Kusion: Intent-driven Platform Orchestrator

Dayuan Li, Forest Chen
Ant Group

About us

Forest Chen, Ant Group

Dayuan Li, Ant Group

Table of Contents

Challenges

Source: David Bell

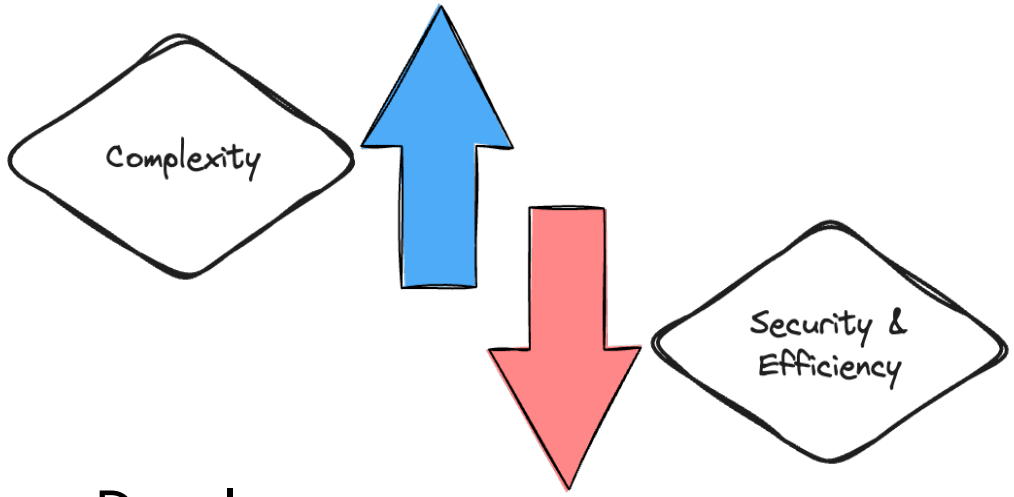
Developers during application delivery:

- How many internal systems are involved?
- How many cloud products?
- How many people do I have to ping?
- How long does it take?

Cloud-native applications are more than Kubernetes…

It's slowing down Innovation

Source: [the awesomer]



Developer

- Overloaded with (unnecessary) infrastructure knowledge

Platform

- Became a bottleneck with endless feature requests

Community

- Slowing down adoption of new technologies

Our Vision

Protocol-driven Golden Path

Agile Innovations & Value Delivery

Effective Collaborations

Kusion as Platform Orchestrator

Key Values

Tailored Golden Path

Build Golden Path tailored to your specific organizational needs

Vendor Agnostic

Write once, deliver anywhere, any cloud with a consistent workflow

Everything As Code

Codified Best Practices are key assets to achieve operational excellence

Extendable Modules

Extendable and reusable modules with developer-friendly schemas

Dynamic Config Mgmt.

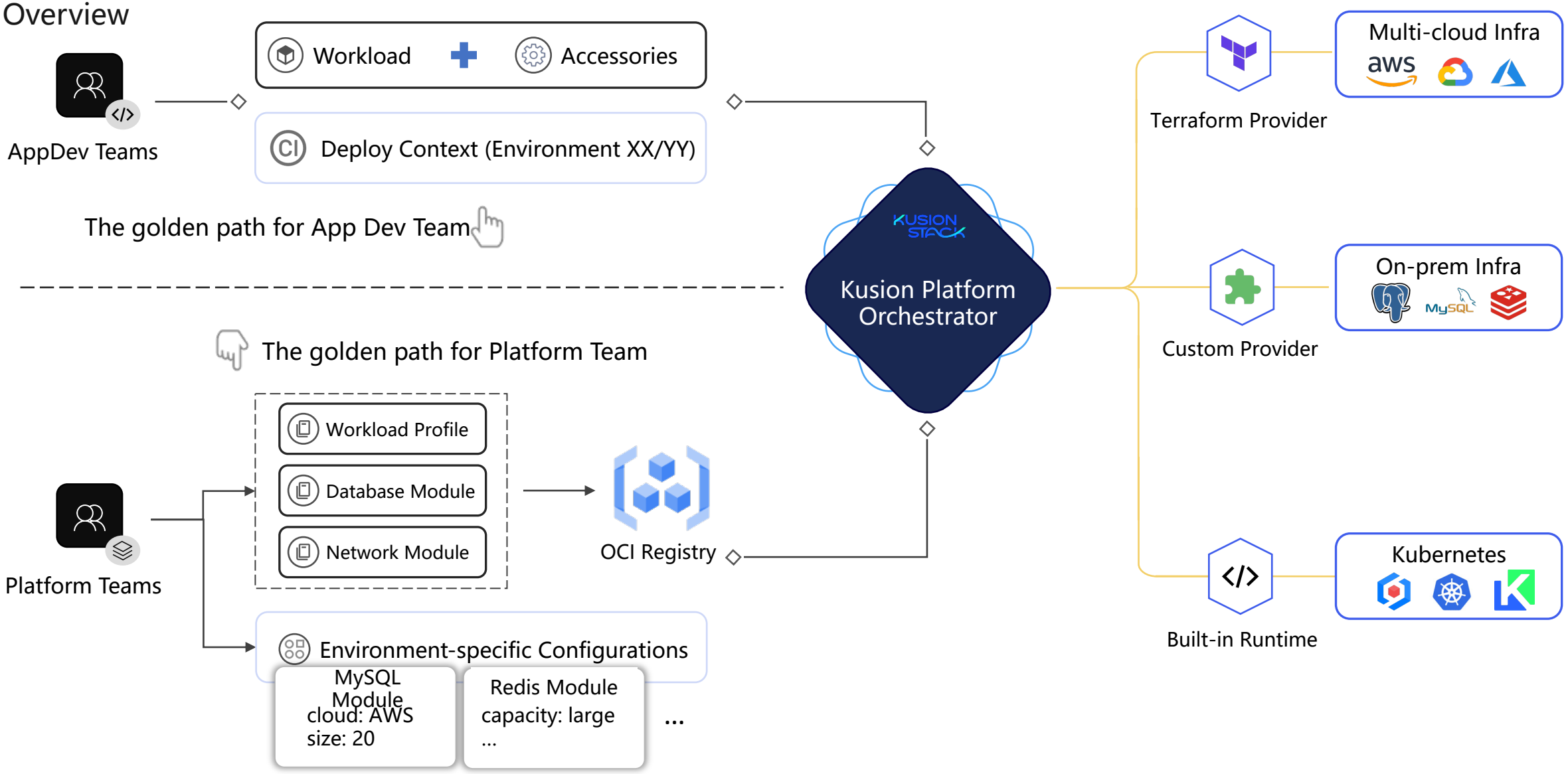
Lower burden for developers via separation of concerns

Hybrid Resource

Orchestrate resources across different runtimes

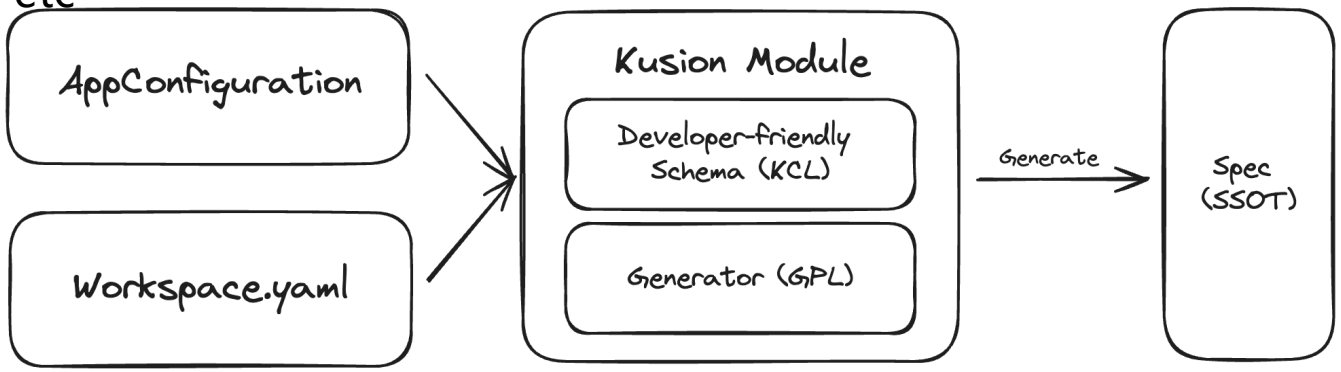
Define Golden Paths with Kusion

Overview



Kusion Modules & Workspaces

Standardized, re-usable building blocks, fronted by KCL(CNCF Sandbox) such as database, storage, monitoring, etc



Platform Configurations

- Environment and vendor-specific configurations should be managed by the platform.
- Application configurations should be decouple with platform ones so they can be written once and applied everywhere.
- The developers choose environments as they see fit.

```
accessories: {
    "mysql": mysql.MySQL {
        type: "cloud"
        version: "8.0"
    }
}
```

App Dev Config Sample

```
modules:
  mysql:
    default:
      provider: aws
      size: 20
      instanceClass: db.t3.micro
      securityIPs:
      - 10.0.0.0/18
    smallClass:
      size: 50
      instanceClass: db.t3.small
      projectSelector:
      - foo
      - bar
    largeClass:
      instanceClass: db.t3.large
      projectSelector:
      - baz
```

Module declaration

Default module config

Overriding default module config with selectors

Platform Config Sample：workspace.yaml

Application Centric Model

```
helloWorld: ac.AppConfiguration {
    workload: wl.Service {
        containers: {
            "main": c.Container {
                image: "ghcr.io/kusion-stack/samples/helloworld:latest"
                readinessProbe: p.Probe {
                    probeHandler: p.Http {
                        url: "http://localhost:80"
                    }
                }
            }
        }
    }

    # a collection of accessories that will be attached to the workload
    accessories: {
        # Built-in module
        "my-prometheus" : m.Prometheus {
            path: "/metrics"
        }
        # Built-in module
        "my-database" : d.MySQL {
            type: "cloud"
            version: "8.0"
        }
        # Customized module
        "my-customize": customizedModule {
            type: "customized"
        }
    }
    pipeline: {
        "step" : Step {
            use: "exec"
            args: ["--test-all"]
        }
    }
    dependency: {
        dependentApps: ["init-kusion"]
    }
}
```

Workload

Core part of the application, represent the compute workload configurations

Accessories

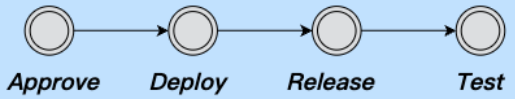
Various runtime capabilities and operational requirements that app needs to function properly

App Developer Configuration

- Enable self-service using modules that are understandable and easy to combine
- A configuration that covers the entire delivery lifecycle of the application

Pipeline

More fine-tuned control and customization on how to deliver an app



Depedency

Describes cross-application dependency.

Kusion Platform Orchestrator Workflow

Drive intent into reality

Team collaboration

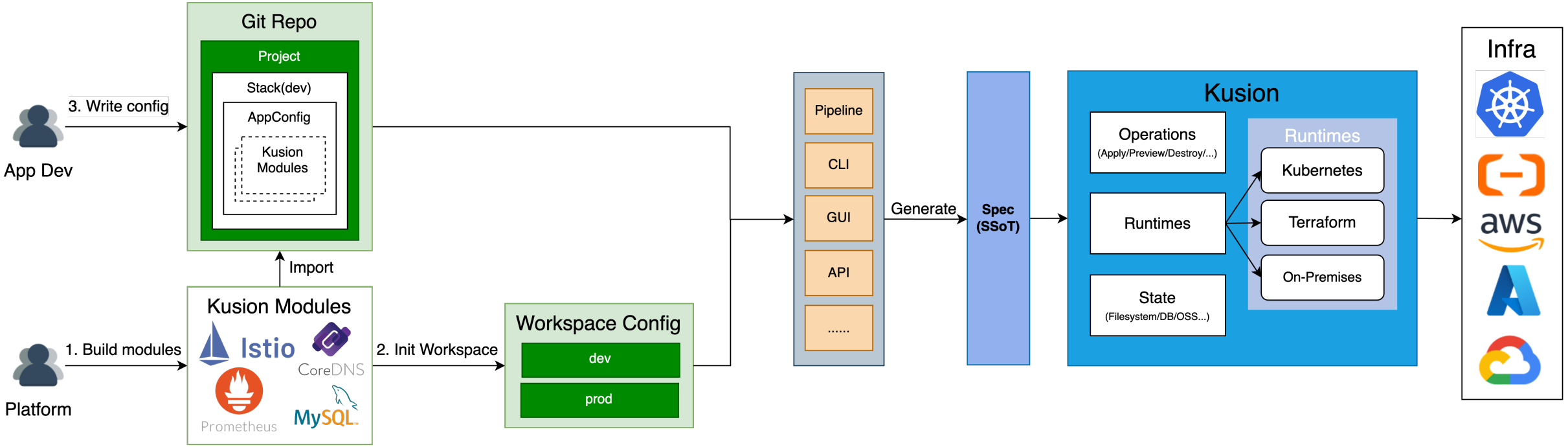
Environment-agnostic configuration enables self-service with Kusion modules

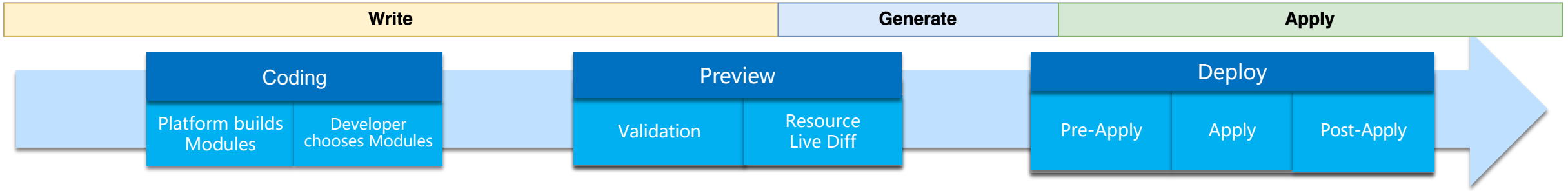
Environment variance management and platform standardization

Seamlessly integrates into current workflow

SSoT (Single Source of Truth) Management

Manage Kubernetes, Cloud and On-Premises in one place with one config file





Demo

Reference

Demo

What is Kusion doing?

1. Provision Kubernetes resources, such as Deployment, Service, Secret, etc

2. Provision cloud resources, such as DBInstance and SecurityGroup

3. Inject environment variables to the workload

Roadmap

1. Expand Kusion Module Ecosystem to meet more scenarios
    - CNCF ecosystem projects
    - Leverage Terraform module ecosystem

2. LLM (Large Language Models) Operation
    - Applications based on Triton inference server
    - Ray ecosystem

3. Kusion Server
    - A long-running service
    - A user-friendly portal

Roadmap Dashboard

Community

GitHub

- 800+ stars
- 75 forks
- 25 contributors

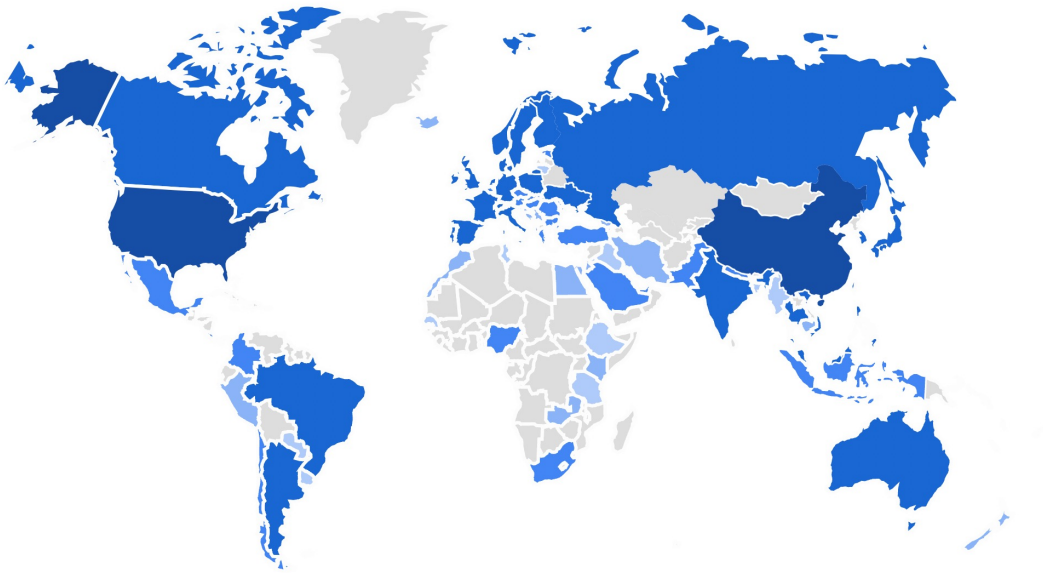
Website

- 5.7K users
  (This year Jan-Today)

End-User

- 5+ companies

- Slack
- Blog
- Community Calls



	Country +	↓ Users
		5,690 100% of total
1	China	2,876
2	United States	1,117
3	United Kingdom	170
4	Germany	147
5	Japan	135
6	Singapore	116
7	India	107
8	France	92
9	Canada	72
10	South Korea	70

Comparison

Radius

Similarity

- Application-centric
- Separation of concerns

Kusion's difference

- Lightweight CLI
- Manage the SSoT for hybrid resources via Spec

Crossplane

Similarity

- Manage resources lifecycle
- Extension mechanism

Kusion's difference

- DSL > YAML for App developers
- Lightweight CLI
- DCM (Dynamic Configuration Management)

OAM/KubeVela

Similarity

- App-centric
- Separation of concerns

Kusion's difference

- Lightweight CLI
- Manage the SSoT for hybrid resources via Spec
- DCM (Dynamic Configuration Management)

For other comparisons: Kusion vs X

Q&A