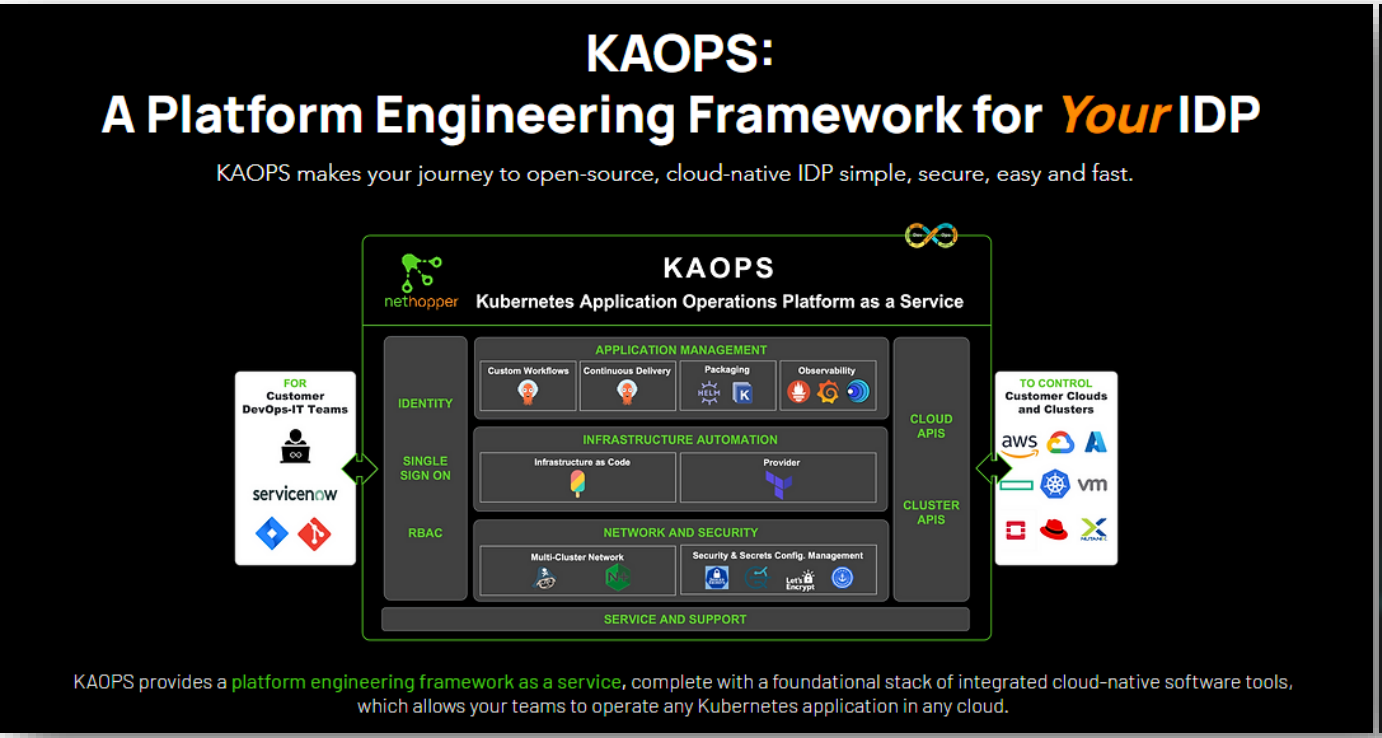


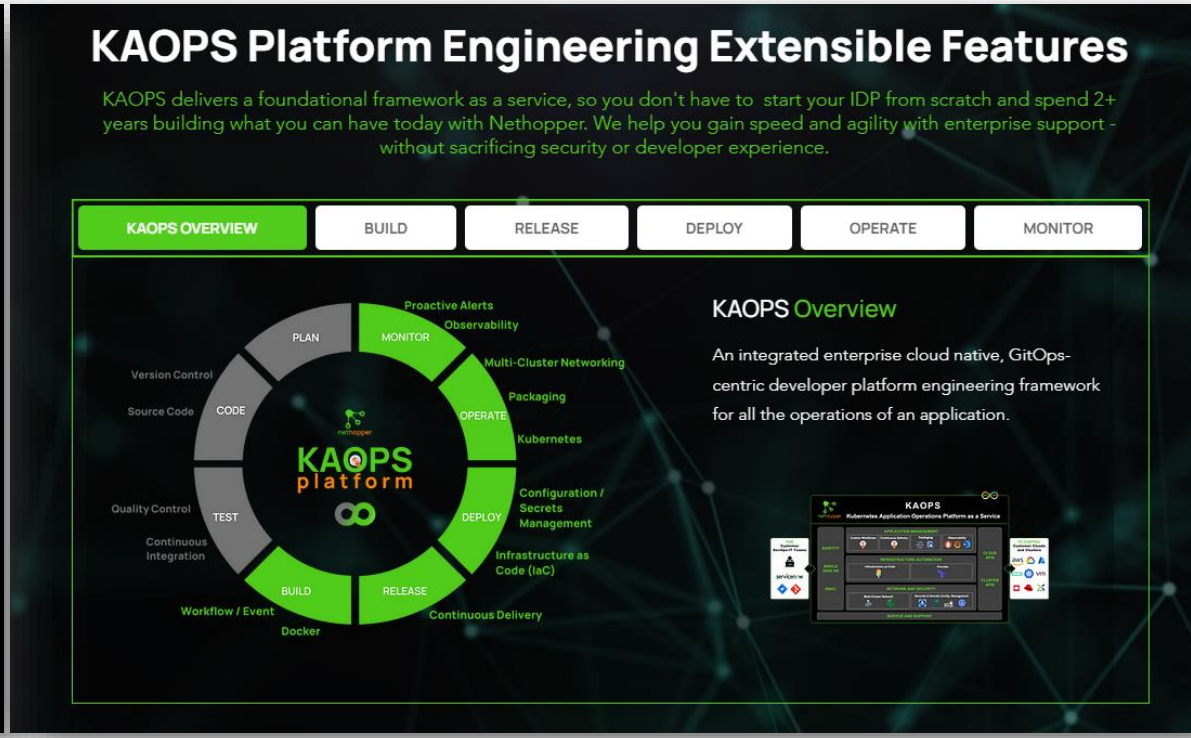


https://static.sched.com/hosted_files/kccncna2023/3b/Kubecon-2023-Backstack.pdf

案例4：Nethopper　　为你的IDP实现提供的平台工程框架

- KAOPS: Kubernetes Application Operations Platform as a Service



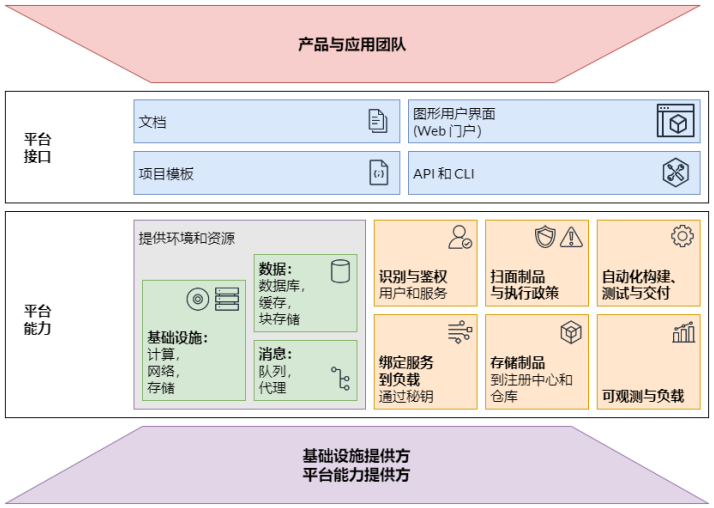


https://www.nethopper.io/

基于当前对平台工程的认知共识，并结合失败与成功案例，

行业内取得了哪些阶段性成果呢?

成果1 CNCF 平台工程 白皮书 & 成熟度模型



方面		暂时性的	可操作	可扩展	可优化
投入	*如何分配工作人员和资金给平台能力?*	自愿或临时的	专职团队	作为产品	已启用的生态
采用	*用户为什么和如何发现和使用内部平台和平台能力?*	不稳定的	外部推动	内部拉力	参与性
接口	*用户如何与平台进行交互并使用平台能力?*	自定义程序	标准工具化	自定义解决方案	综合服务
Operations	*平台及其能力是如何规划、确定优先次序、开发和维护的?*	按需求	集中跟踪	集中启用	管理服务
衡量	-收集、整合反馈和学习的流程是什么? _	临时的	一致的收集	见解	定量与定性

https://tag-app-delivery.cncf.io/zh/wgs/platforms/platforms-maturity-model/v1

成果2 CNOE： Cloud Native Operational Excellence

使命

CNOE旨在通过内置的最佳实践，帮助平台工程师更快、更安全地构建IDP平台

愿景

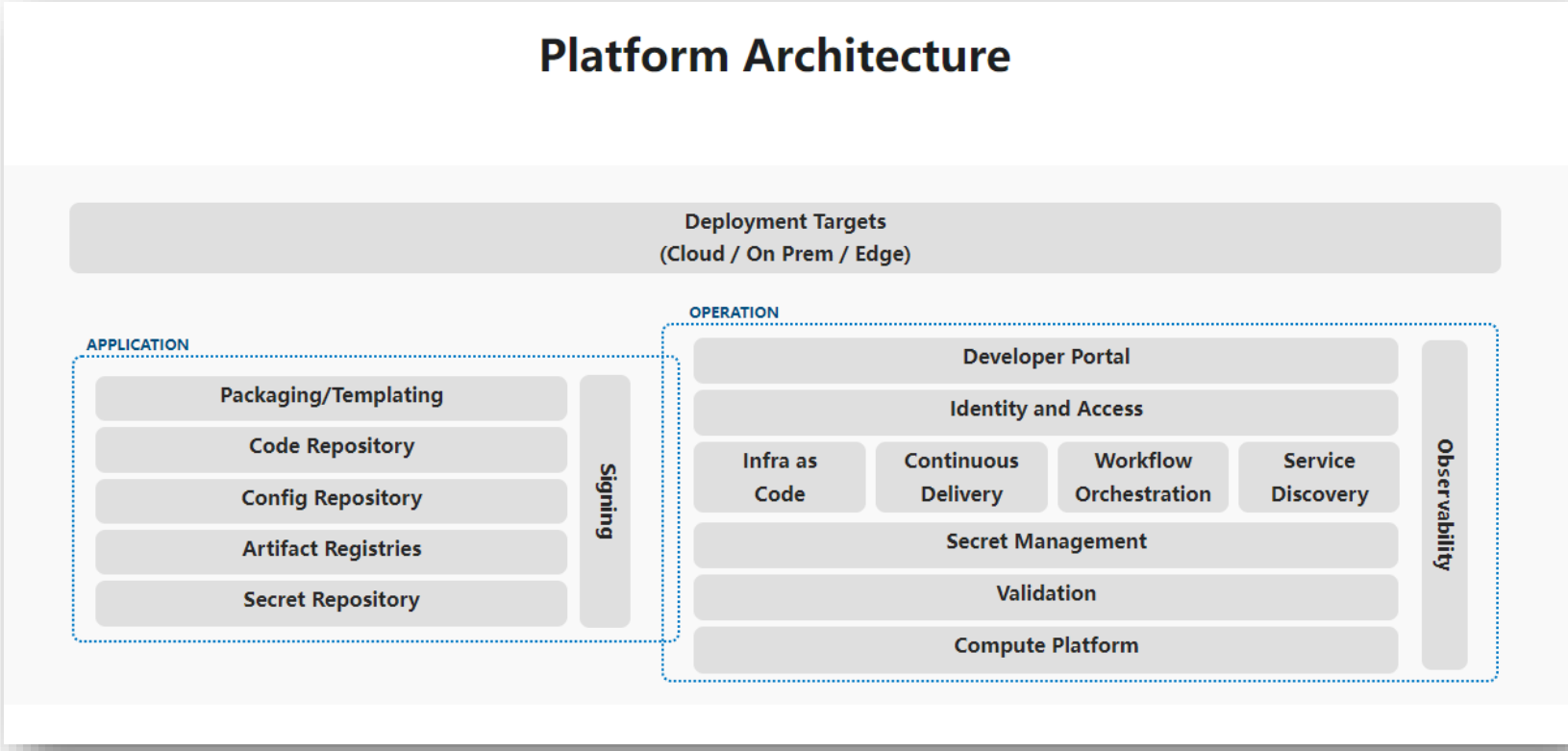
CNOE努力成为领先软件公司构建云原生内部开发者平台的首选框架

价值主张

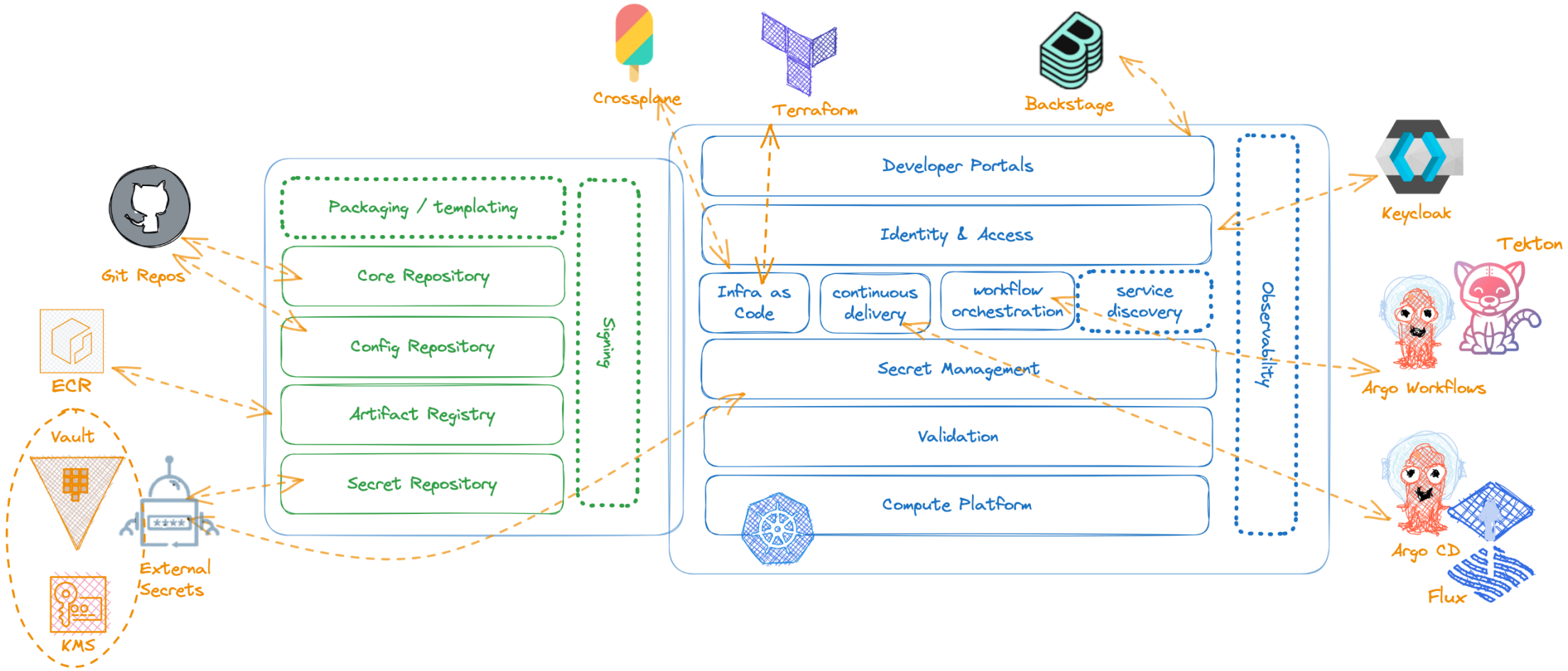
- 云原生
- 社区最佳实践
- 模块化

https://cnoe.io

Adobe, Amazon Web Services, Autodesk, Salesforce, Twilio



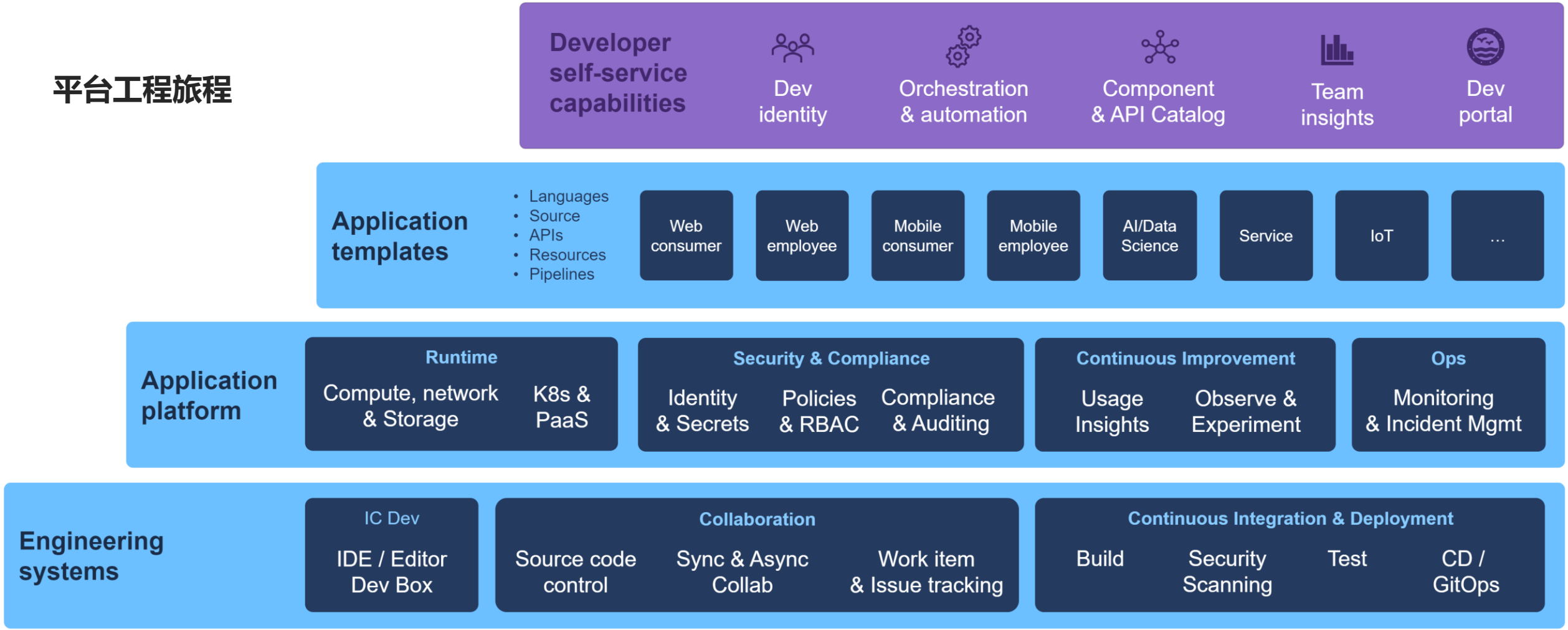
CNOE 参考架构



https://cnoe.io/docs/reference-implementation

学习资源推荐： 微软 平台工程指南

平台工程旅程



https://learn.microsoft.com/en-us/platform-engineering

05 获得的新启示

文化导入，组织架构适配

战略挂钩，业务价值支撑

规模化组织

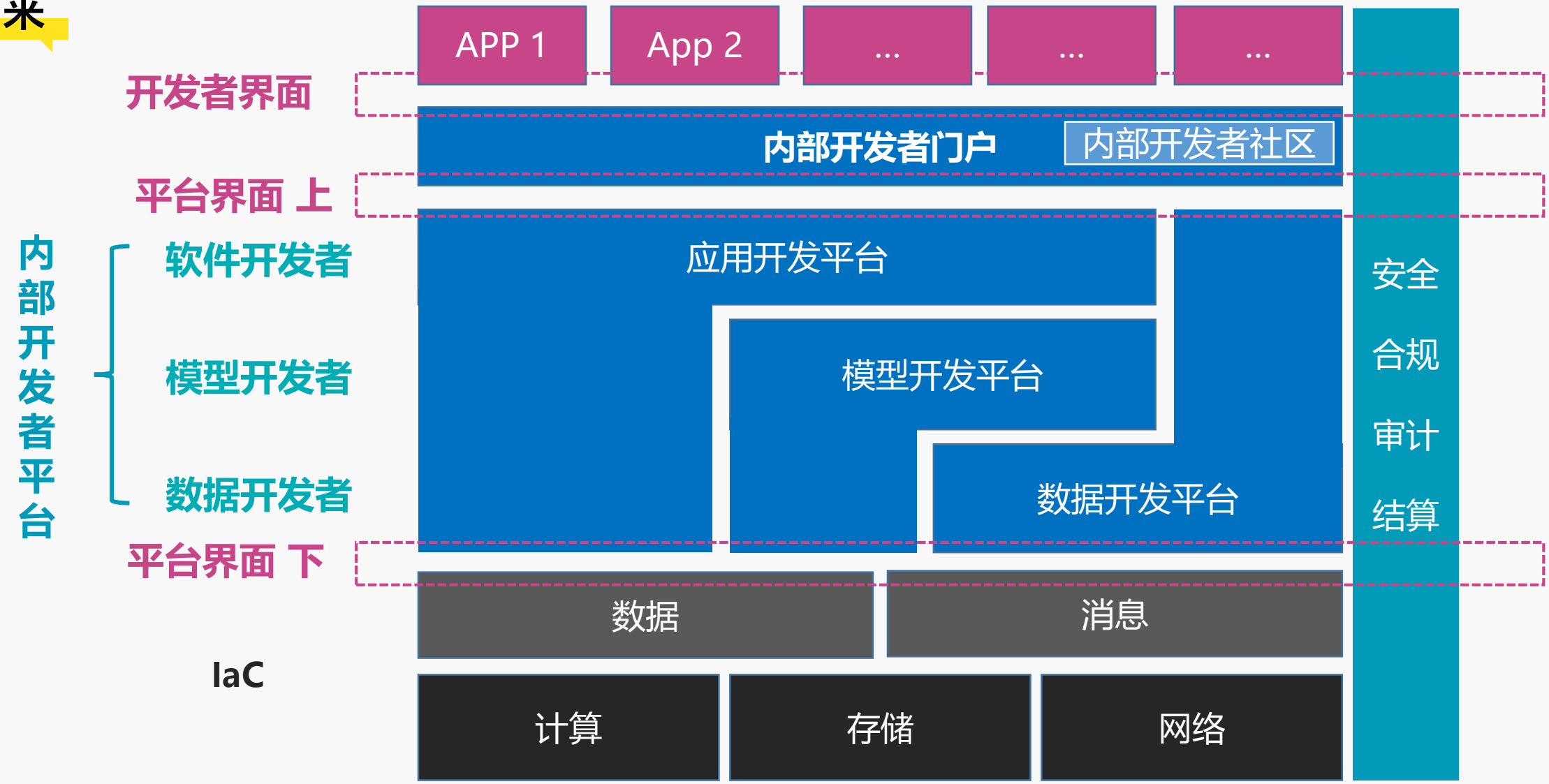
- 加速拥抱云原生
- 优化你的团队拓扑
- 度量平台的成功

- **平台工程会是终点吗?**
- **云原生时代软件工程的理想态**
- **Everything is code !**

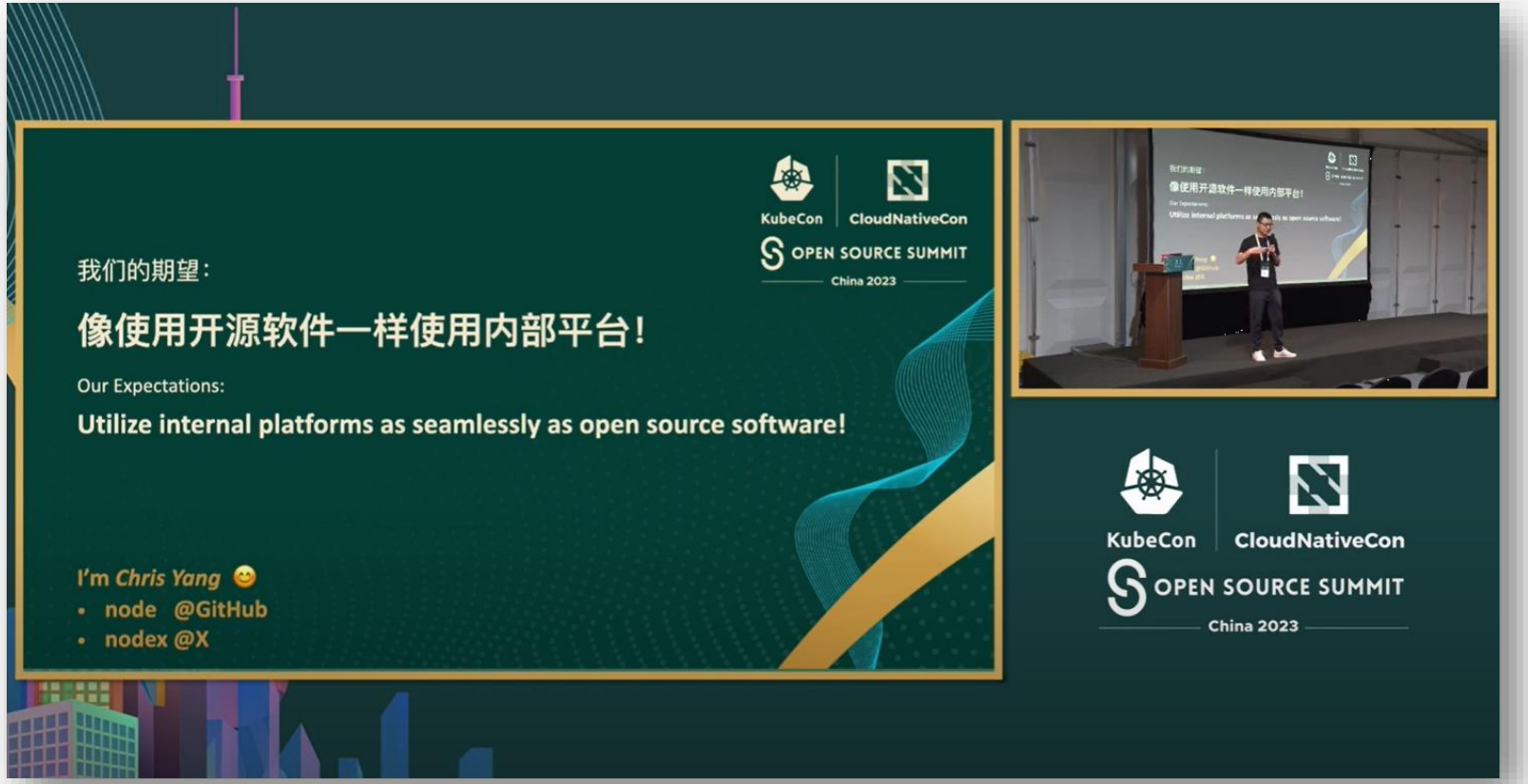
初创团队

- MVP & TVP

未来


APP 1
App 2
...
...
...
开发者界面
内部开发者门户
内部开发者社区
平台界面 上
内部开发者平台
软件开发者
模型开发者
数据开发者
应用开发平台
模型开发平台
数据开发平台
安全
合规
审计
结算
平台界面 下
数据
消息
IaC
计算
存储
网络

未来


KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023
我们的期望：
像使用开源软件一样使用内部平台！
Our Expectations:
Utilize internal platforms as seamlessly as open source software!
I'm Chris Yang
• node @GitHub
• nodex @X
KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

共同参与

For Platformers !

@平台工程洞察

PECommunity.cn

TOP100®

主办方 msup®

微信官方公众号：壹佰案例
关注查看更多年度实践案例