

智能时代的运维

**CNUTCon**

全球运维技术大会

# 小米研发效能演进之路

王福

小米通用架构与工具团队-架构师

主办方 Geekbang > InfoQ  
极客邦科技

# 关于

关于我  
小米人工智能与云平台  
云技术-通用架构与工具团队  
负责分布式链路追踪、研发效能平台

讨论联系  
公众号：(微信搜索)小米云技术

# TABLE OF CONTENTS 大纲

---

- 背景介绍
- 开源工具采用
- 商业采购经验
- 自研研发效能平台

# TABLE OF CONTENTS 大纲

---

- 背景介绍
- 开源工具采用
- 商业采购经验
- 自研研发效能平台

# 背景介绍

## 业务模块

硬件IOT：手机、生态链、小爱智能音响

新零售：小米网、有品

互联网：广告、金融、MiCloud

## 技术栈

通用技术栈：Java

部分电商业务：Golang

AI业务：C&C++

# 背景介绍

- 小米研发效能领域的三个阶段



开源工具使用

商业采购

自研研发效能  
平台

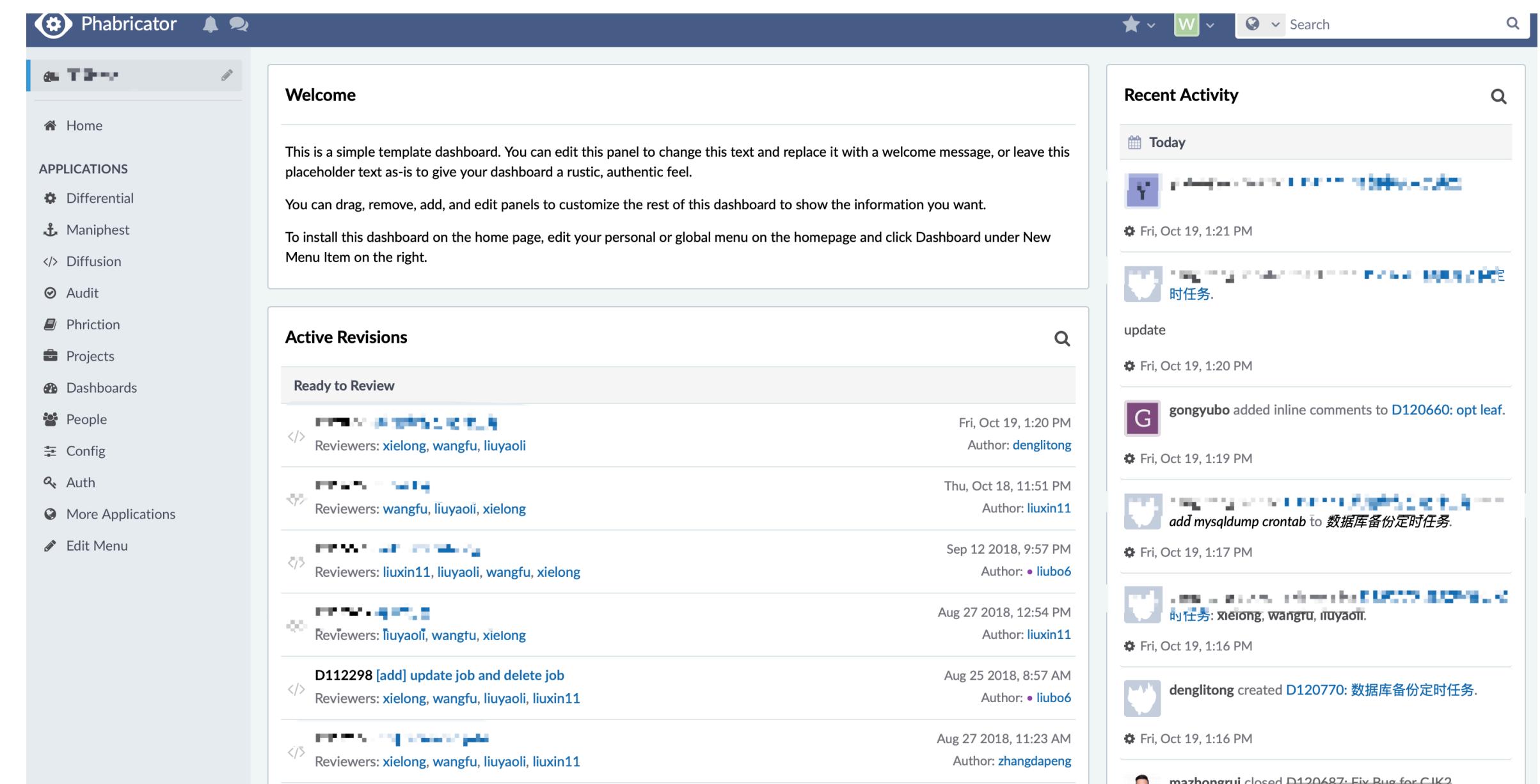
# TABLE OF CONTENTS 大纲

---

- 背景介绍
- 开源工具采用
- 商业采购经验
- 自研研发效能平台

# 开源工具采用

- 代码质量工具
    - Phabricator
      - Facebook出品,小米早期员工引入
      - 更人性化的代码Diff功能
      - 代码检查集成：Sonar、JsLint等



# 开源工具采用

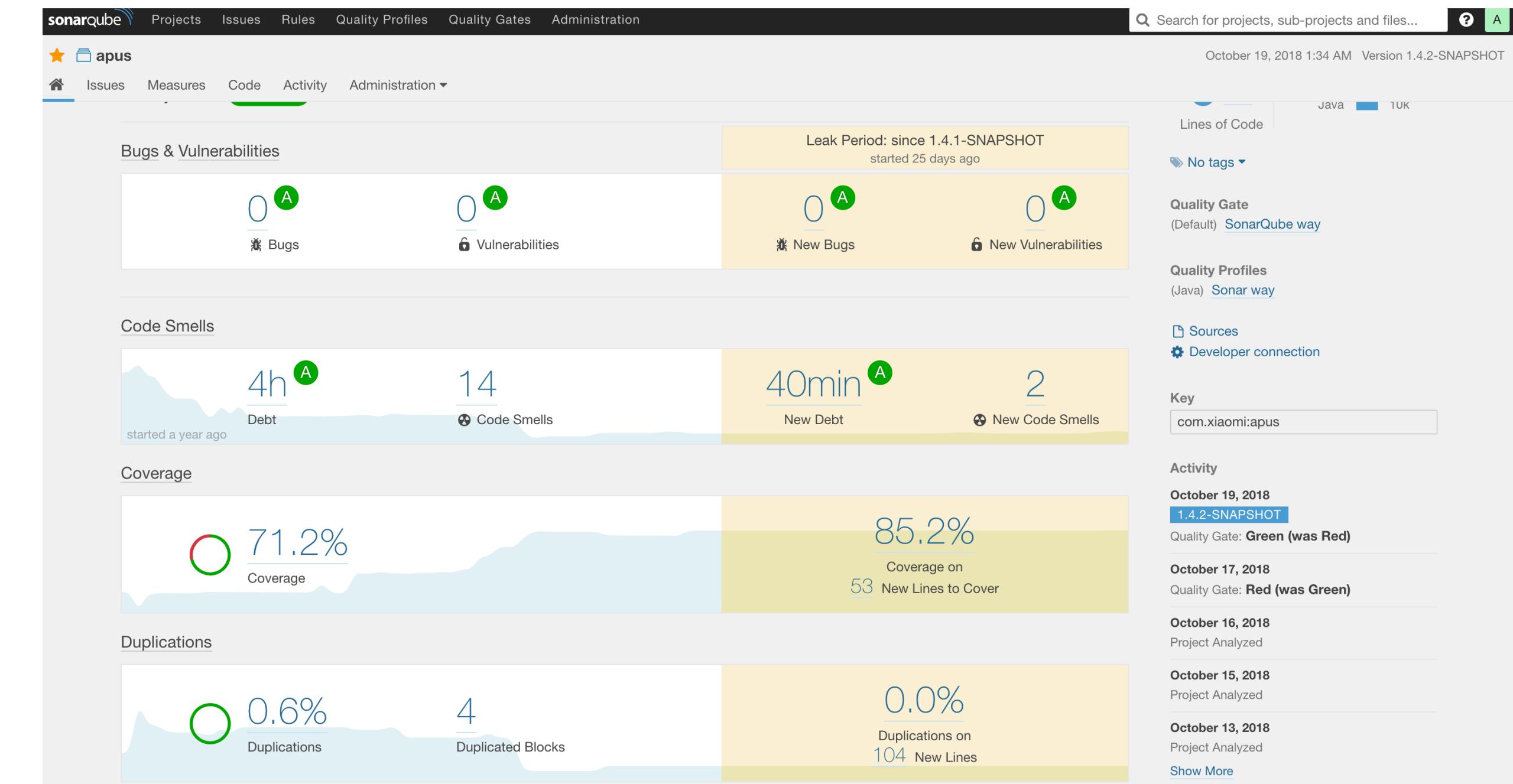
- 代码质量工具

- SonarQube

- 开源代码质量管理平台

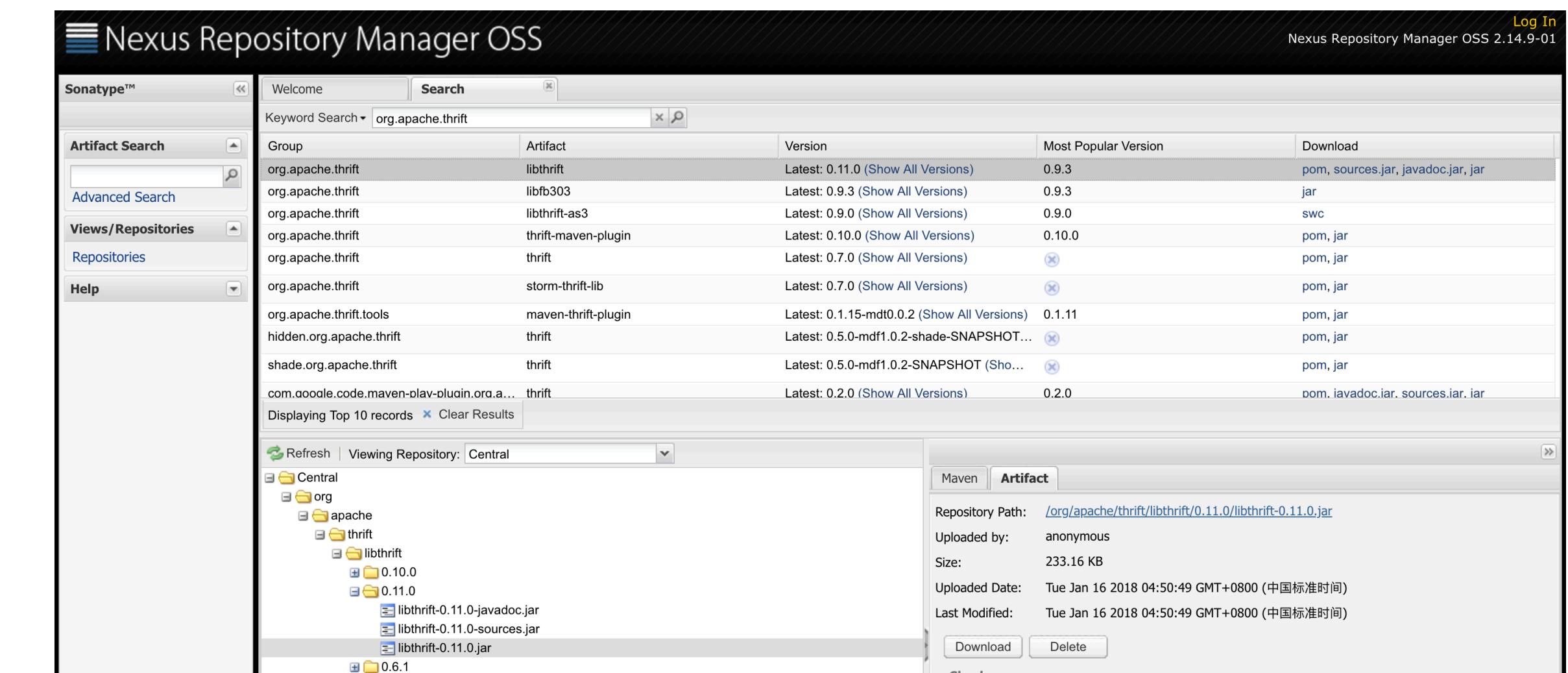
- 支持Java、C\C++、C\#等语言

- SonarLint\Mvn+Jacoco+Jenkins



# 开源工具采用

- 二进制存储与管理
  - Nexus
  - 优秀的Maven仓库管理器
  - 强大的仓库管理与搜索功能



# 开源工具采用

- 持续集成持续交付引擎

- Jenkins

- 扫描、发布类需求

- 监控、测试类需求

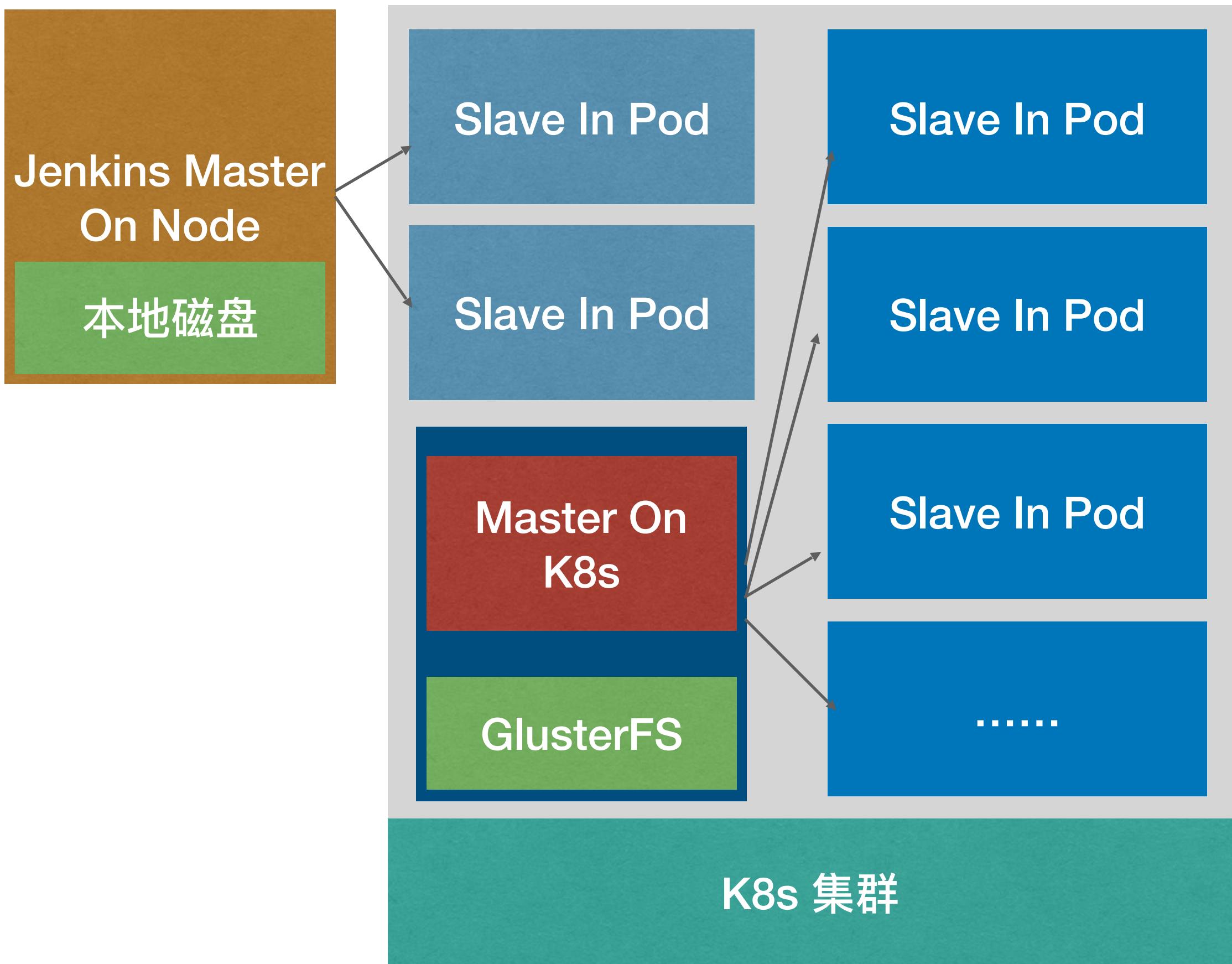
The screenshot shows the Jenkins dashboard with the following details:

- Top Bar:** Jenkins logo, search bar, user info (WANGFU), and a link to 'WANGFU | 注销'.
- Left Sidebar:** Navigation links including '新建', '用户', '任务历史', '项目关系', '检查文件指纹', '系统管理', 'My Views', 'Credentials', and 'Dependency Graph'.
- Middle Section:**
  - A banner for '智能云自动化测试平台' with links to '各个机房jenkins地址' and '使用文档'.
  - A grid of Jenkins jobs categorized by color: MIUI\_INTERFACE\_AUTOTEST (green), MiPhone\_coverity (blue), XMSS (orange), and others like computing-sql-services, emq-ut, etc. (grey).
  - A '构建队列 (8)' section listing jobs: hbase\_bvt\_emq\_staging, messaging\_mimc\_uktopic\_c3\_test, messaging\_resolver\_foreign\_test, messaging\_resolver\_staging\_test, messaging\_micloud\_c3\_test, messaging\_micloud\_staging\_test, messaging\_mixin\_v6\_staging\_test, and messaging\_rts\_relay\_c3\_test.
  - A '构建执行状态' section showing the status of various builds.
- Right Section:** A table of build jobs with columns: S (Status), W (Icon), Name, 上次成功 (Last Success), 上次失败 (Last Failure), 上次持续时间 (Last Duration), and Fav (Favor).

S	W	Name	上次成功	上次失败	上次持续时间	Fav
✓	cloud	ad-delivery-service	1月 27 天 - #39	1月 27 天 - #37	1分 46 秒	☆
✗	rain	ad-index-deploy-old	没有	2月 13 天 - #6	0.78 秒	★
✗	sun	ad-infra	没有	无	无	☆
✓	star	aggregator-framework	2天 21 小时 - #62	无	1分 43 秒	☆
✗	cloud	AI-Bokeh	没有	无	无	☆
!	cloud	aivs-client	4月 13 天 - #76	1月 27 天 - #78	2分 38 秒	☆
!	cloud	aivs-client-dev	2月 10 天 - #31	21 天 - #32	6分 56 秒	☆
✓	star	aivs-client-master	1天 0 小时 - #66	21 天 - #60	4分 51 秒	☆

# 开源工具采用

- 持续集成持续交付引擎
  - Jenkins on K8s
    - 资源：物理机做Slave资源利用率低
    - 隔离：基于Docker提供更好的隔离性



# TABLE OF CONTENTS 大纲

---

- 背景介绍
- 开源工具采用
- 商业采购经验
- 自研研发效能平台

# 商业采购经验

基于开源的方案遇到困难  
性能瓶颈

业务需求日益多元化但团队规模有限

扩展更多的能力  
更多元化的代码质量检查  
审计需求  
二进制、镜像等安全扫描

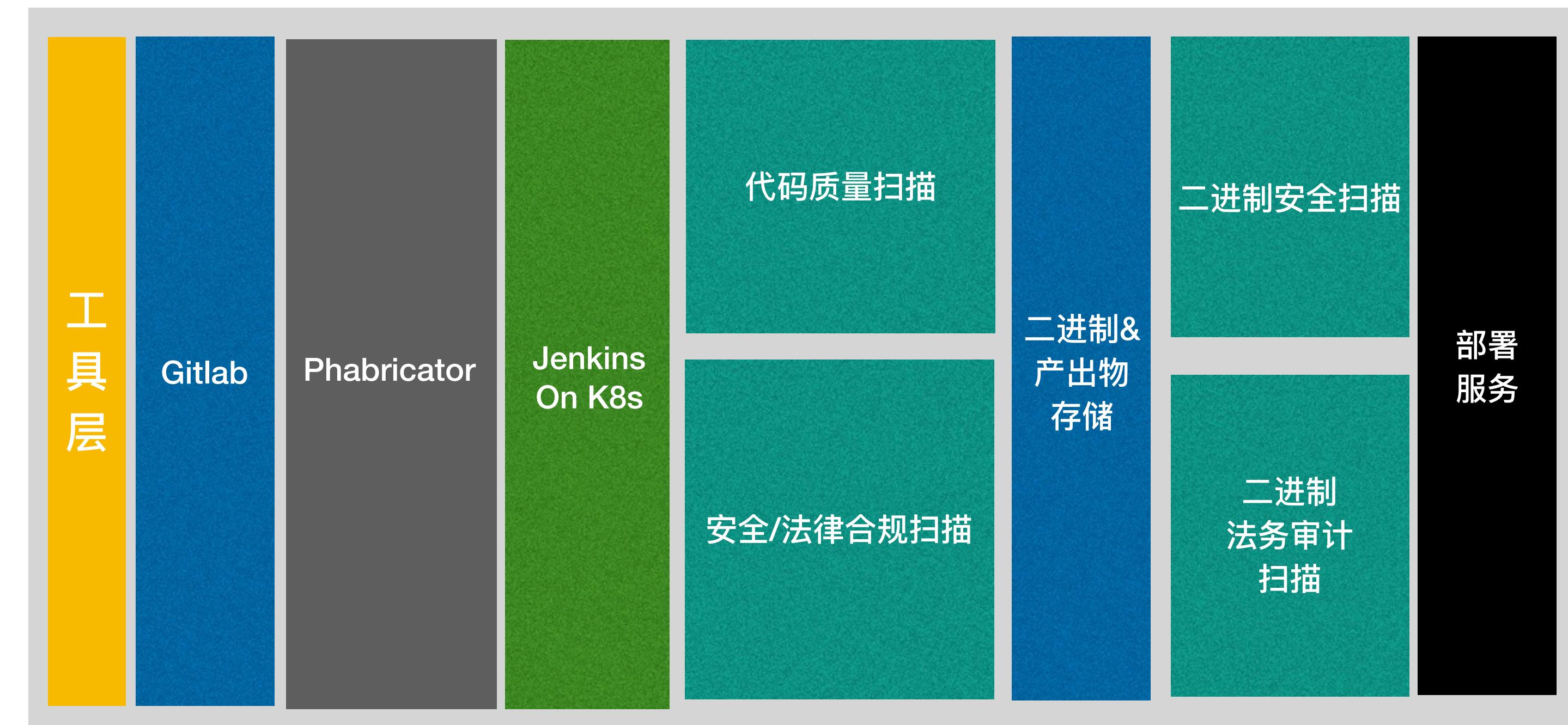
# 商业采购经验

分类	收益	风险
自研	平台功能产出 技术积累	人员变动 技术攻关失败 产品可用性差
采购	更快价值交付与落地 更快获取业界一流经验	数据泄露 供应商不可控因素无法提供服务等

# 商业采购经验

- 经验总结

- 货比三家，一定要多试用比较
- 售后服务与咨询很重要
- 采购可整合性更好的产品
- 尽量离线部署以防止数据泄露



# TABLE OF CONTENTS 大纲

---

- 背景介绍
- 开源工具采用
- 商业采购经验
- 自研研发效能平台

# 自研研发效能平台

理清组织结构、  
项目结构

降低接入成本

标准化打通各个系统  
整合工具，赋能研发

符合某特征的必须审计  
的项目有哪些?  
没人能清楚

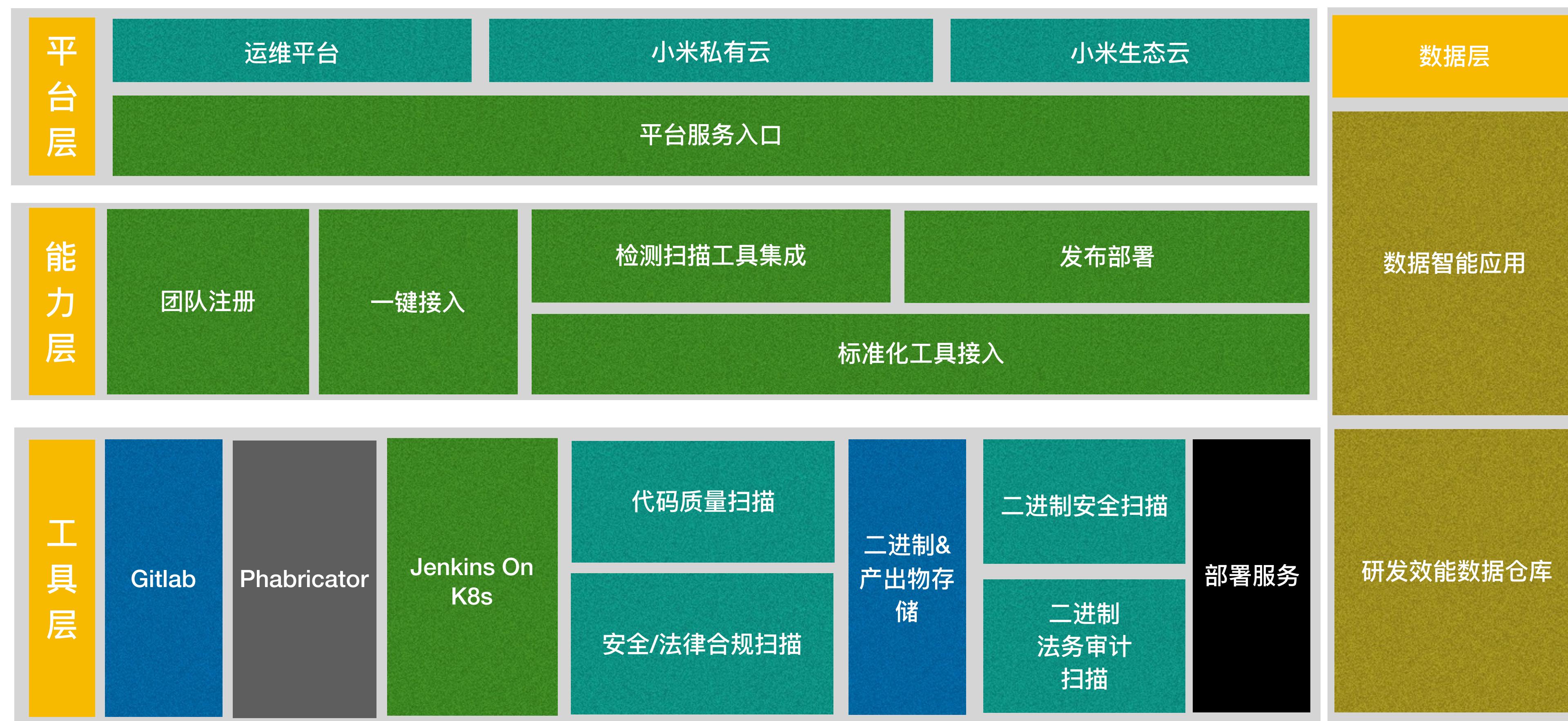
N个平台M个场景J个团  
队，经常做重复接入、  
遇到同样的问题

如何打通部署？如何集  
成诸如压测等其他系统

自研的目标，是标准化整合各工具，而非自研每一个独立工具

# 自研研发效能平台

计划中



# 自研研发效能平台

- 核心实现：Jenkins无状态优化
  - Job数据无状态化
    - Job的参数配置、Jenkinsfile等均由平台管理
    - 每次build时依次选集群、新建或更新Job、执行
    - 执行结果通过插件同步导入ES
  - 集群信息无状态化



# 自研研发效能平台

姓名: wangfu

所属部门: 人工智能...

所属二级部门: 云技术

所属三级部门: 通用架构与工具团队

管理员权限:

git addr:   
请输入要创建pipeline项目的地址,eg: git@git.n.xiaomi.com:cloud-common/xtrace.git

git branch: master

项目类型: web应用

语言和打包: Maven

输出地区: Global

项目名称	类型	语言与打包	简况	
submit_test_web	web应用	Maven	4条	<a href="#">Pipeline列表</a> <a href="#">检测更新pipeline</a>
test-cpp-release	基础库中间件	CPP	1条	<a href="#">Pipeline列表</a> <a href="#">检测更新pipeline</a>
artifactory-tes	基础库中间件	CPP	1条	<a href="#">Pipeline列表</a> <a href="#">检测更新pipeline</a>
cpp-artifactory	基础库中间件	CPP	1条	<a href="#">Pipeline列表</a> <a href="#">检测更新pipeline</a>
zander-apm-agen	web应用	CPP	1条	<a href="#">Pipeline列表</a> <a href="#">检测更新pipeline</a>
build-permission-manager-hostnetwork	web应用	Maven	4条	<a href="#">Pipeline列表</a> <a href="#">检测更新pipeline</a>

submit_test_web	mvnCodeAnalysis	root--->git--->sonar--->coverity	<a href="#">执行列表</a> <a href="#">配置执行</a>
submit_test_web	mvnUnitTest	root--->git--->sonar	<a href="#">执行列表</a> <a href="#">配置执行</a>
submit_test_web	mvnDeployOcean	root--->git--->sonar--->ocean	<a href="#">执行列表</a> <a href="#">配置执行</a>
submit_test_web	mvnDeployXbox	root--->git--->sonar--->xbox	<a href="#">执行列表</a> <a href="#">配置执行</a>

**CNUTCon**  
全球运维技术大会

主办方  
**Geekbang InfoQ**  
极客邦科技

# 谢谢

欢迎关注

公众号：(微信搜索)小米云技术

# 附录

- 核心业务模型抽象
  - Pipeline={Stage1,Stage2...}
  - Stage=[input]{Action1,Action2...}[output]
    - Action=cmd+params
    - input:Code\\*.jar
    - output: \*.jar\镜像\扫描数据

```
node('maven-jdk1.8-bazel') {
    sh 'echo 1 | alternatives --config java'
    sh 'java -version'
    try {
        stage('Preparation') {
            sh 'git pull'
        }
        stage('Build&Test - unit') {
            sh 'mvn test'
        }
        stage('Build&Test - integration') {
            sh 'mvn verify'
        }
        stage('Build Binaries') {
            sh 'bazel build --copt=-Wall'
        }
        stage('Build&Test - rpm') {
            sh 'bazel build //:all'
        }
        stage('Build&Test - deb') {
            sh 'bazel build //:all'
        }
    }
    stages:
        - test
        - build
        - release
    cache:
        paths:
            - .gopath
    before_script:
        - echo "Git Branch is ${GIT_BRANCH}"
    testcase:
        stage: test
        script:
            - echo "begin run test"
            - make test
    build_bin:
        stage: build
        script:
            - make
    release_bin:
        stage: release
        script:
            - echo "begin run release rpm pkg"
            - make rpm
            - make deb
}
url: 'git@github.com:geekbanginfo/cnutchain.git'
:all'
:all'
```

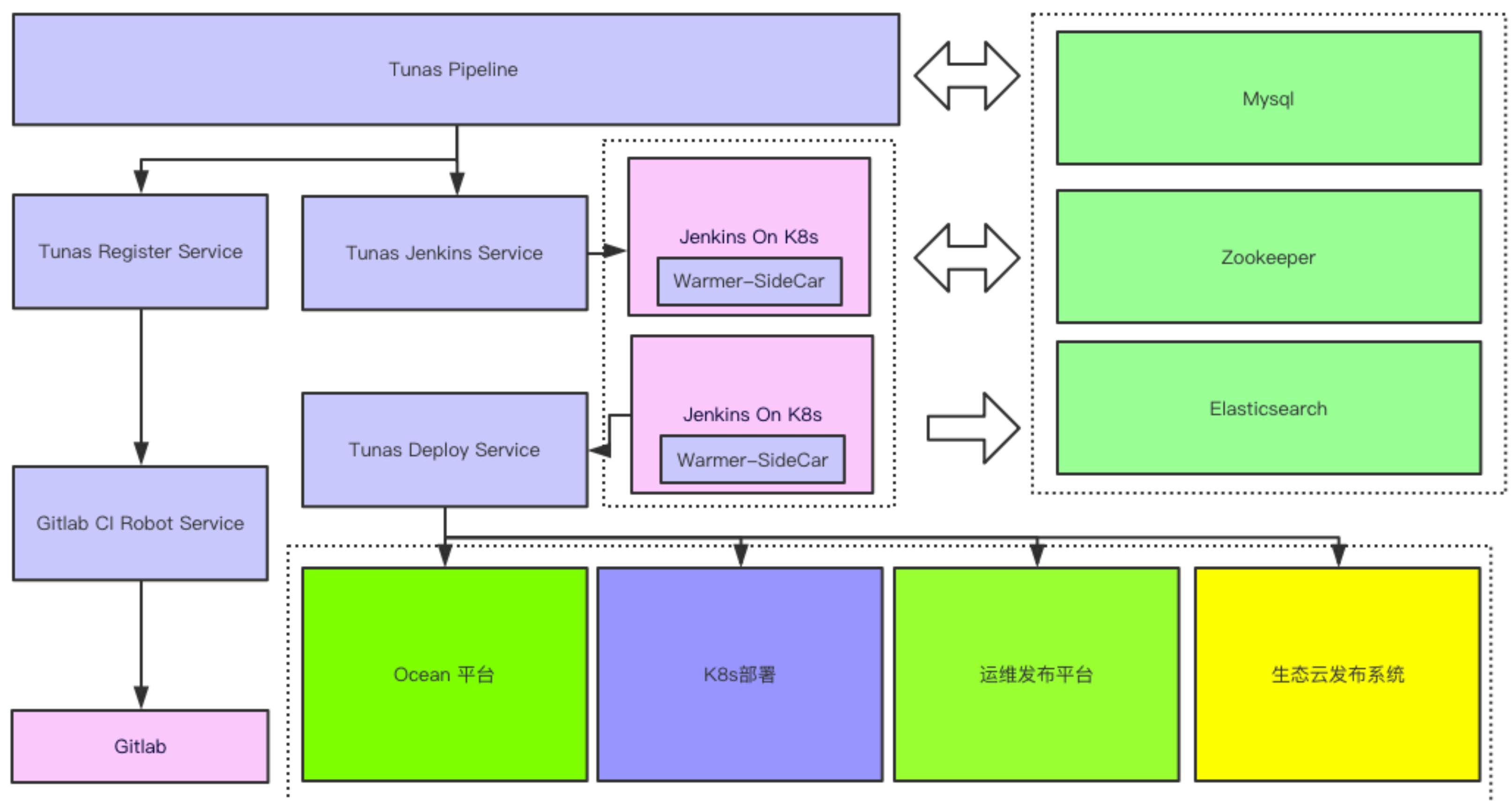
# 附录

- 核心业务模型抽象
  - Pipeline模板：通用Pipeline的抽象
    - 管理若干Stage模板及先后顺序
    - 一个模板对应一类场景：如mvn类代码质量扫描
    - 平台运营提供公司级Pipeline模板
    - 业务经理\架构师制定团队级模板
  - Stage模板：最细粒度执行单元
    - 一个Stage对应一个平台或一组操作（如git pull）
    - Stage中的动态数据抽象为动态参数
    - 提供pre/after condition judge or hook

项目名称	名称	流程概述	执行列表 配置执行
test-cpp-release	Cpp-Commons-Lib-Release	root--->git--->CppBuild--->Deploy2Artifactor	
git_commit:	please input git_commit		
release_dir:	install		
git_branch:	master		
artifactory_namespace:	public		
pipeline_email:	wangfu		
git_project_name:	test-cpp-release		
is_snapshot:	snapshot		
pipeline_node_name:	mvn-jdk1.8		
git_url:	git@v9.git.n.xiaomi.com:cicd-user-test/test-cpp-release.git		
version:	0.0.2.1-snap		
触发类型:	<input type="radio"/> 手动触发 <input type="radio"/> git触发 <input type="radio"/> 定时执行		
<button>取消</button> <button>保存执行</button>			

# 附录

- 核心实现：架构图



# 附录

业务基础  
贴业务  
做标准

推广落地  
出数据  
拜真佛



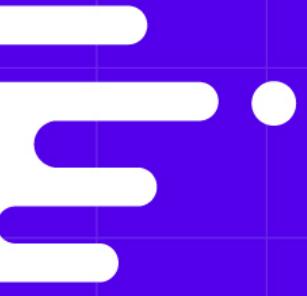
2018.12.20-23 / 北京·国际会议中心

# AI商业化下的技术演进实战干货分享

京东：智能金融



景驰科技：自动驾驶



阿里巴巴：NLP



清华大学人工智能研究院：机器学习



今日头条：机器学习



Twitter：搜索推荐



AWS：计算机视觉



Netflix：机器学习



扫码了解详情

在不同业务场景下架构师如何升级？

听100+位讲师的优质  
架构实践分享

来自Google、Netflix、LinkedIn、BAT等...

2018年12月7-8日 北京·国际会议中心



扫码了解大会详情



全球软件开发大会

北京·2019

更多技术干货分享，北京站精彩继续

提前参与，还能享受更多优惠

识别二维码  
查看了解更多

2019.qconbeijing.com





# 深入剖析 Kubernetes

Kubernetes 原来可以如此简单



扫码了解更多



张磊

Kubernetes 社区

资深成员与项目维护者