



GOPS 2019
ShenZhen



GOPS 全球运维大会

2019
- AIOps 风向标

大会时间：2019年4月12日-13日

指导单位：



主办单位：



支撑滴滴快速发展的代码工具链演进史

杨永强 技术专家

目录



1

滴滴代码工具链的演进

2

演进过程中的挑战——代码基础层

3

演进过程中的挑战——代码引擎层

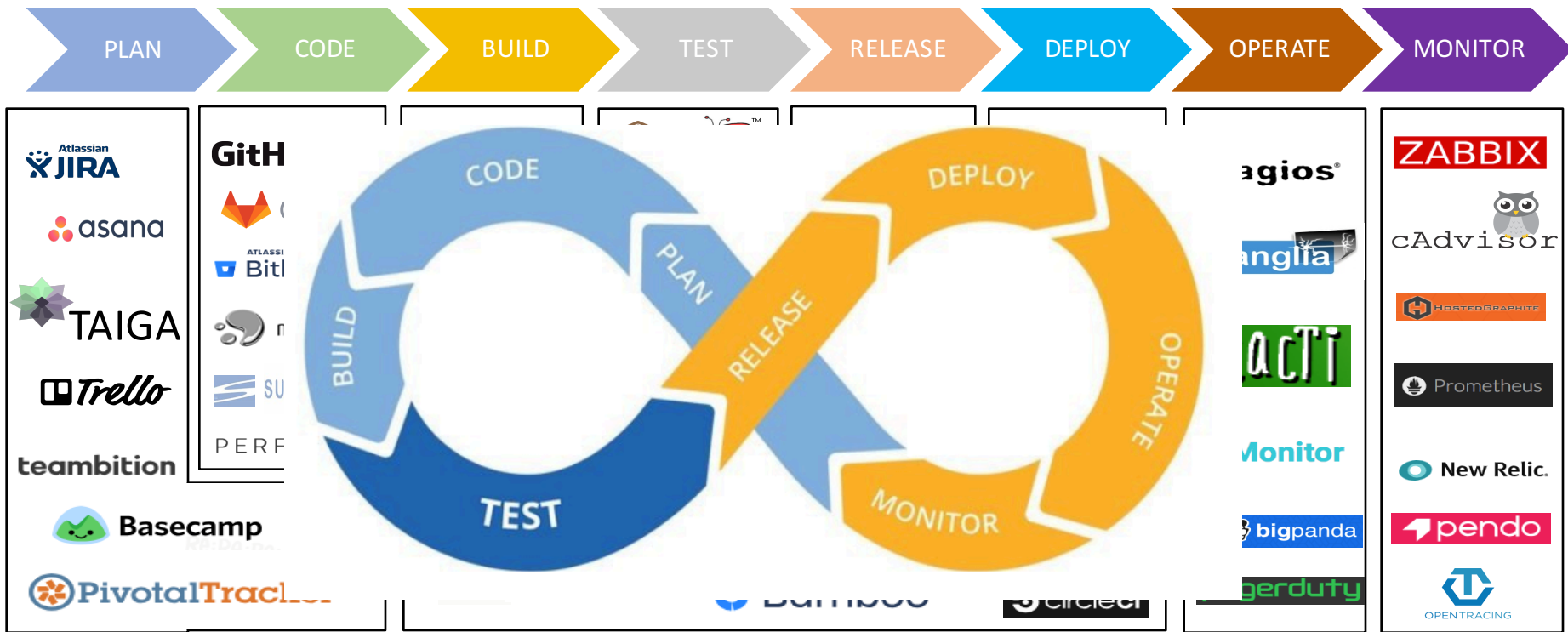
4

演进过程中的挑战——代码应用层

5

未来展望

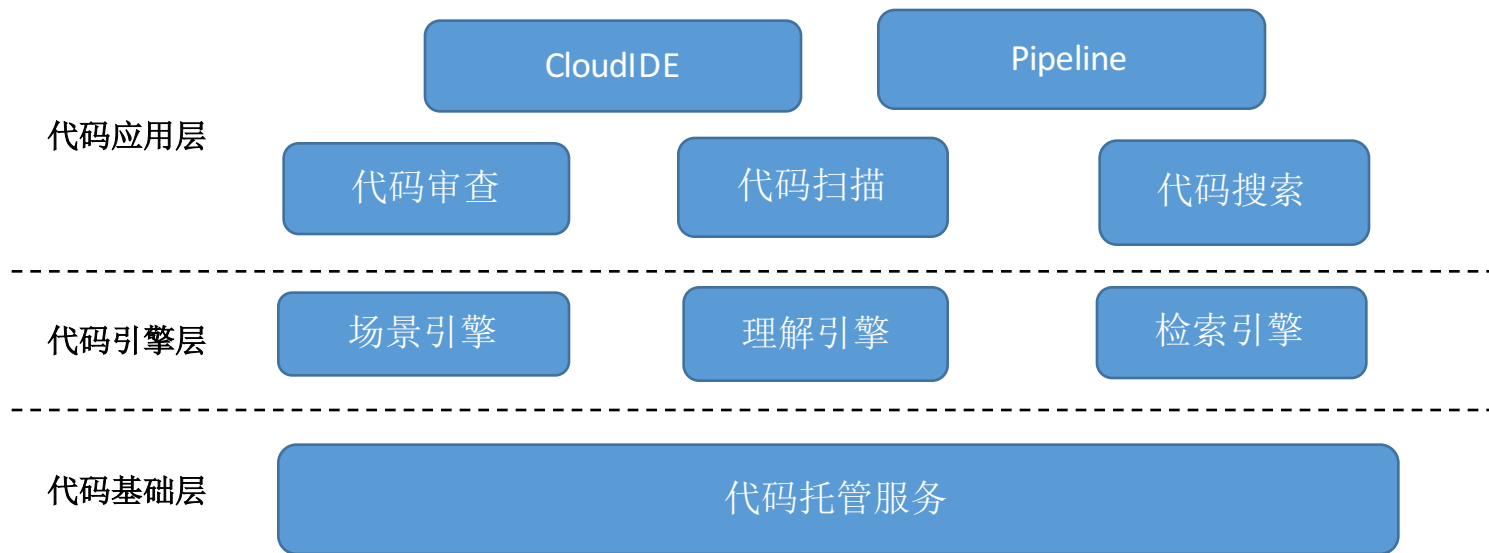
DevOps — everything as code IaC : 1.8x



滴滴代码现状

- 代码量：**80亿行**
- 语言种类：**100+**
- 代码提交次数/天：**2w**
- 代码拉取次数/天：**180w+**

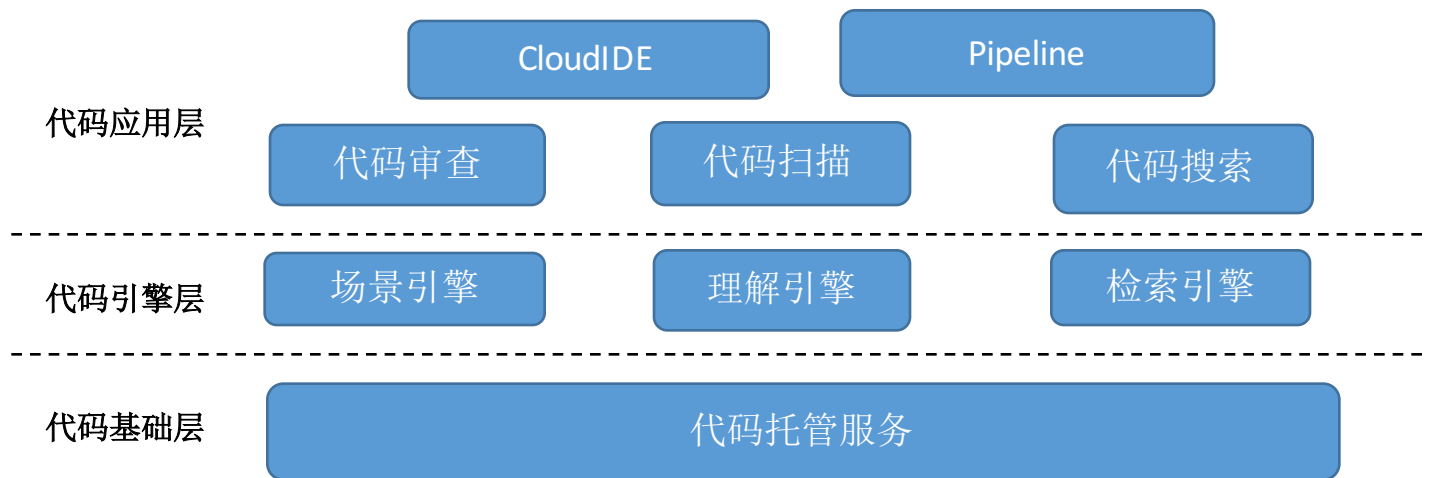
滴滴代码工具链的演进 —— 代码工具链



滴滴代码工具链的演进 —— 快速搭建

- 滴滴代码状况 (2015)

- 近**2.8亿**行代码，**30+**种语言
- 每天 **3w+**的代码拉取次数 & 每天 **~1K**的代码提交次数



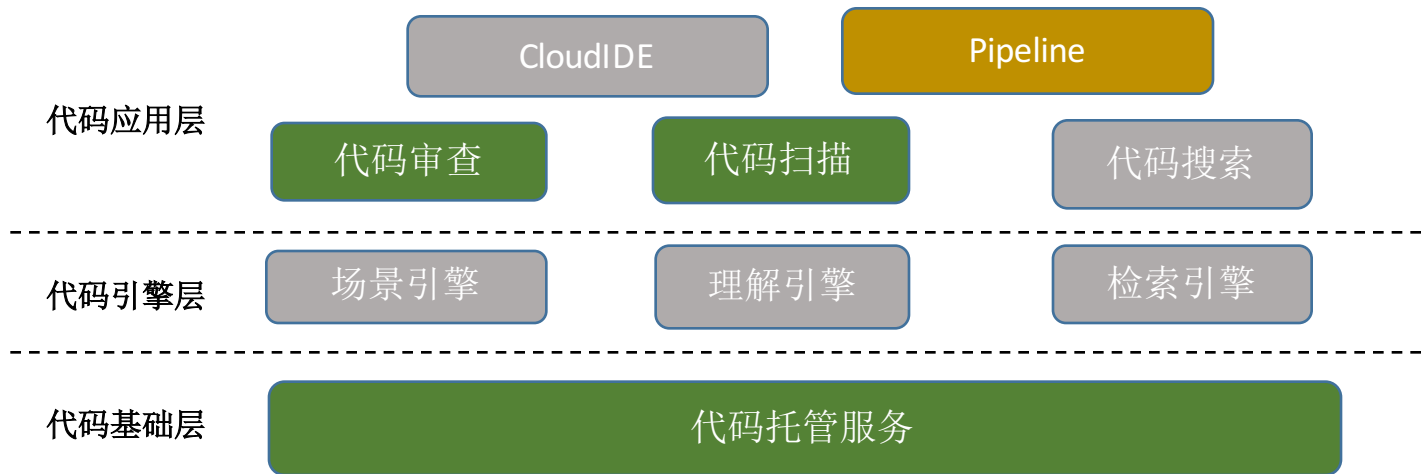
滴滴代码工具链的演进 —— 快速搭建

- 面临的挑战

- 代码基础层：多个SVN和GIT
- 代码应用层：无代码工具

- 解决方案

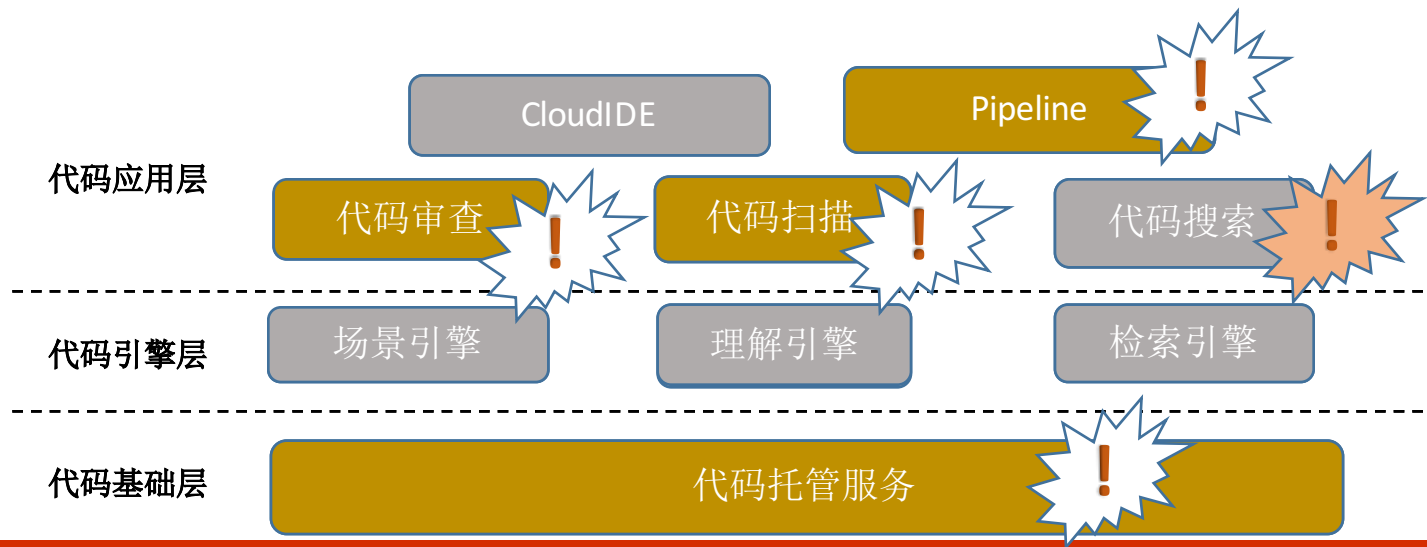
- 代码基础层：统一SVN，迁移GIT
- 代码应用层：代码审查，扫描，Pipeline搭建



滴滴代码工具链的演进 —— 满足业务

- 滴滴代码状况 (2016~2018.06)

- 近60亿 (2.8亿, 20倍) 行代码, 100+ (30+) 种语言
- 每天 120w+ (3w+ , 40倍) 的代码拉取次数, 提交 1.5w+ (1k , 15倍) 次



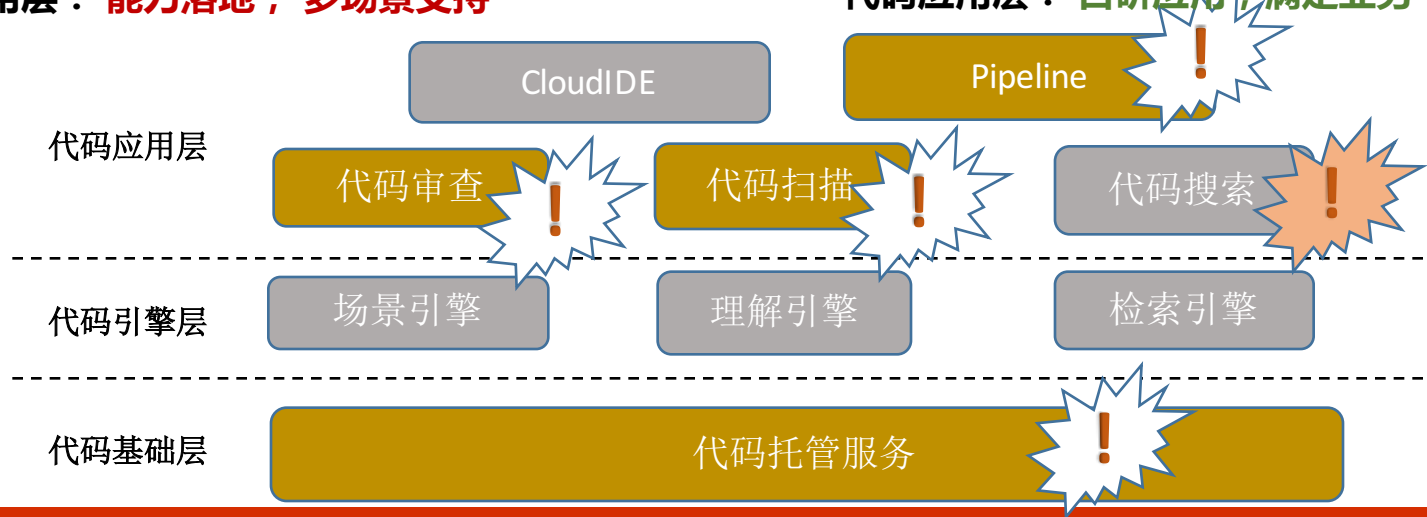
滴滴代码工具链的演进 —— 满足业务

• 面临的挑战

- 代码基础层：**高吞吐率，海量代码**
- 代码引擎层：**多元，灵活，代码理解**
- 代码应用层：**能力落地，多场景支持**

• 解决方案

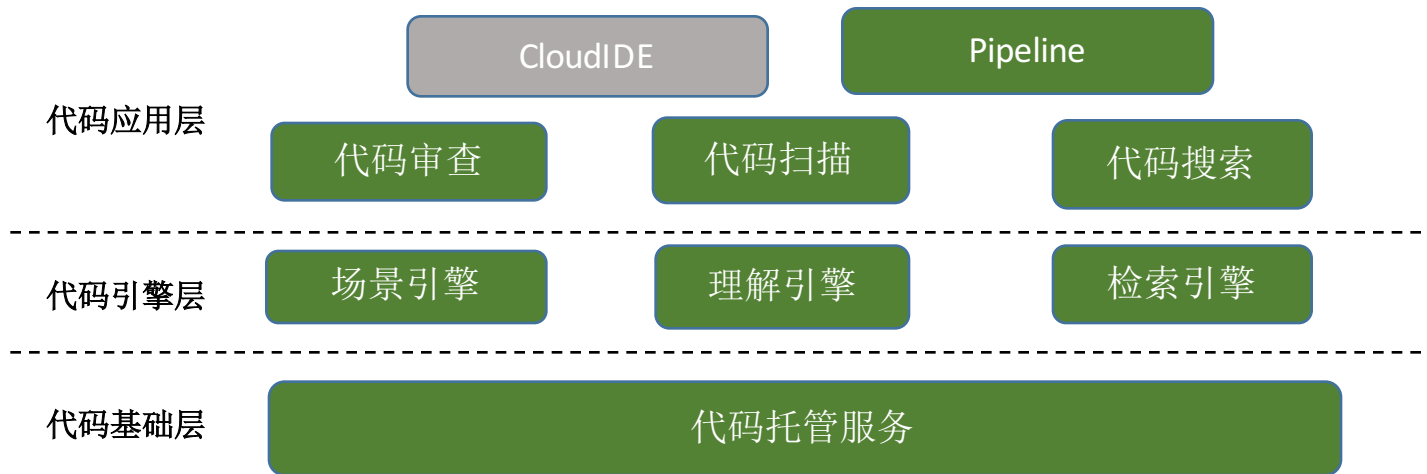
- 代码基础层：**GitLab分布式改造**
- 代码引擎层：**功能抽取，通用模型**
- 代码应用层：**自研应用，满足业务**



滴滴代码工具链的演进 —— 云端赋能

- 面临的挑战 (2018.06~)

- 代码基础层：高吞吐率，海量代码
- 代码引擎层：多元，灵活，代码理解
- 代码应用层：能力落地，多场景支持，加速



目录

1 滴滴代码工具链的演进



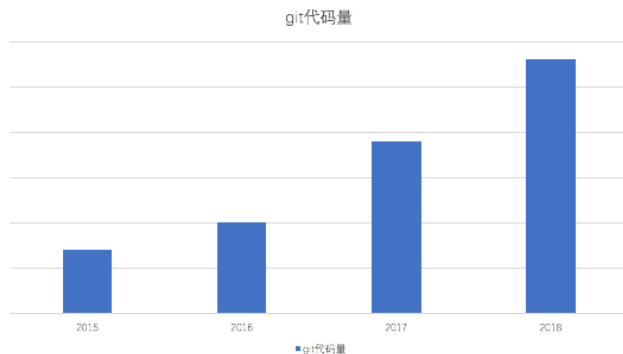
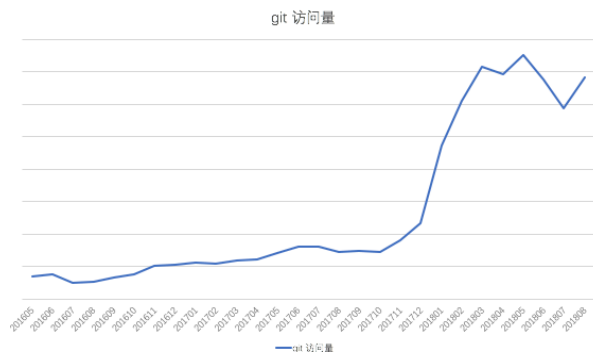
2 演进过程中的挑战——代码基础层

3 演进过程中的挑战——代码引擎层

4 演进过程中的挑战——代码应用层

5 未来展望

滴滴代码工具链的挑战 —— 代码基础层



Git请求数/周: 500w 到 7000w

cpu
idle

OOM

高吞吐率，海量代码

Git容量: 每年60%的增长速度

磁盘
容量

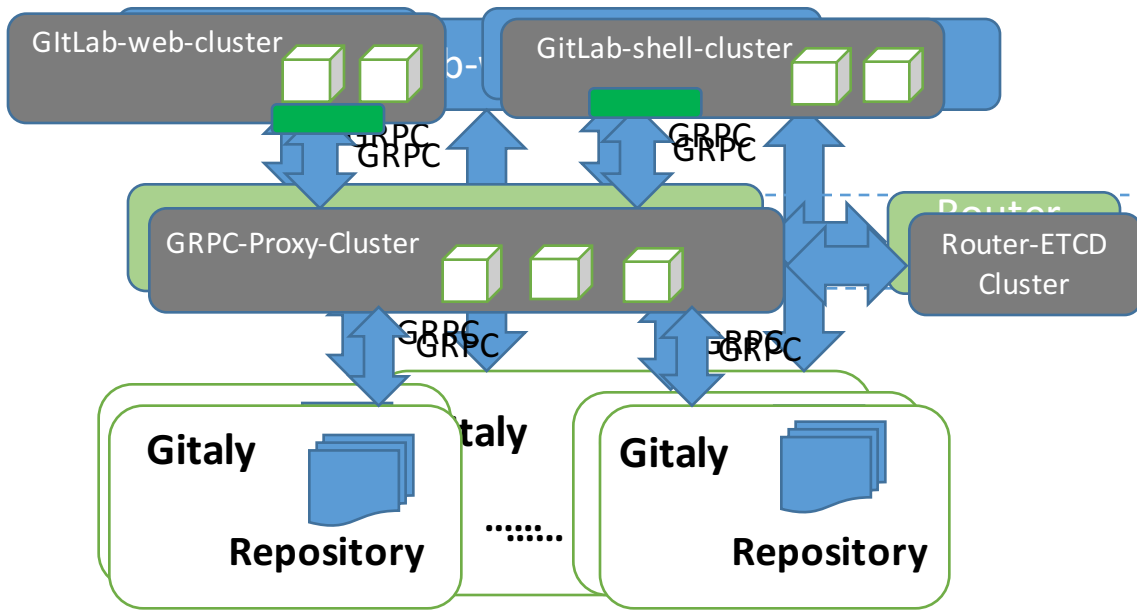
GitLab改造 —— 水平扩展

目标：高吞吐率，海量代码

解决方案：Sharding 改造
(基于golang)

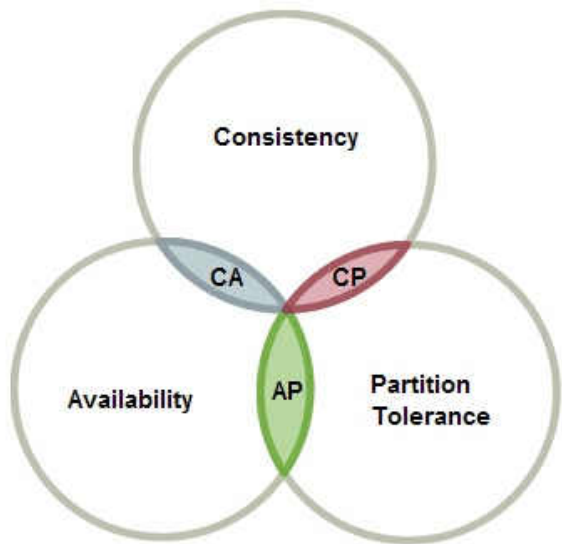
改造思路：

1. 流量标注: grpc (md)
2. 路由计算: grpc-proxy
3. 路由管理: ETCD集群
4. 路由变更: 基于WAL

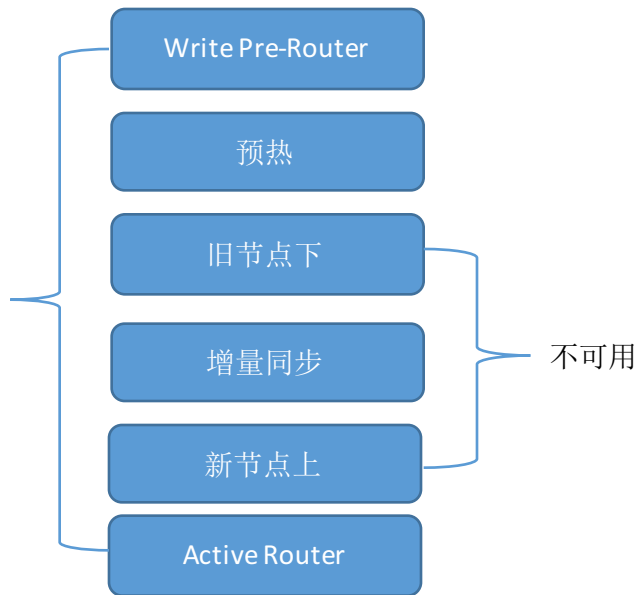


GitLab改造 —— 路由变更

- CAP 原理

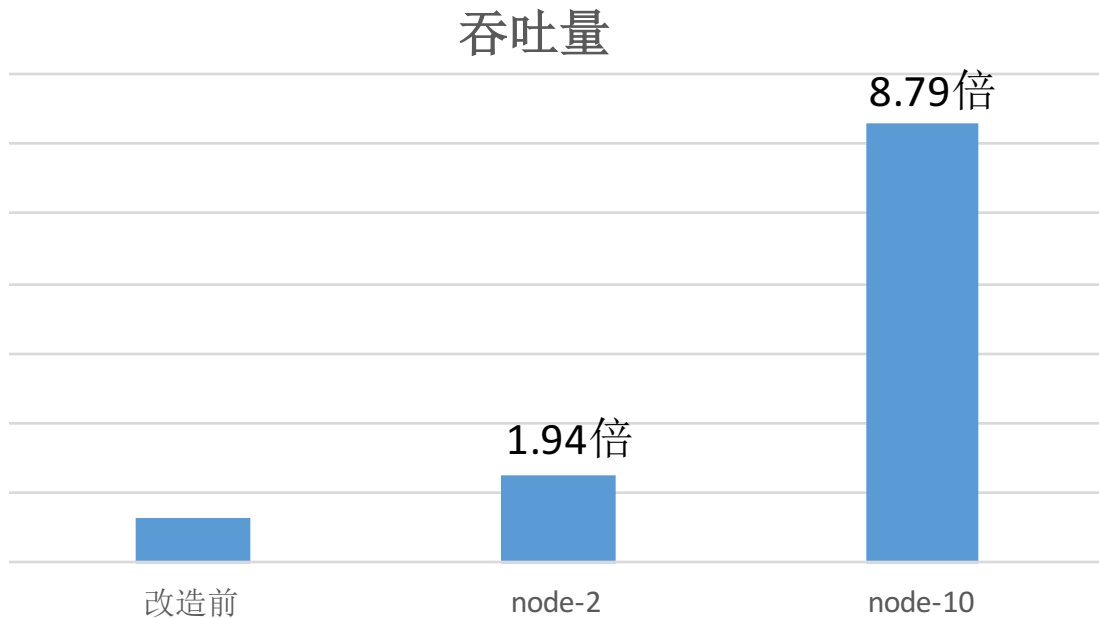


路由变更



GitLab改造 —— 效果

- Sharding 性能提升对比



目录

1 滴滴代码工具链的演进

2 演进过程中的挑战——代码基础层

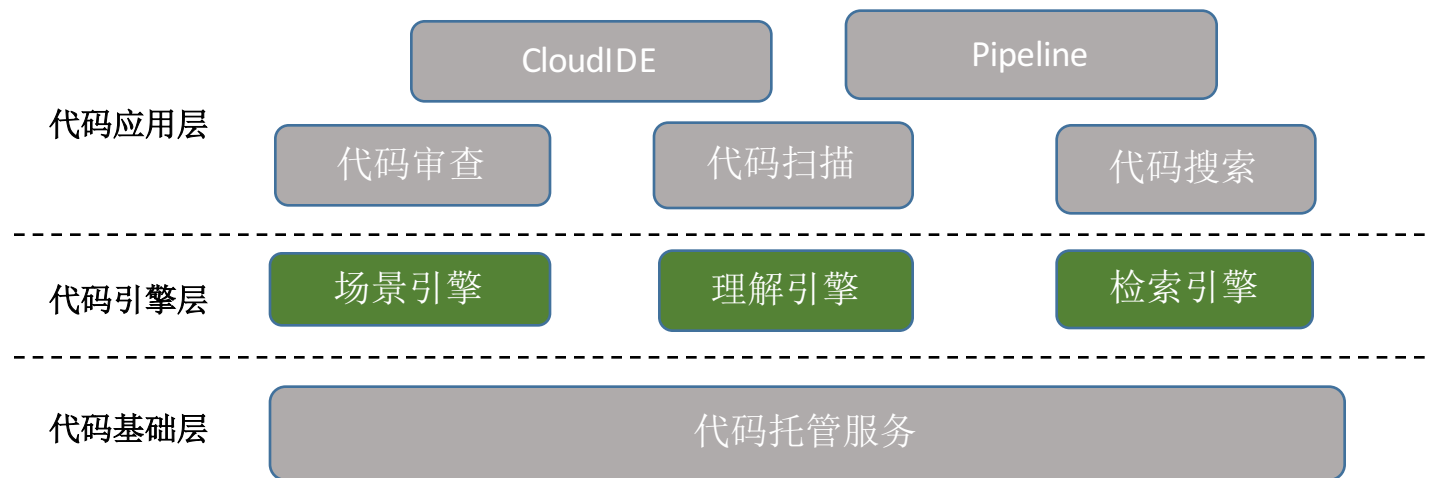
 **3** 演进过程中的挑战——代码引擎层

4 演进过程中的挑战——代码应用层

5 未来展望

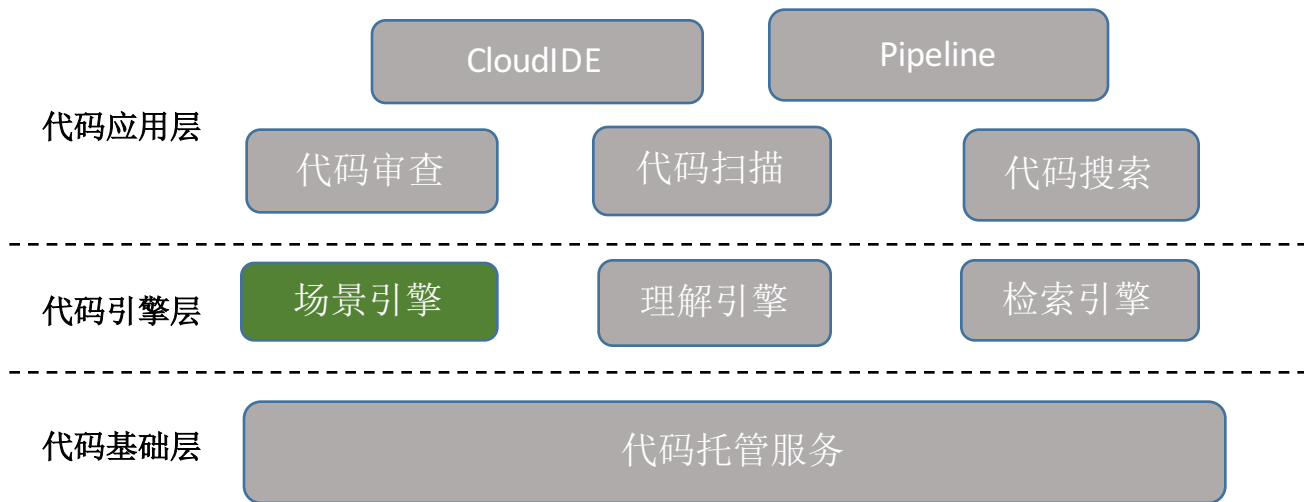
滴滴代码工具链的挑战 —— 代码引擎层

- 代码引擎层改造：多元，灵活，代码理解

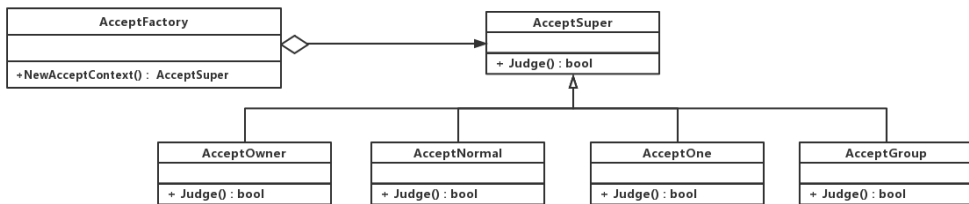


场景引擎 —— 代码审查场景

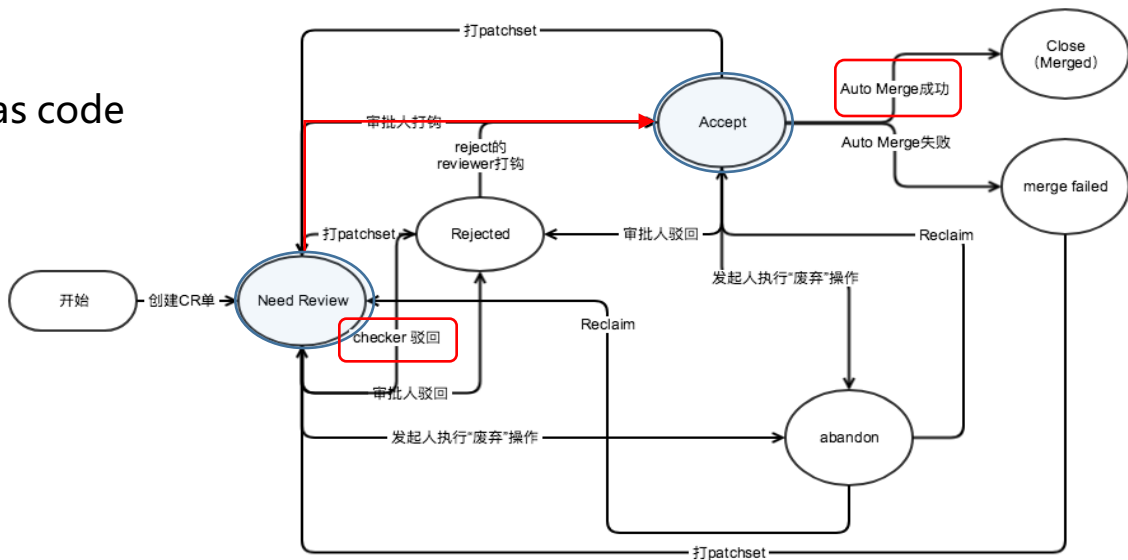
- 流程多样性
- 灵活可配



场景引擎 —— 代码审查场景

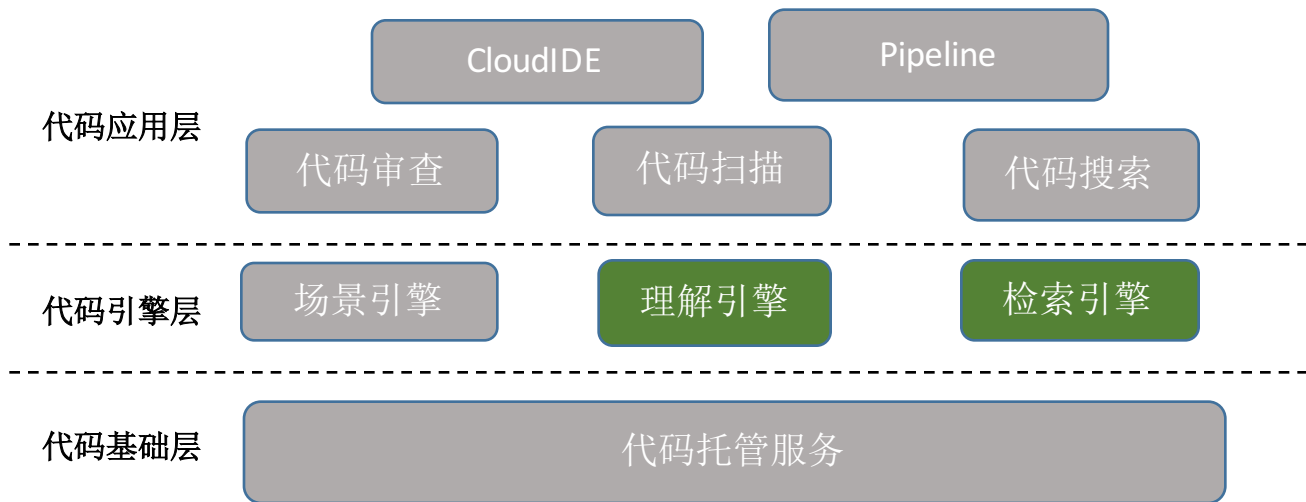


- 流程多样性：策略模式
- 灵活可配：configuration as code



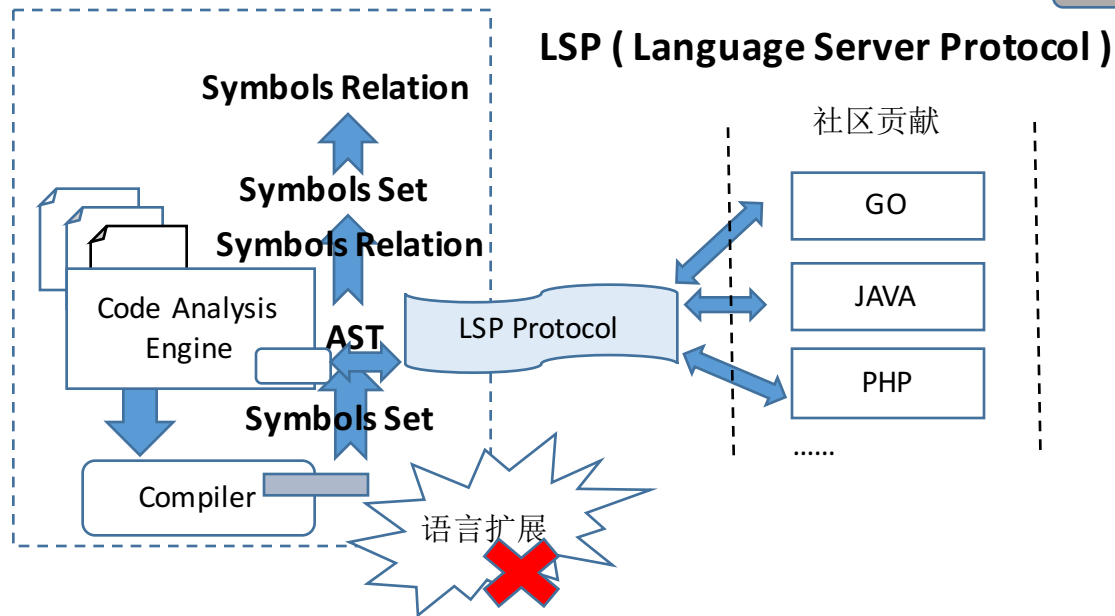
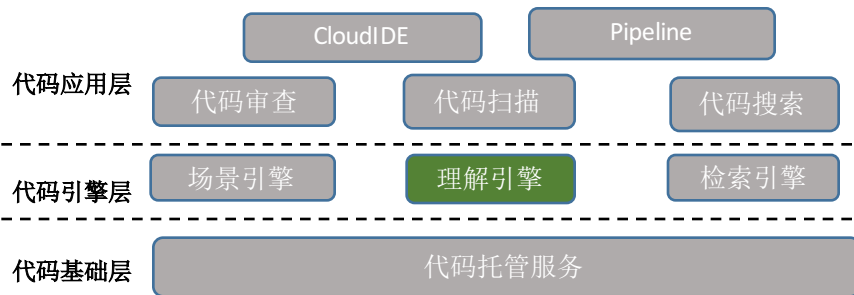
理解引擎 & 检索引擎

- ~80亿 行的代码量
- 代码重用
- 安全检测



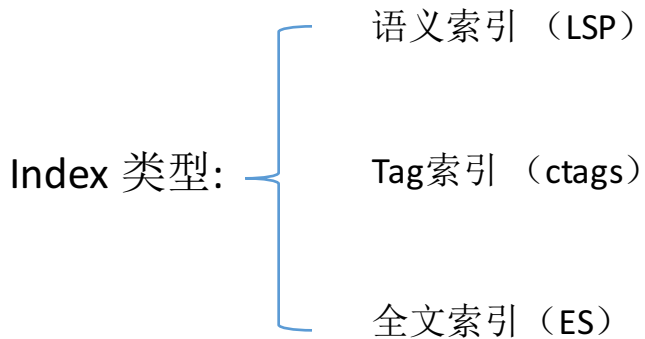
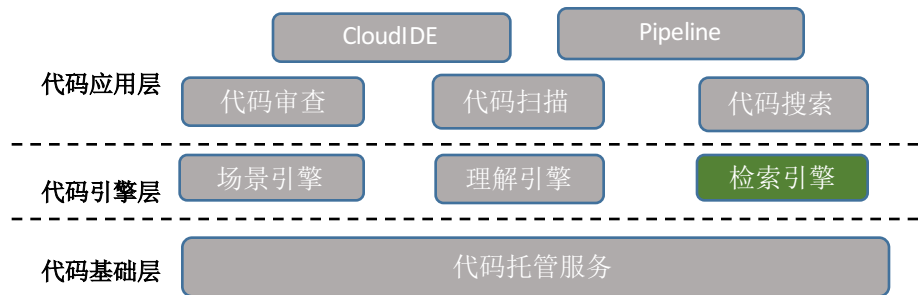
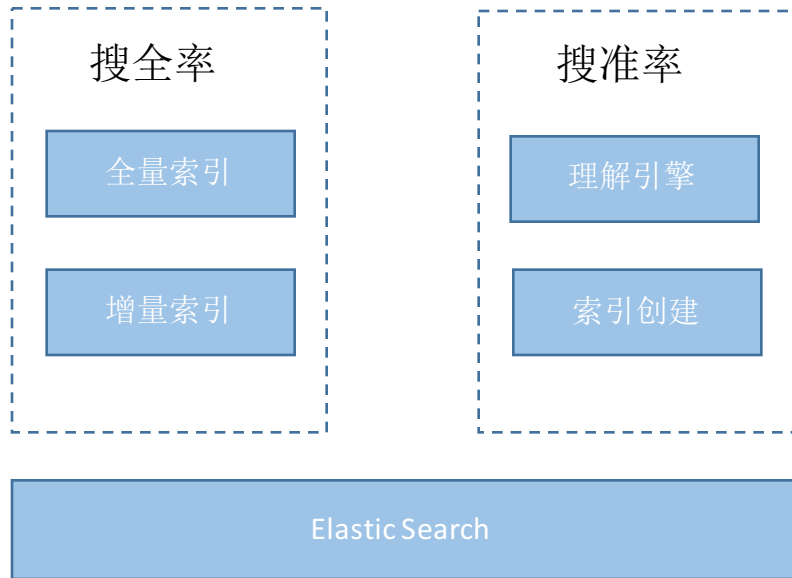
理解引擎 —— 代码语义解析

理解代码: Symbols Set && Symbol Relation



- 无状态 (http) vs 有状态 (LSP)
 - 局部串行化
 - 复用
- LSP进程池管理
 - LRU
 - 监控
 - [go] 主动kill 要wait 防止僵尸进程

检索引擎 —— Elastic Search



目录

1 滴滴代码工具链的演进

2 演进过程中的挑战——代码基础层

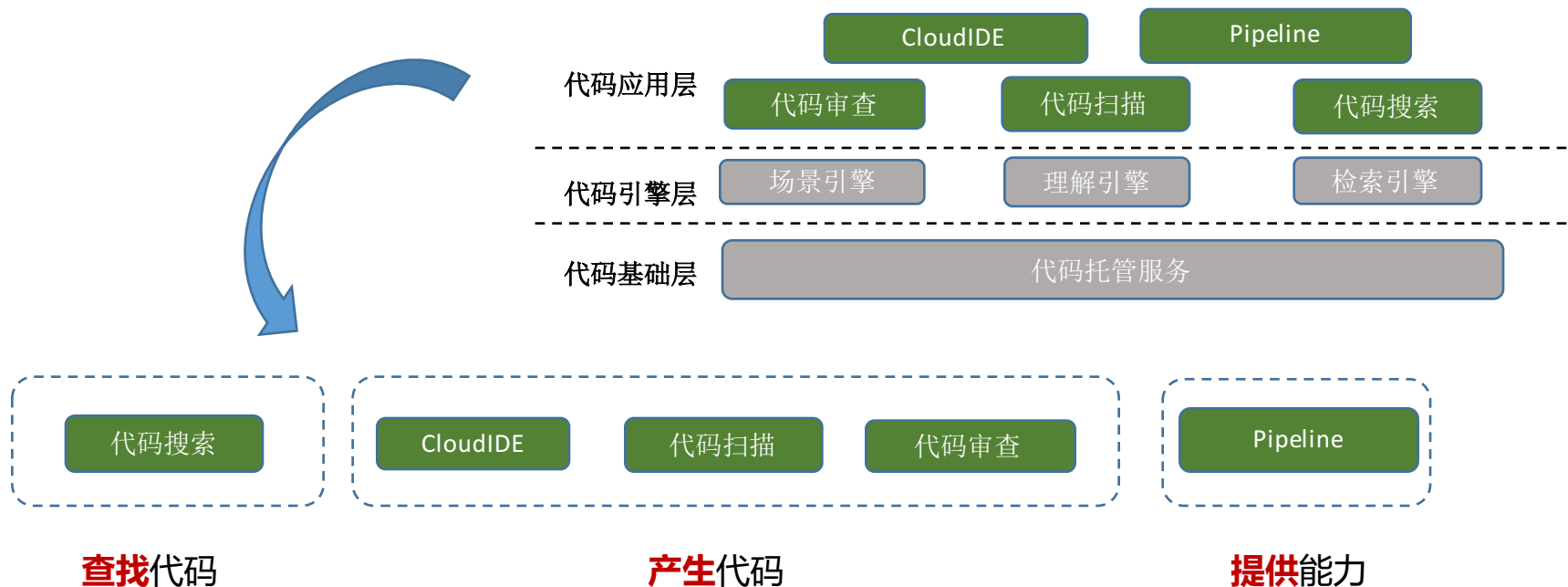
3 演进过程中的挑战——代码引擎层

 **4** 演进过程中的挑战——代码应用层

5 未来展望

滴滴代码工具链的挑战 —— 代码应用层

• 代码应用层改造：能力落地，多场景支持，加速



代码应用层 —— 代码搜索

代码搜索

查找代码

CloudIDE

代码扫描

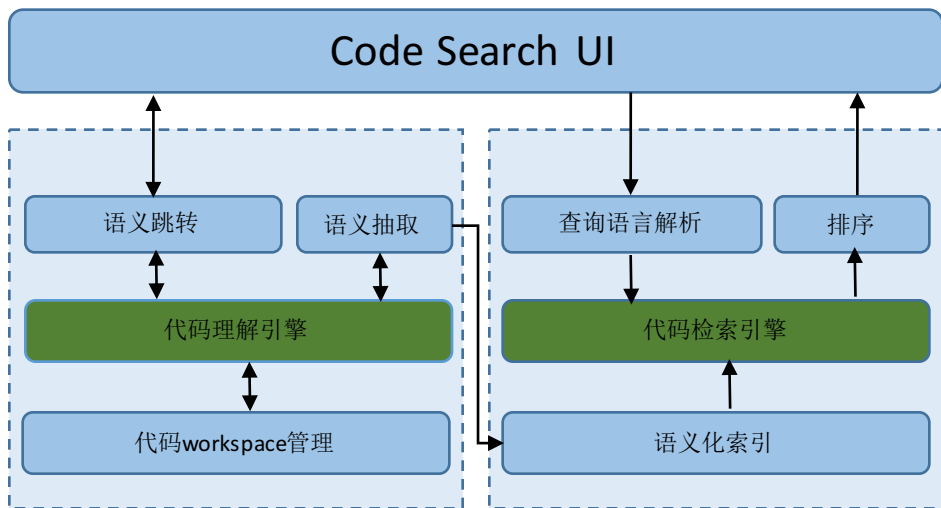
代码审查

产生代码

Pipeline

提供能力

- QL(Query Language)
 - **repo**:kunpeng logger
 - **repogroup**:codeinsight
 - **file**:search.go
 - **tags**:ip **repo**:kunpeng
 -
- Call Graph 有向无环图



代码应用层 —— Cloud IDE

代码搜索

CloudIDE

代码扫描

代码审查

Pipeline

查找代码

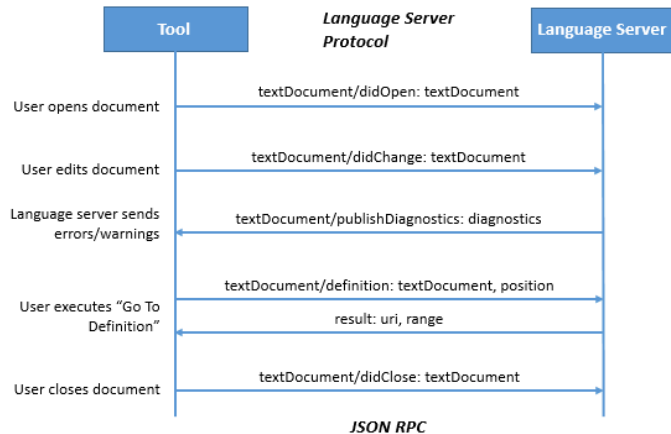
产生代码

提供能力

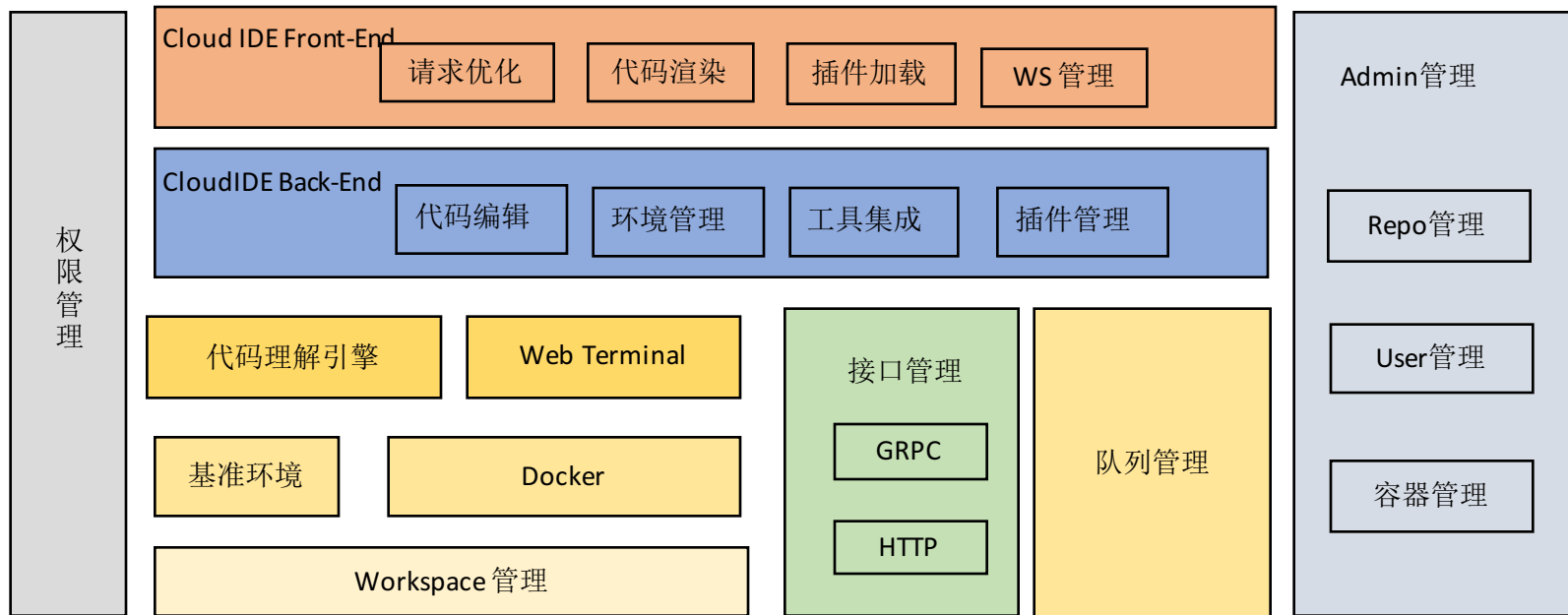
- 传统开发模式 (个人视角):
 - 本地开发, 本地工具, 环境依赖
- CloudIDE (协同视角):
 - 云端开发, 云端工具, 开发环境一键生成

CloudIDE 核心能力

- 理解引擎 (LSP)
- Workspace 容器化管理



Cloud IDE——系统架构



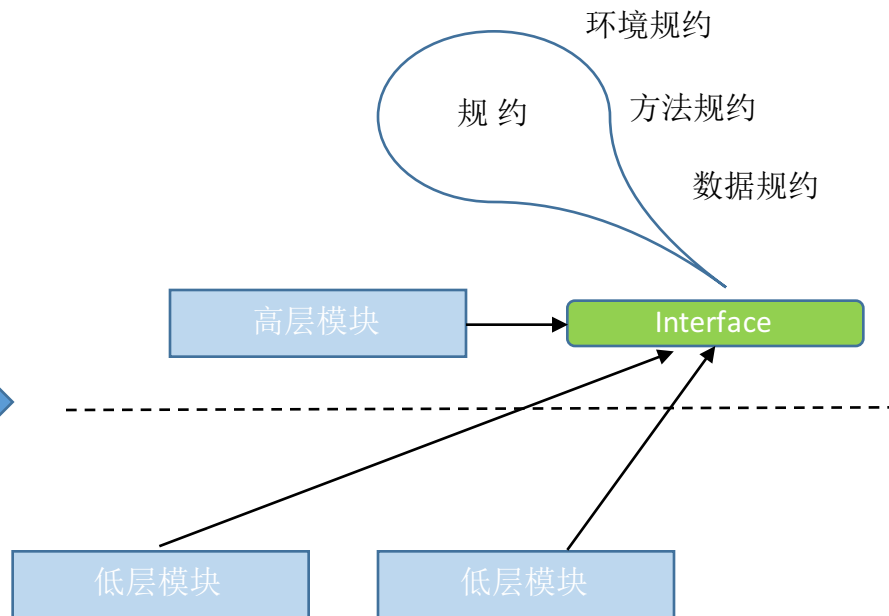
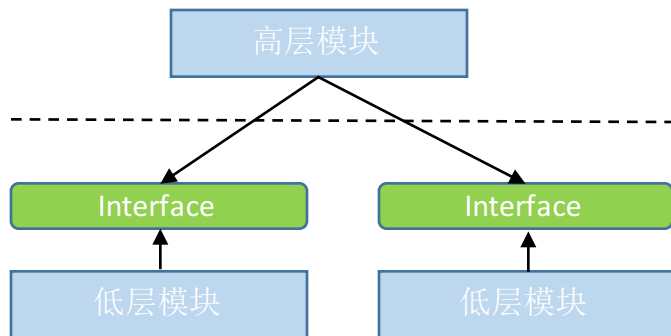
代码应用层 —— 代码扫描



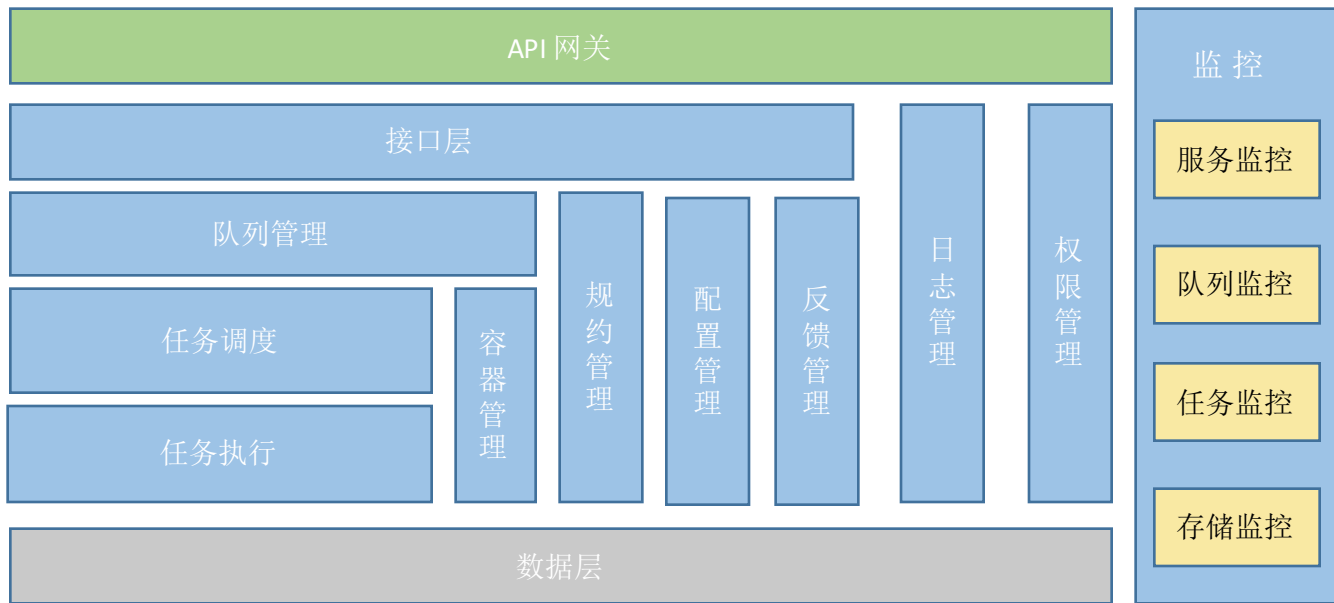
现状：种类繁多 (20+)，规则不定

问题：如何统一化接入和管理

方案：依赖倒置 (DIP)



代码扫描 —— 系统架构



代码应用层 —— 代码审查

代码搜索

CloudIDE

代码扫描

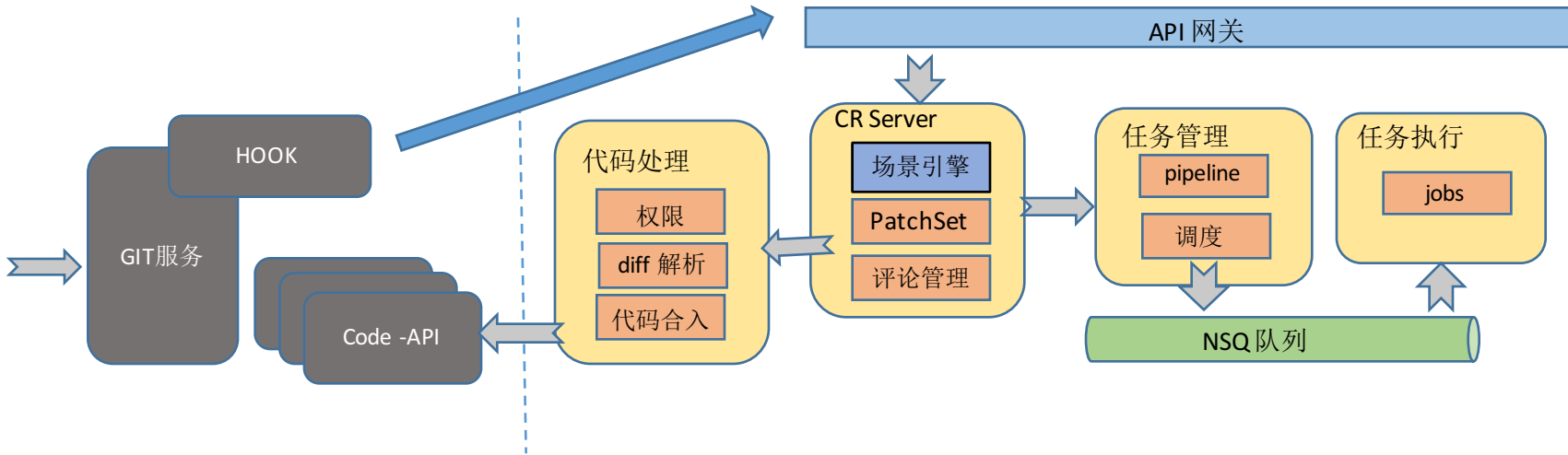
代码审查

Pipeline

查找代码

产生代码

提供能力

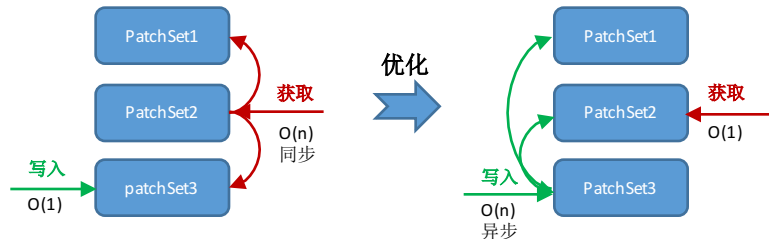


大Diff 解决思路：

- 分而治之：文件粒度拆分
- 计算后置：后端渲染
- 局部性原理：Redis缓存 (TTL 设定)

代码行评论：

对齐计算前置



代码应用层 —— 代码审查

- 上线一年半以来
 - 自动merge了 5亿+ 行代码
 - 总共发起了 20W+ CR
 - 共拦截了 ~1KW 行代码
 - 机器自动拦截 ~300w 行代码
 - 检测问题数 ~200w 个issue

代码应用层 —— Pipeline

代码搜索

CloudIDE

代码扫描

代码审查

Pipeline

查找代码

产生代码

提供能力

需求 -> 开发 -> 构建 -> 部署

- 多场景支持：
 - 第一代：schema 配置 ， 连通性弱 (SCMPF系统)
 - 第二代：pipeline as code ， 连通性强 (OE)

目录

1 滴滴代码工具链的演进

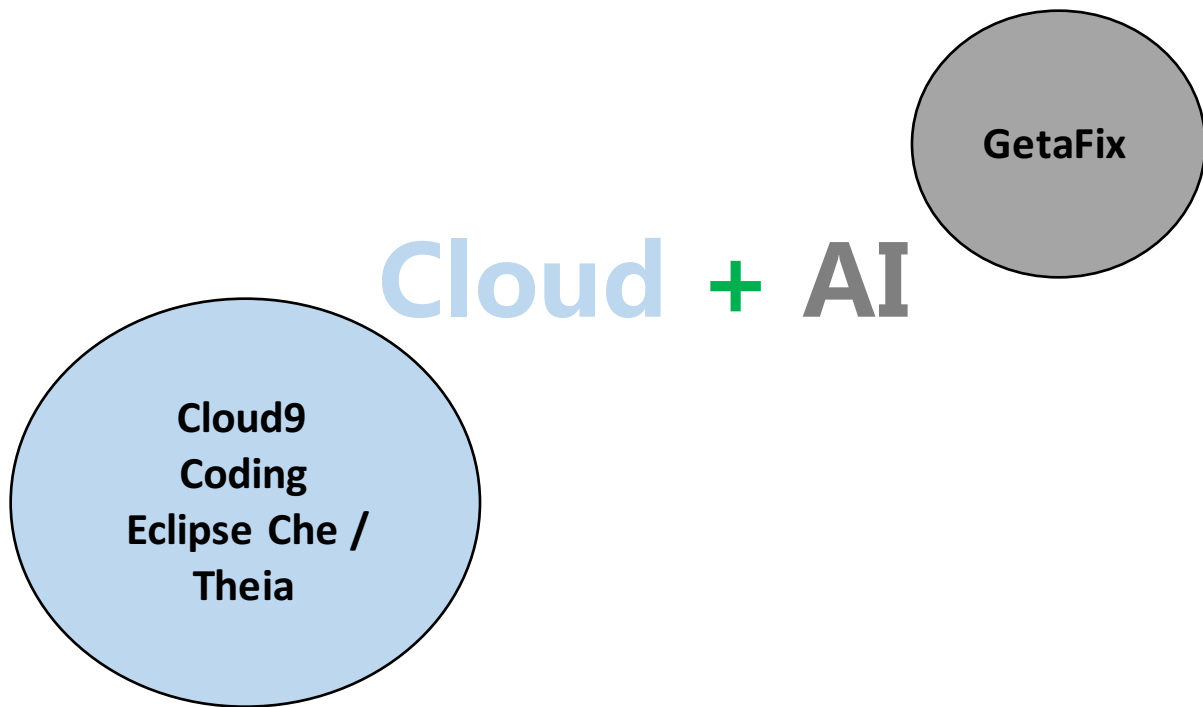
2 演进过程中的挑战——代码基础层

3 演进过程中的挑战——代码引擎层

4 演进过程中的挑战——代码应用层

 **5** 未来展望

未来展望





Thanks

高效运维社区
开放运维联盟

荣誉出品

想第一时间看到高效运维社区的
最新动态吗？

