使用 KCL 重塑配置管理界面

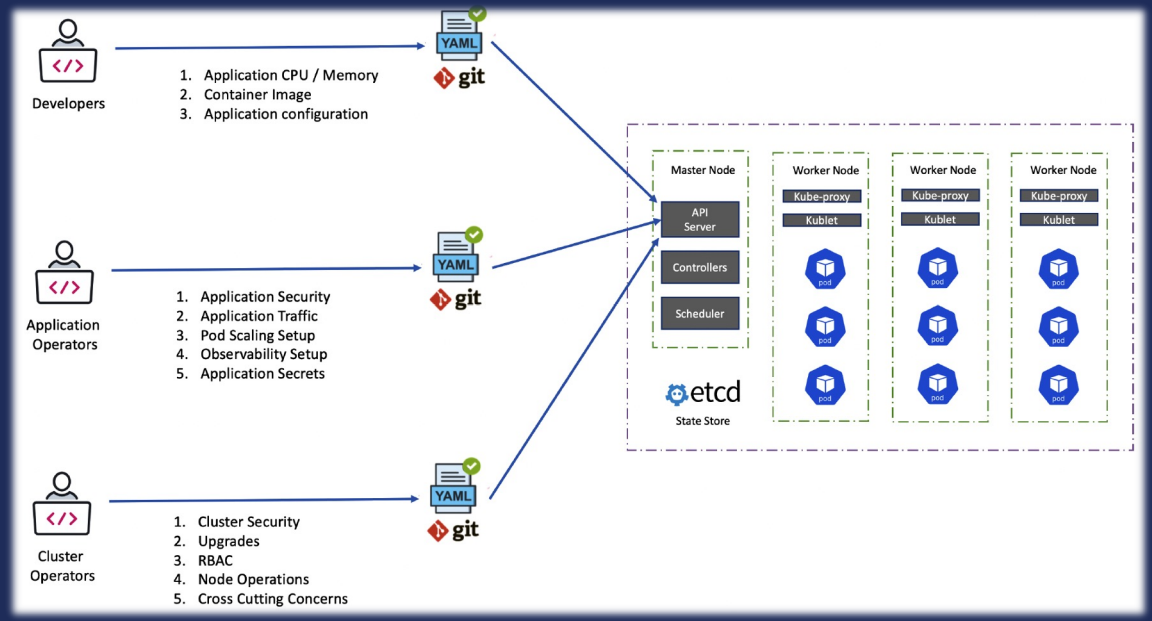
徐鹏飞 | KCL 项目 Maintainer

目录

背景

01

背景



认知负担

- 应用开发人员需要面对复杂的基础设施和平台概念
- 不像基础设施配置有 Terraform 等 IaC 工具，Kubernetes 是平台的平台，特别是在客户端缺乏轻量的配置组合和校验工具

静态配置

- YAML 膨胀，维度爆炸
- 跨团队配置协作负担和配置漂移

效率、可靠性低

- 缺乏标准的测试验证手段，大多是胶水代码或者脚本的拼盘
- 缺乏高效配置协同的工具，大多通过人肉拉群解决

减轻基础设施对开发人员的负担，提高配置管理效率

Kubernetes 中的声明式应用管理：https://docs.google.com/document/d/1cLPGweVEYrVqQvBLJg6sxV-TrE5Rm2MNOBA_cxZP2WU/edit#
CNCF 平台工程白皮书：https://tag-app-delivery.cncf.io/whitepapers/platforms/
Google SRE 工作手册：https://sre.google/workbook/configuration-specifics/

kcl

背景

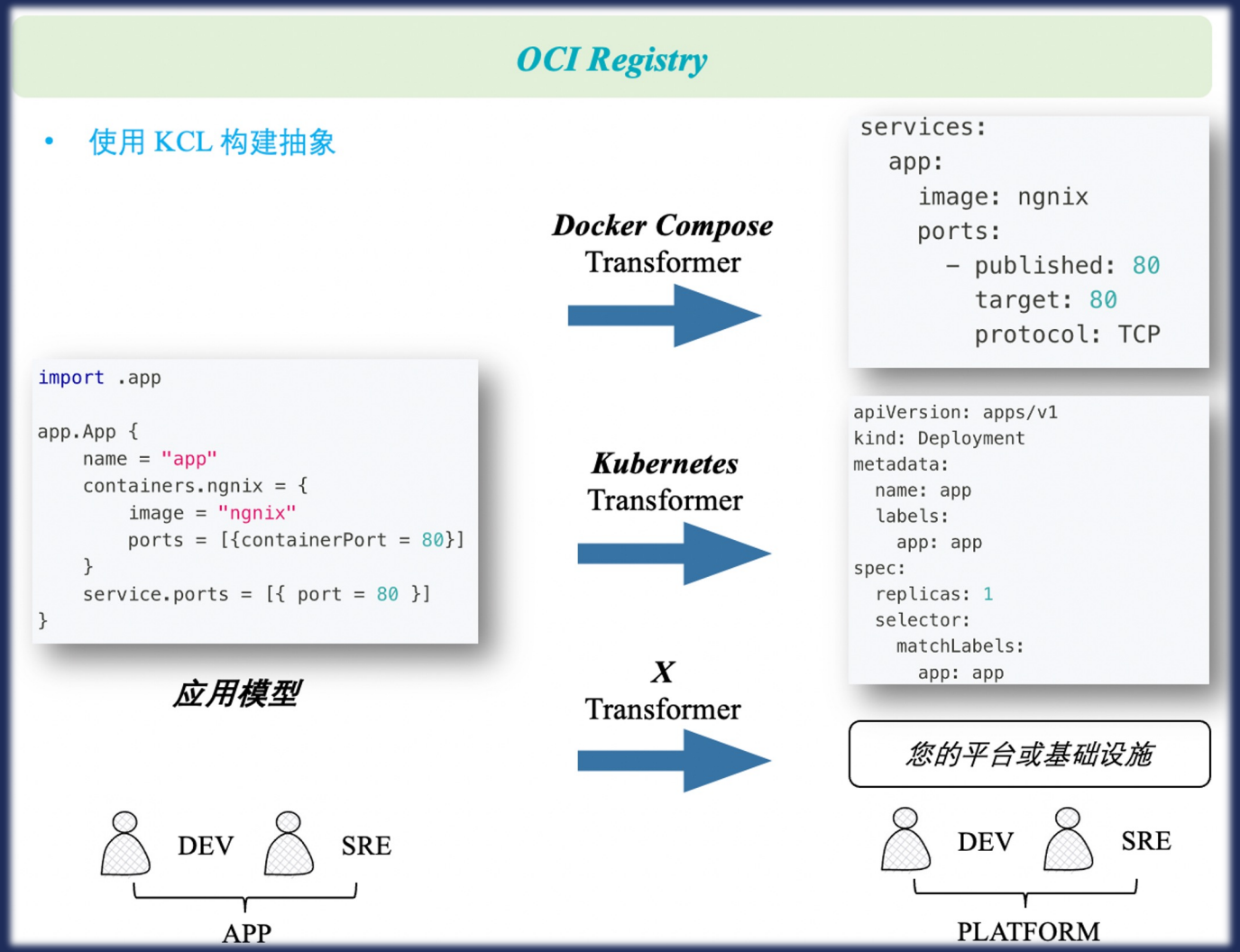
- *屏蔽基础设施和平台细节，降低开发者认知负担*
  - 自定义抽象模型
  - 解决 YAML 或者模版配置维度爆炸问题
  - 可编程支持：逻辑、类型、函数、模块、…
- *无副作用地进行大规模配置管理*
  - 可扩展
  - 风险左移
  - 自动化
  - 高性能
  - 包管理支持
- *通过插件为已有的配置管理工具进行增强*
  - Helm
  - Kustomize
  - kpt
  - …

KCL 可以做什么

概念

02

配置 *Standalone KCL 形式*



- Validation: 通过 KCL Schema/Rule 等结构对配置进行约束校验
- Mutation: 通过 KCL 函数对资源进行变换或注入配置

- ✓ *开发者可以理解的声明式应用配置模型*
- ✓ *动态配置管理 DCM: 可编程可扩展*
- ✓ *关注点分离：应用/平台 Dev/SRE*
- ✓ *风险左移：实时的配置错误提示*
- ✓ *标准的 OCI 配置交付和包管理支持：KPM*

配置 *KRM KCL 形式*

- **Mutation**

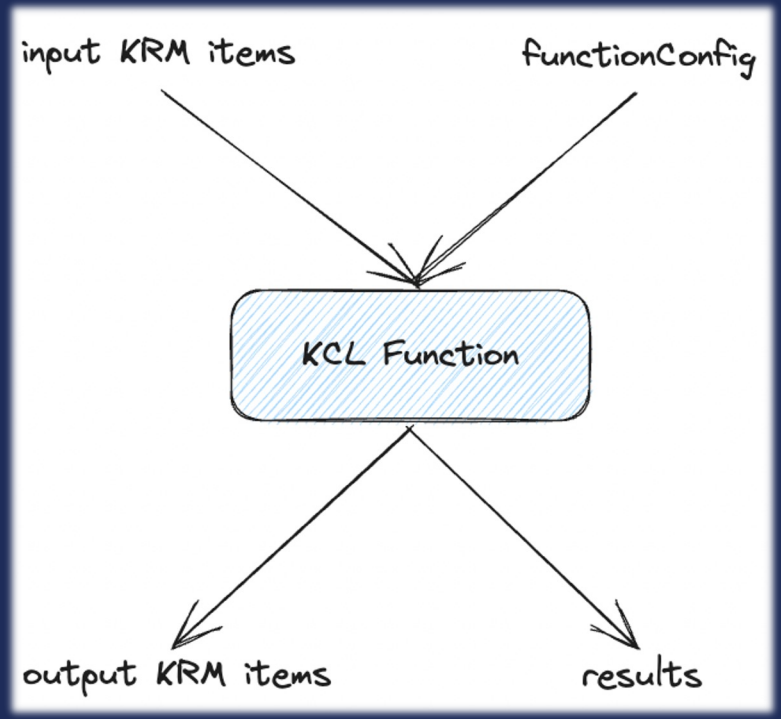
```yaml
apiVersion: krm.kcl.dev/v1alpha1
kind: KCLRun
metadata:
  name: set-annotations
  metadata:
    annotations:
      krm.kcl.dev/version: 0.0.1
      krm.kcl.dev/type: mutation
      documentation: >-
        Add or change annotations
spec:
  params:
    toAdd: addValue
  source: oci://ghcr.io/kcl-lang/set-annotation
```

- **Abstraction**

```yaml
apiVersion: krm.kcl.dev/v1alpha1
kind: KCLRun
metadata:
  name: web-service
  metadata:
    annotations:
      krm.kcl.dev/version: 0.0.1
      krm.kcl.dev/type: abstraction
      documentation: >-
        Web service application abstraction
spec:
  params:
    name: app
    containers:
      ngnix:
        image: ngnix
        ports:
          containerPort: 80
    labels:
      name: app
  source: oci://ghcr.io/kcl-lang/web-service
```

- **Validation**

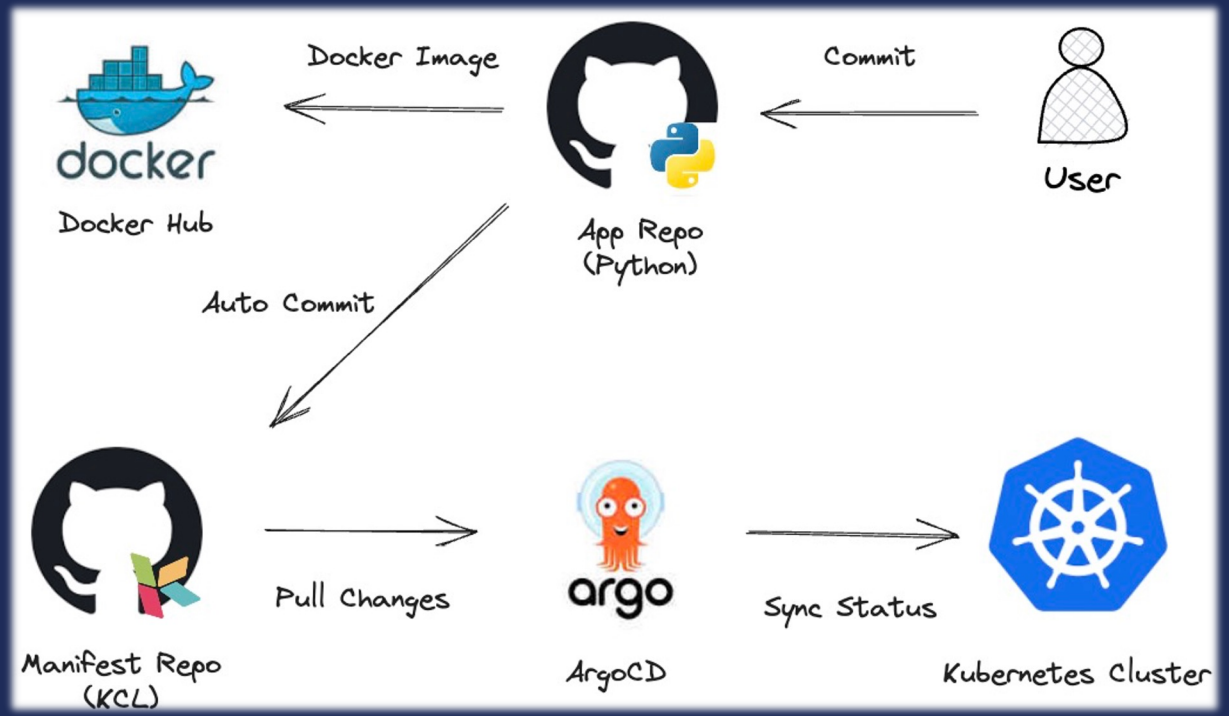
```yaml
apiVersion: krm.kcl.dev/v1alpha1
kind: KCLRun
metadata:
  name: https-only
  metadata:
    annotations:
      krm.kcl.dev/version: 0.0.1
      krm.kcl.dev/type: validation
      documentation: >-
        Requires Ingress resources to be HTTPS only.  Ingress resources must
        include the `kubernetes.io/ingress.allow-http` annotation, set to `false`.
        By default a valid TLS {} configuration is required, this can be made
        optional by setting the `tlsOptional` parameter to `true`.
        More info: https://kubernetes.io/docs/concepts/services-networking/ingress/#tls
spec:
  source: oci://ghcr.io/kcl-lang/https-only
```



- *遵循统一的 KRM Function 规范*
- *可编程可扩展*
- *多种代码源支持: OCI, Git, Https, ...*

https://github.com/kcl-lang/krm-kcl

自动化



Commit

kcl code set image to kcllang/flask_demo:6428cff4309afc8c1c40ad180bb9...
…cfd82546be3e

main

github-actions[bot] committed 3 minutes ago

Showing 1 changed file with 1 addition and 1 deletion.

2 main.k

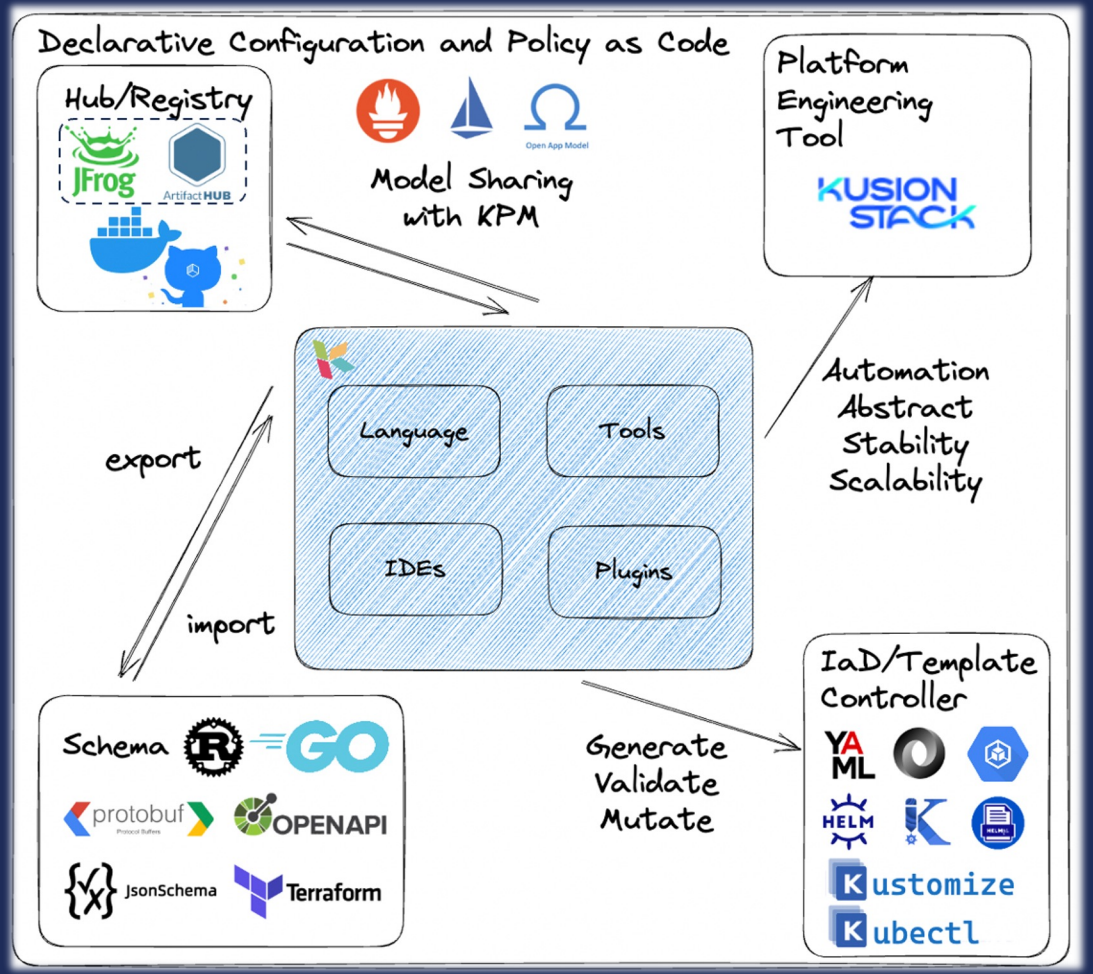
```
@@ -3,7 +3,7 @@ config = app.App {
3  3      name = "flask_demo"
4  4      containers: {
5  5          flask_demo = {
6     -            image = "kcllang/flask_demo:f1f2cbc0c4555d141e9f642fbd12edaf34d0b723"
   6  +            image = "kcllang/flask_demo:6428cff4309afc8c1c40ad180bb9cfd82546be3e"
7  7              ports = [{containerPort = 5000}]
8  8          }
9  9      }
```

配置驱动的工作流：同时支持 Standalone KCL 和 KRM KCL 配置格式
多种 CI/CD 和 GitOps 工具支持

https://kcl-lang.io/docs/user_docs/guides/gitops/gitops-quick-start

kcl

集成



客户端

运行时

SDKs

- *多语言 SDK: Go, Rust, Python, Java*
- *包管理支持: KPM 工具和多种 Registry 支持*
- *数据和 Schema 集成：KCL Import 工具*
- *运行时集成: 使用 KCL Operator 而不是重复开发 Admission Webhook*
- *KRM 支持: 统一的规范和插件支持 e.g., kubectl-kcl plugin, helm-kcl plugin, kustomize-kcl plugin, kpt-kcl-plugin ...*
- *平台工程工具支持: 作为 DCM 语言配合不同引擎/编排器进行应用交付*

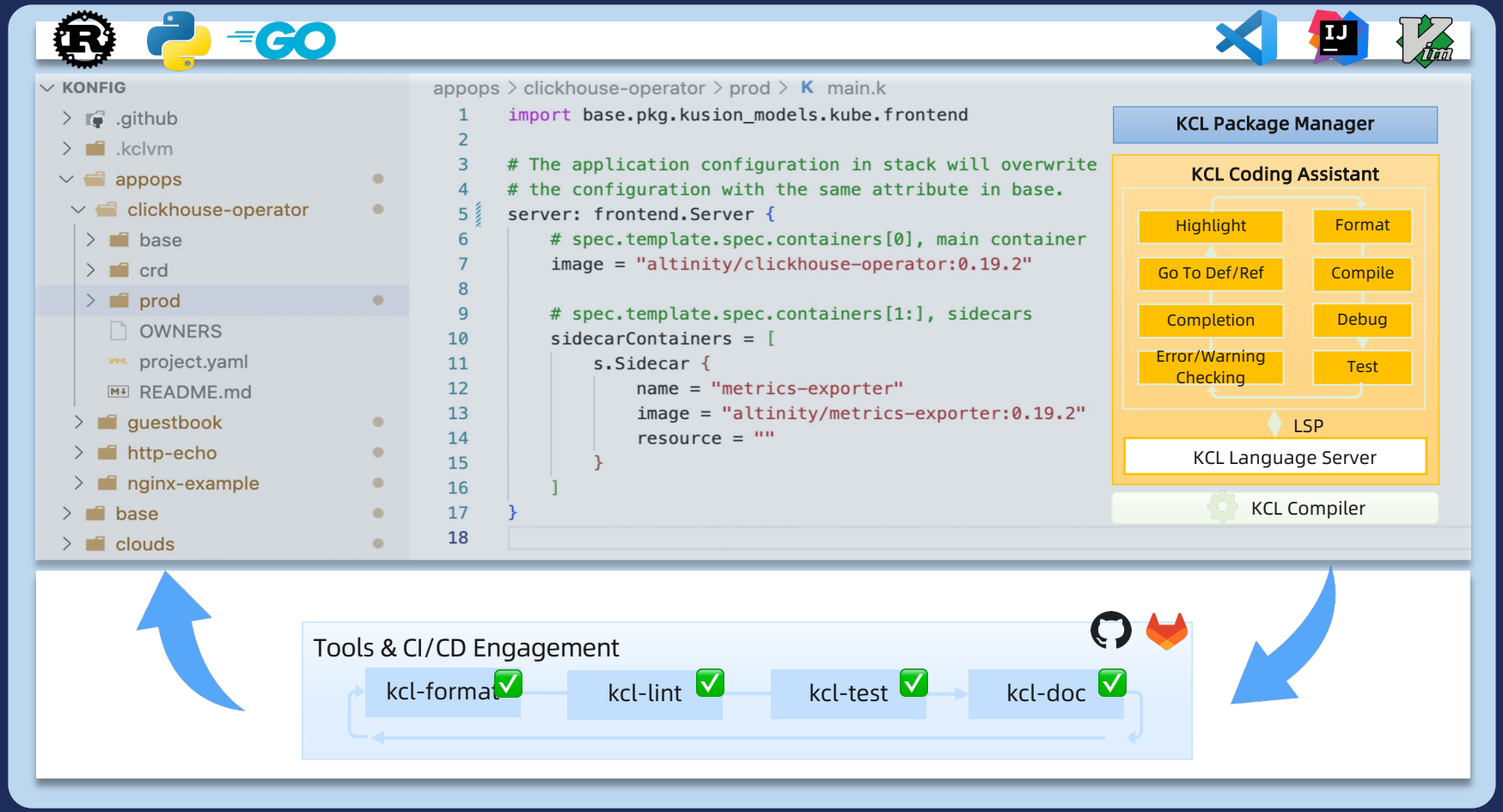
架构

03

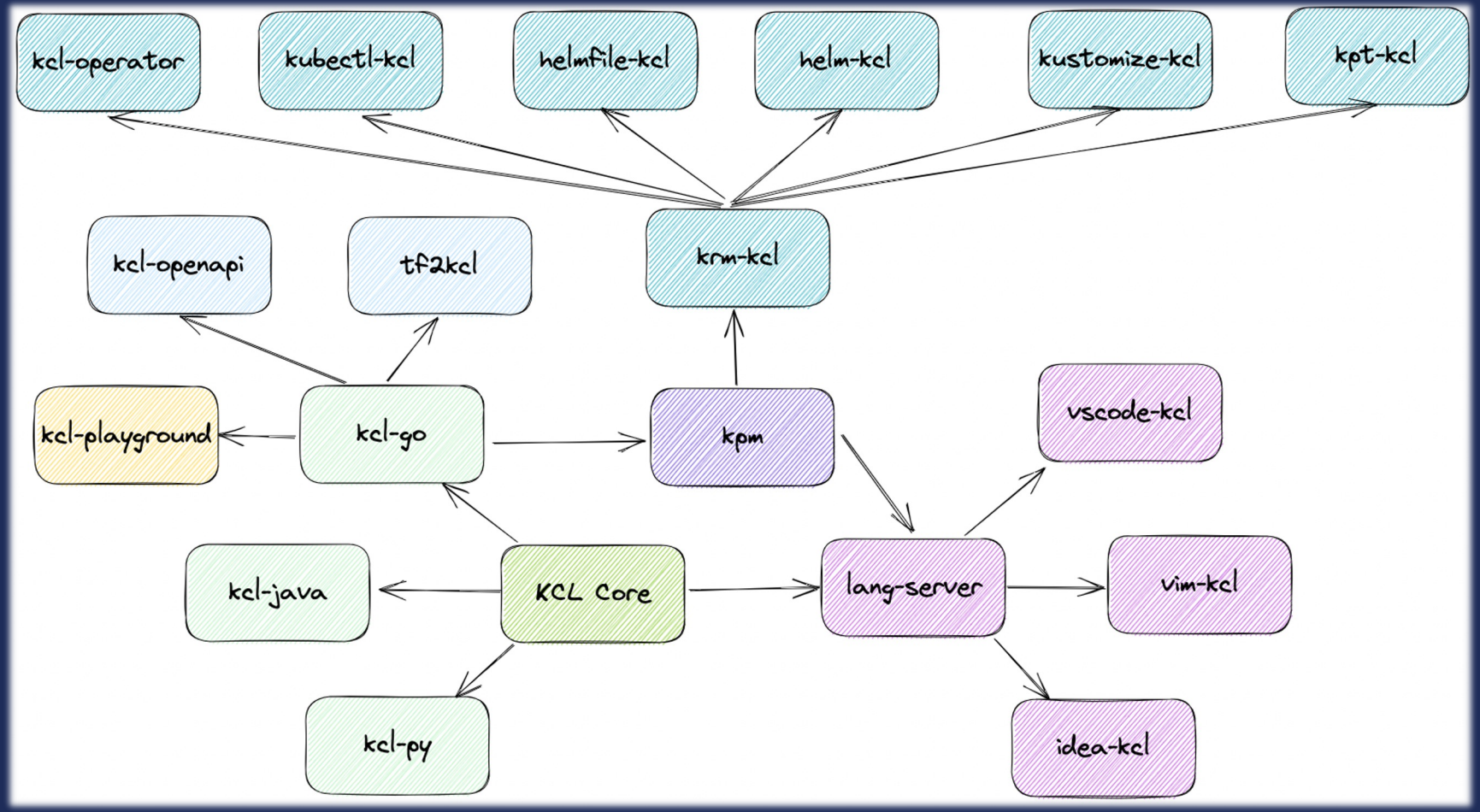
kcl

工作空间

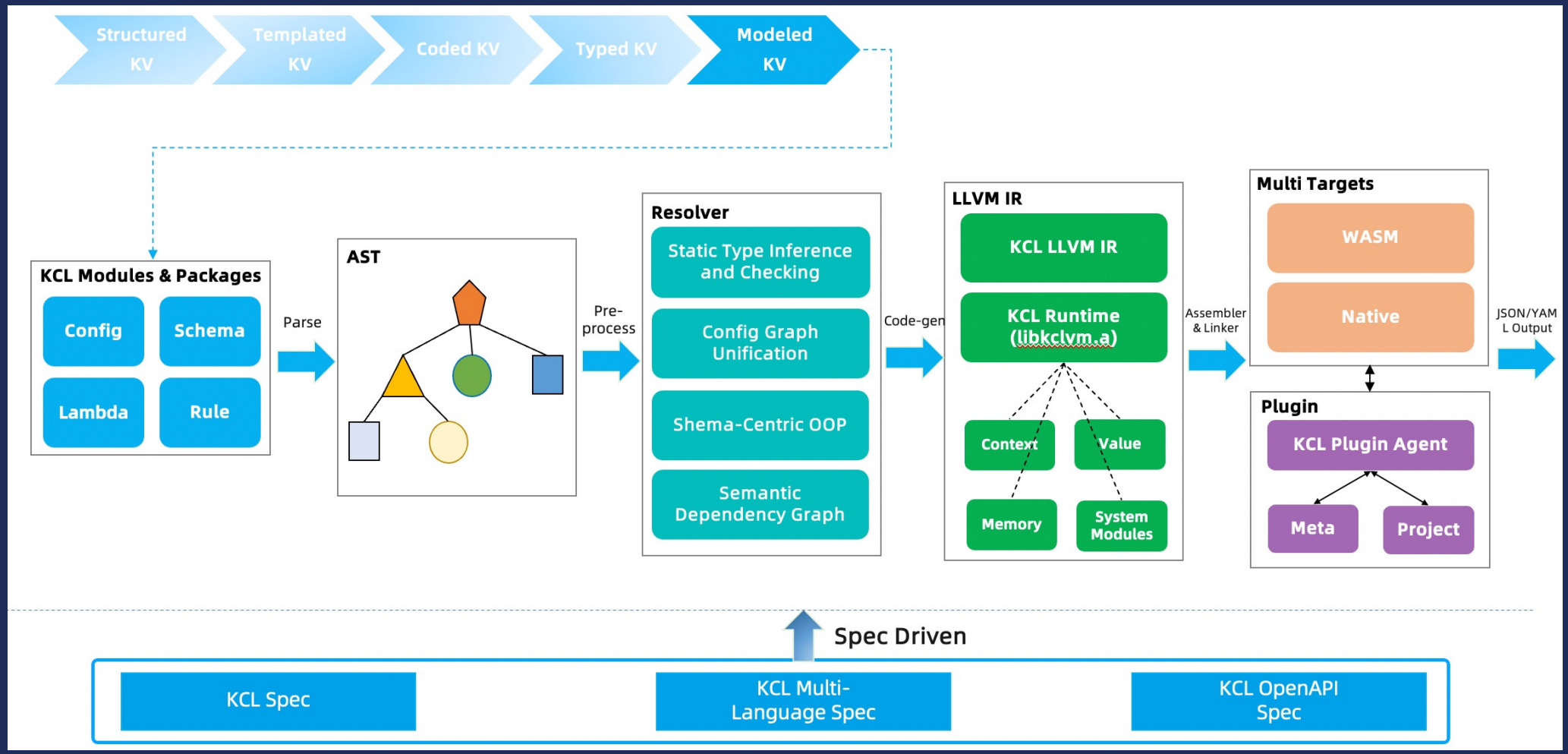
Language + Tools + IDEs + SDKs + Plugins



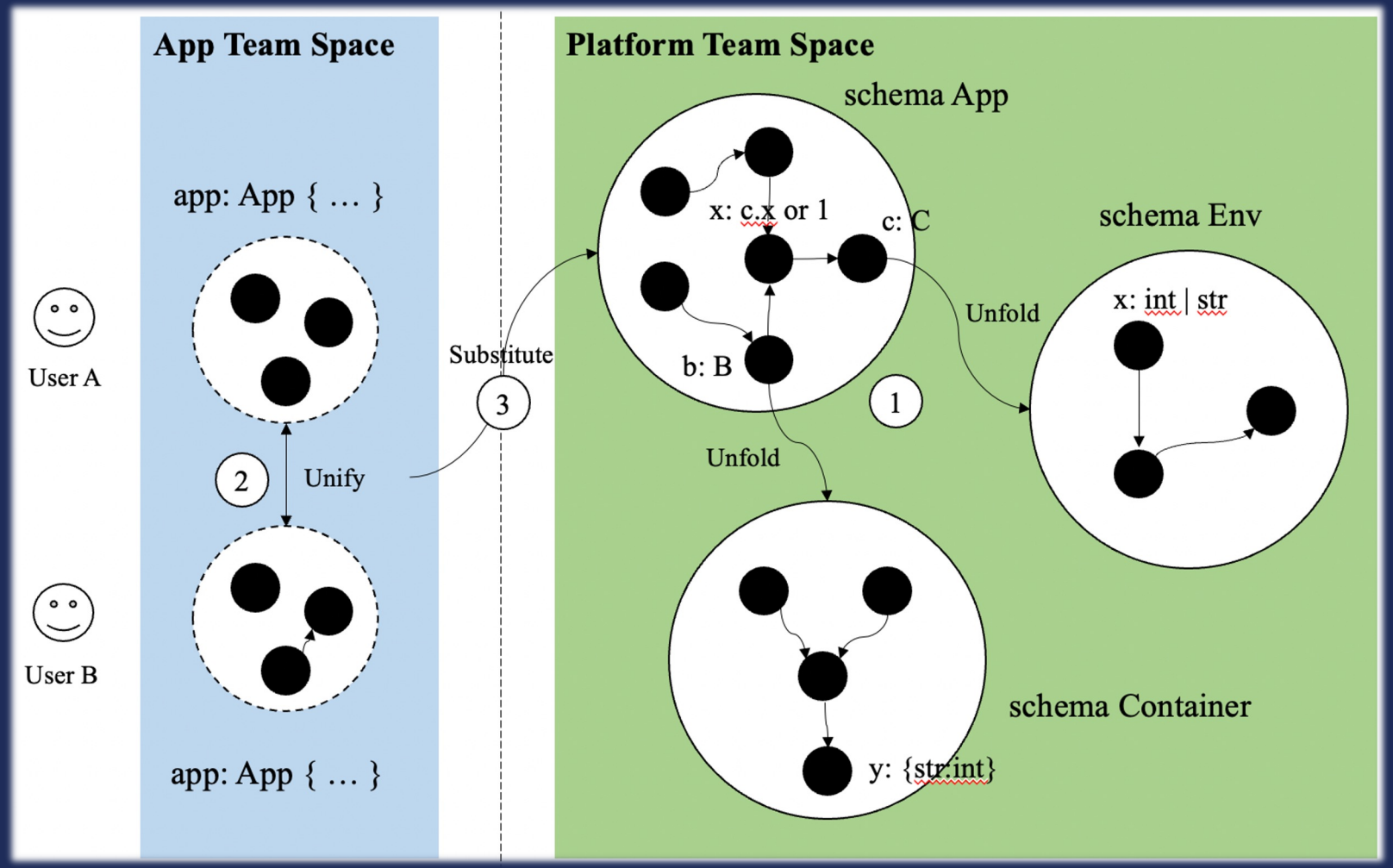
组件


kcl-operator
kubectl-kcl
helmfile-kcl
helm-kcl
kustomize-kcl
kpt-kcl
kcl-openapi
tf2kcl
krm-kcl
kcl-playground
kcl-go
kpm
vscode-kcl
kcl-java
KCL Core
lang-server
vim-kcl
kcl-py
idea-kcl

技术架构


Structured KV
Templated KV
Coded KV
Typed KV
Modeled KV
KCL Modules & Packages
Config
Schema
Lambda
Rule
Parse
AST
Pre-process
Resolver
Static Type Inference and Checking
Config Graph Unification
Shema-Centric OOP
Semantic Dependency Graph
Code-gen
LLVM IR
KCL LLVM IR
KCL Runtime (libkclvm.a)
Context
Value
Memory
System Modules
Assembler & Linker
Multi Targets
WASM
Native
Plugin
KCL Plugin Agent
Meta
Project
JSON/YAML Output
Spec Driven
KCL Spec
KCL Multi-Language Spec
KCL OpenAPI Spec

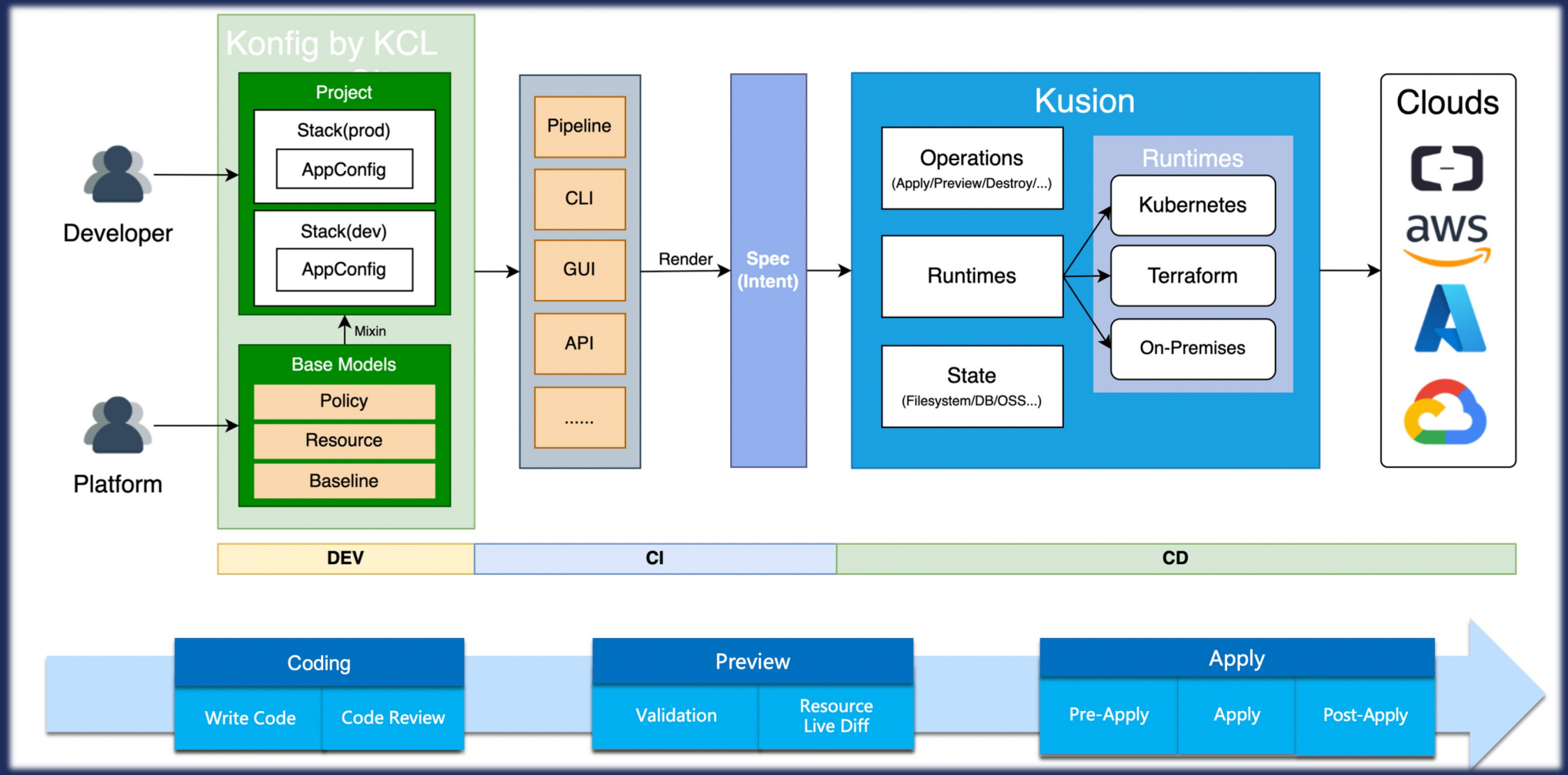
配置图模型



- ✓ 配置生成
  - ✓ 自动合并
  - ✓ 多种合并策略支持
- ✓ 配置变换
  - ✓ 数据/Schema 集成
  - ✓ CRUD API
- ✓ 配置校验
  - ✓ 静态类型
  - ✓ 约束规则
  - ✓ 不可变性
- ✓ 配置抽象
  - ✓ 关注点分离

KCL SETTA 2022 Paper：https://link.springer.com/chapter/10.1007/978-3-031-21213-0_6

Kusion 引擎


Konfig by KCL
Project
Stack(prod)
AppConfig
Stack(dev)
AppConfig
Mixin
Base Models
Policy
Resource
Baseline
Developer
Platform
Pipeline
CLI
GUI
API
.....
Render
Spec (Intent)
Kusion
Operations (Apply/Preview/Destroy/...)
Runtimes
State (Filesystem/DB/OSS...)
Runtimes
Kubernetes
Terraform
On-Premises
Clouds
aws
DEV
CI
CD
Coding
Write Code
Code Review
Preview
Validation
Resource Live Diff
Apply
Pre-Apply
Apply
Post-Apply

社区项目

Structured KV

Pros.

- Easy to write and read
- Rich multi-language API
- Various Path Tools

Cons.

- Redundant information
- Insufficient functionality e.g. abstraction, constraint, …

Tech.

- JSON
- YAML

Product

- Kustomize
- …

Templated KV

Pros.

- Simple config logic support
- Dynamic argument input

Cons.

- Increase of argument makes it difficult to maintain
- Insufficient functionality e.g. abstraction, constraint, …

Tech.

- Velocity
- Go Template

Product

- Helm
- Helmfile
- …

Programmable KV

Pros.

- Required programming features
- Code modularity
- Templates & Data abstraction

Cons.

- Insufficient type constraints
- Insufficient restraint ability
- Runtime error

Tech.

- GCL
- HCL
- Starlark
- Jsonnet…

Product

- Terraform
- Tanka
- ytt

Typed KV

Pros.

- Rich config constraint syntax
- Unified type & value constraint
- Configuration conflict checking

Cons.

- Difficult to configuration override for multi-environment scenarios
- Runtime checks and limited performance

Tech.

- CUE
- …

Product

- KubeVela
- …

Modeled KV

Pros.

- Model-centric & constraint-centric
- Scalability on separated block writing with rich merge strategies
- Static type system & analysis
- High Performance

Cons.

- Expansion of different models requires investment in R&D

Tech.

- **KCL**
- …

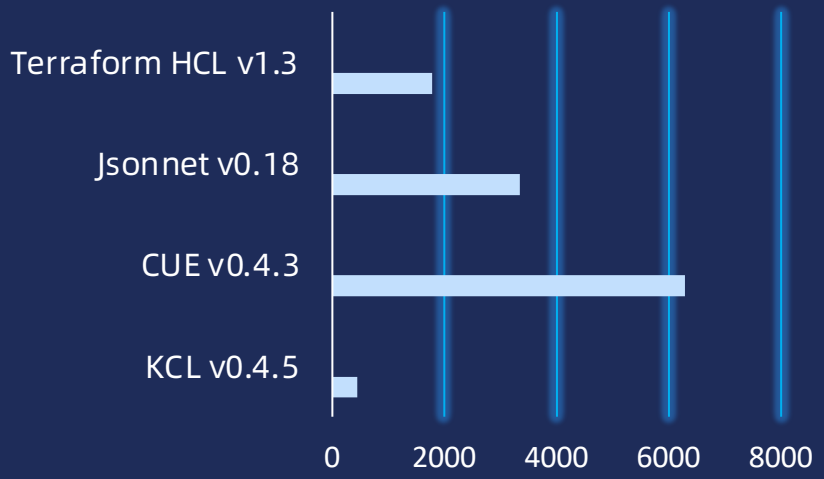
Product

- KusionStack
- KRM-KCL Tools and Operators

性能对比

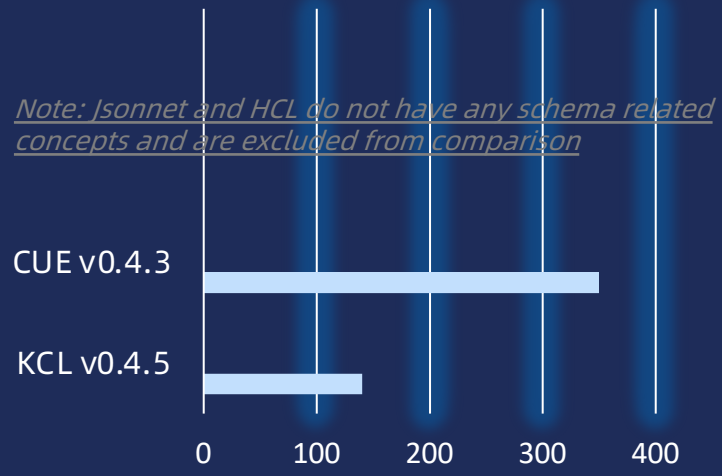
Loop and Function

```
a = lambda x: int, y: int -> int {
    max([x, y])
}
temp = {"a${i}": a(1, 2) for i in range(10000)}
```



Kubernetes Configuration

```
import kubernetes.api.apps.v1

deployment = v1.Deployment {}
```

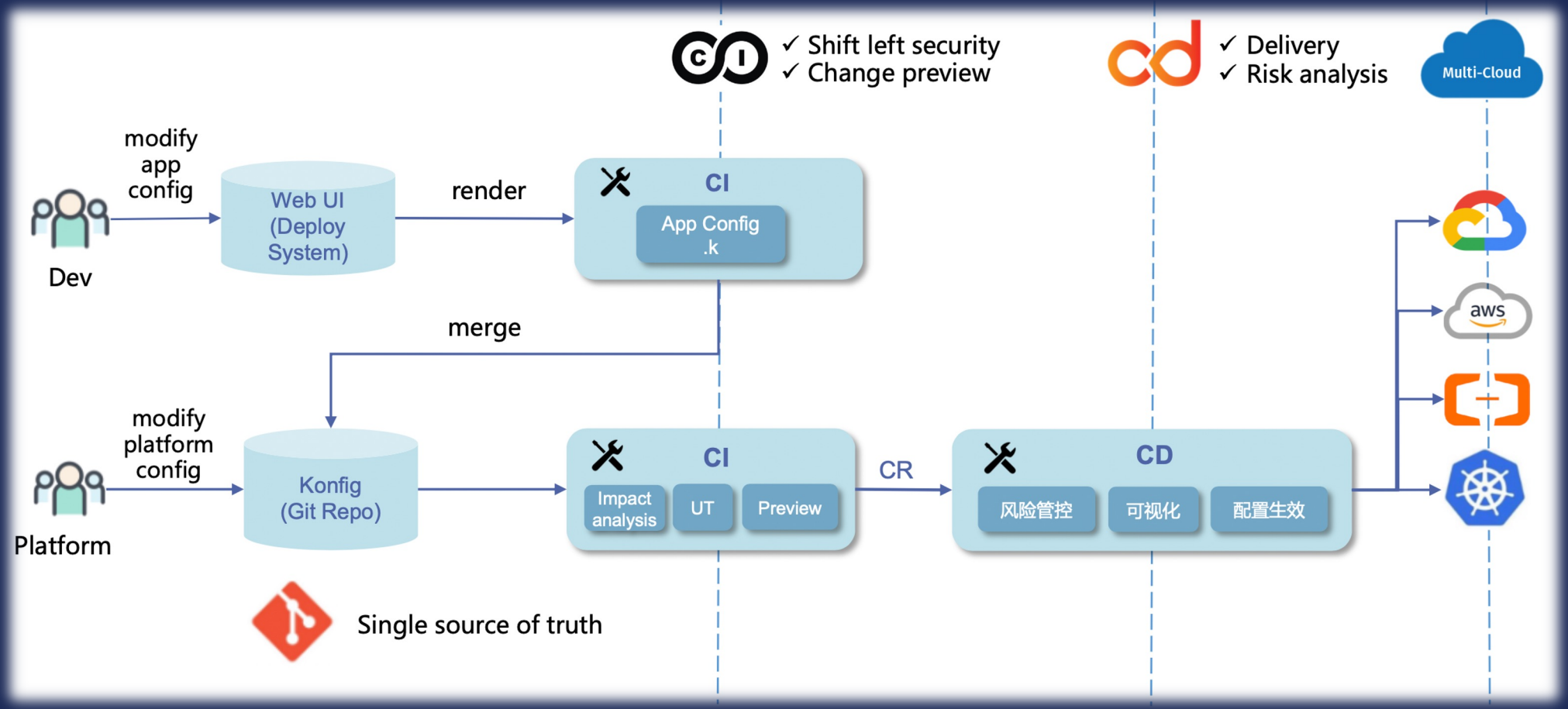


Test environment: single core macOS 10.15.7 CPU: i7-8850H 2.6GHz 32GB 2400Mhz DDR4 No NUMA, e2e run time (ms)

实践

04

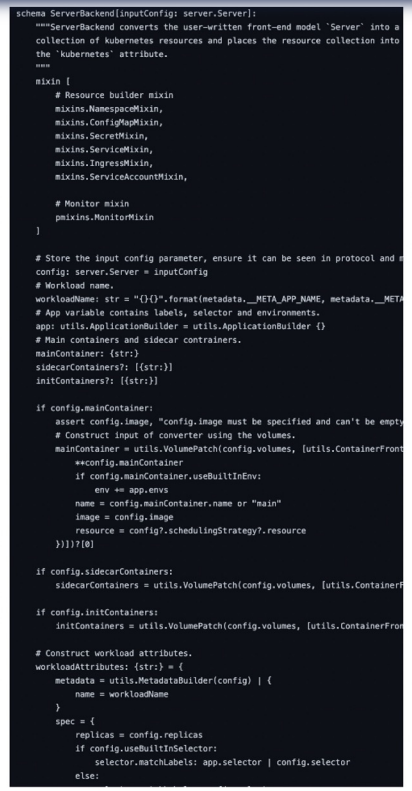
实践


✓ Shift left security
✓ Change preview
✓ Delivery
✓ Risk analysis
Multi-Cloud
modify app config
Dev
Web UI (Deploy System)
render
CI
App Config .k
merge
modify platform config
Platform
Konfig (Git Repo)
CI
Impact analysis
UT
Preview
CR
CD
风险管控
可视化
配置生效
aws
Single source of truth

kcl

实践

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

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



Spec

- **Deployment**
- **Service**
- **ConfigMap**
- **Database**
- **Monitor**
- **... ...**

Front-end model (Developer)　　Back-end model (Platform)　　K8s/clouds/on-prem resources

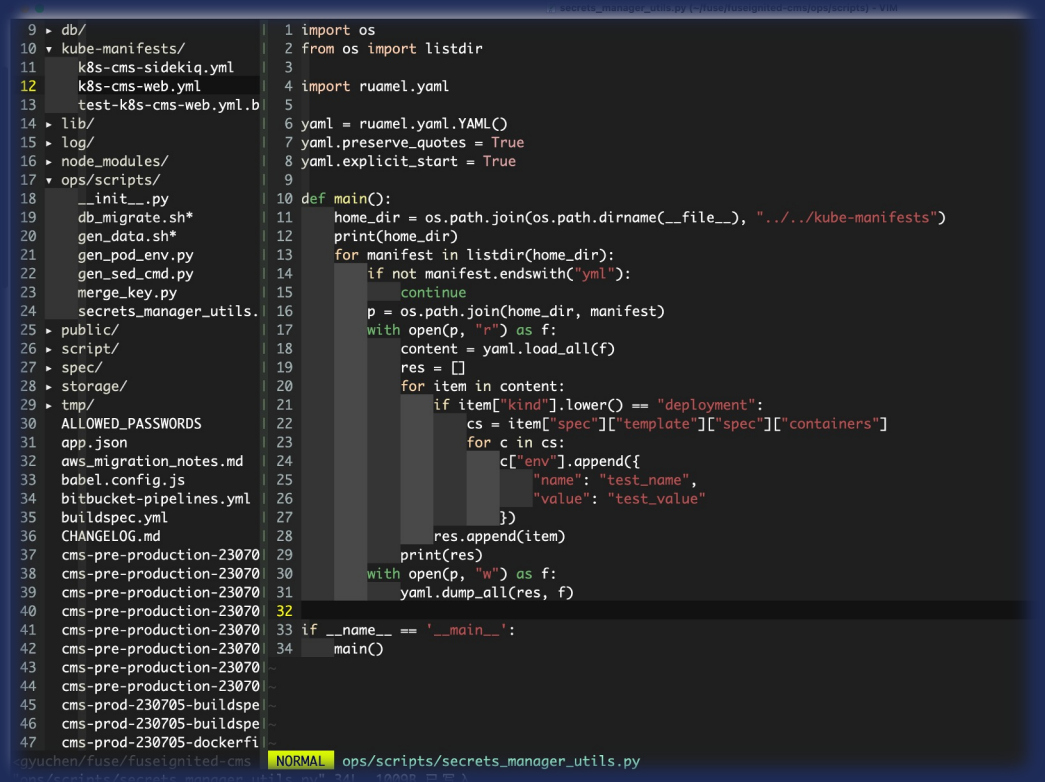
kcl

实践

```
➜  bin git:(watch-ui) ./kusion apply -w /Users/yuanhao/opensource/KusionStack/konfig/base/examples/server/app_service/prod --watch --yes
 ✓  Compiling in stack prod...
```

kcl

实践



- Code

```
apiVersion: krm.kcl.dev/v1alpha1
kind: KCLRun
spec:
  params:
    env:
    - name: test_name
      value: test_value
  source: |
    [r | {if resource.kind == "Deployment": spec.template.spec.containers: [{
        env += option("params").env
    }] * len(r.spec.template.spec.containers)} for r in option("items")]
```

- OCI

```
apiVersion: krm.kcl.dev/v1alpha1
kind: KCLRun
spec:
  params:
    env:
    - name: test_name
      value: test_value
  source: oci://ghcr.io/kcl-lang/append-env
```

实践

OpenAPI spec schemas in KCL - https://kcl-lang.io/

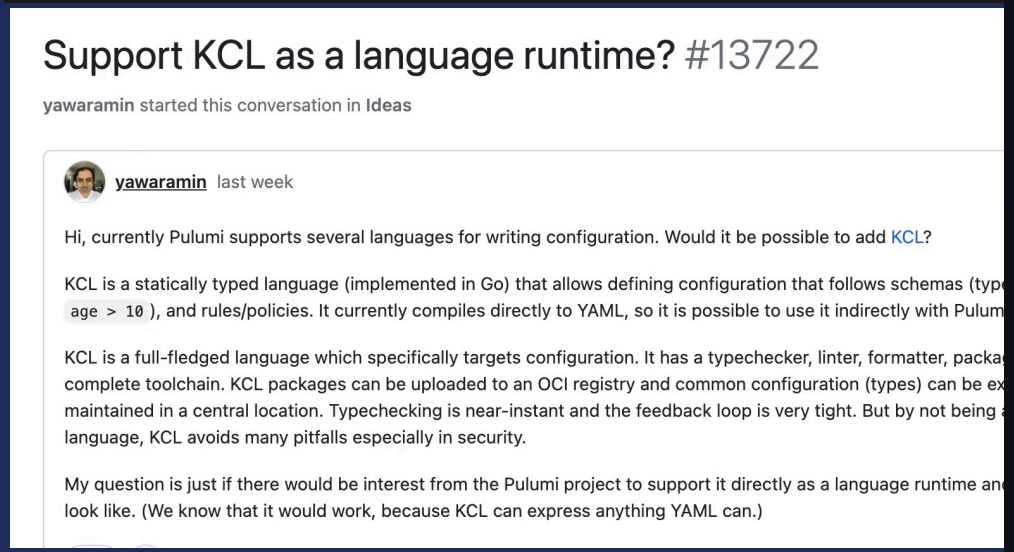
openapi.k

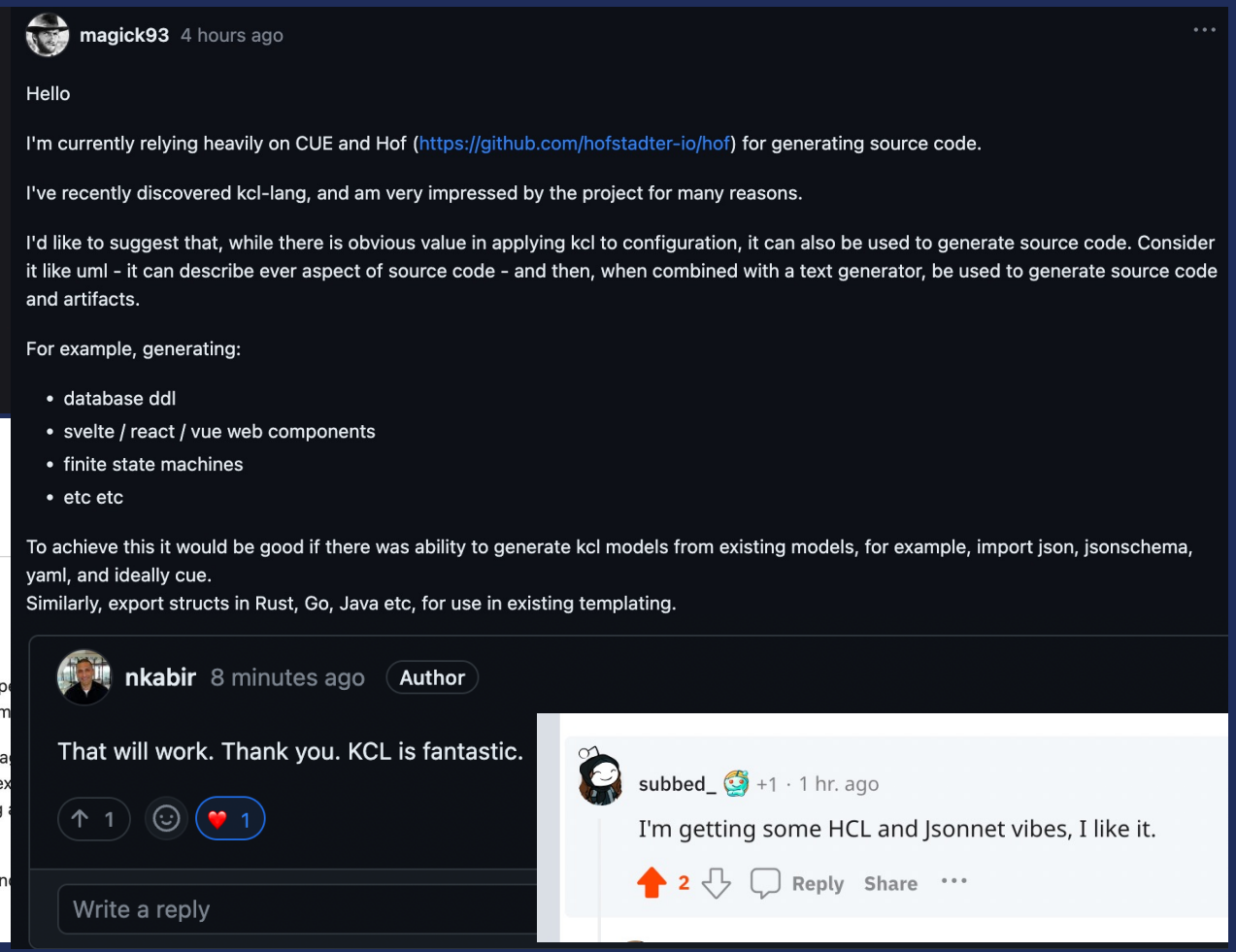
```
"""
Bindings for OpenAPI 3.0.0 - https://swagger.io/specification/#schema-object

ℹ JSON Schema validation for OpenAPI 3.0.0 must conform to this version:
https://datatracker.ietf.org/doc/html/draft-wright-json-schema-validation-00
"""

schema Ref:
  $ref: str

schema Contact:
  """
  https://swagger.io/specification/#contact-object
  """
  name?: str
  url?: str
  email?: str

schema Description:
  description?: str

schema SchemaString(Description):
  type: "string" = "string"
  enum?: [str]
  default?: str
  format?: "byte" | "binary" | "date" | "date-time" | "password"

schema SchemaNumber(Description):
  type: "number" = "number"
  minimum?: float
  maximum?: float
  exclusiveMinimum?: bool
  exclusiveMaximum?: bool

schema SchemaInteger(Description):
  type: "integer" = "integer"
  minimum?: int
```

- ✓ JSON/YAML 支持
- ✓ 双向 Open API 转换支持
- ✓ 自定义错误信息

https://gist.github.com/yawaramin/6f56f4b053d2b090353d7b7d40a722ce

用户声音

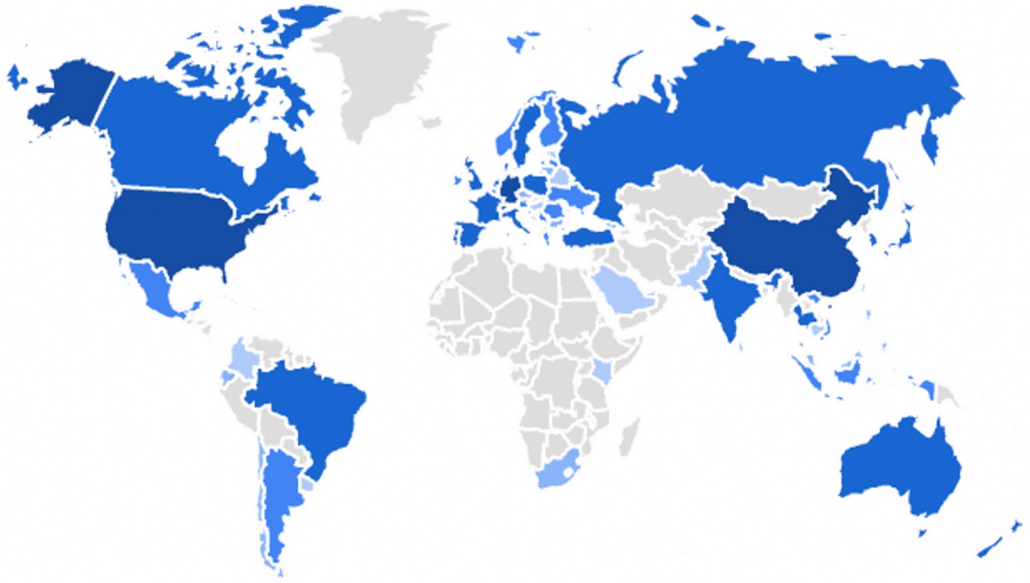




更多玩法，欢迎加入 KCL 社区

kcl

社区



COUNTRY ID

- United States
- China
- Germany
- United Kingdom
- Canada
- France
- India

~30 Releases

20+ Presentation & Blogs

8 Maintainers

~20 Contributors

Biweekly Community Meeting and Blogs

https://github.com/kcl-lang/community

2022 年 5 月开源

规划

05

技术规划

- **v0.4.6 Release**
- Ease of Use and Error Improvement
- Helm KCL Schema Plugin Design
- KPT KCL SDK Deign
- KCL KPM Pre-release (Git Repo, Lock, Dependency management)
- Go SDK: Better Installation, Go Struct/KCL Schema Conv API，List Instance API, etc.
- KCL New IDE Plugin Pre-release
- KCL Playground Code Share

2023.4

- **v0.5.0 Release**
- Improvement of Grammar and Standard Libraries
- KCL Abstract/Constraint Specification Standards and Best Practices Manual
- Helm KCL Plugin
- KPT KCL Plugin
- Kustomize KCL Plugin
- KPM
- KCL New Test Tool Design
- More SDK Support e.g.，Java
- KCL New IDE Extension
- More KCL Playground Code Samples

2023.7

- **v0.6.0 Release**
- KCL New UI Design
- Helmfile KCL Plugin
- KCL Operator
- KCL UI and Backstage Plugin Design
- KCL New Doc Tool Release
- KPM Registry Design
- More KCL IDE Extension Support e.g., Vim

2023.9

- **v0.7.0 Release**
- KCL New UI Release
- KCL UI and Backstage Plugin Release
- KCL Import Tool
- KPM Registry Release

2023.12

Language
Tool
SDK
Scene & Integration
IDE Extension

https://github.com/kcl-lang/kcl/issues/29

下一步是什么？

AI Based Programming Language for the Cloud?

AIGC ≈ AI + DSL + Engine + CoT

平台/云能力**抽象**封装成 DSL 统一 App 和 Infra

引擎消费 DSL 输入

将 DSL 的组合方式（链式思维的方式，**CoT**）提炼成 **Prompt** 喂给 **AI 模型**并返回结果

其他资源

- **官方网站**
  - *https://kcl-lang.io/*
  - *https://kusionstack.io/*
- **GitHub**
  - *https://github.com/kcl-lang*
  - *https://github.com/KusionStack*
- **Twitter**
  - *@kcl language*
  - *@kusionstack*
- **Slack**
  - *https://kcl-lang.slack.com*
  - *https://kusionstack.slack.com*

钉钉(DingTalk ID 42753001)

微信搜一搜

规模化云原生运维

Thank You!