KCL: A constraint-based record & functional language mainly used in cloud-native configuration and policy scenarios.

Pengfei Xu (Ant Group)

Aug. 2023

Agenda

01 Background

02 Concepts

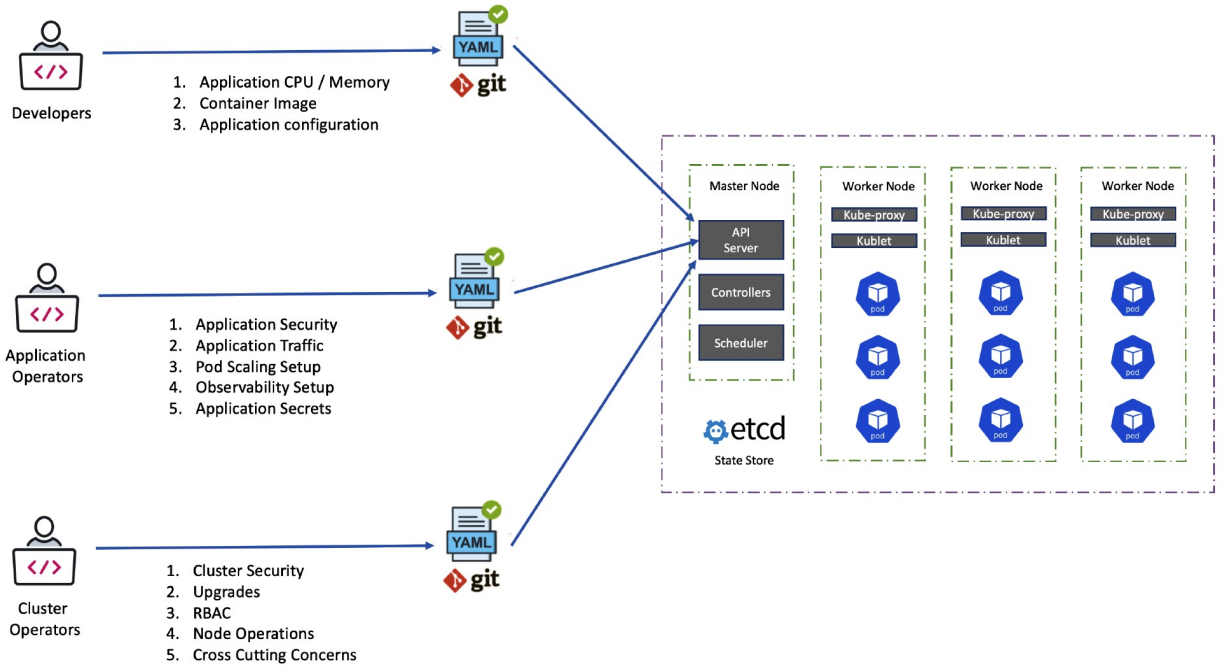
03 Architecture

04 Evaluation

Background

01

Background



Cognitive Loading

- Complex infrastructure and platform concepts
- Kubernetes is a platform of platform, lack of lightweight declarative configuration composition methods at client side

Static Config

- YAML Bloat
- Burdensome and configuration drift for all involved teams

Low Efficiency/Reliability

- Lack of standardized testing and validation methods
- Lack of effective tools for team collaboration

Reduce the burden of infrastructure for developers and improve the efficiency of configuration management

Declarative Application Management in Kubernetes: https://docs.google.com/document/d/1cLPGweVEYrVqQvBLJg6sxV-TrE5Rm2MNOBA_cxZP2WU/edit#
CNCF Platforms White Paper: https://tag-app-delivery.cncf.io/whitepapers/platforms/
Google SRE Workbook: https://sre.google/workbook/configuration-specifics/

Why KCL

- ***Hide infrastructure and platform details to reduce the burden of developers.***
  - Abstraction
  - Solve issues on YAML/Template bloat
  - Language enhancement: logic, type, function and package.

- ***Large-scale configuration management without side effects cross teams.***
  - Stability
  - Scalability
  - Automation
  - High performance
  - Package management

- ***Enhancement for configuration tools e.g., Helm, Kustomize, kpt.***
  - Mutation
  - Validation

Concepts

02

Configuration *Standalone KCL*

OCI Registry

- Abstraction

```
import .app

app.App {
    name = "app"
    containers.ngnix = {
        image = "ngnix"
        ports = [{containerPort = 80}]
    }
    service.ports = [{ port = 80 }]
}
```

Application Schema

Docker Compose
Transformer

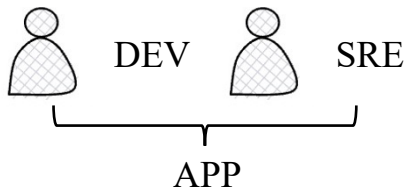
```
services:
  app:
    image: ngnix
    ports:
      - published: 80
        target: 80
        protocol: TCP
```

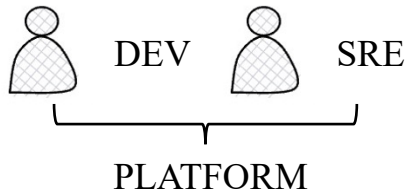
Kubernetes
Transformer

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app
  labels:
    app: app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: app
```

X
Transformer

Your Infra or Platform Spec





- Validation: Use KCL schema and rule to validate resources
- Mutation: Use KCL Function to transform resources.

- ✓ *Declarative and application-centric configuration interface that developers can understand*
- ✓ *Scalable and dynamic configuration management*
- ✓ *Separation of concerns*
- ✓ *Real time errors/warnings with schema and constraints*
- ✓ *Package management and OCI Registry support*

Configuration *KRM KCL*

- **Mutation**

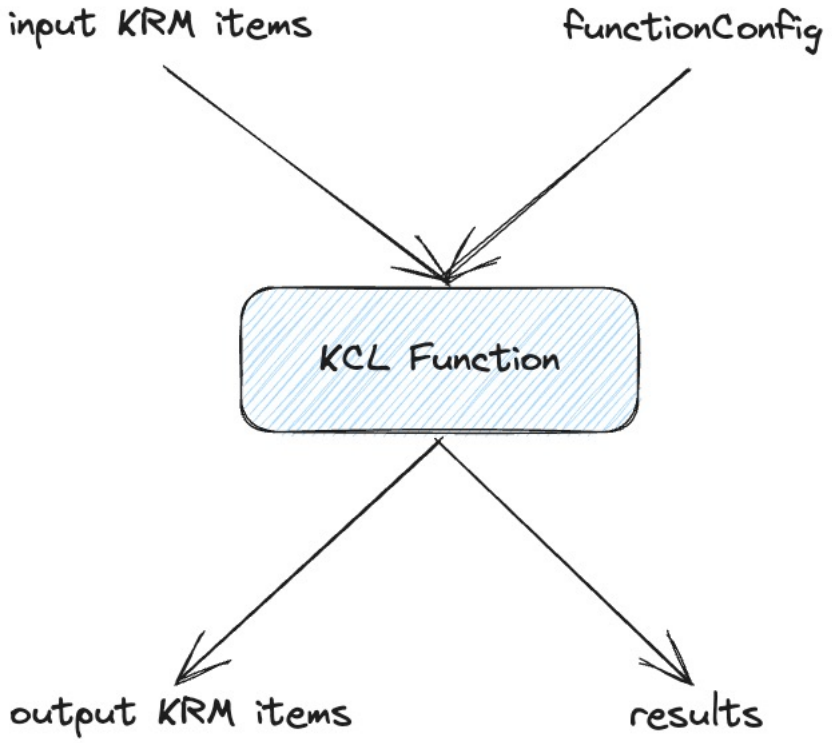
```yaml
apiVersion: krm.kcl.dev/v1alpha1
kind: KCLRun
metadata:
  name: set-annotations
  metadata:
    annotations:
      krm.kcl.dev/version: 0.0.1
      krm.kcl.dev/type: mutation
      documentation: >-
        Add or change annotations
spec:
  params:
    toAdd: addValue
  source: oci://ghcr.io/kcl-lang/set-annotation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

- **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**

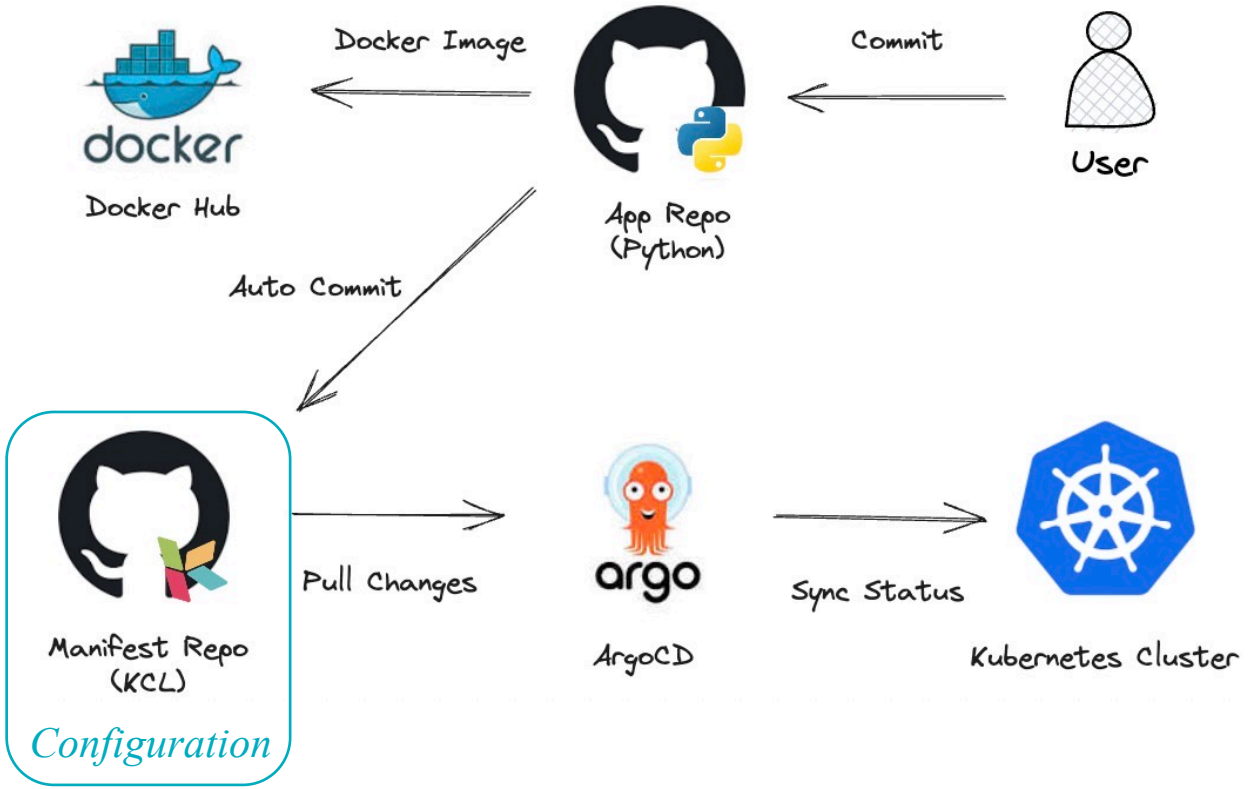
```yaml
apiVersion: krm.kcl.dev/v1alpha1
kind: KCLRun
metadata:
  name: https-only
  metadata:
    annotations:
      krm.kcl.dev/version: 0.0.1
      krm.kcl.dev/type: validation
      documentation: >-
        Requires Ingress resources to be HTTPS only.  Ingress resources must
        include the `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
```



- ***Unified Spec with KCL Function***
- ***Programmable and Extensible***
- ***Multiple Source Support****: OCI, Git, Https, …*

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

Automation



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

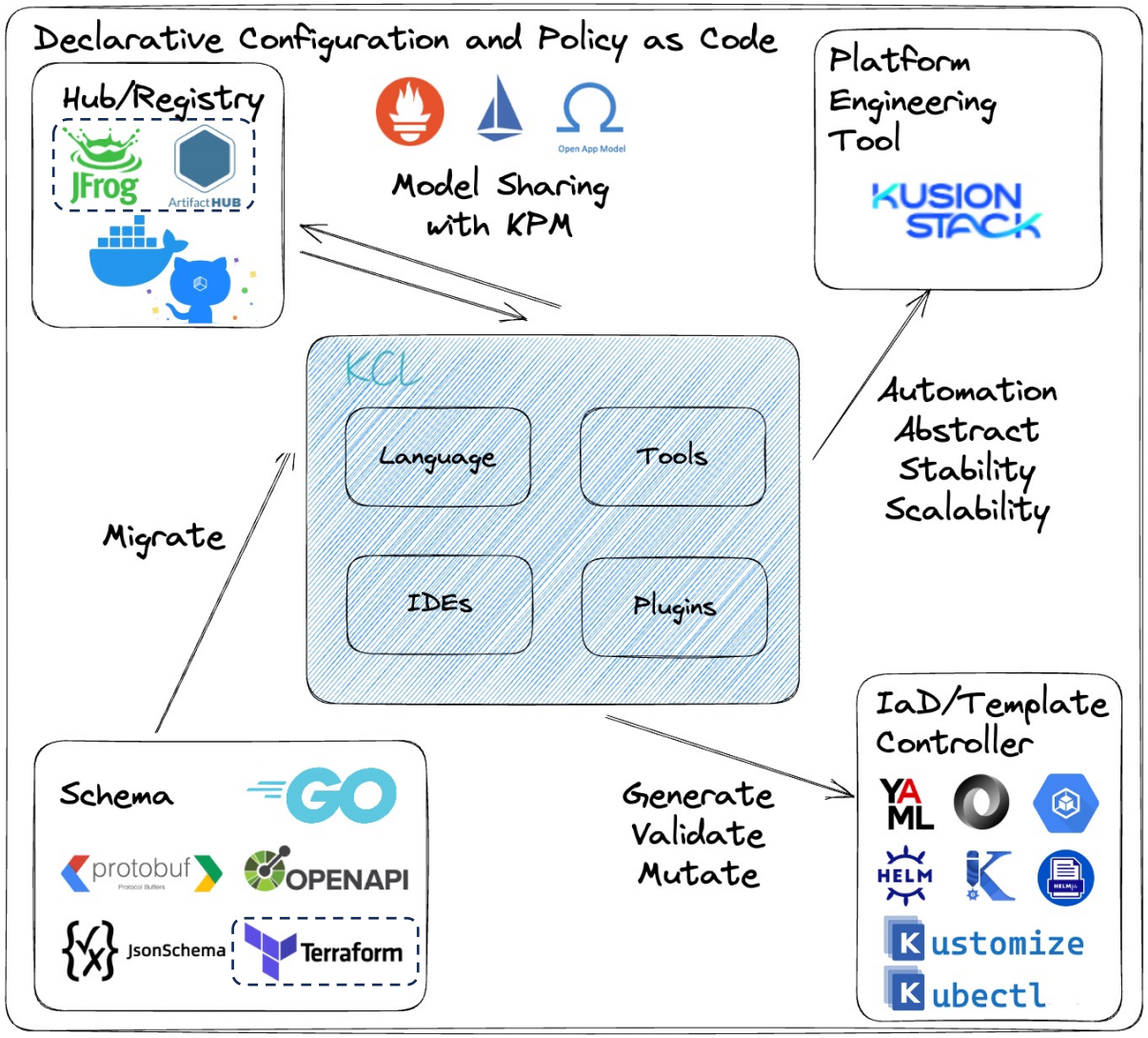
```
@@ -3,7 +3,7 @@ config = app.App {
    name = "flask_demo"
    containers: {
        flask_demo = {
-           image = "kcllang/flask_demo:f1f2cbc0c4555d141e9f642fbd12edaf34d0b723"
+           image = "kcllang/flask_demo:6428cff4309afc8c1c40ad180bb9cfd82546be3e"
            ports = [{containerPort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

Configuration source driven workflow: Supports both standalone and KRM KCL form

CI/CD Integration, GitOps Friendly

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

Integrations



Client

Server

SDKs

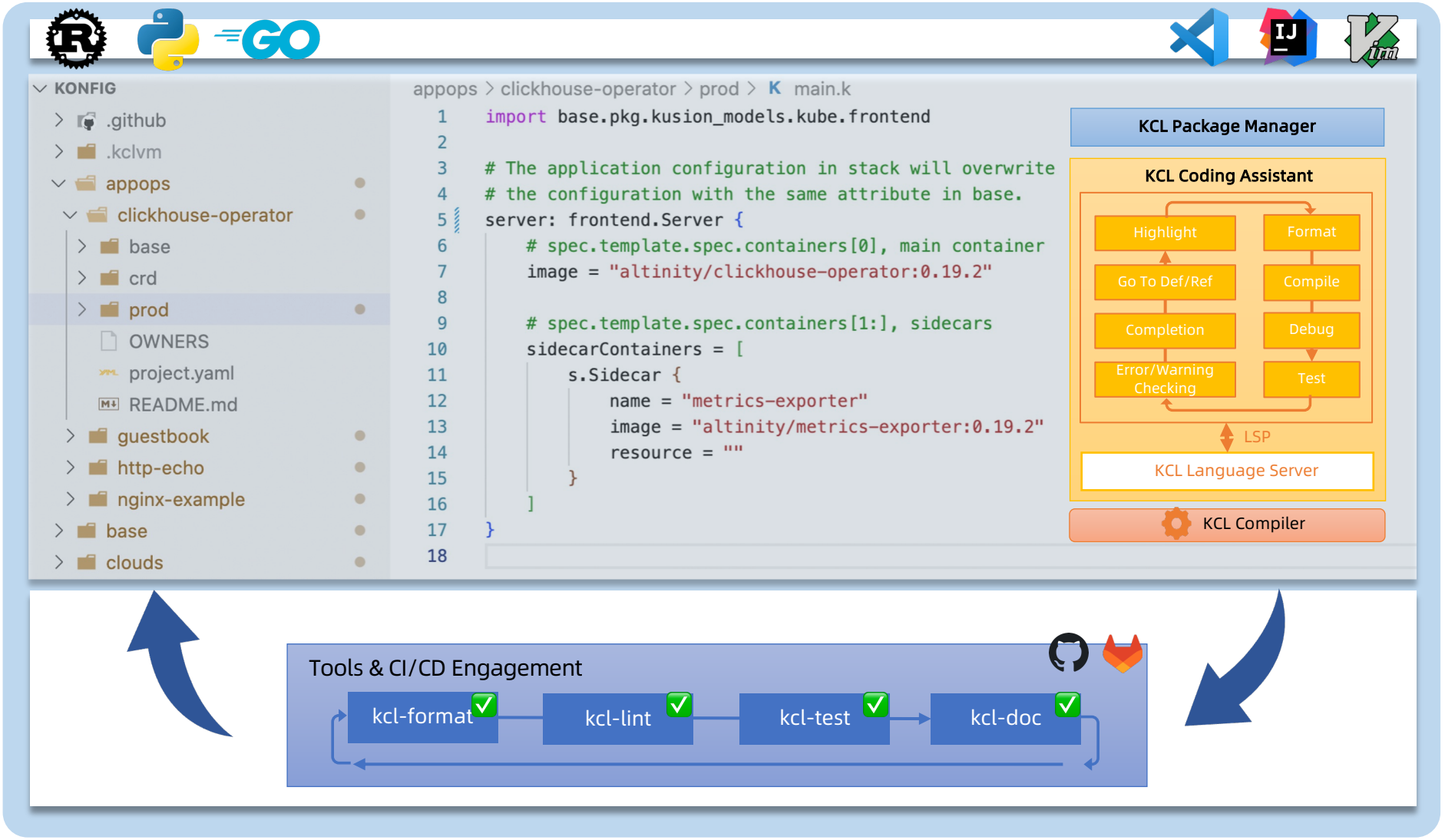
- ***Multi-Lang Binding***: *Go SDK, Python SDK, Java SDK, etc.*
- ***Package Management***: *KPM Tool & Registry*
- ***Data and Schema Migration***: *KCL Import tool*
- ***Cluster Integration***: *KCL Operator instead of deploying webhooks.*
- ***KRM Support***: *Unified spec and multiple tools e.g., kubectl-kcl plugin, helm-kcl plugin, kustomize-kcl plugin, kpt-kcl-plugin …*
- ***Platform Engineering:*** *As the Dynamic Configuration Management (DCM) language to deliver applications with KusionStack*

Architecture

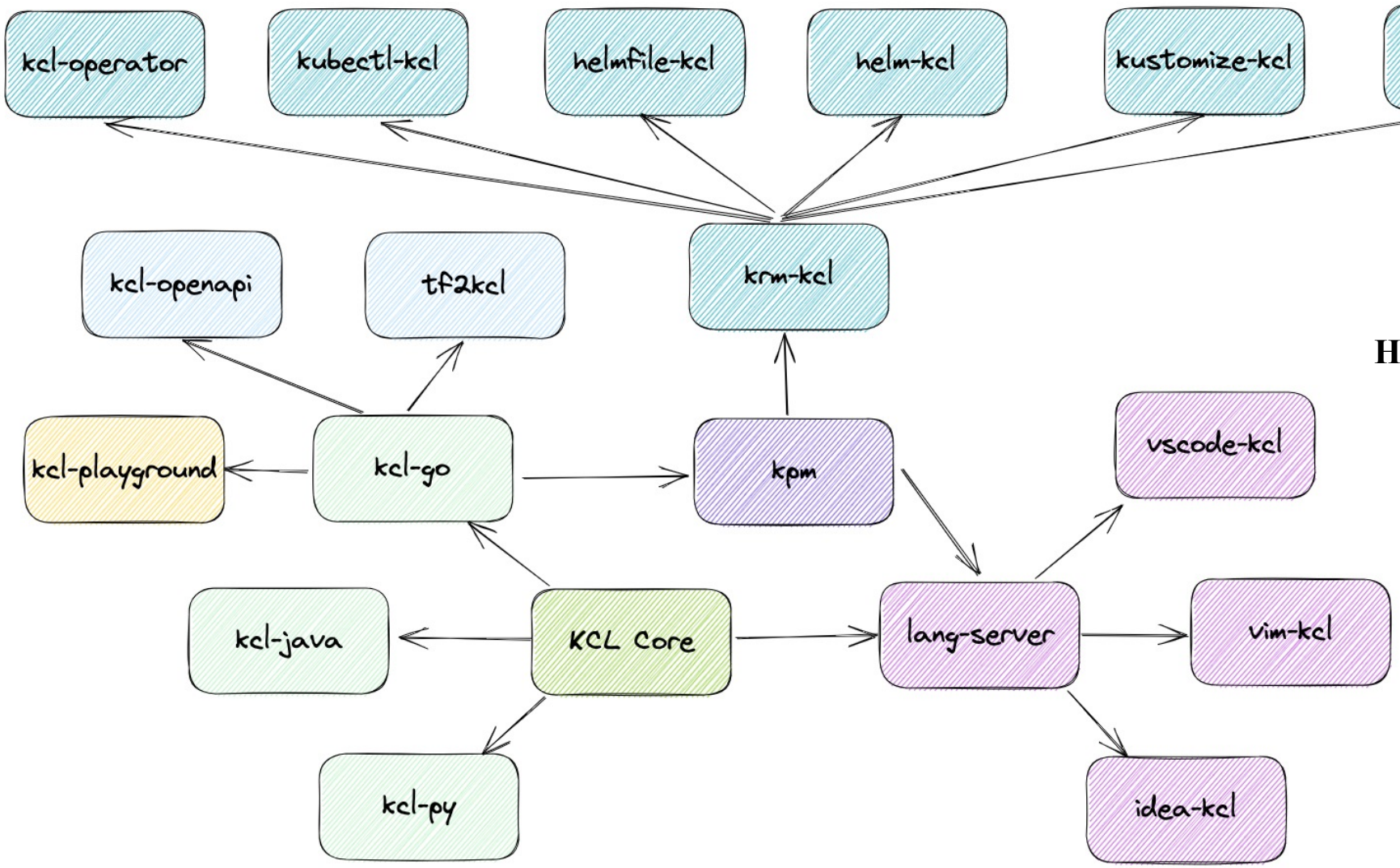
03

Workspace

Language + Tools + IDEs + SDKs + Plugins



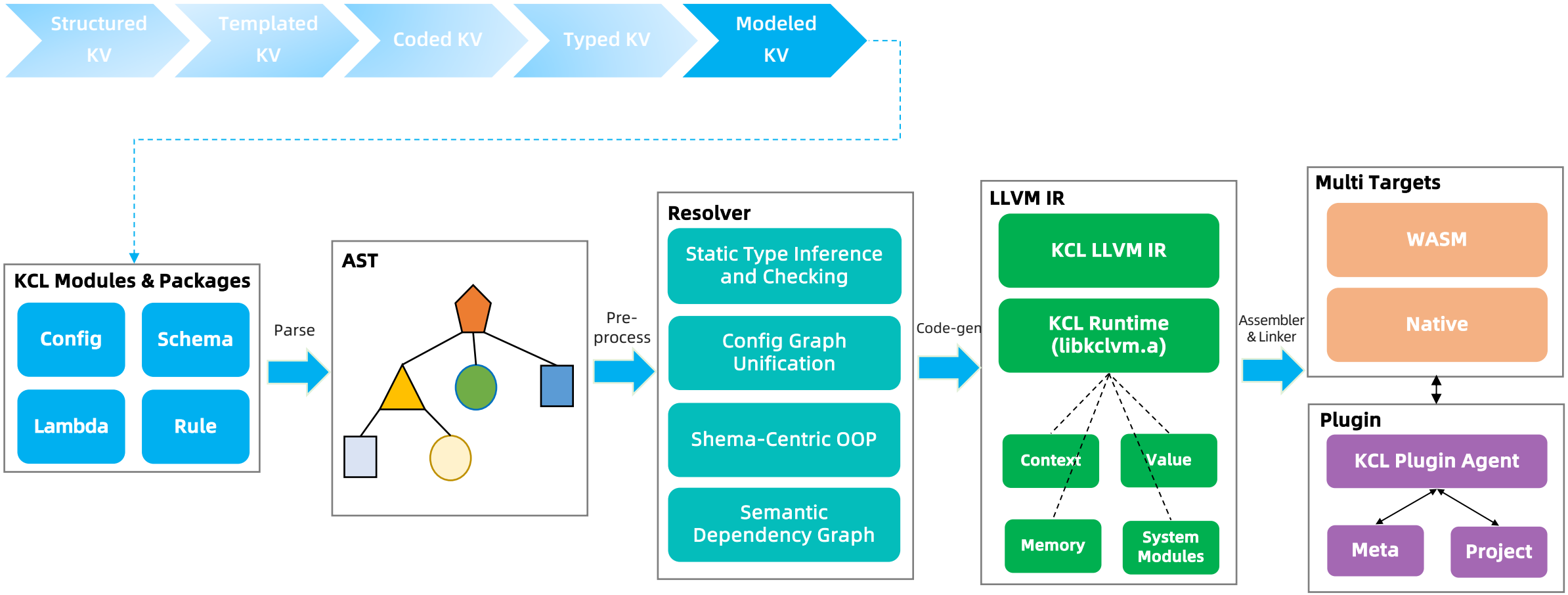
Components



How to (For App/Platform Dev & SRE)

- Lang
- Tools
- IDE
- SDKs
- Playground
- Operators
- UI

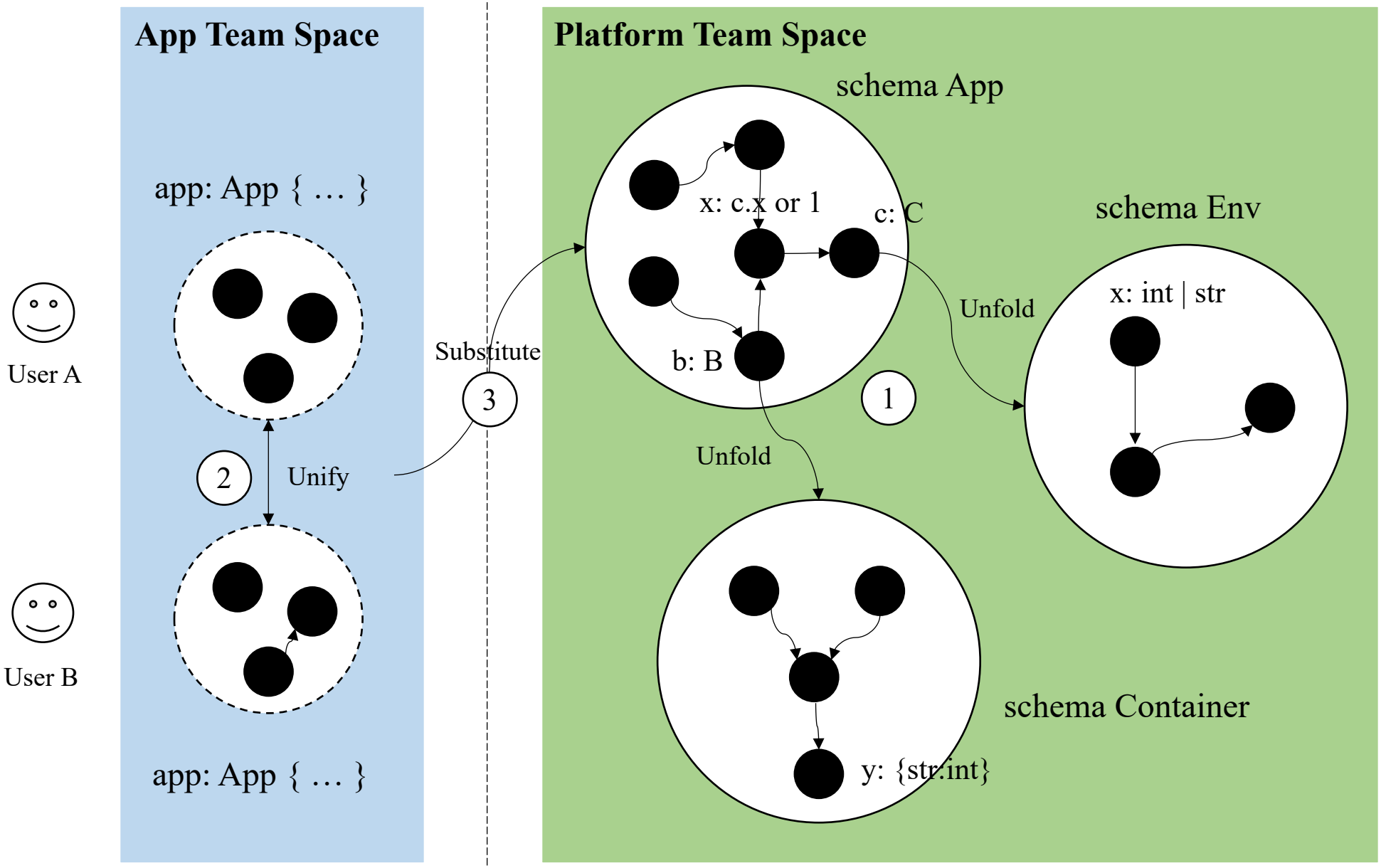
Language Core



Spec Driven

KCL Spec | KCL Multi-Language Spec | KCL OpenAPI Spec

Graph Model



- ✓ **Config Generation**
  - ✓ Automatic merge
  - ✓ Multiple Merge Strategy
- ✓ **Config Mutation**
  - ✓ Data Integration
  - ✓ CRUD API
- ✓ **Config Validation**
  - ✓ Static Type
  - ✓ Validation
  - ✓ Immutability
- ✓ **Config Abstraction**
  - ✓ Role based collaborative

Dependency

Attribute

Data

Schema Definition

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

Evaluation

04

Related Projects

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
- …

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
- …

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

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
- …

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**
- …

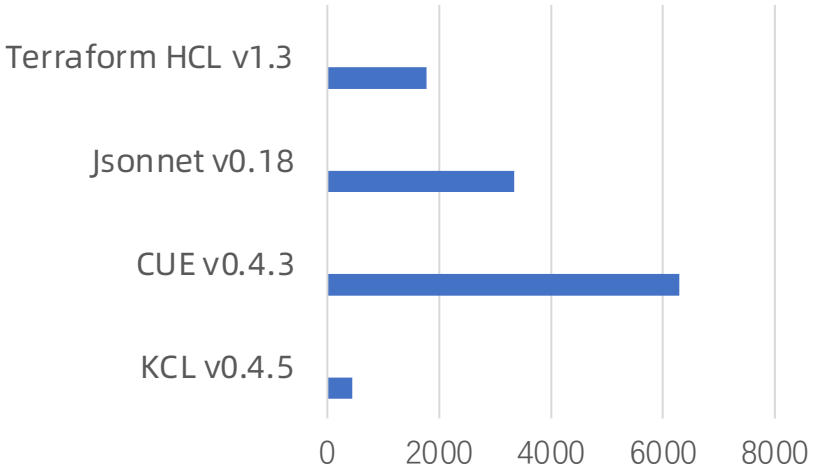
Product

- KusionStack
- KRM-KCL Tools and Operators

Performance

Loop and Function

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

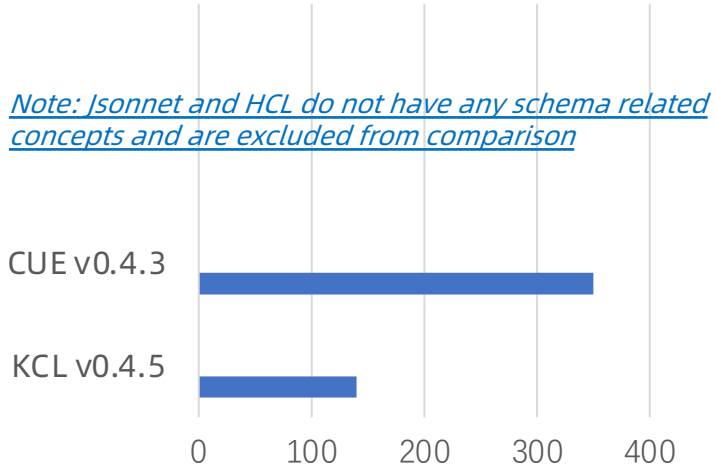


Kubernetes Configuration

```
import kubernetes.api.apps.v1

deployment = v1.Deployment {}
```

Note: Jsonnet and HCL do not have any schema related concepts and are excluded from comparison



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

Roadmap

- **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

Community

Open source in ***May 2022***

~30 Releases

20+ Presentation & Blogs

8 Maintainers

~20 Contributors

Biweekly Community Meeting and Blogs

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

Adopters

ANT GROUP

700+

Committers

~1M

KCL Codes

10K+/day

KCL Compilations

有赞

Use the KCL OpenKruise -> Application abstraction to support the middleware containerization configuration management in over 100+ clusters

HUAWEI

Use KCL to solve the problems such as the difficulty in abstracting Terraform HCL and the lack of scalability can be solved to meet its SaaS configuration UI definition requirements with 2000+ KCL model code, 3+ KCL Plugins

tusimple

KYLIGENCE

Add more…

Resources

- **Website**
  - *https://kcl-lang.io/*
- **GitHub Organization**
  - *https://github.com/kcl-lang/kcl*
- **Community**
  - *https://github.com/kcl-lang/community*
- **Twitter**
  - *@kcl language*
- **Slack**
  - *https://kcl-lang.slack.com*
  - *https://kusionstack.slack.com*

THANKS