Ant Group's Platform Engineering Practice at Scale

Dayuan Li
AntGroup

Agenda

01 Background

02 Classic PaaS is no longer applicable

03 Will platform engineering be the answer?

04 Our practice

05 Architecture and technologies

06 Product and culture

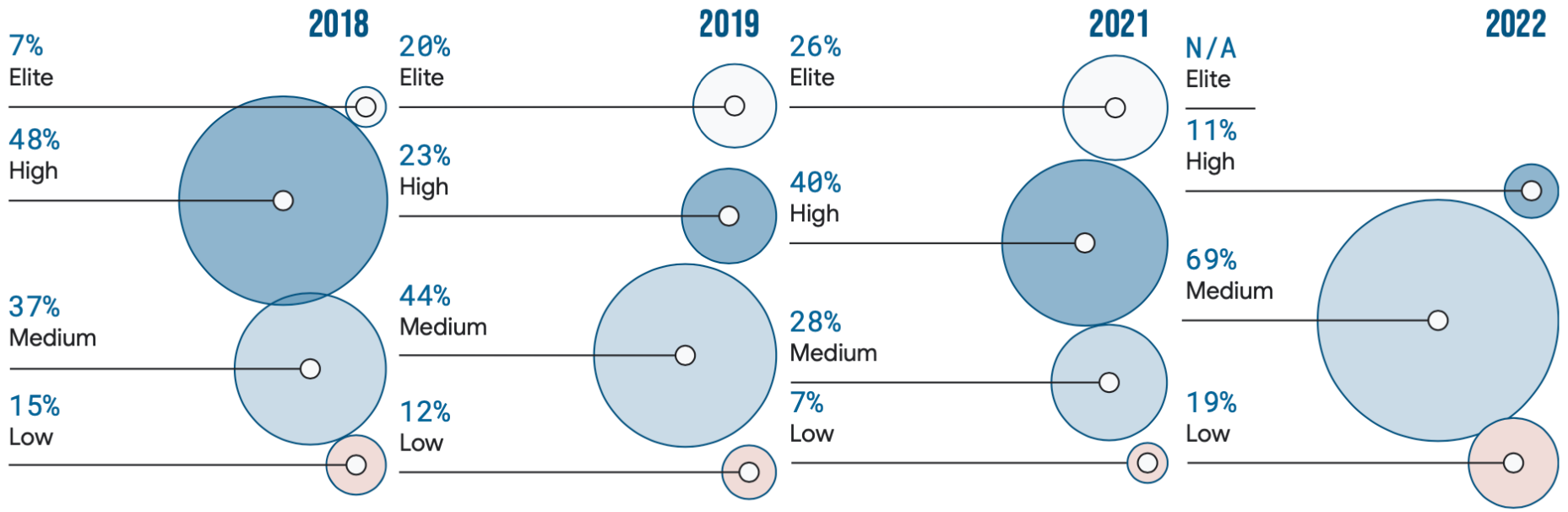
07 Challenges

08 Practice in AntGroup and Other Companies

Background

Industry trends

Software delivery and operational performance[1]



A clear drop in high and elite performers. Delivery and operational performance is decreasing.

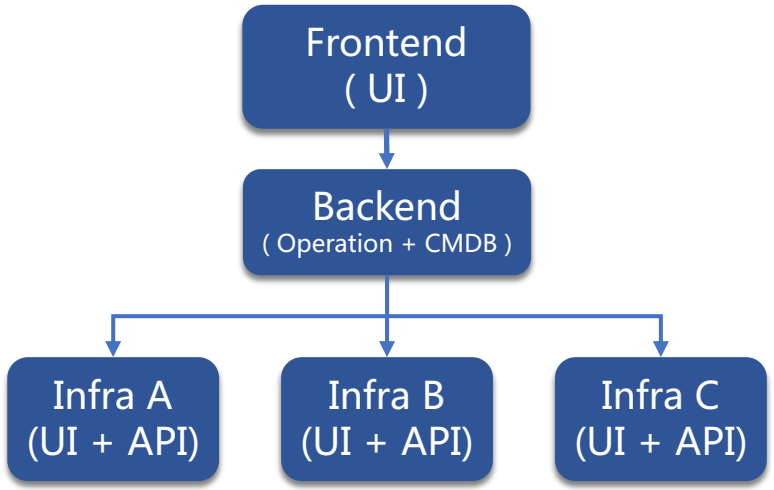
[1] Google 2022 Accelerate State of DevOps Report

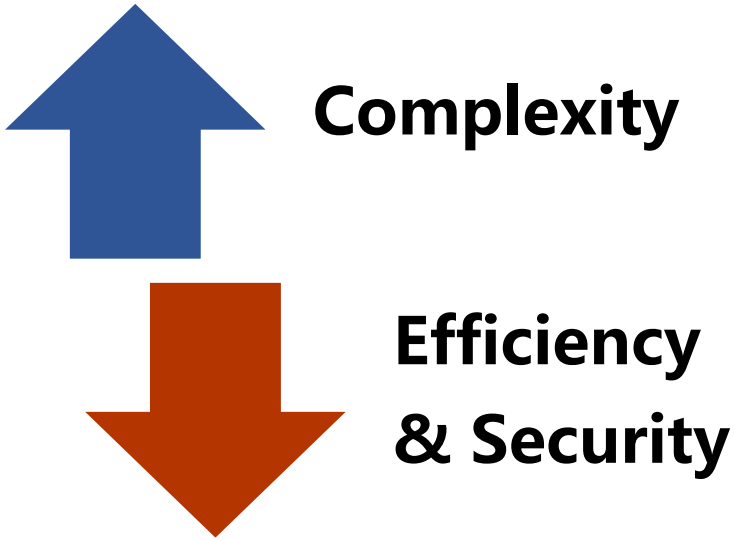
Background

What we already have? Compared with autonomous driving

Level		Description	Products
L0	No Automation	Manual operation	Hands🤘
L1	Operator Assistant	Limited automation operations through scripts	Shell
L2	Partial Automation	Solve specific problems in certain scenarios by a platform	Heroku
L3	Conditional Automation	Making simple decisions during declarative operations. Still require human override	Kubernetes
L4	High Automation	Fully declarative operations, in most cases, do not focus on the process	AIOps
L5	Full Automation	Do not require human attention. No Ops, No SRE	?

Classic PaaS is no longer applicable





Developer

- High cognitive load
- High lead time for changes

Platform

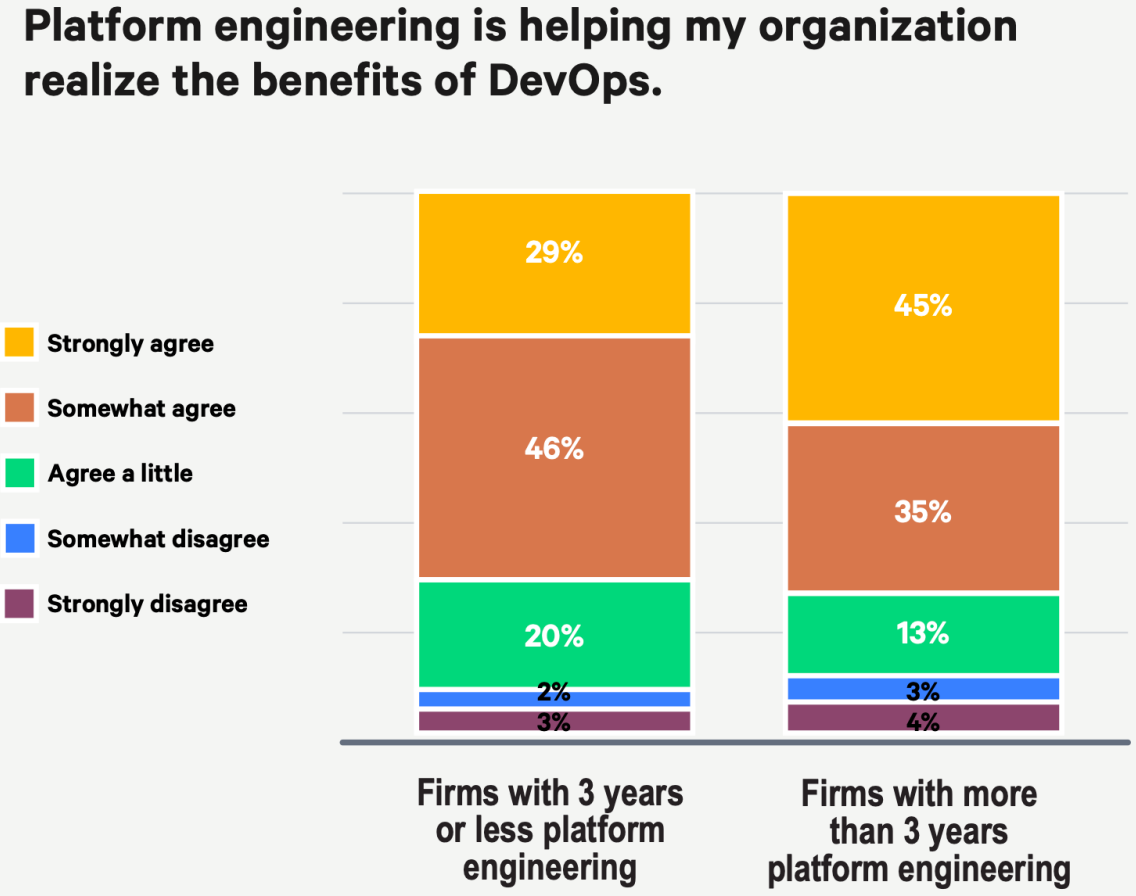
- Increasing types and quantity of demands
- Increasing communication cost with numerous infrastructure teams

Security

- Multiple discrete infrastructure platforms lead to a high security risk scope
- Declarative operations reconciliation is powerful but dangerous

Will platform engineering be the answer?

Platform engineering is the **discipline** of designing and building Internal Developer Platforms to address the **conflict** between the increasing complexity of operations and the need for enterprise efficiency and security.



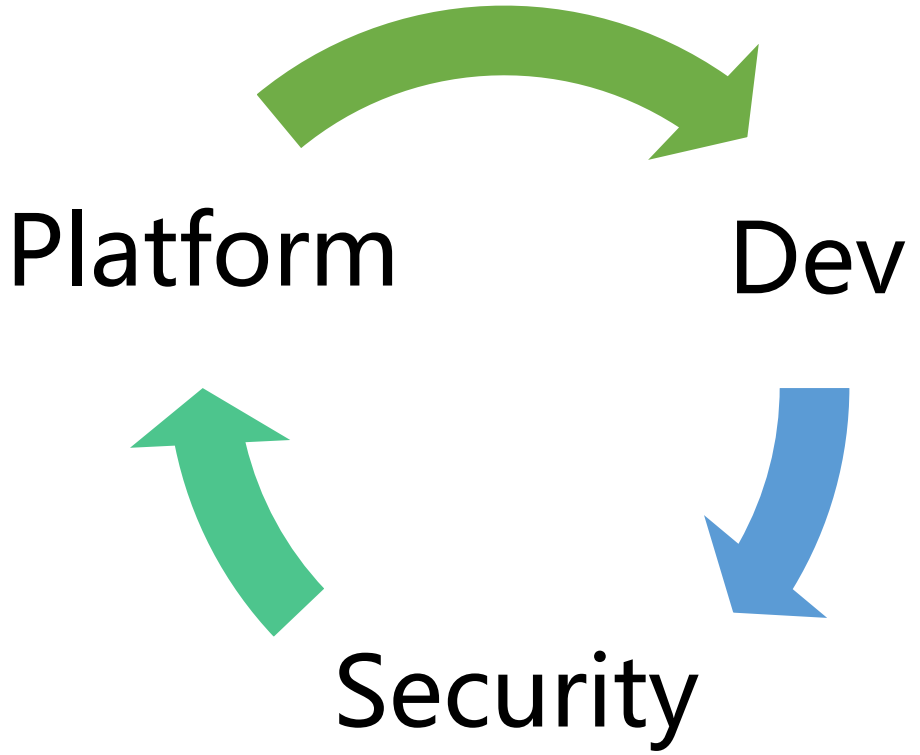
DevOps is not Dead

Platform engineering is the new DevOps

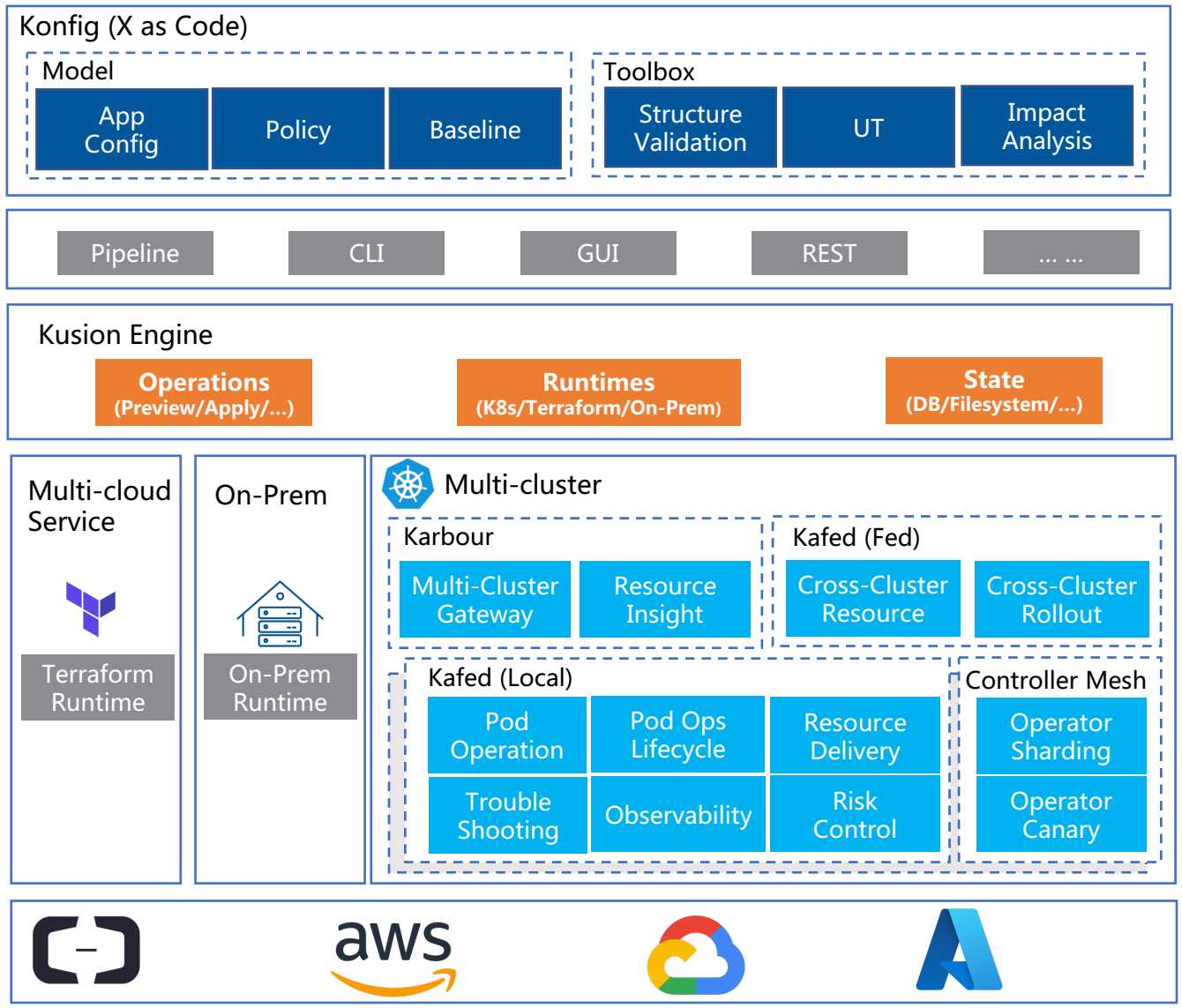
Puppet 2023 State of DevOps Report

Our thinking

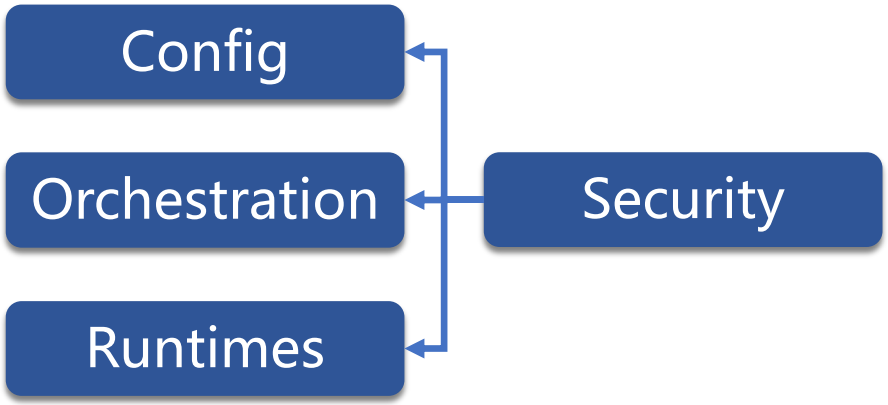
- **Reduce user cognitive load:** Masking infrastructure complexity by high level abstracting
- **Transforming production relations:** Developers can solve their own needs through self-service tools provided by PaaS
- **Policy and Shift risks left:** Guarantee security at the earliest stages with policy codes



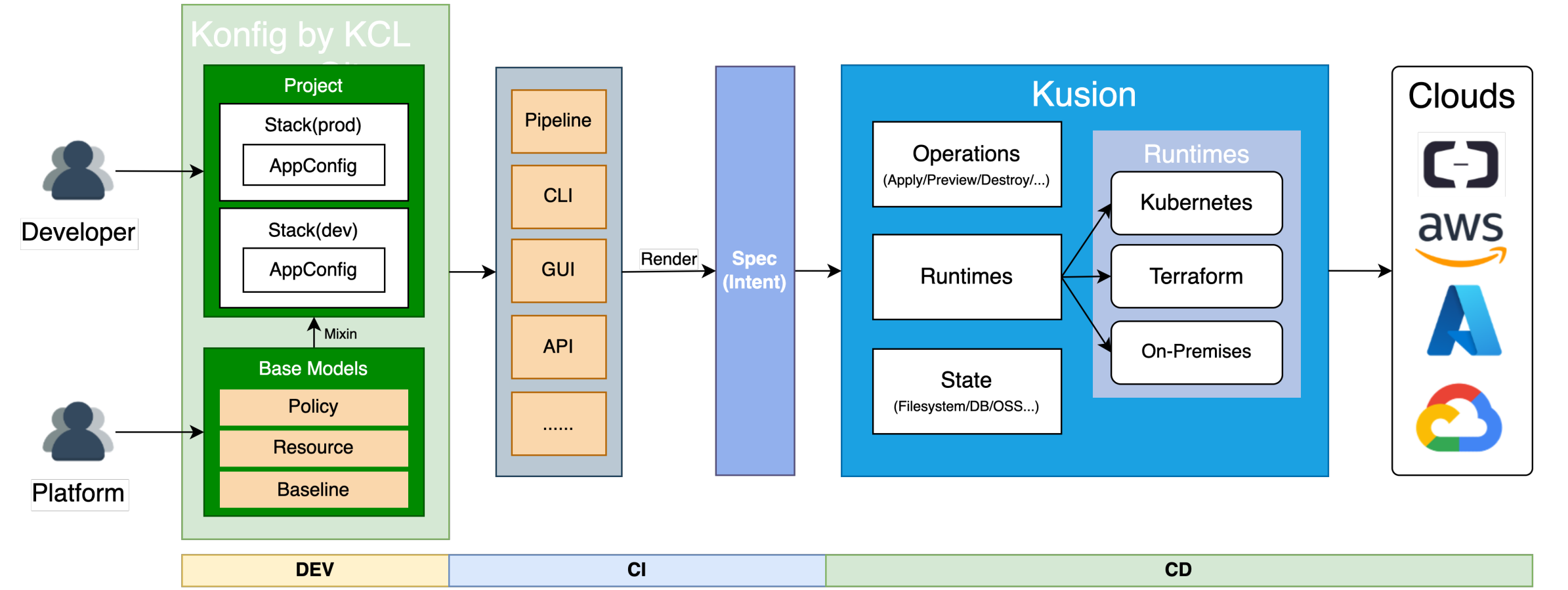
Our practice — KusionStack Arch

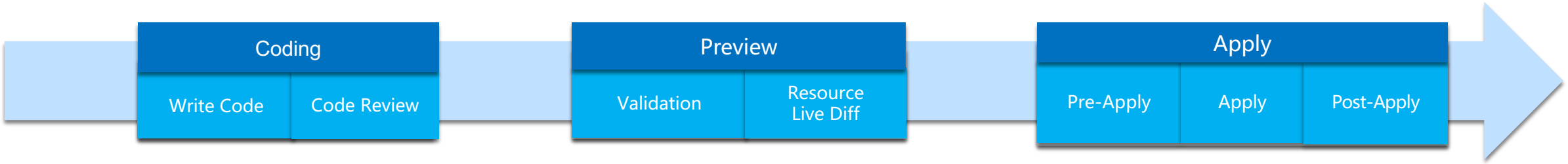


Help you to build your Internal Developer Platform more effectively and safely



Our practice — KusionStack Workflow





Architecture and technologies

Konfig: A git repo stores operation intents and serves as collaboration platform between Platform and Developers.

```
cd appops/wordpress && tree
.
├── README.md
├── base                            // Common configuration for all stacks
│   └── base.k
├── dev                             // Stack directory
│   ├── ci-test                     // Test data
│   │   ├── settings.yaml
│   │   └── stdout.golden.yaml
│   ├── kcl.yaml                    // Compile configuration for current stack
│   ├── main.k                      // App Configs matained by App Dev
│   ├── platform.k                  // App Configs matained by Platform Dev
│   └── stack.yaml                  // Stack metadata
└── project.yaml                    // Project metadata
```

Platform

Developers

Architecture and technologies

KCL: a DSL for operation configurations and policies

```
import base.pkg.kusion_models.kube.frontend

appConfiguration: frontend.Server {
    image = "howieyuen/gocity:latest"
}
```

```
schema ServerBackend[inputConfig: server.Server]:
    """ServerBackend converts the user-written front-end model `Server` into a
    collection of kubernetes resources and places the resource collection into
    the `kubernetes` attribute.
    """
    mixin [
        # Resource builder mixin
        mixins.NamespaceMixin,
        mixins.ConfigMapMixin,
        mixins.SecretMixin,
        mixins.ServiceMixin,
        mixins.IngressMixin,
        mixins.ServiceAccountMixin,

        # Monitor mixin
        pmixins.MonitorMixin
    ]

    # Store the input config parameter, ensure it can be seen in protocol and m
    config: server.Server = inputConfig
    # Workload name.
    workloadName: str = "{}{}".format(metadata.__META_APP_NAME, metadata.__META
    # App variable contains labels, selector and environments.
    app: utils.ApplicationBuilder = utils.ApplicationBuilder {}
    # Main containers and sidecar contrainers.
    mainContainer: {str:}
    sidecarContainers?: [{str:}]
    initContainers?: [{str:}]

    if config.mainContainer:
        assert config.image, "config.image must be specified and can't be empty
        # Construct input of converter using the volumes.
        mainContainer = utils.VolumePatch(config.volumes, [utils.ContainerFront
            **config.mainContainer
            if config.mainContainer.useBuiltInEnv:
                env += app.envs
            name = config.mainContainer.name or "main"
            image = config.image
            resource = config?.schedulingStrategy?.resource
        ])]?[0]

    if config.sidecarContainers:
        sidecarContainers = utils.VolumePatch(config.volumes, [utils.ContainerF

    if config.initContainers:
        initContainers = utils.VolumePatch(config.volumes, [utils.ContainerFron

    # Construct workload attributes.
    workloadAttributes: {str:} = {
        metadata = utils.MetadataBuilder(config) | {
            name = workloadName
        }
        spec = {
            replicas = config.replicas
            if config.useBuiltInSelector:
                selector.matchLabels: app.selector | config.selector
            else:
```

Spec

Deployment

Service

ConfigMap

Database

Monitor

… …

Front-end model (Developer)

Back-end model (Platform)

K8s/clouds/on-prem resources

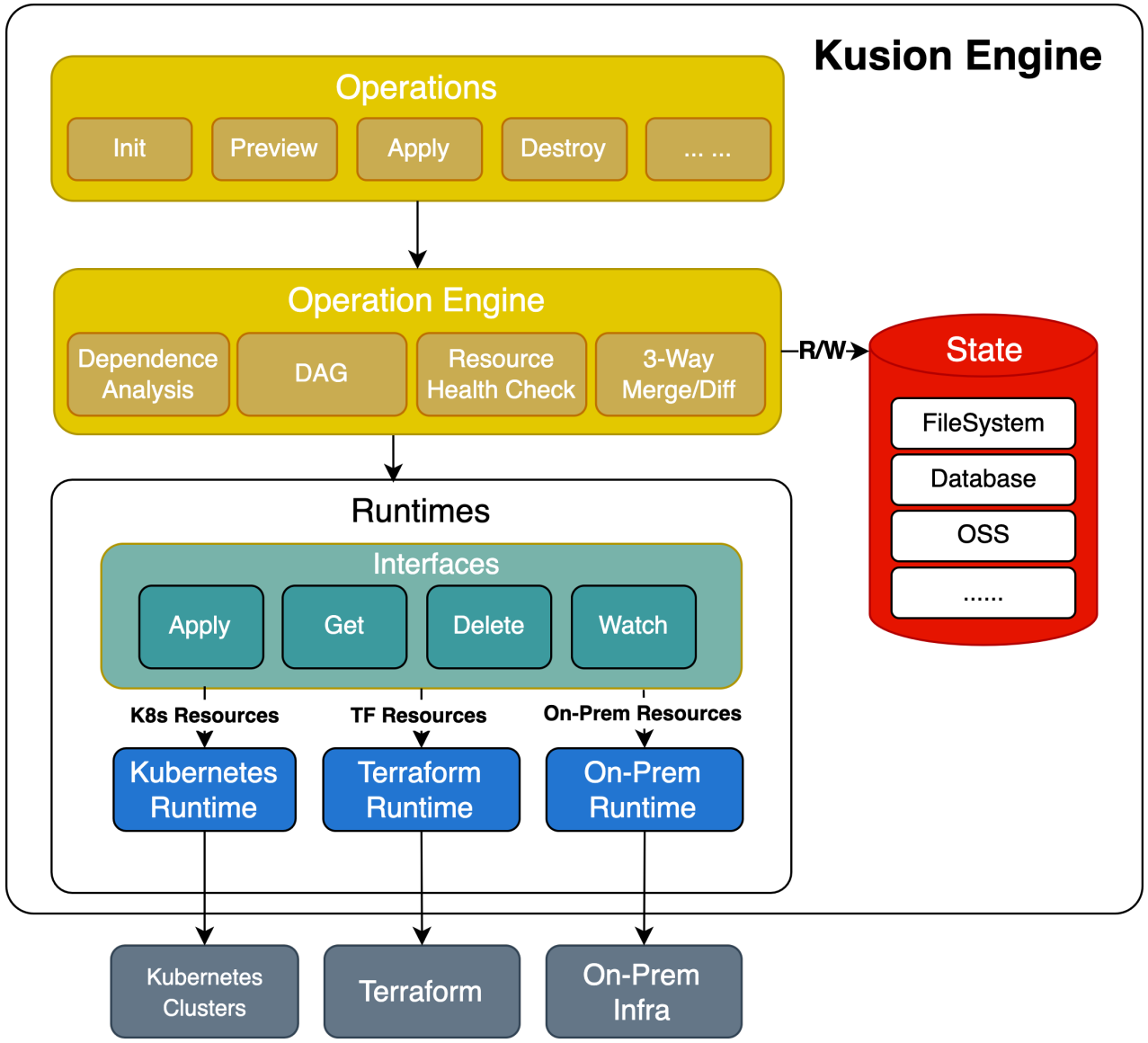
Architecture and technologies

Kusion Engine: Platform engineering engine, responsible for all operations

Operations: Provide core capabilities such as resource management, orchestration, and live-diff for all Kusion operations commands.

Runtimes: represents infrastructures managed by Kusion, which interacts with heterogeneous infrastructure through a unified interface

State: The mapping of real resources to Kusion used to resource management



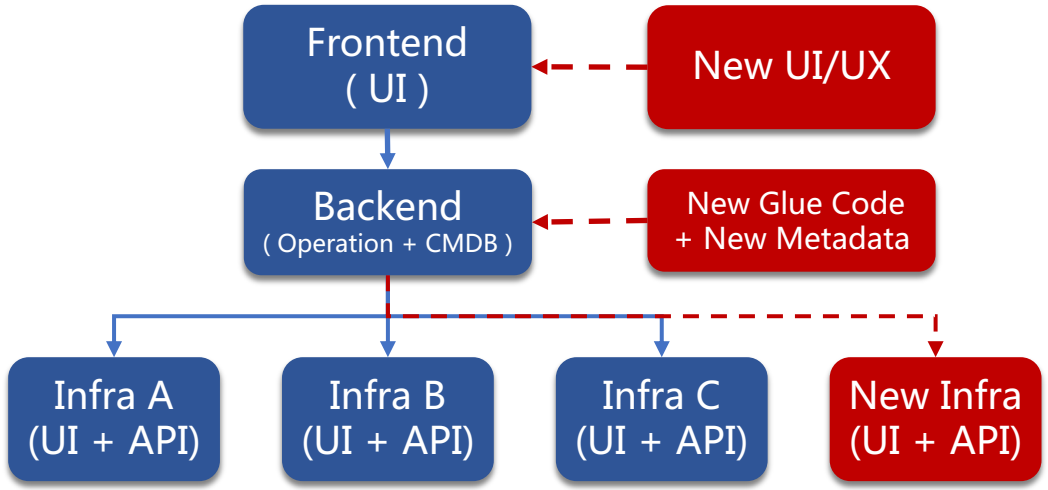
Highlights

- **Application-Centric:** Managing all application operations in one place, in a unified way
- **Enable Self-Service:** Developers fulfill its own needs by using the capabilities provided by the platform
- **Shift left security:** Guarantee security at the earliest stages to make operation more confidence
- **Kubernetes-friendly:** Provide features such as observability and health checks for K8s resources to improve the user experience

Use Case — Support a new kind of IaaS resource

Classic PaaS

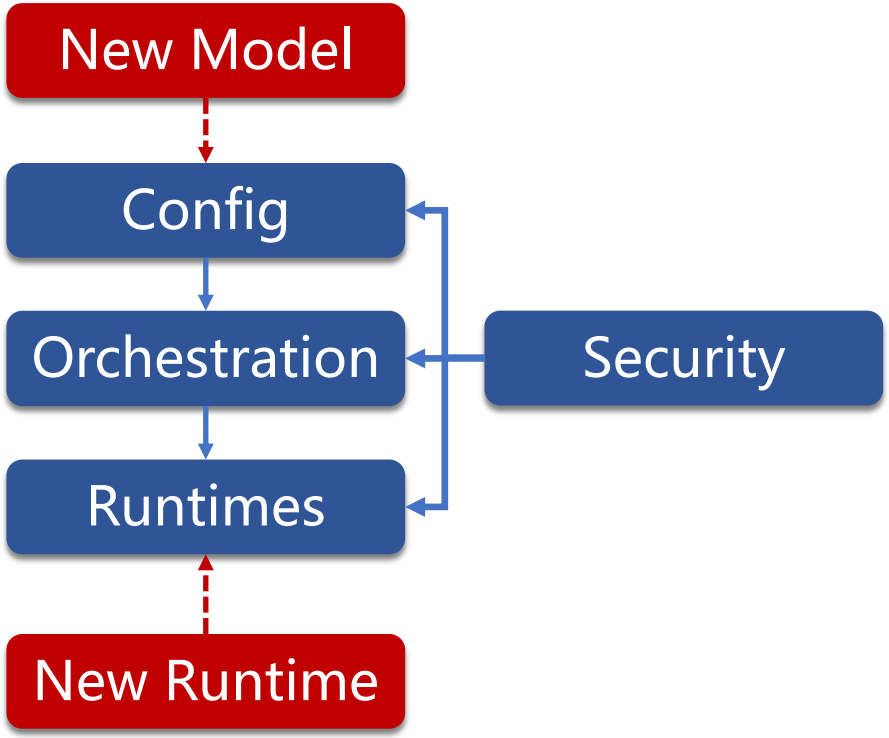
1. **Platform** design a new UI/UX
2. **Platform** add the new resource metadata in CMDB and write codes to invoke the new Infra API
3. **Platform** abstract infra details to user's perspective API



Use Case — Support a new kind of IaaS resource

KusionStack

1. **Infra/SRE team** design a new config model
2. **Infra team** add a new Runtime (or existed TF provider)



Transform production relations through self-service tools provided by Platform

Use Case — Code Detail

Developer’s Config

1. Application-Centric
2. Reduce cognitive load

```
# dev/main.k
import base.pkg.kusion_models.kube.frontend
import base.pkg.kusion_models.kube.frontend.storage

# The application configuration in stack will overwrite
# the configuration with the same attribute in base.
# And main.k is for the configurations in concern of application developers.

# defination of wordpress application frontend model
wordpress: frontend.Server {
    # specify application image
    image = "wordpress:4.8-apache"

    # use cloud database for the storage of wordpress
    database = storage.DataBase {
        # choose aliyun_rds as the cloud database
        dataBaseType = "aliyun_rds"
        dataBaseAttr = storage.DBAttr {
            # choose the engine type and version of the database
            databaseEngine = "MySQL"
            databaseEngineVersion = "5.7"
            # choose the charge type of the cloud database
            cloudChargeType = "Serverless"
            # create database account
            databaseAccountName = "root"
            databaseAccountPassword = option("db_password")
            # create internet access for the cloud database
            internetAccess = True
        }
    }
}
```

Use Case — Code Detail

Platform’s Config

1. Infrastructure-related configuration
2. Policy
3. Dependency

```
aliyunVPC = alicloud.AlicloudVPC {
    vpc_name = _alicloudResourceName
}
provider = [*provider, alicloud_backend.VPCRender(aliyunVPC).provider]

aliyunVswitch = alicloud.AlicloudVswitch {
    vpc_id = _alicloudDependencyPrefix + alicloud_config.alicloudVPCMeta.type + ":"
    + aliyunVPC.vpc_name + ".id"
    vswitch_name = _alicloudResourceName
    zone_id = alicloud_config.alicloudProviderMeta.region + "-h"
}
provider = [*provider, alicloud_backend.VswitchRender(aliyunVswitch).provider]

if config.database.dataBaseAttr.cloudChargeType == "Serverless":
    assert config.database.dataBaseAttr.databaseEngine == "MySQL",
    "databaseEngine must be set to MySQL when creating a serverless instance"
    aliyunDBInstance = alicloud.AlicloudDBInstance {
        engine = config.database.dataBaseAttr.databaseEngine
        engine_version = config.database.dataBaseAttr.databaseEngineVersion
        instance_type = "mysql.n2.serverless.1c"
        instance_charge_type = config.database.dataBaseAttr.cloudChargeType
        instance_name = _alicloudResourceName
        vswitch_id = _alicloudDependencyPrefix + alicloud_config.alicloudVswitchMeta.type
        + ":" + aliyunVswitch.vswitch_name + ".id"
        category = "serverless_basic"
        security_ips = ["0.0.0.0/0"]
        serverless_config = [alicloud.serverlessConfig{}]
    }
```

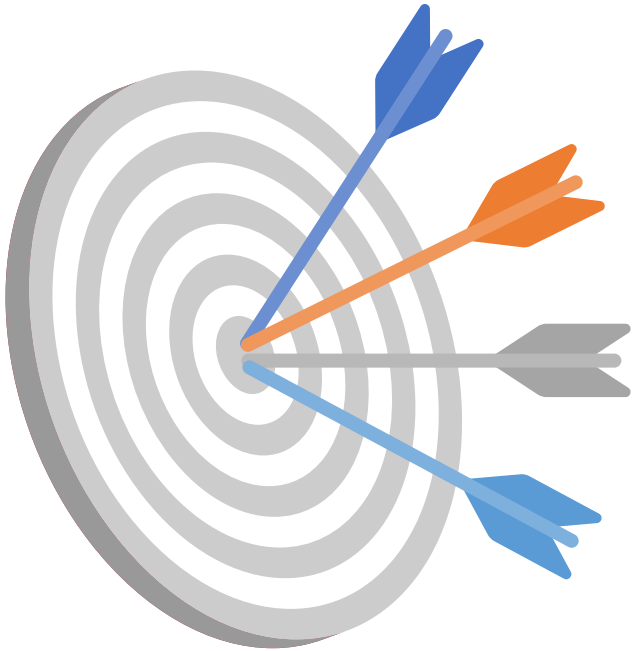
Demo

Video

https://kusionstack.io/docs/user_docs/getting-started/usecases/deliver-the-wordpress-application-on-kubernetes-and-clouds

Product and culture

Developer’s Value



User research

- Surveys
- Interviews
- Focus groups

Data analysis

- User behavior analysis
- Data mining analysis
- A/B testing analysis

Marketing

- Periodic release note
- User feedback report
- Evangelical promotion

Leadership support

- Value orientation
- Industry trends
- benefits the entire organization

Challenges

- Dev experience
- User-friendly document
- New user onboarding

- Configuration correctness assurance
- CI pipeline performance
- CMDB REST Service

- Code GPT
- Abnormal Detection
- Intelligent Decision

Practice in AntGroup and Other Companies

1K/day	10K+/day	1 : 9	100K+
Pipelines	KCL Compilations	Plat : Dev	Commits
600+	5.7K+	1.2M+	10M+
Contributors	Projects	KCL Codes	YAML

Adopted by

HUAWEI

KYLIGENCE®

有赞

Welcome to join us

- Web Site
  - https://kusionstack.io/
- Github
  - https://github.com/KusionStack/kusion
  - https://github.com/KusionStack/KCLVM
  - https://github.com/KusionStack/konfig
- Twitter
  - @KusionStack

WeChat

Thank You

Dayuan Li