

面向WASM的编程语言探索实践

(以KCL和凹语言为例)

柴树杉 (蚂蚁集团)









- 凹语言联合发起人
- KCL配置语言创始团队成员
- PLOC编程语言开放社区联合发起人
- 《国产编程语言蓝皮书》联合策划和编撰
- 《WebAssembly标准入门》联合作者







国产编程语言蓝皮书 2023



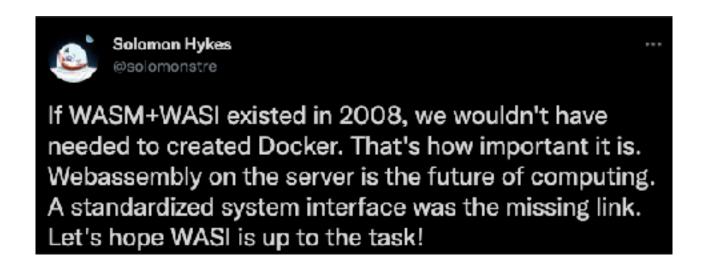


- 1. 为何关注WASM技术
- 2. KCL语言: WASM对技术选型的影响
- 3. 凹语言:简化了WASM编程
- 4. IDE和现代编程语言
- 5. 编程语言未来挑战

1. 为何关注WASM技术



- •WASM 是第一个世界级虚拟机标准,跨浏览器/云
- •WASM 比JVM简单太多,几乎每个程序员都可以手撸一个
- •WASM 给了程序员计算的自由





Ending定律: 所有可被编译为WebAssembly的,终将被编译为WebAssembly!

2. KCL语言: 简介



名字含义: Kubernetes+Configure+Language

• 设计目标:解决云原生时代的YAML爆炸问题

• 发展历程: 2020年立项, 2022年开源, 2023年捐给CNCF

• 落地用户: 蚂蚁、华为、有赞等

• KCL官网: https://kcl-lang.io/



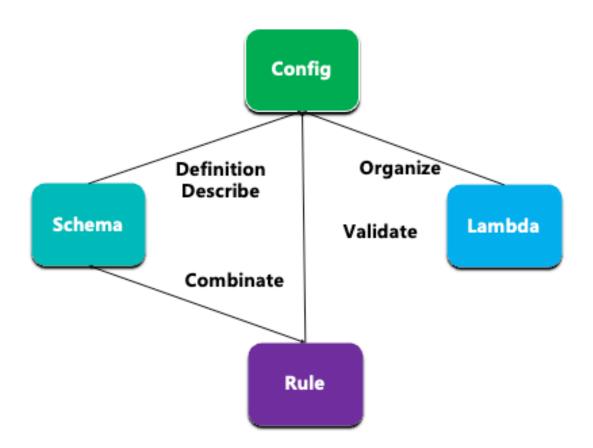
注:本页商标均为引用,版权属于原作者。

Automation & Configuration airship رگي Apollo **BOSH** CFEngine CHEF CLOUDIFY Çidence Juju KUSION MAAS openstack.

2.1 KCL语言: 更简洁安全的方式声明K8S资源



Config + Schema + Rule + Lambda

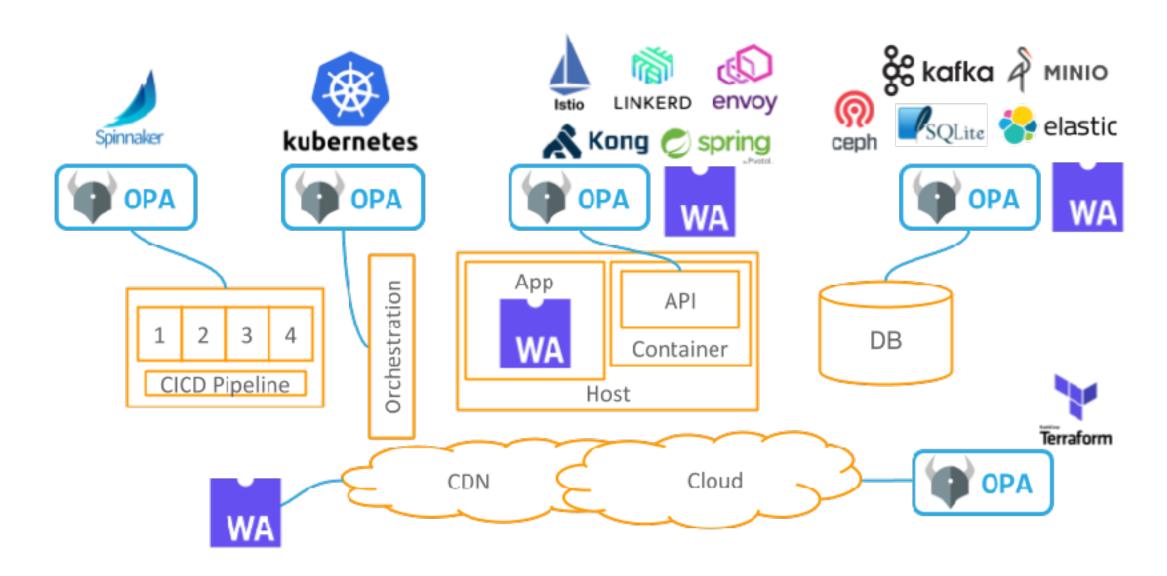


Pattern: k = (T)v

```
import k8s.core.v1
# Create a Kubernetes Deployment resource.
v1.Deployement {
   metadata.name = "nginx"
   metadata.labels.app = metadata.name
   spec = {
       replicas = 3
       selector.matchLabels.app = metadata.name
       template = {
           metadata.labels.app = metadata.name
           spec.containers = [{
               name = metadata.name
               image = "nginx"
               ports = [\{ containerPort = 80 \}]
```

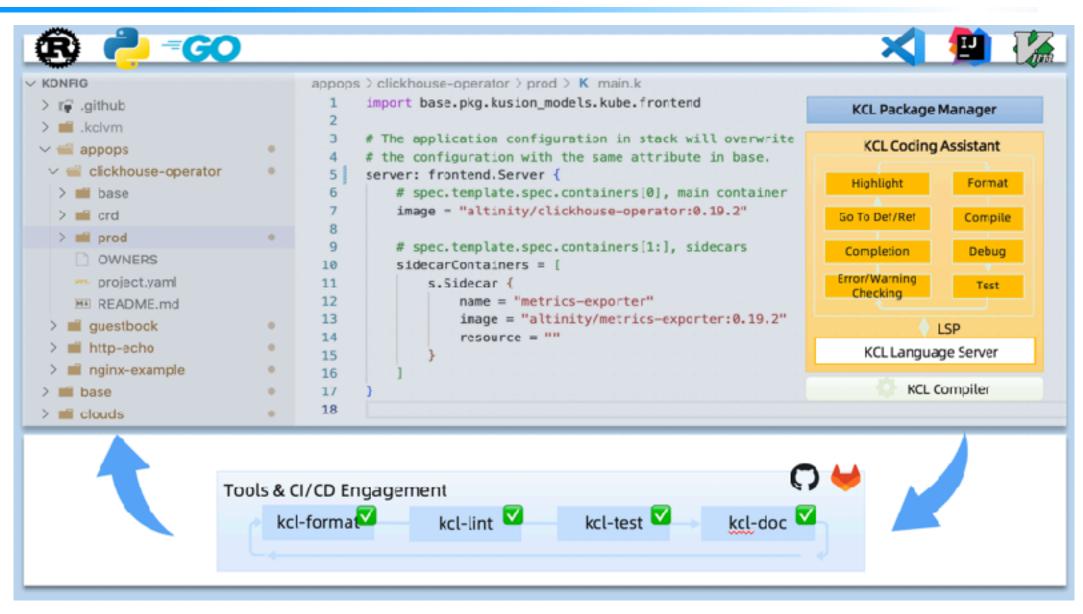
2.2 KCL语言: 同类项目技术选型参考-OPA基于WASM





2.3 KCL语言: WASM给云开发带来潜在技术支撑





2.4 KCL语言: 打通WASM是技术选型的重要因素



- 最早Python实现,后希望用静态类型语言重构
- 希望工具链能在纯浏览器执行,Go对WASM支持不足退出
- 后端采用了LLVM,工具链太重导致WASM纯浏览器困难
- 影响了SmartIR项目选型,实现了KCL最初的WASM目标。

3. 凹语言: 弥补Go对WASM支持的不足



•名字由来: WASM语言 -> WA语言 -> 凹语言

•设计目标:简单易用、强表达力、高性能、可用于工业开发

•发展历程: 2018年底立项, 2022年开源, 2023年MVP发布

•凹语言主页: https://wa-lang.org/

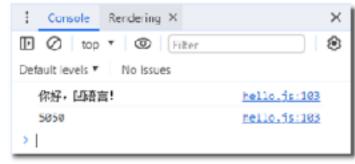


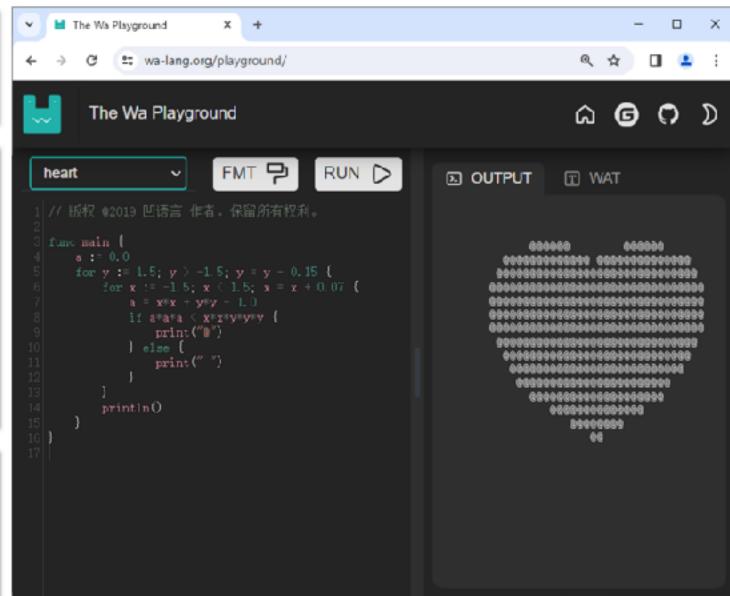


3.1 凹语言简化了WASM编程-脚手架/Playground



```
$ wa init
$ wa run hello
listen at http://localhost:8000
```





3.2 凹语言简化了WASM编程-NES模拟器



```
global nes_ctx: canvas.Context2D
     // 初始化ROM
                                                           ■ 四音宣表・NES 模拟器
     func NES_InitGame(id: int) {
                                                                  *: wa-lang.pig/nei/
         err := nes.InitConsole(id)
         if err !- nil {
                                                                                 Wa dame <sup>2023</sup>
             panic("NES InitGame failed")
                                                                                                            Game Name
                                                                                                             職斗罗
         println("NES_InitGame ok")
21
         canvas_nes, ok := canvas.QueryCanvas("#nes")
         if !ok {
             return
         nes ctx, ok = canvas nes.GetContext2D()
         if !ok {
             return
     // 玩家1 键盘调用
     func NES_SetButtons1(btnA, btnB, btnSelect, btnStart, btnUp, btnDown, btnLeft, btnRight: bool) {
         nes.Console_SetButtons1([8]bool{
             btnA, btnB, btnSelect, btnStart, btnUp, btnDown, btnLeft, btnRight,
```

3.3 凹语言简化了WASM编程-工业场景



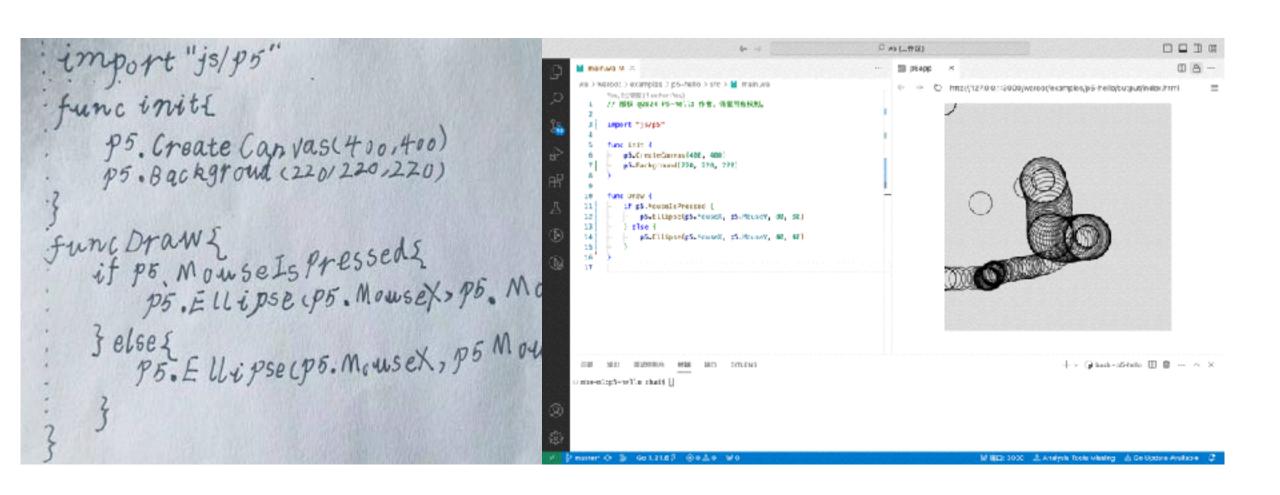
```
Func maán [
  printin("你好, 凹语言! ")
                                       V @ MERSHELDINGE
  c, _ := canvas.QueryCanvas("canvas
  ctx, _ := c.GetContext2D()
                                                                         章 🗇 🚢

← → C ② localhest8000
  ctx.SetFillStyle("green")
  ctx.FillRect(50, 10, 300, 10)
   ctx.FillText("https://wa-lang.org/
                                                                       drawWalogo(ctx)
unc drawwaLogo(ctx: canvas.context2D)
  ctx_MeveTo(50, 50)
  ctx.LineTo(150, 50)
  ctx.LineTo(150, 150)
  ctx.LineTo(250, 150)
  ctx.LineTo(250, 50)
  ctx.LineTo(350, 50)
  ctx.LineTo(350, 350)
  ctx.LineTo(50, 350)
  ctx.LineTo(50, 50)
   ctx.Rect(95, 95, 10, 10)
  ctx.Rect(295, 95, 10, 10)
  CTX_MoveTo(200, 230)
  ctx.LineTo(200-34, 230+34)
  ctx.LineTo(200-34-34, 230+34-34)
  ctx.MoveTo(200, 230)
  ctx.LineTo(200+34, 230+34)
  ctx.LineTo(200+34+34, 230+34-34)
   ctx.Stroke()
```

```
"gpu"
mport "matrix"
mport "math"
import "math/rand"
                                              C O locahost8000
                                                                          京 日 🏝
   CarvasWidth - 512
   CarvasHeight = 512
                                       凹语言WebGPU示例--RenderBundle
   uniformBufferSize - 4 * 16
                                       Repo: https://gitee.com/wa-lang/webgpu
func main {
   println("你好, GPU")
   vec3.T()
const shaderCode -
struct Uniforms (
 viewProjectionMatrix : mat4x4f
@group(0) @binding(0) var<uniform> unif-
@group(1) @binding(0) var<uniform> mode
struct VertexInput [
 @location(0) position : vec4f,
 @location(1) normal : vec3f,
 @location(2) uv : vec2f
Struct VertexDutput {
 @builtin(position) position : vec4f,
 Miocation(0) normal: vec3f.
 @location(1) uv : vec2f,
fn vertexMain(input: VertexIrput) -> VertexOutput {
```

3.4 凹语言简化了WASM编程-P5儿童编程

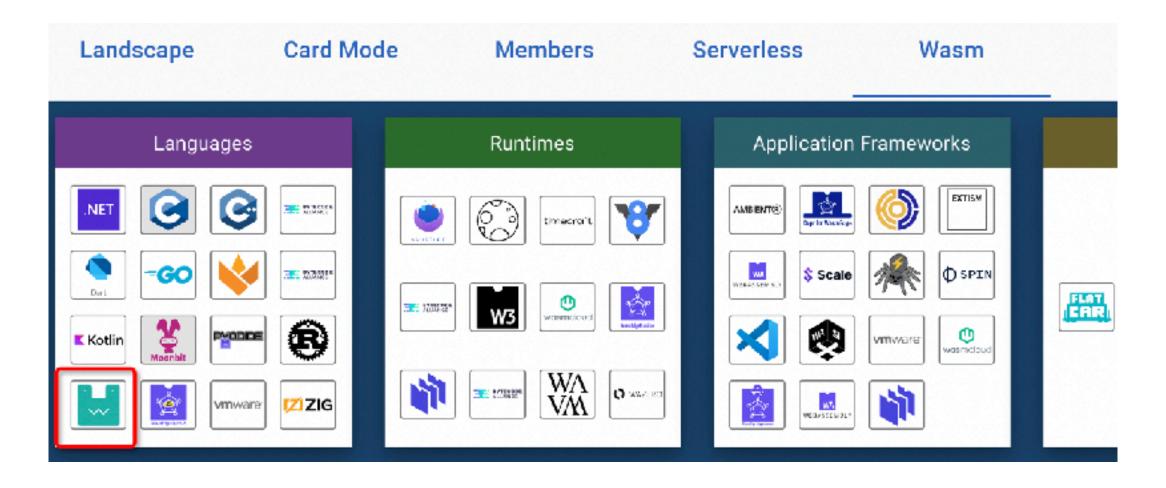




来自杭州一小学三年级小学生用凹语言写的程序

3.5 凹语言简化了WASM编程-CNCF wasm全景图

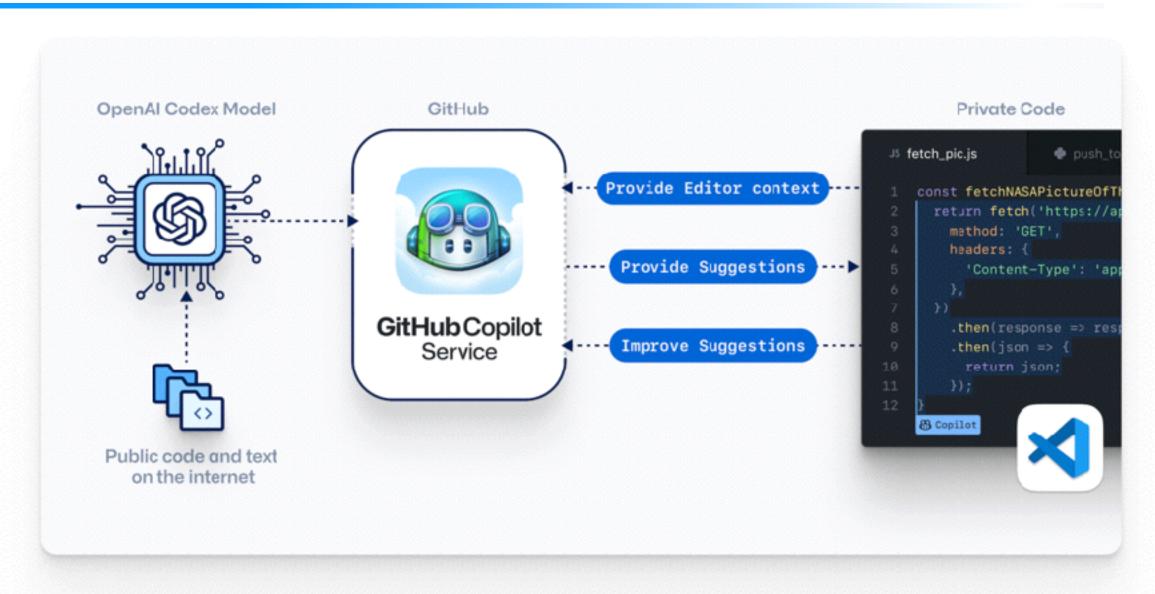




https://landscape.cncf.io/?group=wasm

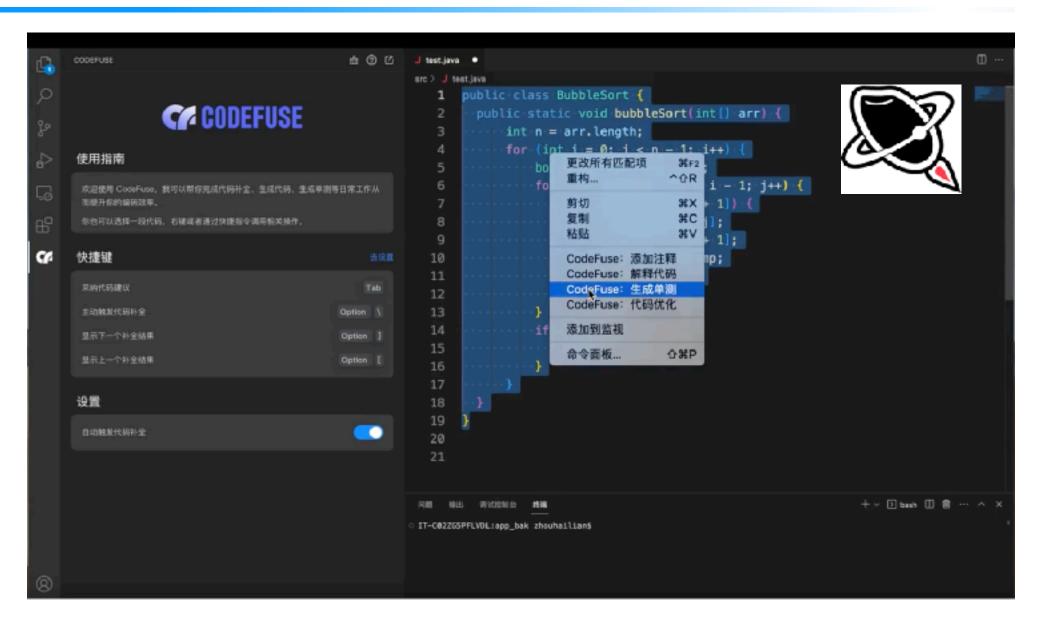
4. IDE和现代编程语言





4.1 IDE和现代编程语言: OpenSumi+CodeFuse





5. 编程语言未来挑战







THANKS / 感谢!





