

# Project Harbor Introduction

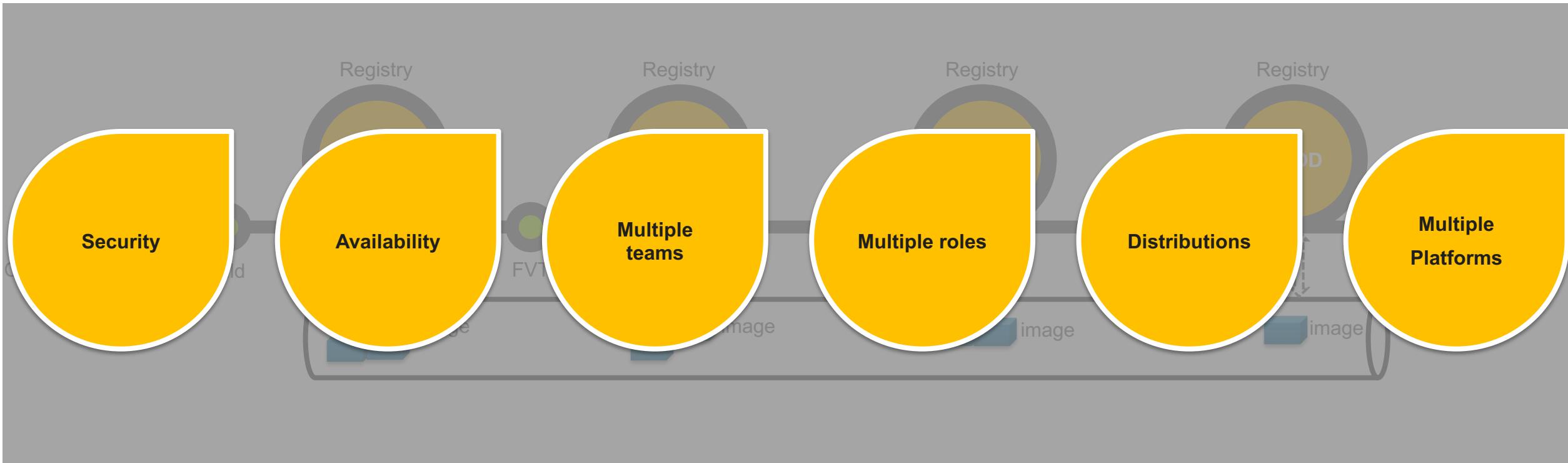
Open source trusted cloud native registry

Henry Zhang, Chief Architect, VMware R&D China

Steven Zou, Staff Engineer, VMware R&D China

Nov. 2018

# Image Management through Pipeline





## 开源的可信云原生仓库项目

[goharbor.io](https://goharbor.io)



由 VMware 中国  
研发团队发起,  
社区共同维护



集成到多个企业级  
产品中:VIC和PKS

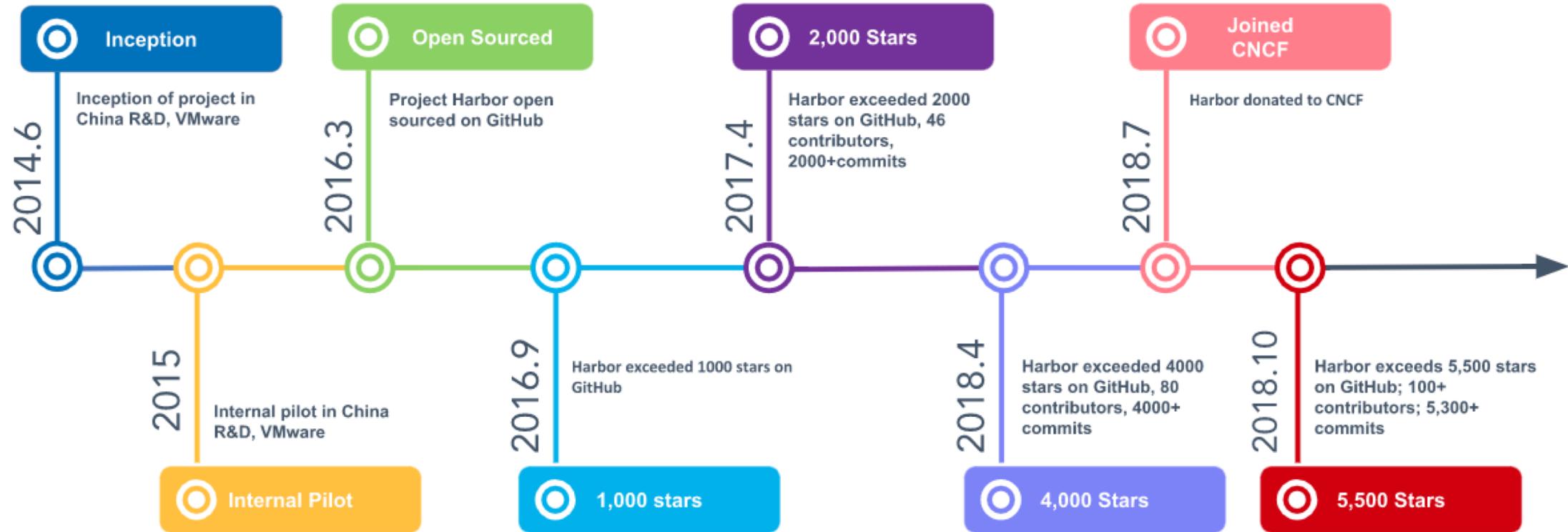


Apache 2.0  
使用许可



GitHub Repo:  
[https://github.com/goharbor/  
harbor/](https://github.com/goharbor/harbor/)

# Harbor项目简史



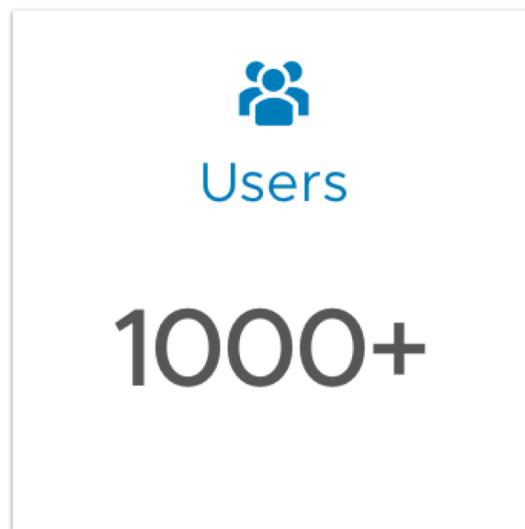
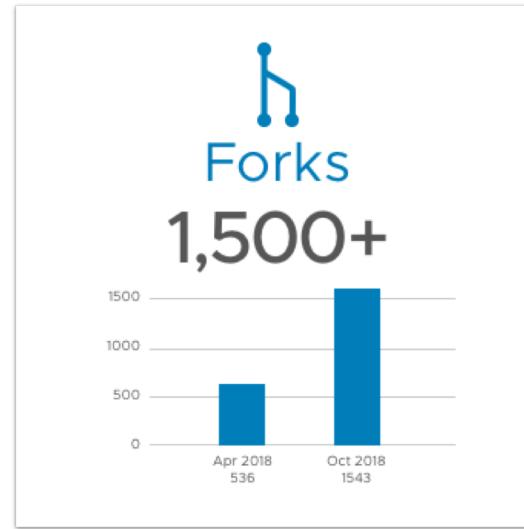
未来更多集成方向



OPEN SERVICE BROKER API™



# Harbor社区概况



# Harbor部分用户



# Agenda

## OVERVIEW

### SECURITY

- Isolation
- Access Control
- Vulnerability
- Content Trust

### DISTRIBUTION

- Replication
- Control Policy

### RELIABILITY

- HA Supporting

### DEPLOYMENT

- Helm Chart Repo
- Deployments

# OVERVIEW

Main  
Features

Architecture



# 主要特性



## GUI支持

基于开源Clarity构建  
完备镜像运管能力批处理操作支持



## Restful API

完善的API支持集成  
Swagger API 文档



## 远程复制

多种过滤器支持  
定时,即时和手动触发



## 访问控制

基于角色的访问控制  
AD/LDAP 用户集成



## 审计日志

操作日志记录以审计



## 分发控制

基于内容信任  
基于漏洞扫描  
基于RBAC

# 主要特性（续）



## 漏洞扫描

多种漏洞扫描策略  
详尽的漏洞扫描报告



## 内容信任

数字签名镜像



## HA高可用

高可用性支持

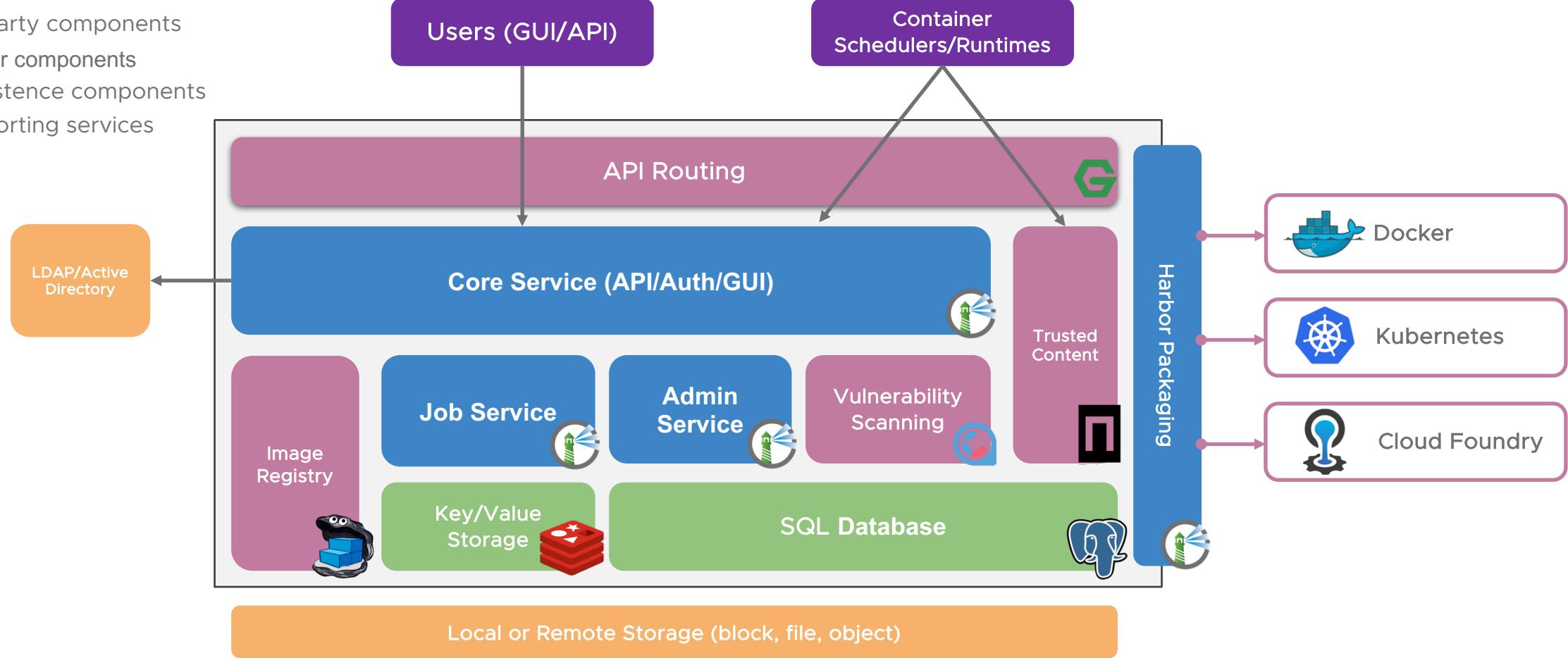


## Helm Chart支持

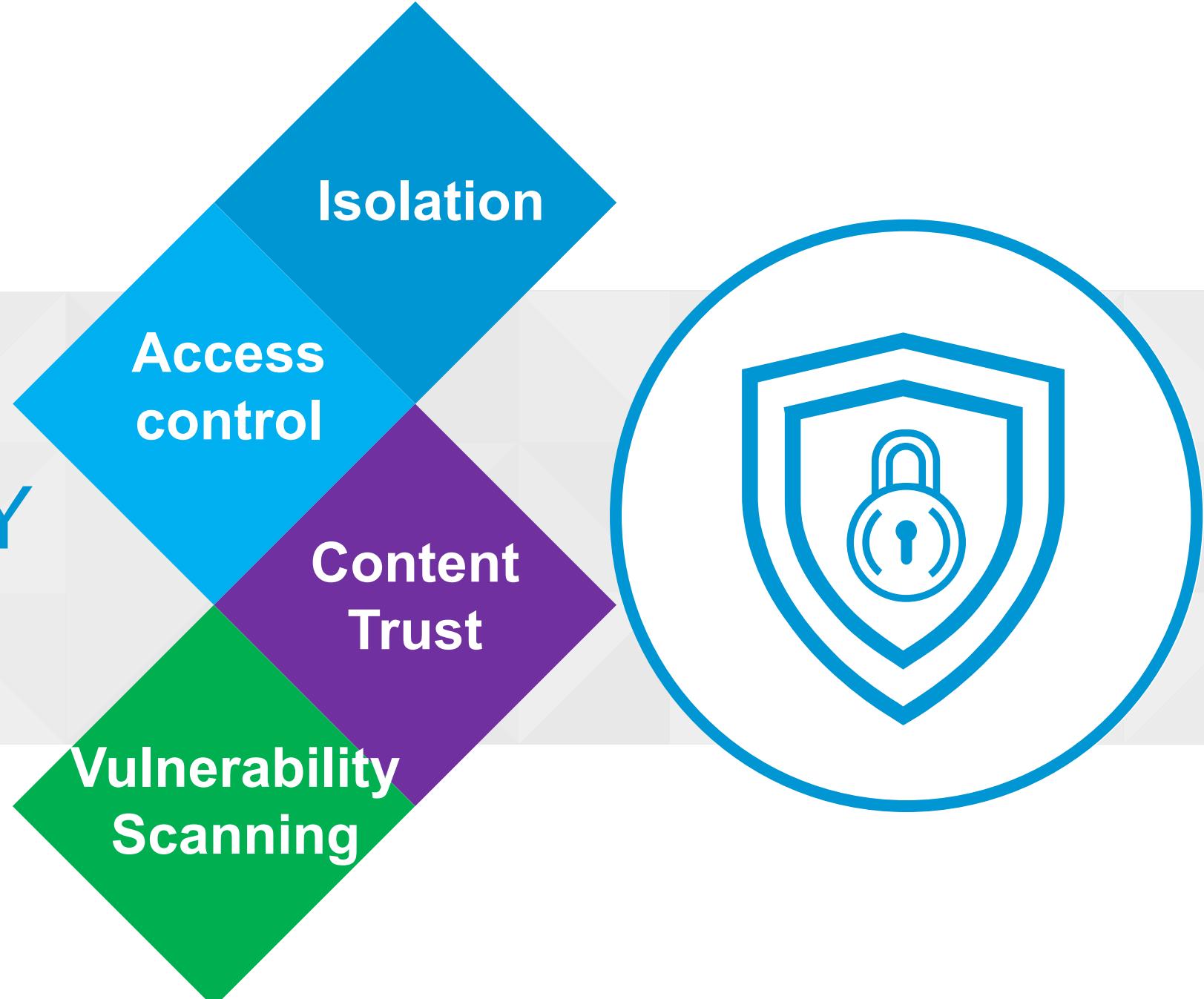
Helm Chart仓库  
与镜像相同的管理机制和体验

# Harbor 架构

- Consumers
- 3rd party components
- Harbor components
- Persistence components
- Supporting services



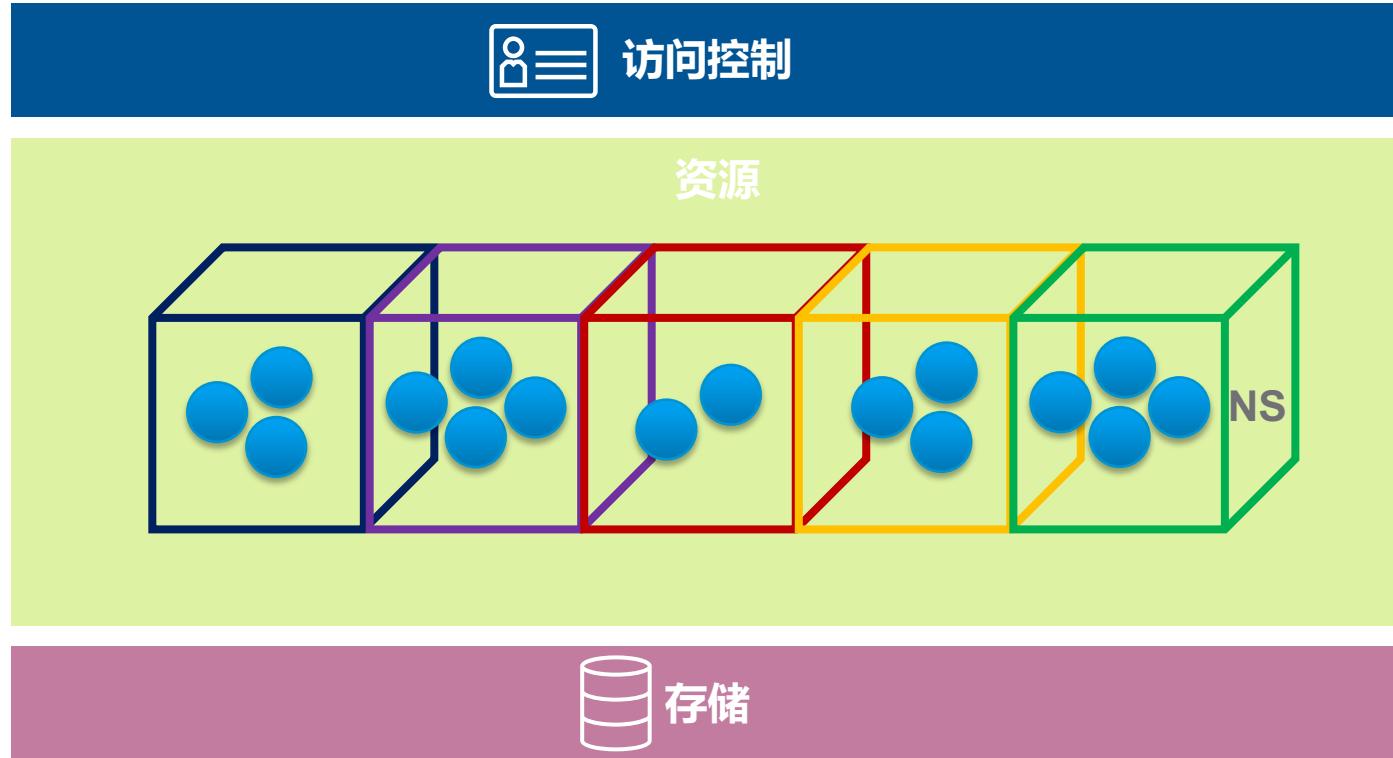
# SECURITY



# 资源隔离



## 资源隔离

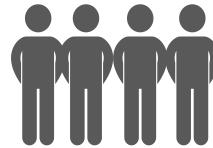


- 基于项目提供独立的NS
- 逻辑隔离，存储共享
- 访问控制的基础
- 为多租户提供可能

# 访问控制

成员  
Members

访客 ( Guest ) :



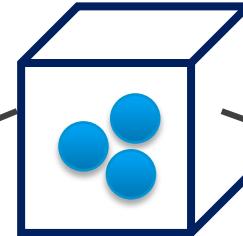
开发者 ( Developer ) :



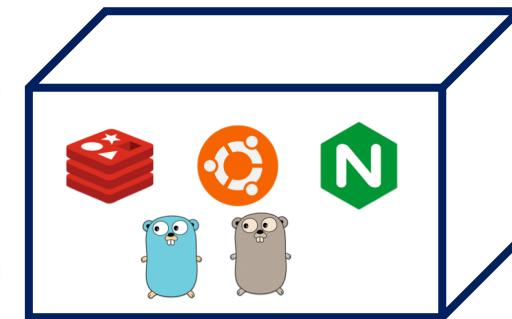
管理员 ( Admin ) :



项目



镜像  
Images



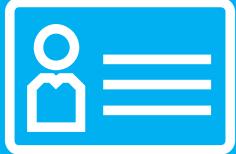
docker pull ...

docker pull/push ...

operation & management



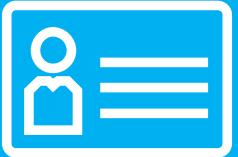
Settings



## 访问控制

- 企业用户通常把镜像存放在组织内部
- 不同角色人员应有不同的访问权限
- 不同环境人员的角色不同
- 与已有的LDAP/AD用户系统集成

# 访问控制



## 访问控制

< 项目

### library

镜像仓库 成员 日志 复制 配置管理

+ 新建成员 设置角色 × 移除成员

姓名	角色
<input type="checkbox"/> admin@harbor.local	开发人员
<input type="checkbox"/> wangyan01	访客

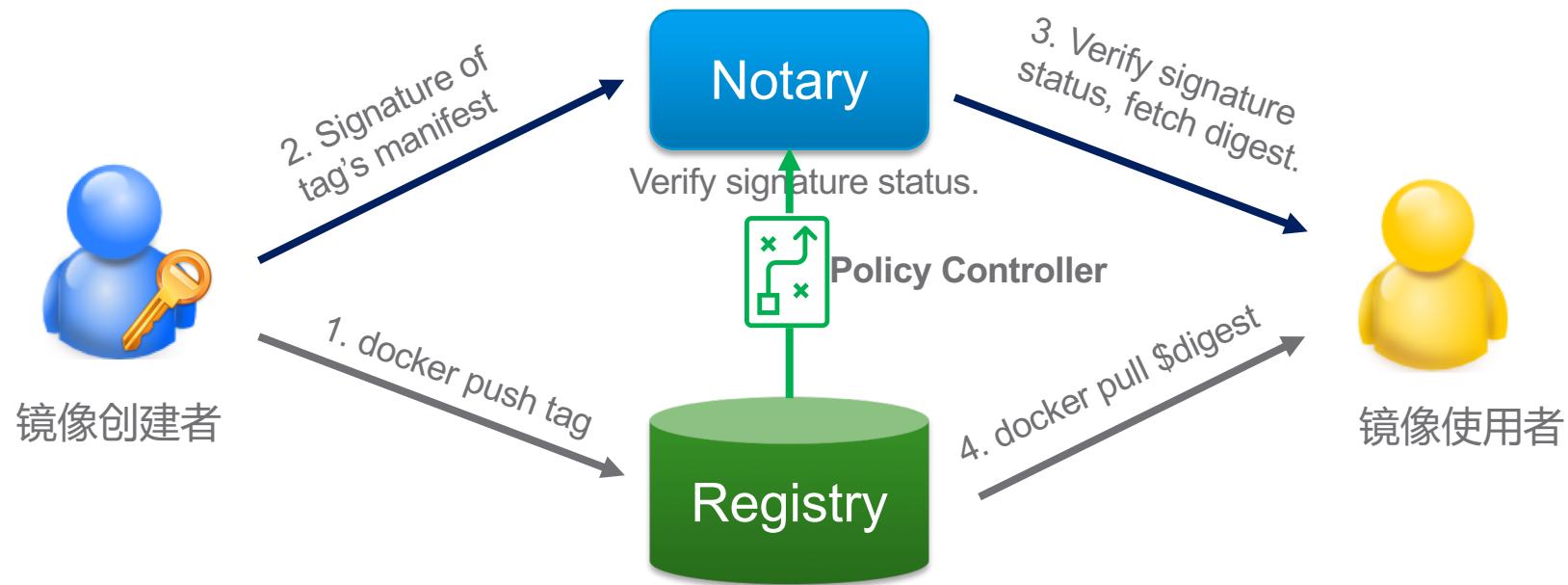
0 条记录

- 企业用户通常把镜像存放在组织内部
- 不同角色人员应有不同的访问权限
- 不同环境人员的角色不同
- 与已有的LDAP/AD用户系统集成

# 内容信任



## 内容信任



- 发布者对镜像签名
- 下载镜像时使用签名摘要 ( Digest )

# 内容信任



内容信任

描述信息 镜像

扫描 复制摘要 删除

Q | C

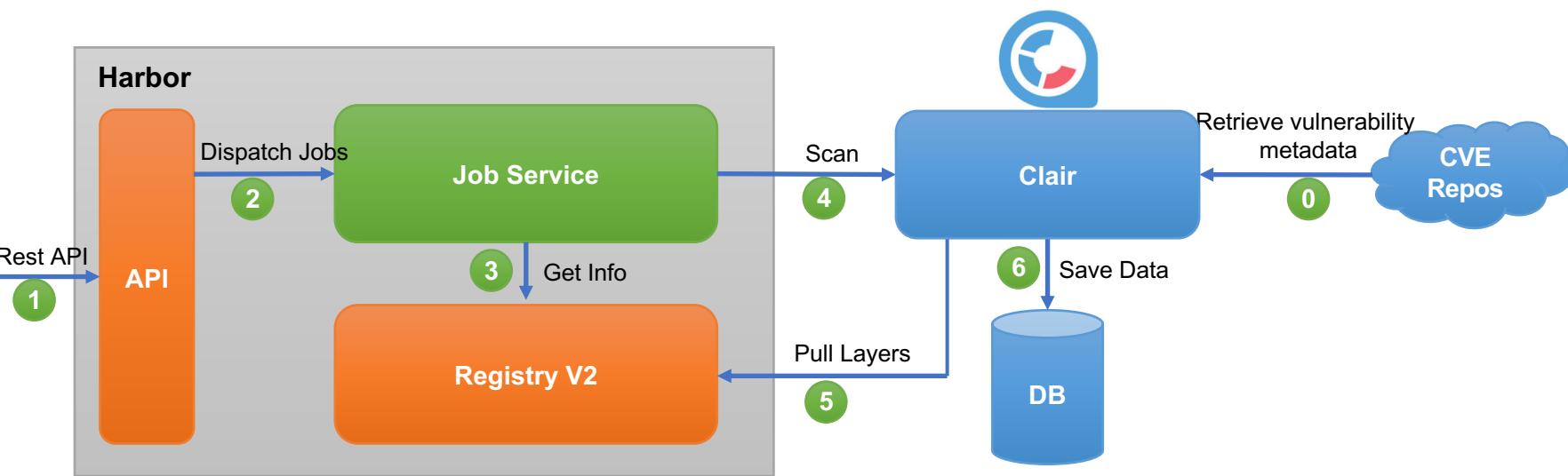
标签	大小	Pull命令	漏洞	已签名	作者	创建时间
v2	2.95KB					2017/11/21 上午8:23
latest	2.95KB					2017/11/21 上午8:23
v1	2.95KB					2017/11/21 上午8:23

- **发布者对镜像签名**
- **下载镜像时使用签名摘要 ( Digest )**

# 漏洞扫描



## 漏洞扫描



- 对镜像文件做静态分析
- 多种扫描模式支持
  - 定时
  - 事件
  - 手动
- 漏洞数据库定期更新
- 多重漏洞数据来源
  - Debian Security Bug Tracker
  - Ubuntu CVE Tracker
  - Red Hat Security Data
  - Oracle Linux Security Data
  - Alpine SecDB



## 漏洞扫描

project061529900025/tomcat:latest

Author: anonymity  
Architecture: amd64  
OS: linux  
Docker Version: 17.06.2-ce  
Scan Completed: Jun 25, 2018

Vulnerability Level Summary:

- 7 high Level Vulnerabilities
- 11 medium Level Vulnerabilities
- 13 low Level Vulnerabilities
- 2 unknown Level Vulnerabilities

Vulnerability	Severity	Package	Current version	Fixed in version
CVE-2017-17942	medium	tiff	4.0.8-2+deb9u2	
Description: In LibTIFF 4.0.9, there is a heap-based buffer over-read in the function PackBitsEncode in tif_packbits.c.				
CVE-2018-5784	medium	tiff	4.0.8-2+deb9u2	
CVE-2018-8905	medium	tiff	4.0.8-2+deb9u2	
CVE-2017-17973	negligible	tiff	4.0.8-2+deb9u2	

- 对镜像文件做静态分析
- 多种扫描模式支持
  - 定时
  - 事件
  - 手动
- 漏洞数据库定期更新
- 多重漏洞数据来源
  - Debian Security Bug Tracker
  - Ubuntu CVE Tracker
  - Red Hat Security Data
  - Oracle Linux Security Data
  - Alpine SecDB

# DISTRIBUTION

Replication

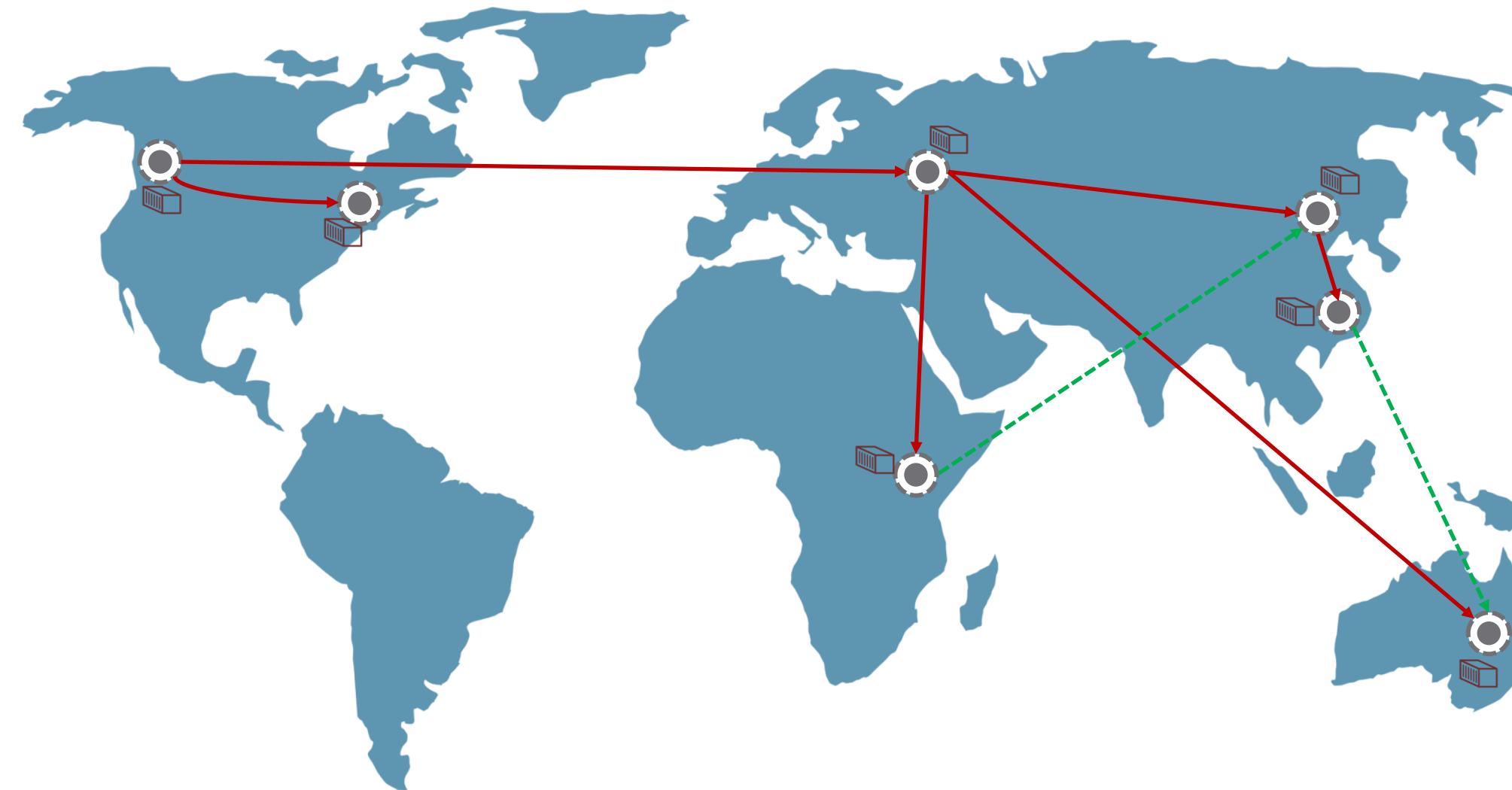
Policy





## 镜像分发

- 多点保持镜像一致
- 镜像备份与恢复
- 就近访问与下载
- 负载分担

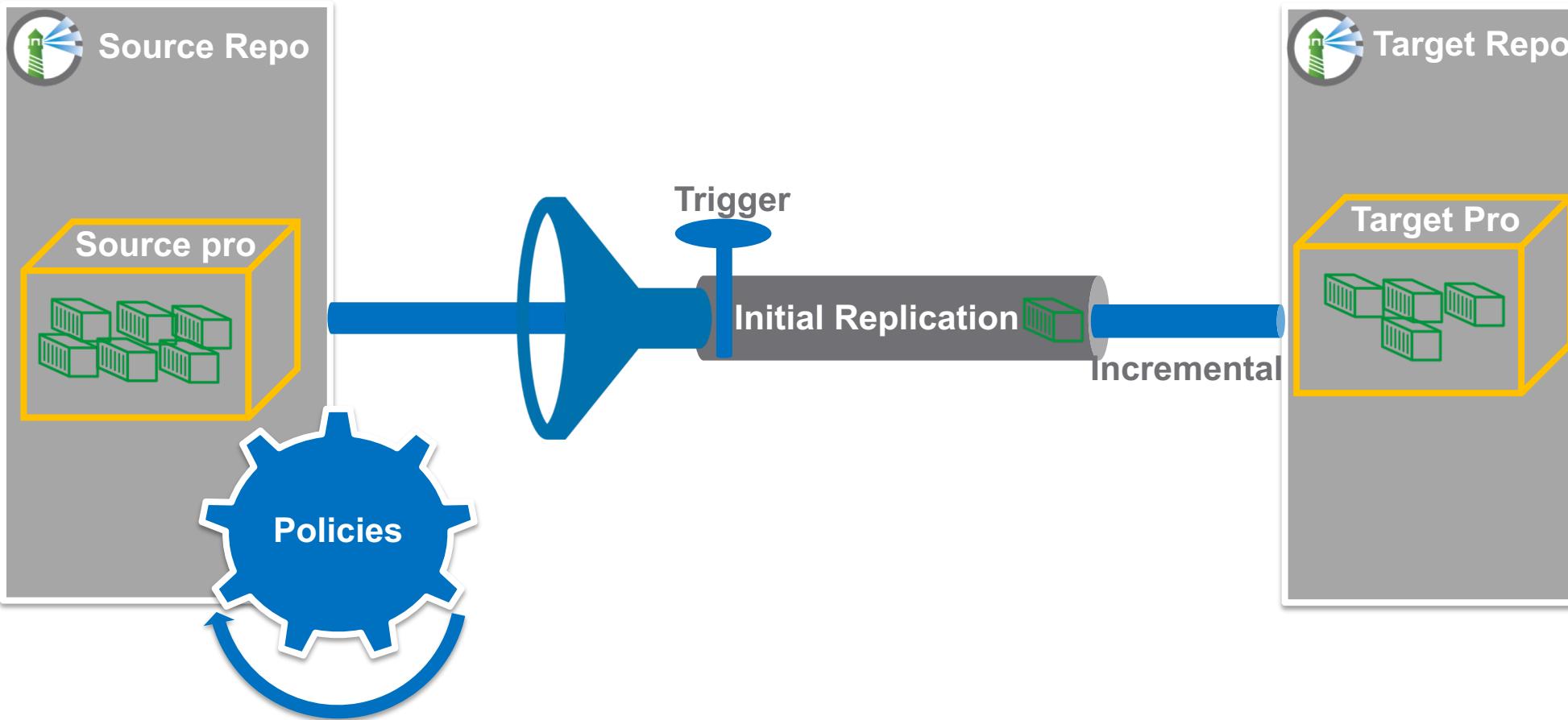


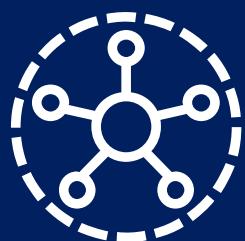


# Image Replication

镜像复制

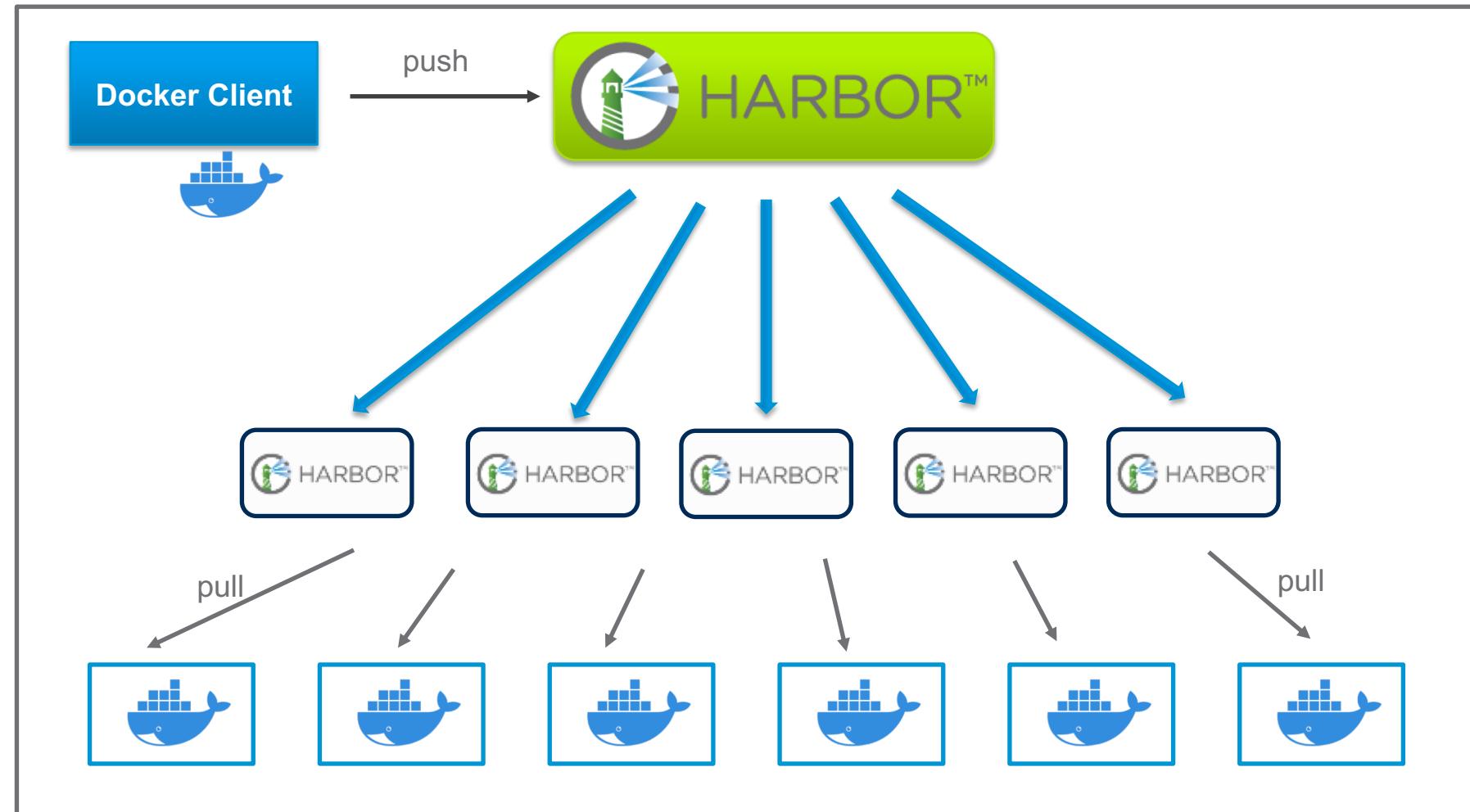
- 基于策略
- 面向项目
- 增量复制
- 支持过滤器
- 多种触发策略





## 负载分担

# 镜像分发-负载分担



- 容器镜像通常从registry分发
- 在大规模集群场景下，Registry 是镜像分发瓶颈
- 扩展 registry 服务
  - 多实例共享存储
  - 多实例不共享存储



## 复制管理

- 可在系统与项目级别管理复制策略
- 查看复制历史
- 查看复制日志

«

### 复制管理

项目 日志 系统管理 用户管理 仓库管理 复制管理 配置管理

+ 新建规则 修改 × 删除 复制

名称	项目	描述	目标名	触发模式
111111	54321	111	test111	Scheduled
123123	library	1233333333	test111	Manual
123123123	library	-	test111	Manual

1 - 3 共计 3 条记录

高级检索 搜索

### 复制任务

停止任务

名称	状态	操作	创建时间	更新时间	日志
library/hello-world	retrying	transfer	2018/1/29 下午6:01	2018/1/29 下午7:21	日志
library/hello-world	stopped	transfer	2018/1/29 下午4:53	2018/1/29 下午4:54	日志

1 - 2 共计 2 条记录



# 复制策略构建

< 复制管理

## 新建规则

名称\*

my rule

描述

Just a test case

源项目\*

repository  test\*



tag  1.\*

目标 \*

khans3: http://10.112.122.203



user Name:

password:

触发模式

手动

删除本地镜像时同时也删除远程的镜像。

立即复制现有的镜像。

保存

取消

## 复制管理

- 支持Repo过滤
- 支持Tag过滤
- 多种触发模式
  - 手动
  - 定期
  - 事件

# 控制策略



## 控制策略



- 设置项目访问级别：公有/私有
- 设置漏洞级别阈值：超过阈值的镜像无法下载
- 设置内容信任：未认证镜像无法下载
- 设置自动扫描：上传即扫描

# 控制策略



## 控制策略

< 项目

### library

镜像仓库 成员 日志 复制 配置管理

项目仓库  公开  
所有人都可访问公开的项目仓库。

部署安全  内容信任  
仅允许部署通过认证的镜像。  
 阻止潜在漏洞镜像  
阻止危害级别 较低 以上的镜像运行。

漏洞扫描  自动扫描镜像  
当镜像上传后，自动进行扫描

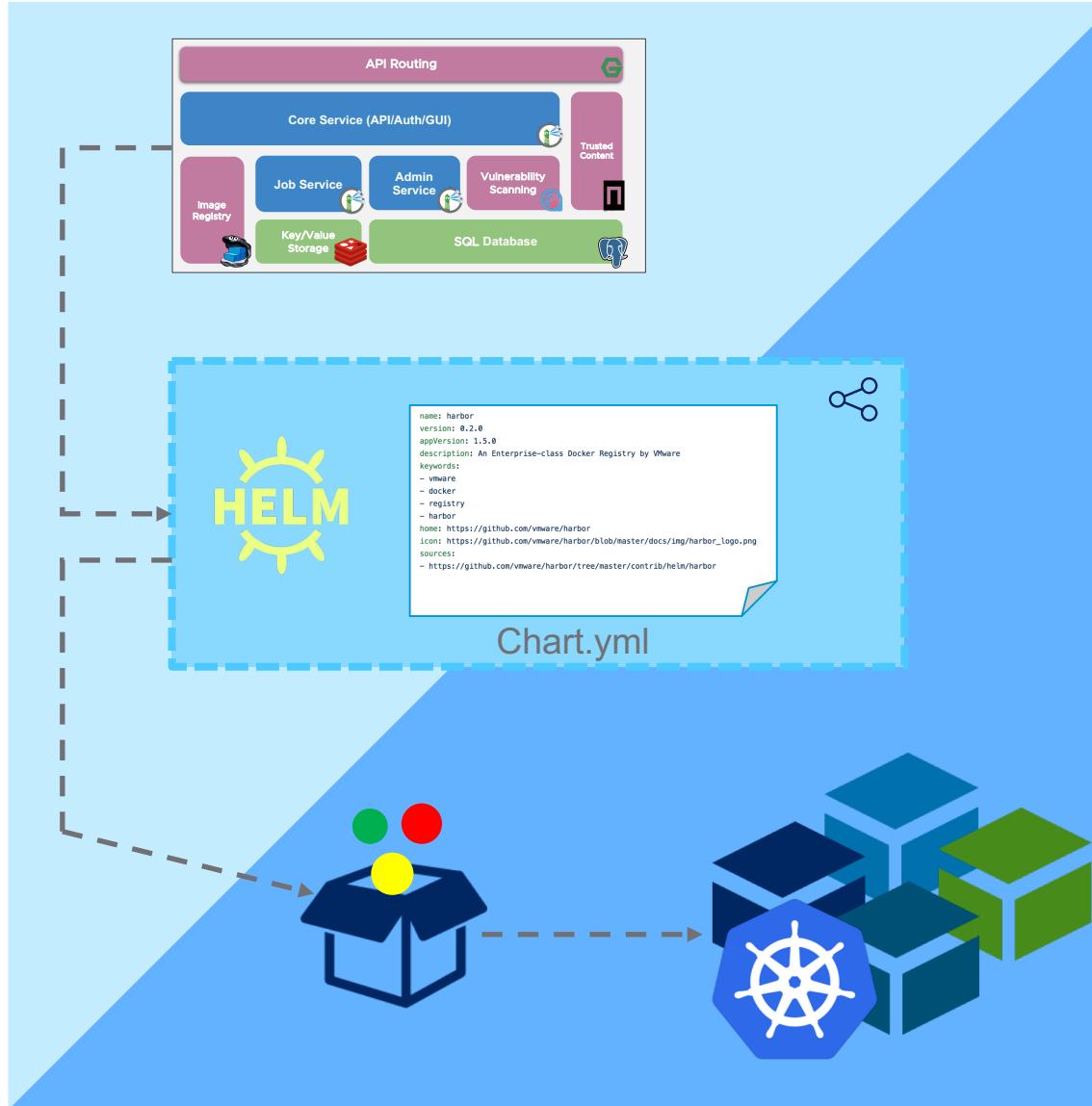
保存 取消

- **设置项目访问级别：公有/私有**
- **设置漏洞级别阈值：超过阈值的镜像无法下载**
- **设置内容信任：未认证镜像无法下载**
- **设置自动扫描：上传即扫描**

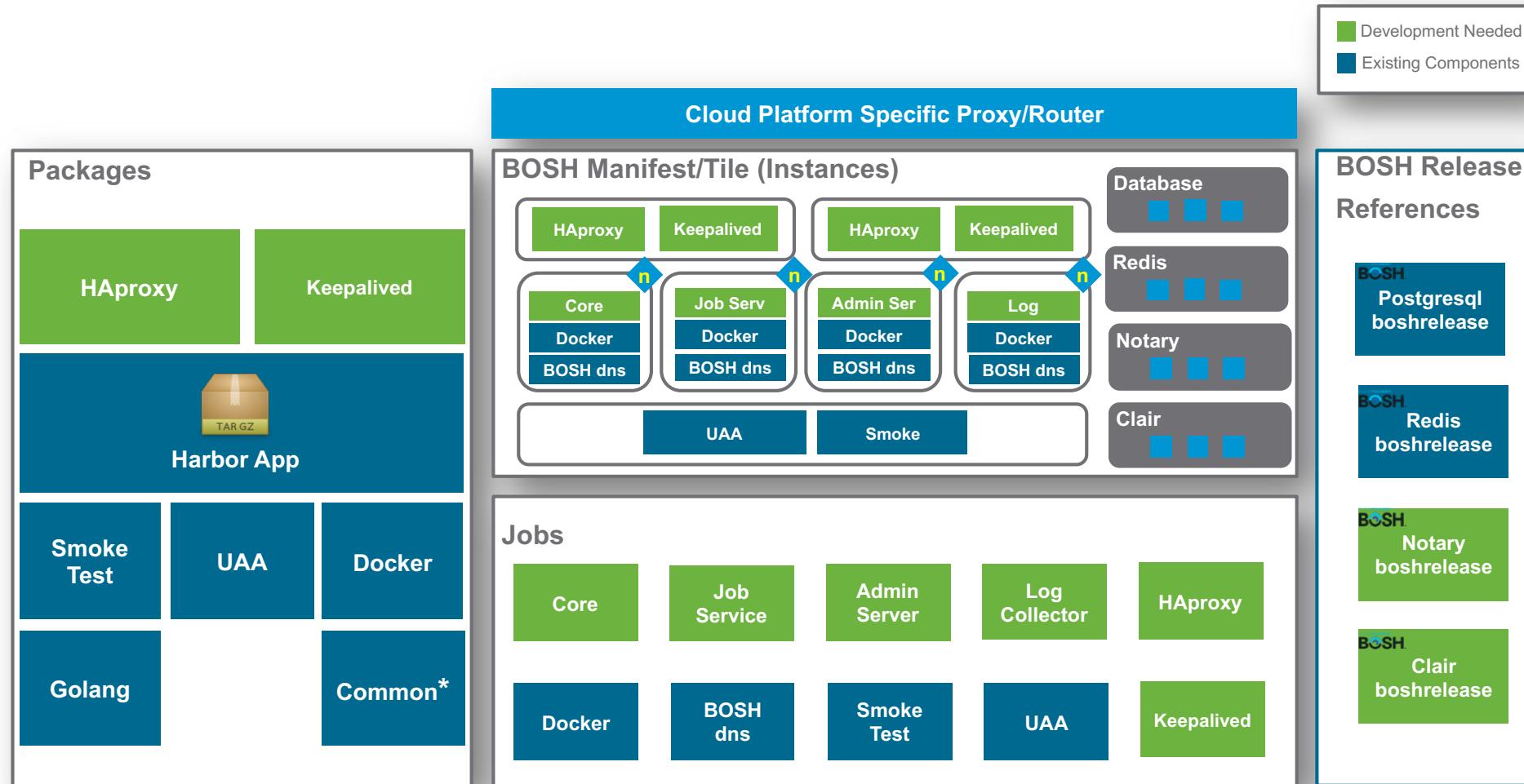
# RELIABILITY



# Deploy Harbor HA via Harbor Helm chart



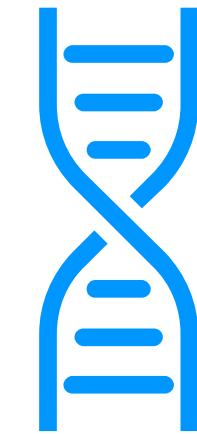
# Deploy Harbor HA via BOSH (Planning)



# DEPLOYMENT

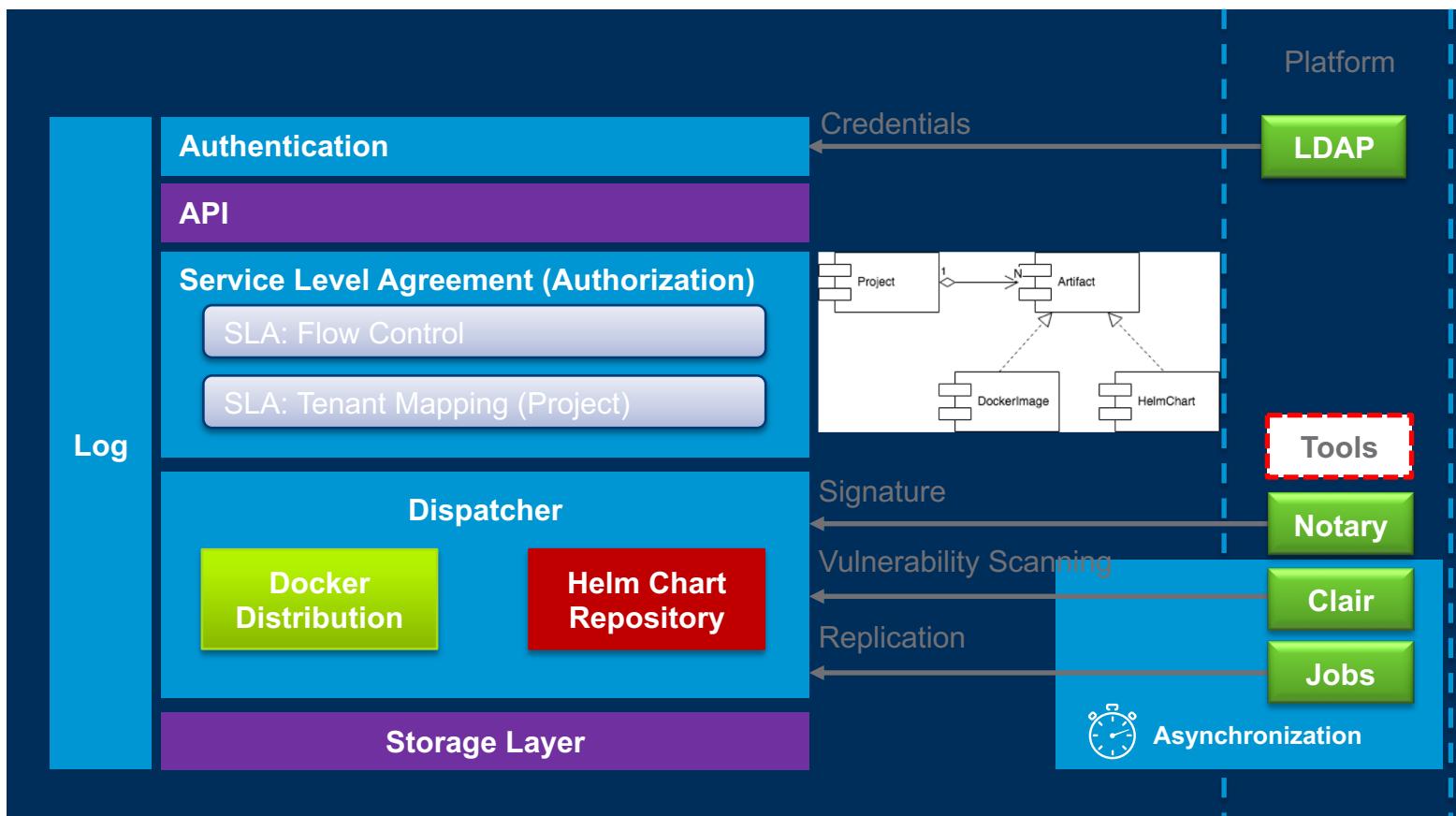
Chart Repo

Deployments



# Helm Chart仓库支持

- Updated Components
- Planning Components



在容器镜像管理基础之上，Harbor将实现Helm Chart仓库的能力以支持对Chart的管理，并将通过创新方式打通Chart与镜像管理的通道。

- Helm Chart事实上的是Kubernetes包管理标准
- 企业级应用与服务编排部署的有效模式
- 简化难度，大幅提升生产效率
- 与容器镜像紧密关联，利用Harbor优势
- 拥抱云原生与Kubernetes

# Helm Chart仓库支持 ( 续 )

Supported



MANAGE



DOWNLOAD



UPLOAD



DELETE

版本列表

To be support



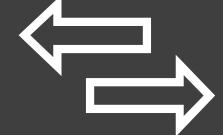
TRUST



VERIFY&amp;SCAN



REPLICATE

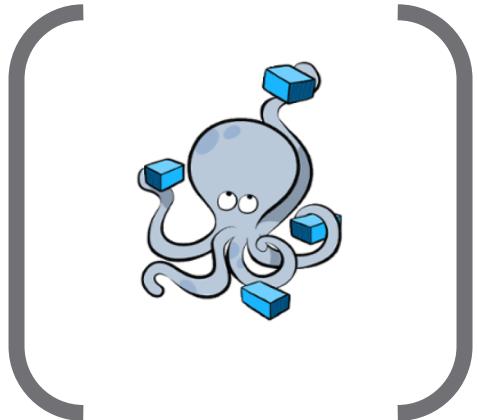


EXPORT/IMPORT

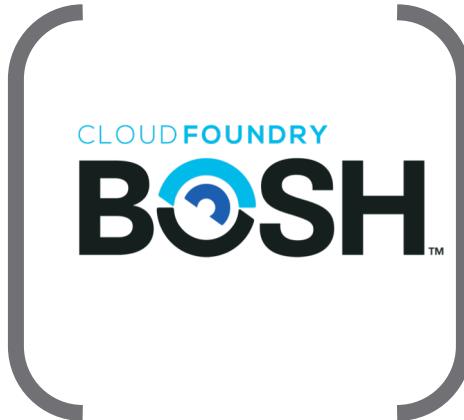
详情信息

# Deploy Harbor via various ways

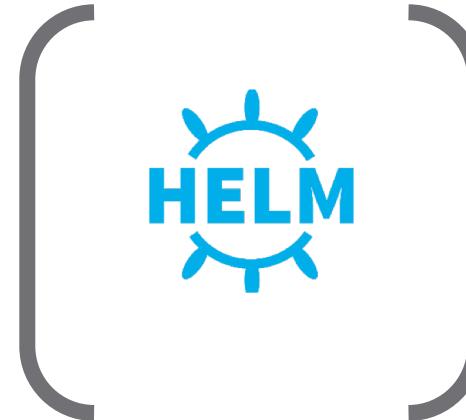
Click to edit optional subtitle



DOCKER COMPOSE



BOSH RELEASE/TILE



HELM CHART\*

NOTES: \* HA deployment supported



**Website:** <https://github.com/goharbor/harbor>

**Twitter:** @Project\_harbor

**Slack:** #harbor/#harbor-dev (register via slack.cncf.io)

**Email group:** (Refer README on GitHub for the subscription way )

[harbor-users@googlegroups.com](mailto:harbor-users@googlegroups.com)

[harbor-dev@googlegroups.com](mailto:harbor-dev@googlegroups.com)

**WeChat group:** Contact speakers

# Thank You