KubeCon | CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

Large-Scale Financial-Grade Platform Engineering Practice in China Merchants Bank

Jiahang Xu,China Merchants Bank
Qingguo Zeng, *Alibaba Cloud*

Speaker

Jiahang Xu

China Merchants Bank(招商银行)

System Architect

Jiahang Xu is a System Architect at China Merchants Bank. He has over 14 years of unique cross-domain experience working in telecom, automotive, financial industry, startup as a co-founder, and KubeVela maintainer. He's mainly focused on cloud-native application technology practice in recent years. He is seasoned in the application management system and responsible for building large-scale financial platform engineering China Merchants Bank

Agenda

KubeCon | CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

Intro：A Reference Case

Intro | Automotive Platform

Automotive Modularization Trend Research Report, 2021

May 2021

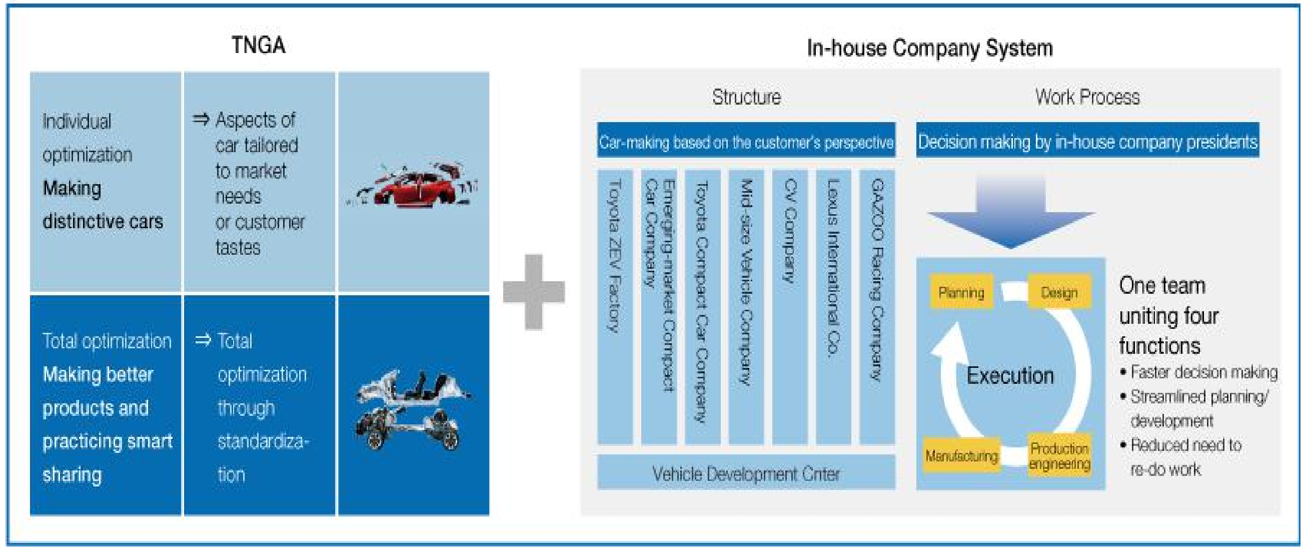
The demand for **low-cost and short development cycles** of new models promotes the development of automotive modular platforms.
An automotive modular platform includes the design and assembly of all sub-systems of a car in a modular manner, and the **standardized** design and production of auto parts in the form of **modules**, and the final "assembly" according to the positioning of **models**.

Car Modeling & Separation of Concerns

variable
variable
uniform
variable
variable

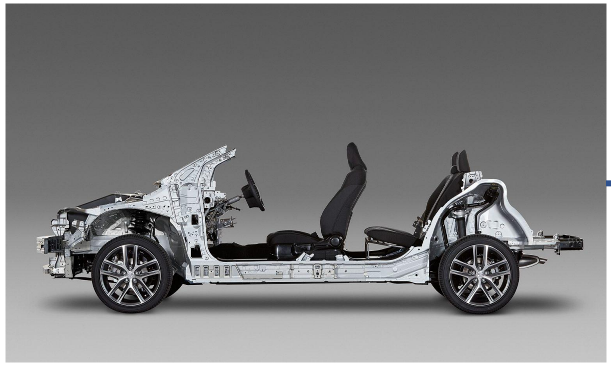
SOP based on Model

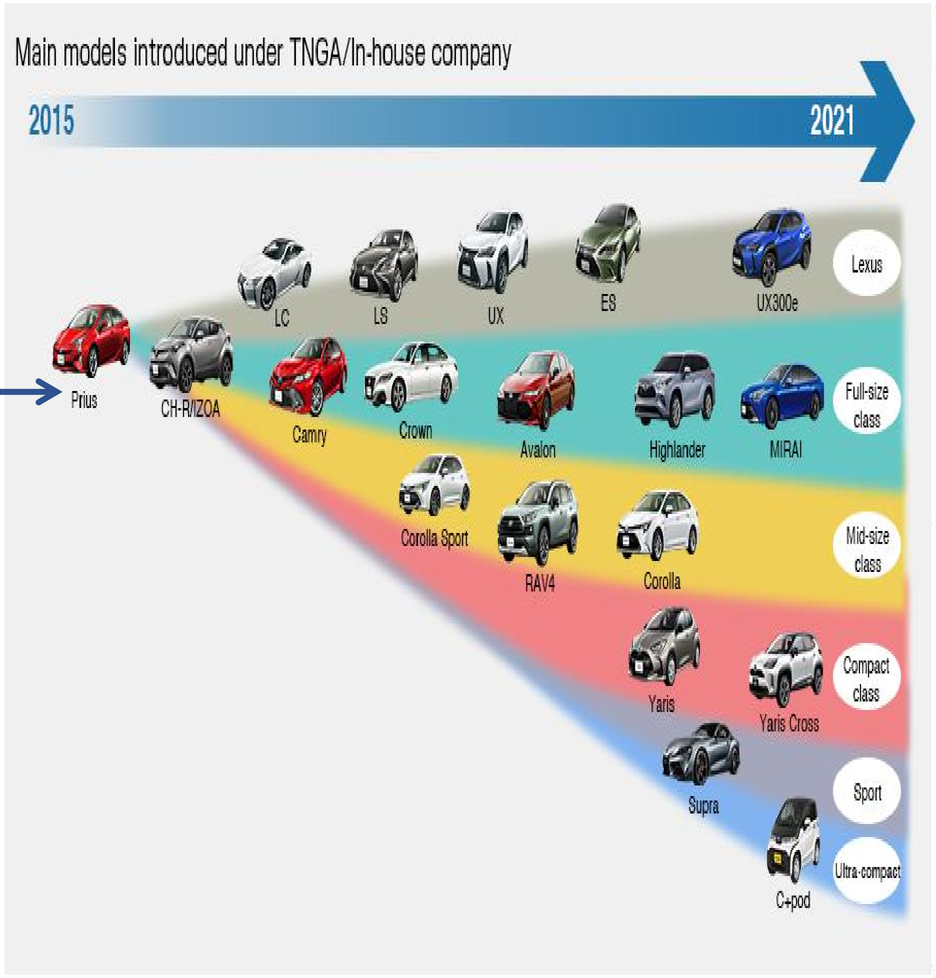
Making Ever-better Cars: Toyota's Approach and Structure


TNGA
Individual optimization
Making distinctive cars
⇒ Aspects of car tailored to market needs or customer tastes
Total optimization
Making better products and practicing smart sharing
⇒ Total optimization through standardiza-tion
In-house Company System
Structure
Car-making based on the customer's perspective
Toyota ZEV Factory
Emerging-market Compact Car Company
Toyota Compact Car Company
Mid-size Vehicle Company
CV Company
Lexus International Co.
GAZOO Racing Company
Vehicle Development Cnter
Work Process
Decision making by in-house company presidents
Planning
Design
Execution
Manufacturing
Production engineering
One team uniting four functions
• Faster decision making
• Streamlined planning/ development
• Reduced need to re-do work

Automotive Platform | Implementation of Platform Engineering

Toyota New Global Architecture (TNGA) platform





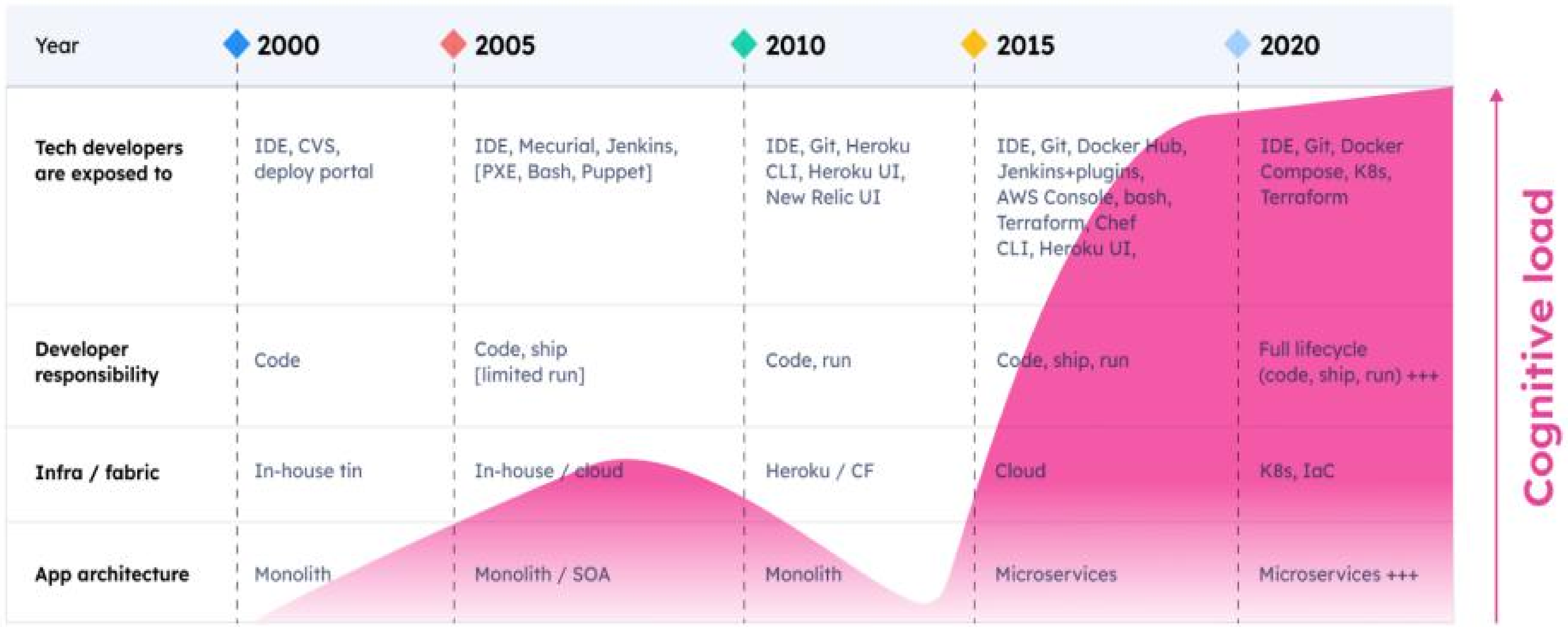
Automotive Modular Platforms of Major Automakers in the World

	Automakers	Main Modular Platforms/Architectures
Foreign	Volkswagen	MQB, MLB, MSB, PPE, MMB, MEB
	Toyota	TNGA, e-TNGA
	Renault-Nissan	CMF, CMF-EV
	BMW	CLAR, ULK
	Benz	MFA, MRA, MHA, MSA, MEA
	Hyundai	E-GMP, i-GMP
	GM	BEV3
	PSA	CMP, EMP2
Domestic	Geely	AMA, BMA, CMA, DMA, PMA, SEA
	GAC	GPMA, GEP
	Great Wall	Lemon, Tank
	Chery	@LIFE, T1X, M1X, A3X, M3X, NEV
	Changan	P1, P2, P3, P4, Ark, EPA0, EPA1, EPA2
	BYD	e-Platform
	BAIC	BMFA, BE22, IMC
	FAW Car Besturn	FMA

Source: ResearchInChina

Background

Why now?

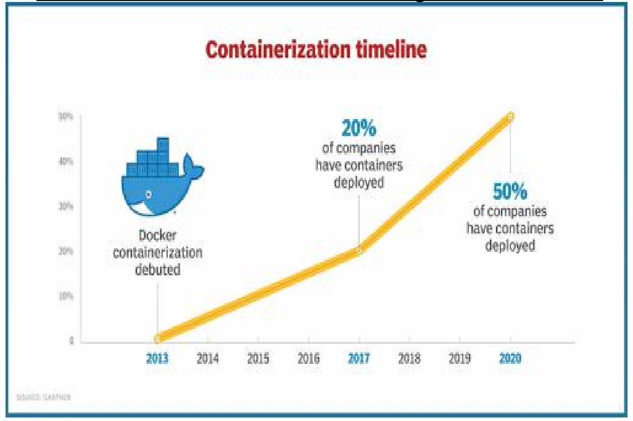


Year	2000	2005	2010	2015	2020
Tech developers are exposed to	IDE, CVS, deploy portal	IDE, Mecurial, Jenkins, [PXE, Bash, Puppet]	IDE, Git, Heroku CLI, Heroku UI, New Relic UI	IDE, Git, Docker Hub, Jenkins+plugins, AWS Console, bash, Terraform, Chef CLI, Heroku UI,	IDE, Git, Docker Compose, K8s, Terraform
Developer responsibility	Code	Code, ship [limited run]	Code, run	Code, ship, run	Full lifecycle (code, ship, run) +++
Infra / fabric	In-house tin	In-house / cloud	Heroku / CF	Cloud	K8s, IaC
App architecture	Monolith	Monolith / SOA	Monolith	Microservices	Microservices +++

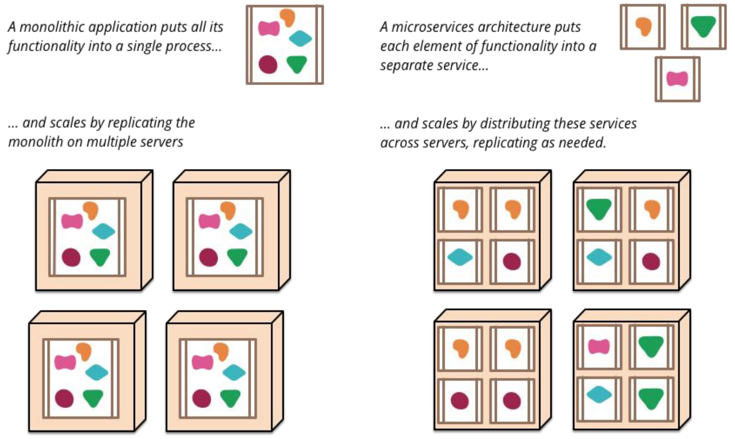
Inspired by Daniel Bryant at PlatformCon 2022

Why now? 2015-now

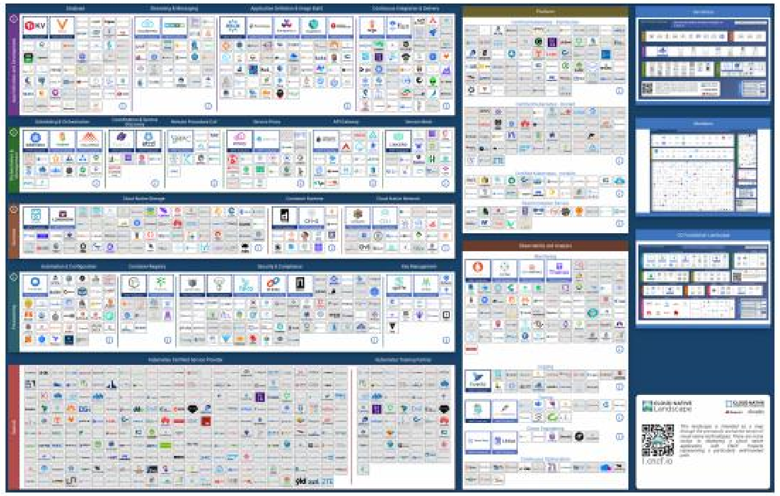
Containers: the standard delivery medium



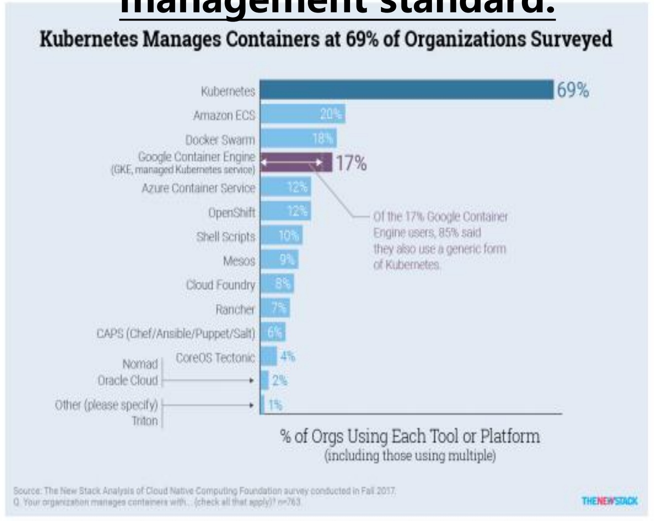
Microservices: standard practice for cloud application


A monolithic application puts all its functionality into a single process...
A microservices architecture puts each element of functionality into a separate service...
... and scales by replicating the monolith on multiple servers
... and scales by distributing these services across servers, replicating as needed.

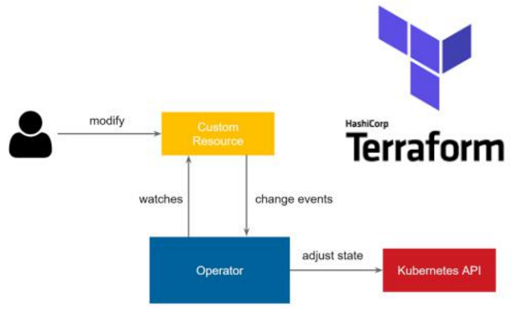
CNCF: Community Evolution



K8s: cloud application resource management standard.

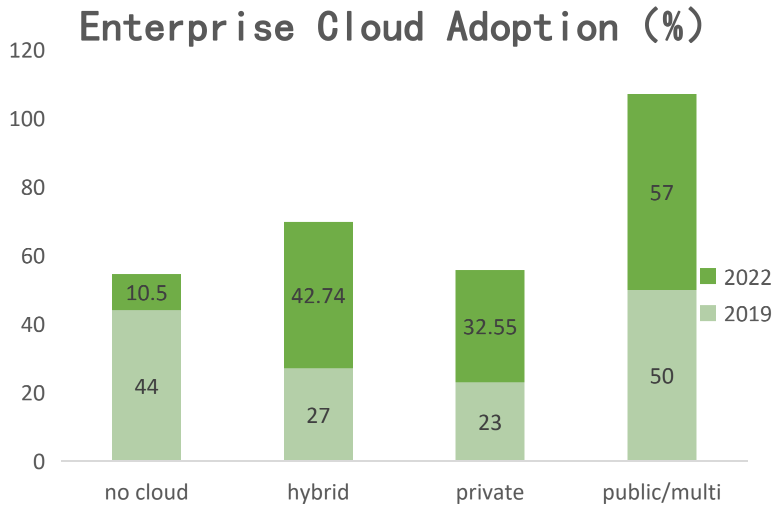


Infrastructure as code: standard practice for automation



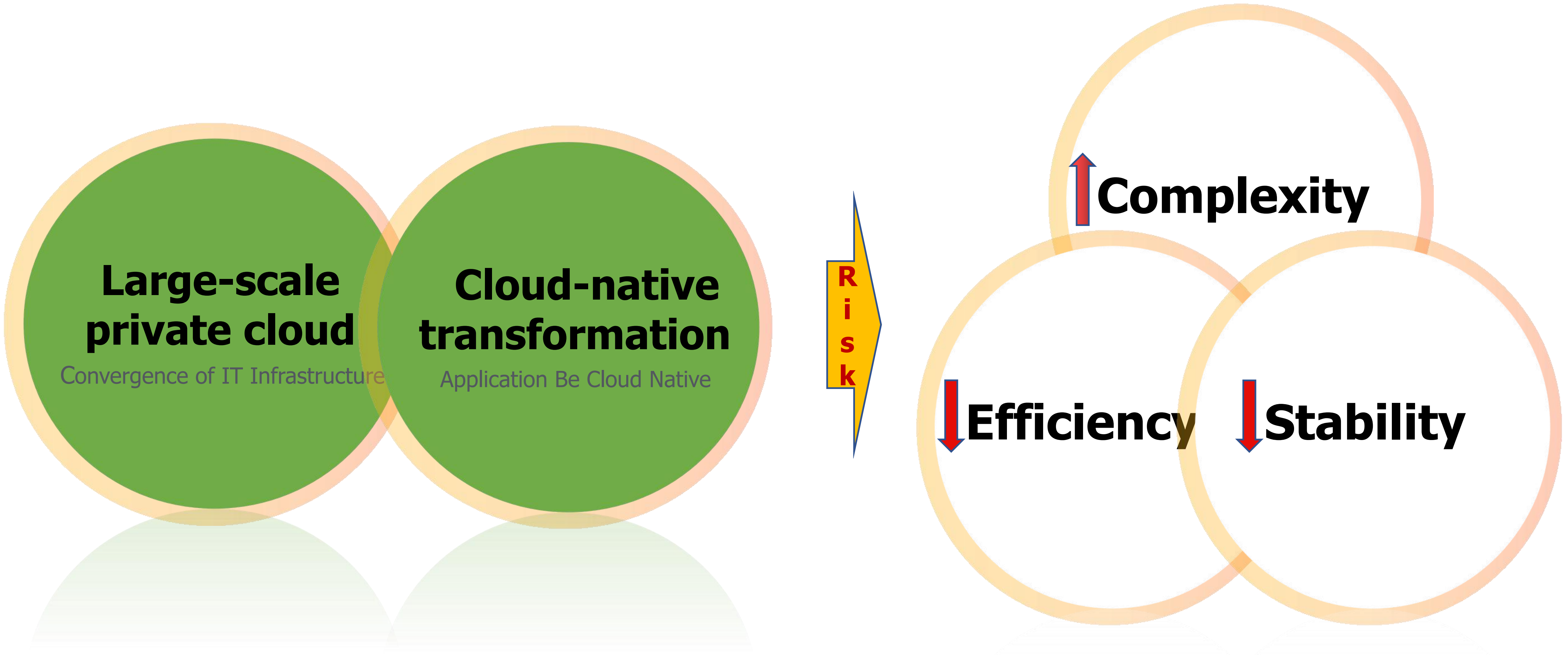
K8s CRD Operator：自动化的运维模型

Infrastructure as Code：基础设施即代码



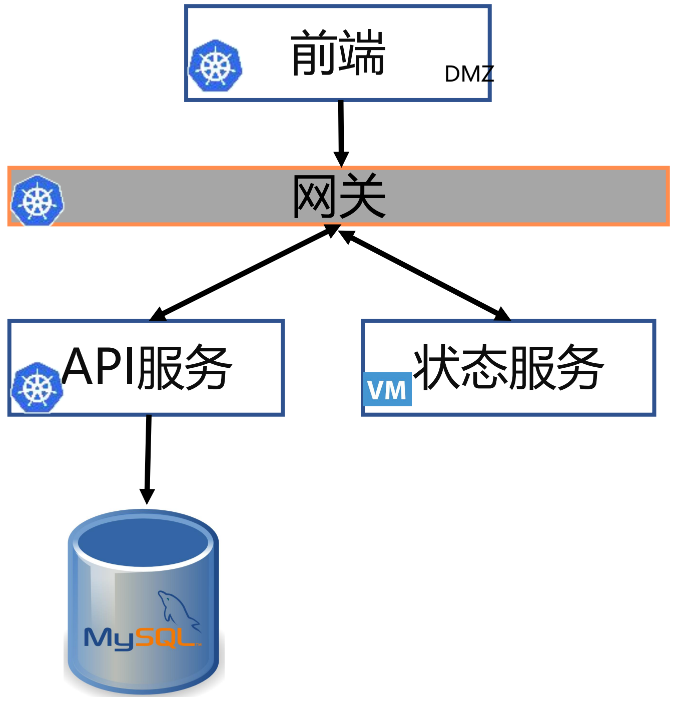
DORA 2022 State of DevOps Report

CMB Challenge


Large-scale private cloud
Convergence of IT Infrastructure
Cloud-native transformation
Application Be Cloud Native
Risk
Complexity
Efficiency
Stability

CMB Challenge

- In the current cloud environment, developers have more and more concerns. They are responsible for application development, delivery, and operation, and their functional boundaries are becoming longer.
- The complexity of cloud infrastructure is exposed to developers, who must face various technical details of the cloud. Microservices have long associated links and involve multiple organizations, which affects operational efficiency and troubleshooting.



举例： 简单业务场景

应用管理生命周期	虚拟机应用	容器化应用	数据库、中间件等	应用用云复杂度
应用初始化	系统\子系统、发布单元、服务单元、代码仓、流水线、制品库、流程中心、部署策略、重要性等级等			行内应用开发运维人员目前至少要掌握60+（不含编程领域）不同门类用云领域知识，使用超过多个应用相关平台，才可能胜任云应用开发运维。用云复杂性，造成应用业务连续性保障难度大
	按租户、Quota申请计算（虚机）、网络（Vnet、SLB、SubNet、PIP、NAT）存储	按容器租户、Quota申请命名空间，申请微服务租户。	mysql架构选型、资源评审，资源申请。	
应用配置	配置文件路径及文件格式，SLB、后端池、出入站规则、健康探测规则等。应用配置自行管理	ENV、Secret、ConfigMap、Mount、ENVREf、CMD、内部路由svc、路由route、容器端口、健康探针、优雅停、微服务配置等等各类yaml	应用正确配置db链接池，解决数据库sql兼容性问题。	
应用部署	编写部署脚本，设置自启动服务，切换回退自行实现。	depoyment、job、cronjob等，弹性、灰度、域名访问、网络安全扫描、DNS上线	sql发布等、防火墙开通	
应用运维、监控、排障	根据自身经验知识查看分析应用日志、应用链路、应用指标。提权运维变更并验证。如遇网络问题需联系平台及网络团队排查。		联系DBA进行故障定位分析	

开发人员需要掌握的用云知识

KubeCon | CloudNativeCon

OPEN SOURCE SUMMIT

China 2023

Platform Engineering 101

Why Platform? What is it?

Platform Aims to Solve:

1. **Reduce the cognitive load on product teams** and thereby accelerate product development and delivery
2. **Improve reliability and resiliency of products** relying on platform capabilities by dedicating experts to configure and manage them
3. **Accelerate product development and delivery** by reusing and sharing platform tools and knowledge across many teams in an enterprise
4. **Reduce risk of security, regulatory and functional issues** in products and services by governing platform capabilities and the users, tools and processes surrounding them
5. **Enable cost-effective and productive use** of services from public clouds and other managed offerings by enabling delegation of implementations to those providers while maintaining control over user experience

A cloud-native computing platform is an integrated collection of capabilities defined and presented according to the needs of the platform's users. It is a cross-cutting layer that ensures a consistent experience for acquiring and integrating typical capabilities and services for a broad set of applications and use cases. A good platform provides consistent user experiences for using and managing its capabilities and services, such as Web portals, project templates, and self-service APIs.

Attributes of platforms

- **Platform as a product.** A platform exists to serve the requirements of its users and it should be designed and evolved based on those requirements, similar to any other software products.

- **User experience.** A platform should offer its capabilities through consistent interfaces and focus onthe user experience.

- **Self-service with golden path.** A platform should be self-serviceable. Such a workflow could be offered with an initial project template and documentation, a bundle often described as a golden path.

- **Reduced cognitive load for users.** An essential goal of a platform is to reduce the cognitive load on product teams.

- **Optional and composable.** Platforms are intended to make product development more efficient, so they must not be an impediment.

- **Secure by default.** A platform should be secure by default and offer capabilities to ensure compliance and validation based on rules and standards defined by the organization.

Platform Team

Platform teams are responsible for

- the interfaces to and experiences with platform capabilities - like Web portals, custom APIs, and golden path templates.
- On one hand, platform teams work with those teams implementing infrastructure and supporting services to define consistent experiences;
- on the other, they work with product and user teams to gather feedback and ensure those experiences meet requirements.

Platform Engineering CMB Practice

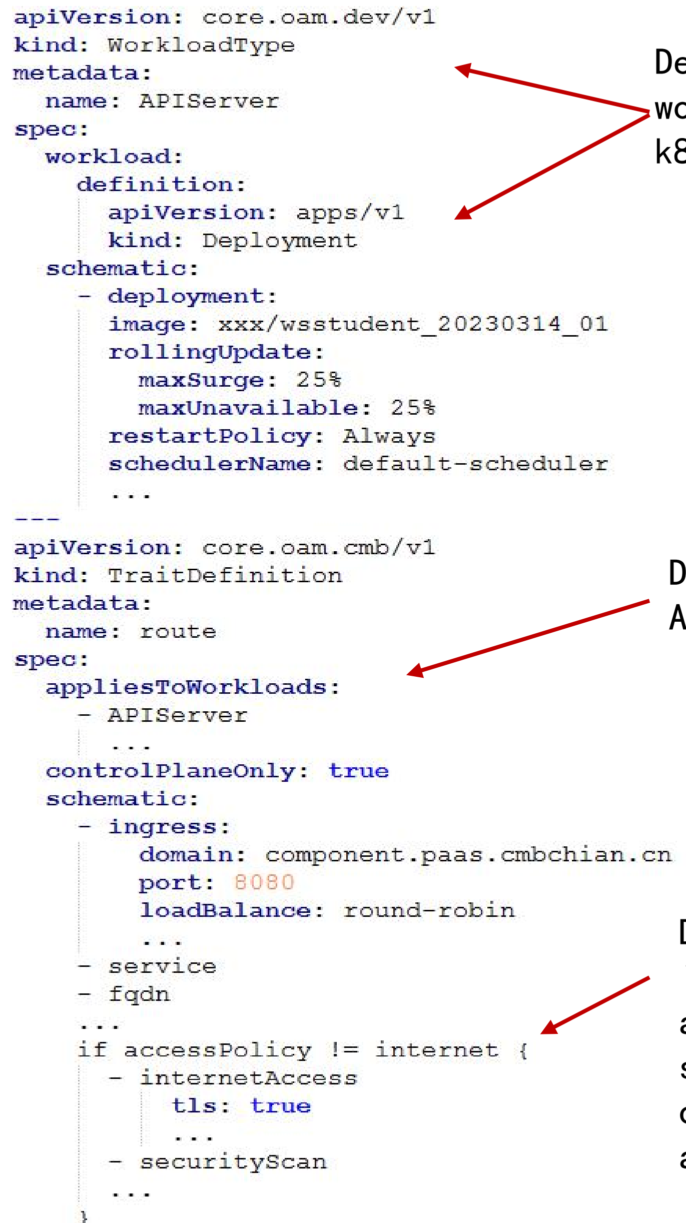
Keypoints:

- ✓ **Standardzation**
- ✓ **SOP(Golden Path)**
- ✓ **Reverse Conway Manoeuvre**

Standardzation | OAM

➢ OAM provides a **standardized cloud capability supply paradigm** for the **Platform team**. The **Platform team** studies user demands, abstracts and encapsulates cloud capabilities, and uses OAM for standardized supply:

- **Defining workloads:** abstracting and encapsulating the basic capabilities and implementation details of the cloud;
- **Defining operational characteristics:** abstracting and encapsulating the basic capabilities of the cloud and operational automation;
- **Implementing controller**: implementing corresponding controllers through the OAM engine.



```
apiVersion: core.oam.dev/v1
kind: WorkloadType
metadata:
  name: APIServer
spec:
  workload:
    definition:
      apiVersion: apps/v1
      kind: Deployment
  schematic:
    - deployment:
      image: xxx/wsstudent_20230314_01
      rollingUpdate:
        maxSurge: 25%
        maxUnavailable: 25%
      restartPolicy: Always
      schedulerName: default-scheduler
      ...
---
apiVersion: core.oam.cmb/v1
kind: TraitDefinition
metadata:
  name: route
spec:
  appliesToWorkloads:
    - APIServer
      ...
  controlPlaneOnly: true
  schematic:
    - ingress:
        domain: component.paas.cmbchian.cn
        port: 8080
        loadBalance: round-robin
        ...
    - service
    - fqdn
    ...
    if accessPolicy != internet {
      - internetAccess
          tls: true
          ...
      - securityScan
      ...
    }
```

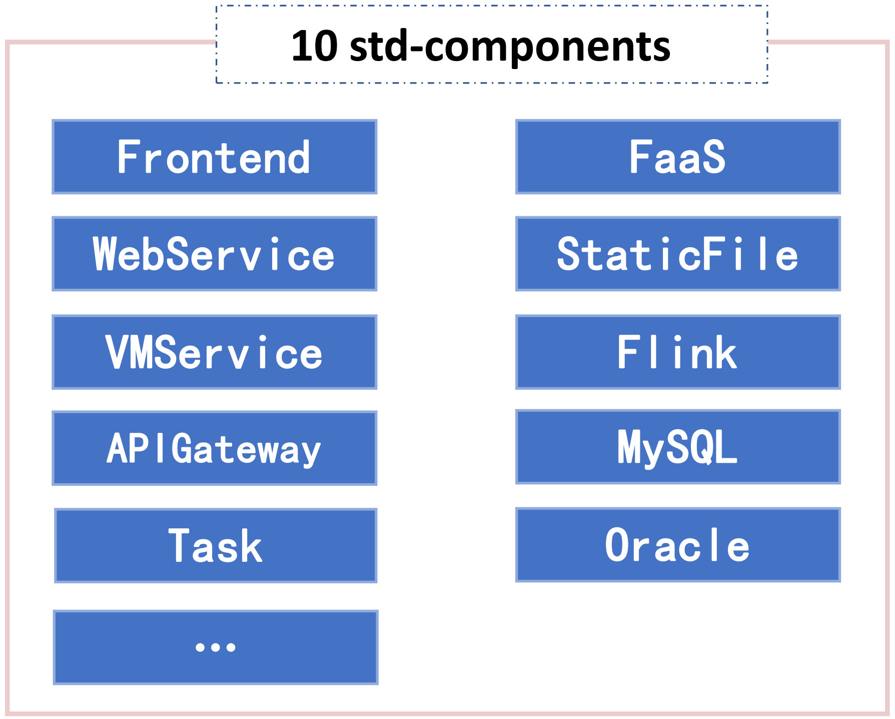
Define APIServer workload base on k8s的deployment

Define route trait for APIServer

Define a route with internet access will automatically initiate security scanning and open DNS for public access.

Standardzation | App Modeling

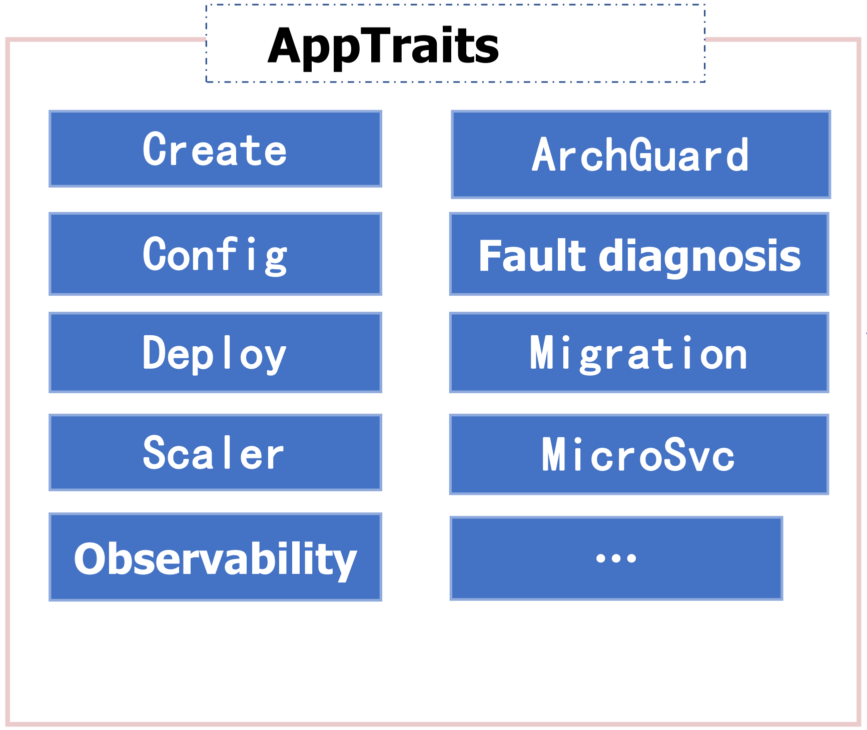
- Application developers select workloads based on business application scenarios.
- For example, for a backend service that accepts API requests, WebService can be used.



- **Frontend**: Developed using frontend frameworks like React, runs through containers, and can be accessed through domain names.
- **WebService**: Developed using cloud frameworks, accepts API requests, has no state, and runs through containers.
- **APIGateway**: A microservice gateway with full capabilities, supporting configuration and custom extensions, and can be accessed through domain names.
- **Task**: Provides standardized and easy-to-use "one-time" and "periodic" tasks, but does not provide API access.
- **FaaS**: Provides standardized and easy-to-use function-as-a-service.
- **StaticFile**: Provides standardized and easy-to-use static file server capabilities, supports OSS and CDN.
- **Flink**: Implemented based on Flink on Kubernetes, with low threshold for using Flink capabilities.
- **MySQL** and **Oracle**: Databases providing application, script publishing, and observability.
- **VMService**: Implements ACS virtual machine publishing using Kubernetes mechanisms, with standardized processes and low usage thresholds.
- **...**

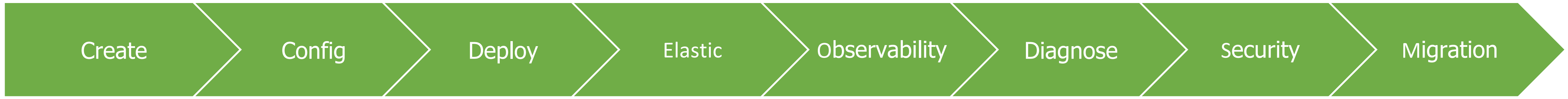
Standardzation | Ops Capabilities

- Traits are application management capabilities that are attached to standard workloads, carrying standardized cloud and operational automation capabilities.



- **Application creation**: Supports "one-stop" creation of cloud-native applications, service units, release units, and resources for the entire environment (DEV-ST-UAT-PRD).
- **Configuration management**: Supports versioned configuration management for full configuration hosting.
- **Gray release**: Provides standardized platform-level gray release capabilities with real-time observation during the process.
- **Elastic scaling**: Supports periodic and metric elasticity with dynamic JVM parameter adjustment.
- **Application observability**: Integrates logs, links, metrics, and alarms for observability.
- **Runtime architecture guardianship**: Dynamically updates runtime call topology and system architecture.
- **Fault diagnosis**: Quickly locates faults and guides switching and isolation.
- **Application migration**: Supports migration at the service unit level within and across systems with automation and resource recycling.
- **Microservice governance**: Provides API-level control, observable traffic governance, and transaction management.

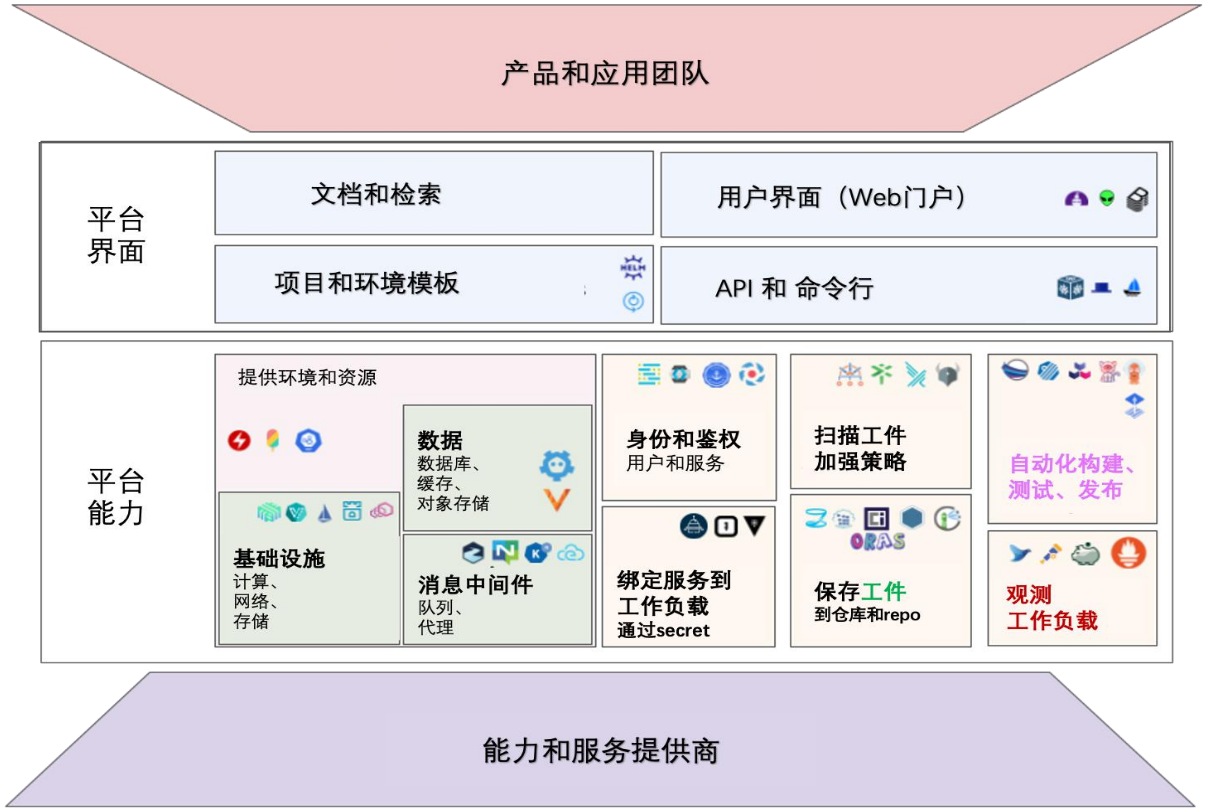
SOP(Golden Path)



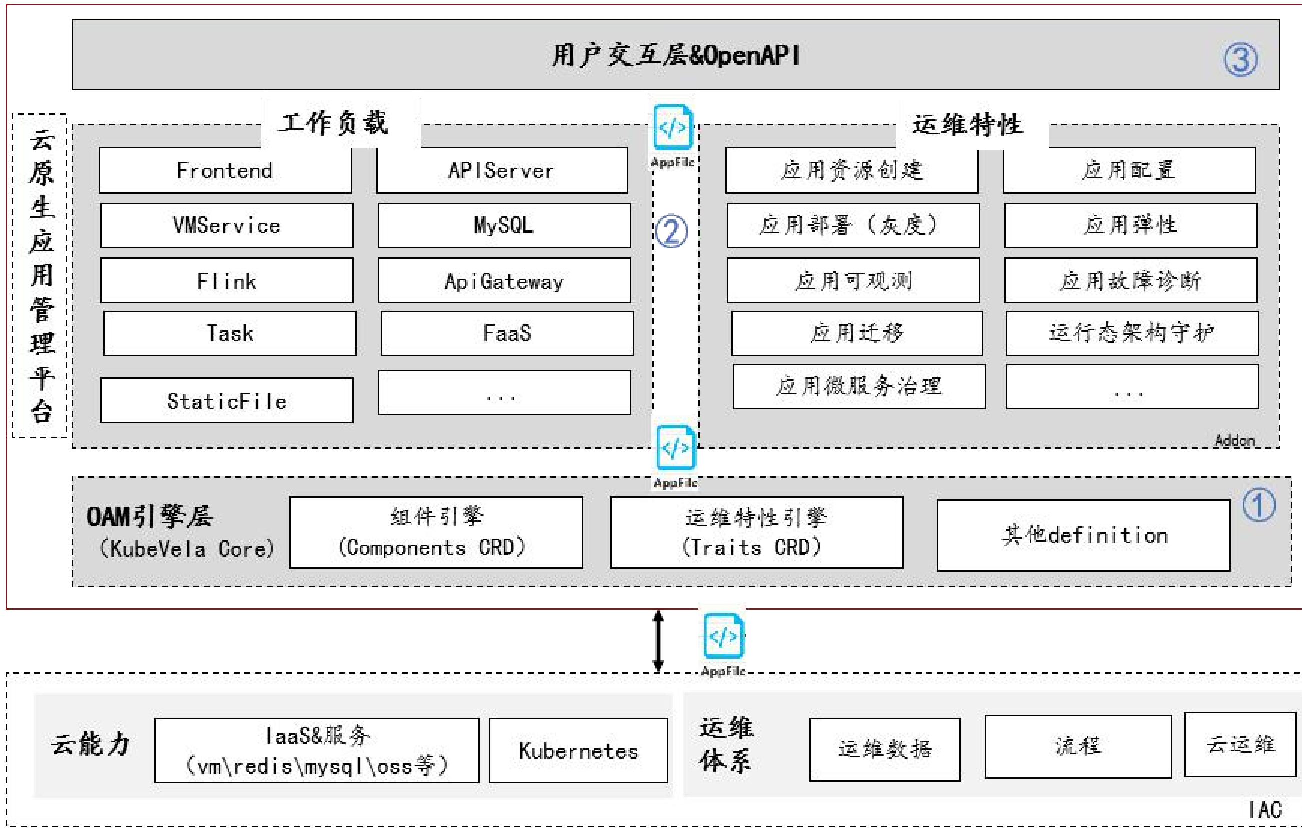
- Create App by Type
- Config App with Version
- Deploy with Policy(Canary, Rolling, Assembly)
- AppCost with Elastic
- Observability with telemetry std-out
- Issue diagnose and performance profile,
- Security Policy As Code(IAM)
- Migration Automation

Reference Architecture

The platform serves as a bridge from the underlying "capability provider" to platform users such as application developers, and implements and executes the practices required for security, performance, cost governance, and consistent experience in the process.



RefArch CNCF Platform
Engineering White-Paper



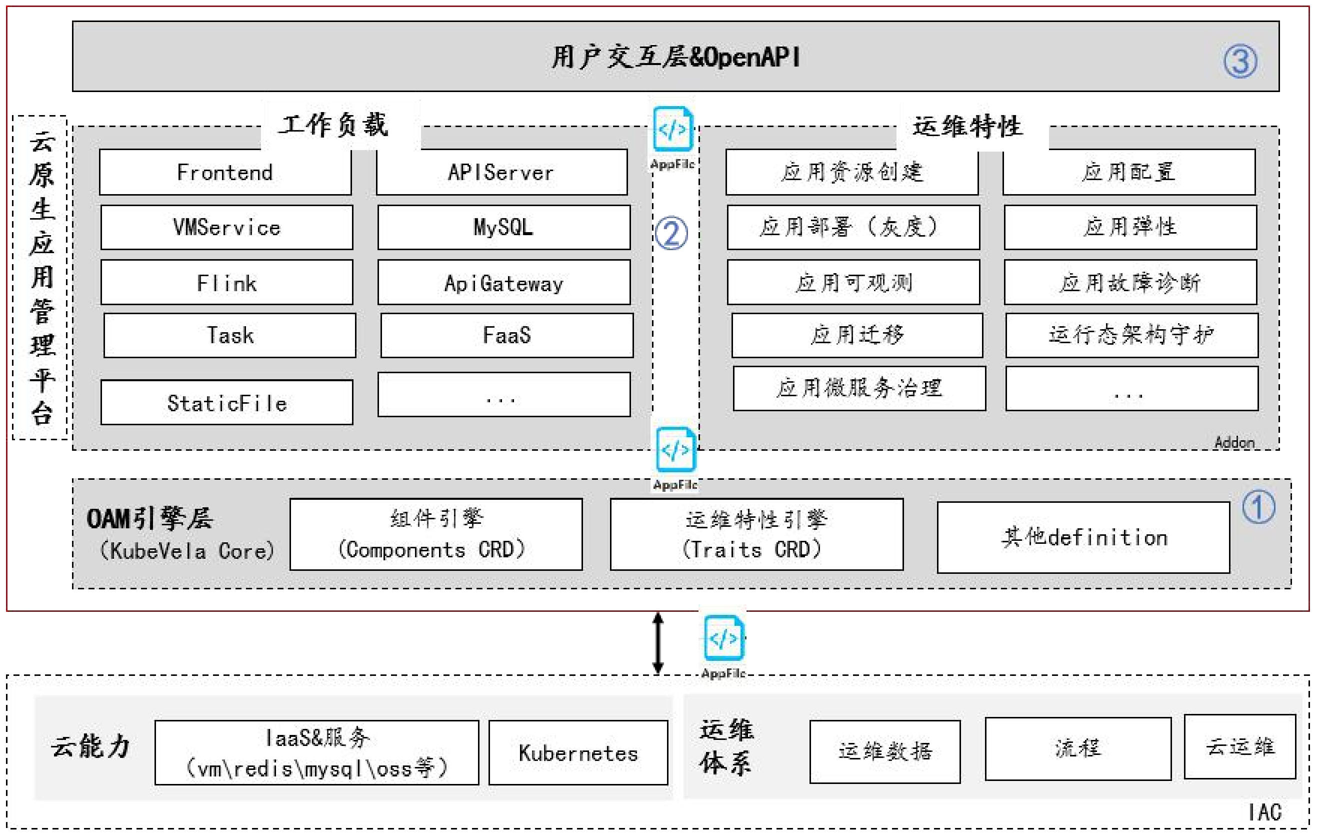
TargetArch
CMB Cloud App Platform

Team Topology | Reverse Conway Manoeuvre

Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.

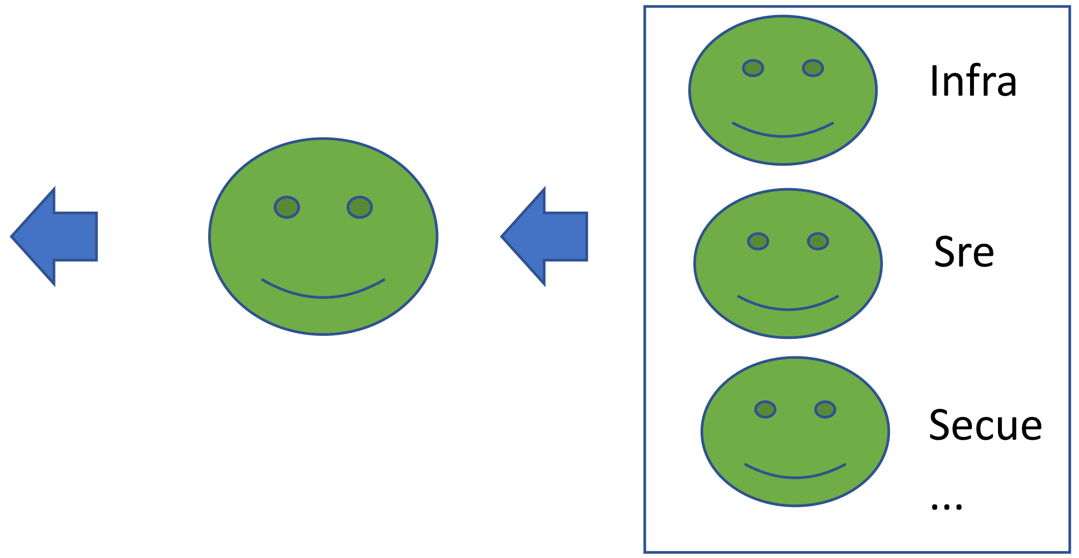
Conway's law

Product Architecture



Team Topology (PlatformTeam)

Team Topology (Enabling Teams)



"Adjusting logic organizational structure for software (product) architecture"

Reverse Conway Manoeuvre

Benefit

Complexity

Stability

Efficiency

AppManagement
Hrs->Mins
（Rookie)

LegacyTool
All in One

OpsTime
Down

MTTR
Down

AdoptionRate
PRD>63%
(Half-Year)

Converage
>5500 Devs

Capabilities Matrix

Trait / compon	Init	Config	CI	Deploy	Elastic	MicroSvc	Migration	MultiEnv	Observe	Dignostics	Arch Guard	...
Frontend	✓	✓	✓	✓*	✓		✓	✓	✓	✓	✓	
APIServer	✓	✓	✓	✓*	✓	✓	✓	✓	✓	✓	✓	
VMService	✓	✓	✓	✓*	✓		✓	✓	✓	✓	✓	
MySQL		✓	✓	SQL				✓	✓	✓	✓	
Oracle		✓	✓	SQL				✓	✓	✓	✓	
Flink	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	
APIGateway	✓	✓	✓	✓*	✓	✓	✓	✓	✓	✓	✓	
Task	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	
FaaS	✓	✓	✓	✓					✓			
StaticFile	✓	✓	✓	✓					✓			
...												

KubeCon | CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

Platform Engineering Engine （KubeVela）

Speaker

Qingguo Zeng

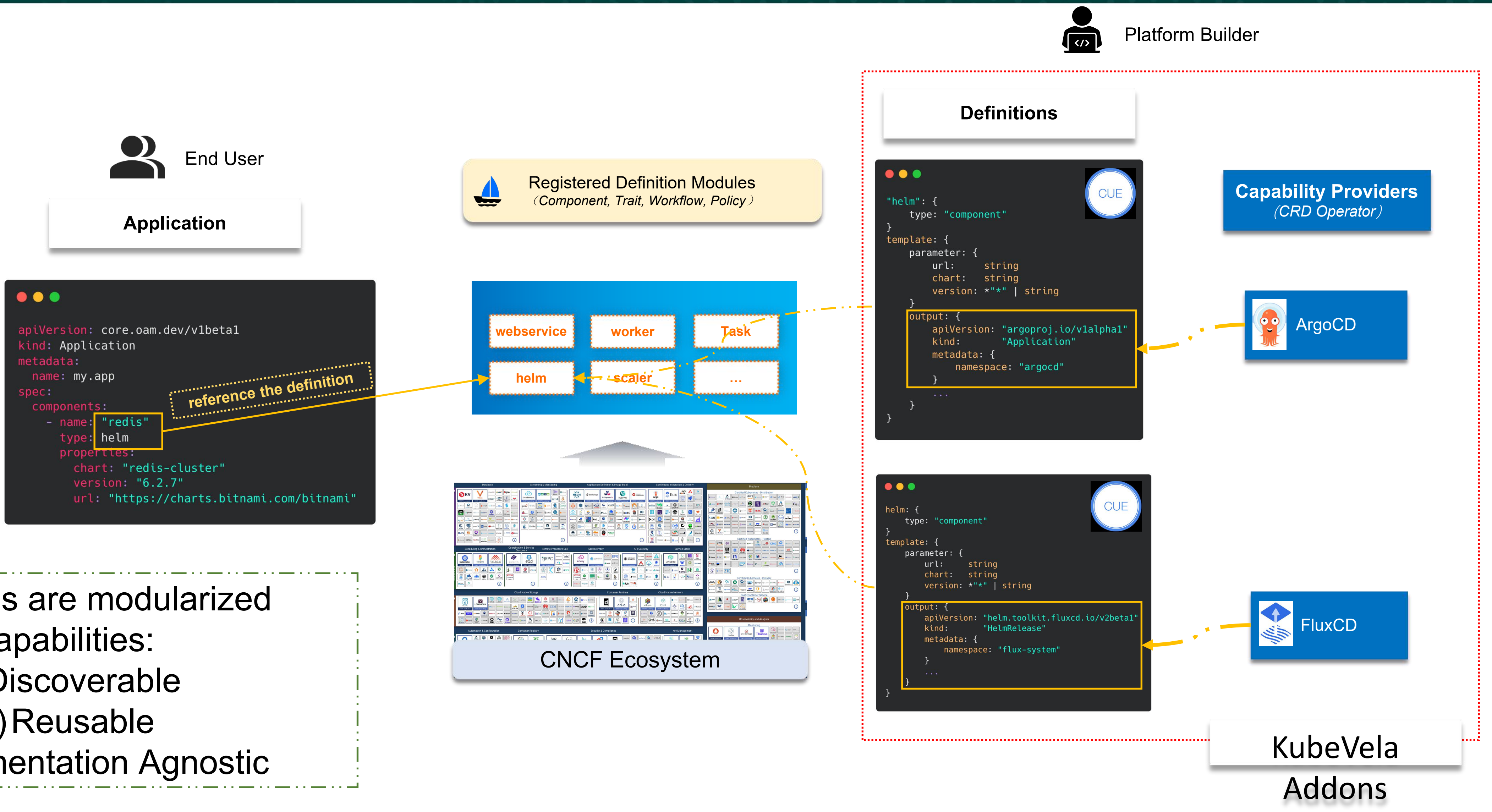
Alibaba Cloud(阿里云智能)

Senior Engineer

Qingguo Zeng (nickname: Yueda) is a Senior Engineer at Alibaba Cloud, currently responsible for OAM/KubeVela products and open source community evangelism, has been engaged in cloud-native, application delivery, observability, open-source field research and practice for many years, and is committed to promoting cloud-native application standardization and cloud-native technology "Personalization".

How KubeVela extension works

KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2023



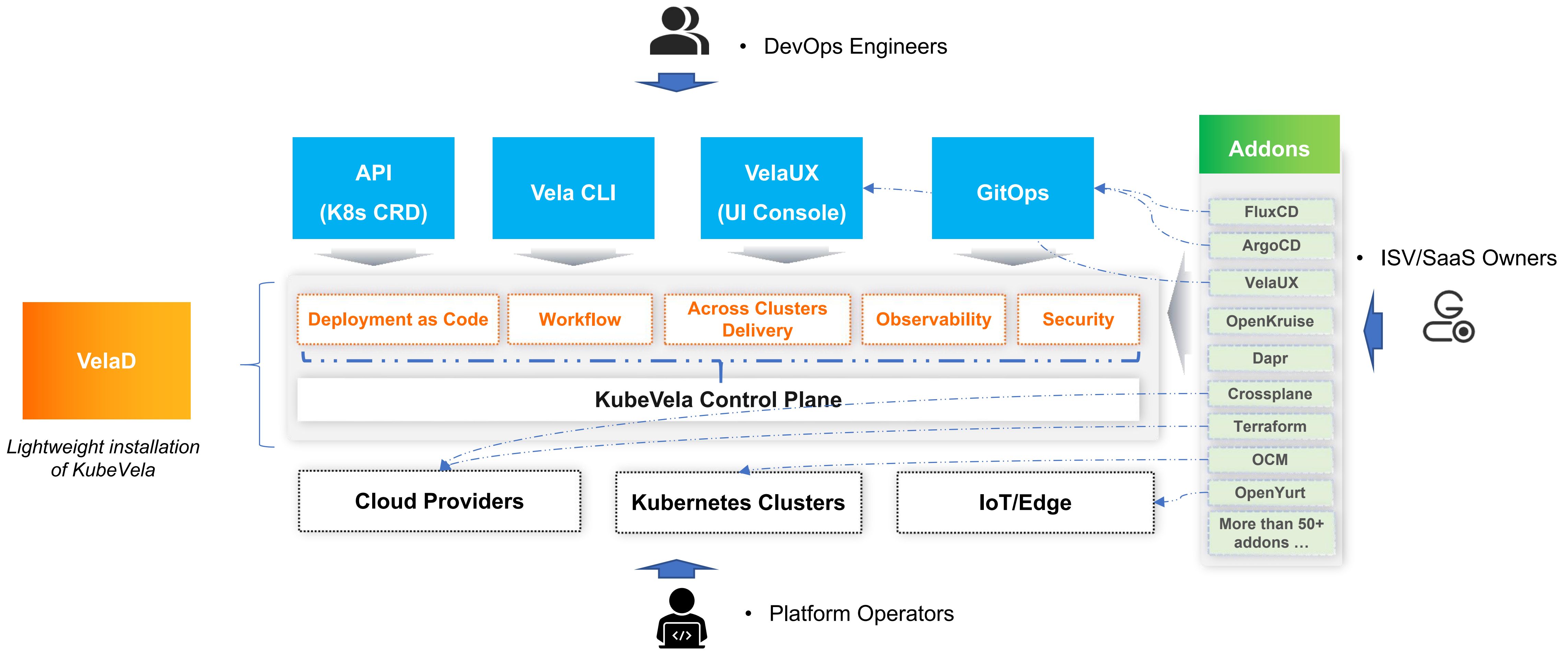
```
apiVersion: core.oam.dev/v1beta1
kind: Application
metadata:
  name: my.app
spec:
  components:
    - name: "redis"
      type: helm
      properties:
        chart: "redis-cluster"
        version: "6.2.7"
        url: "https://charts.bitnami.com/bitnami"
```

```
"helm": {
    type: "component"
}
template: {
    parameter: {
        url:     string
        chart:   string
        version: *"*" | string
    }
    output: {
        apiVersion: "argoproj.io/v1alpha1"
        kind:       "Application"
        metadata: {
            namespace: "argocd"
        }
        ...
    }
}
```

```
helm: {
    type: "component"
}
template: {
    parameter: {
        url:     string
        chart:   string
        version: *"*" | string
    }
    output: {
        apiVersion: "helm.toolkit.fluxcd.io/v2beta1"
        kind:       "HelmRelease"
        metadata: {
            namespace: "flux-system"
        }
        ...
    }
}
```

Definitions are modularized capabilities:
1) Discoverable
2) Reusable
3) Implementation Agnostic

How KubeVela built on top of the extensibility



KubeVela Demo

KubeCon | CloudNativeCon

OPEN SOURCE SUMMIT

China 2023