Managing Cloud Native Artifacts for Large Scale Kubernetes Cluster

Henry Zhang, VMware & Mingming Pei, NetEase

About Us

Haining Henry Zhang

- Technical Director, Cloud Native Lab, VMware China R&D
- Creator and maintainer of Harbor
- Former evangelist of Cloud Foundry China community
- Contributor of FATE/KubeFATE
- Coauthor of Harbor Authoritative Guide (in Chinese)
- Current interest: cloud computing, AI/ML, blockchain etc.

vmware

Mingming Pei

- Architect, NetEase
- Responsible for Qingzhou Cloud Native DevOps platform design and implementation
- Rich experience and solid understanding of cloud native technologies
- Harbor maintainer
- Kraken contributor
- Coauthor of Harbor Authoritative Guide (in Chinese)
- Current interest: cloud native DevOps, mircoservice architecture, distributed system etc.

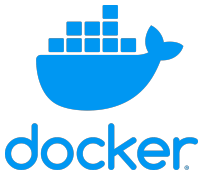
網易 NETEASE 网易轻舟 NETEASE QINGZHOU

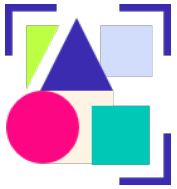
Agenda

- Artifact management of cloud native apps
- Artifact management capabilities of Harbor
- Case study of NetEase

Cloud Native Apps Management

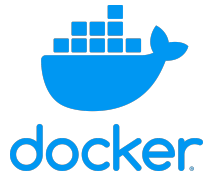
- Two aspects of cloud native applications
  - ✓ Dynamic - runtime
  - ✓ Static - artifacts

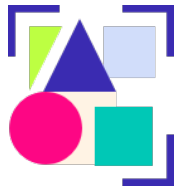




Harbor for Artifact Management

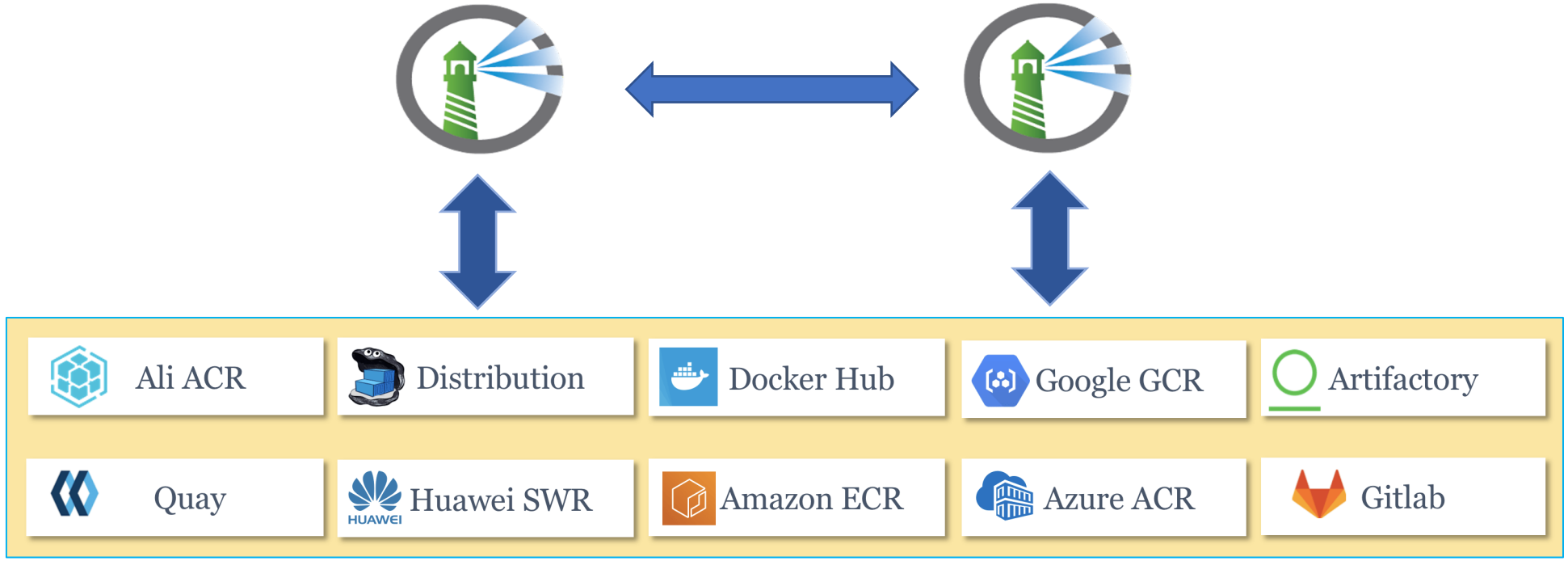
- OCI Artifact Support
  - ✓ Container images, Helm charts, CNAB, OPAs, Singularity, …
- Multitenancy
  - ✓ RBAC, Project Isolation
- Policy
  - ✓ Quotas, Retention, Immutability, Signing, Vulnerability
- Security & Compliance
  - ✓ Identity & Access Mgmt, Scanning, CVE Exceptions
- Extensibility
  - ✓ Webhooks, Replication, Pluggable Scanners, REST API, Robot Accounts, CLI Secrets





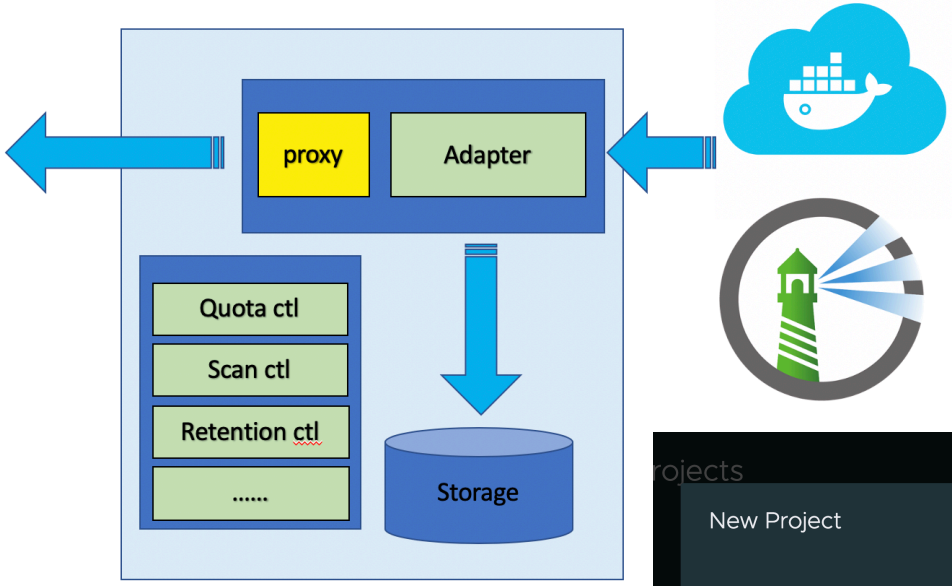
Artifact replication

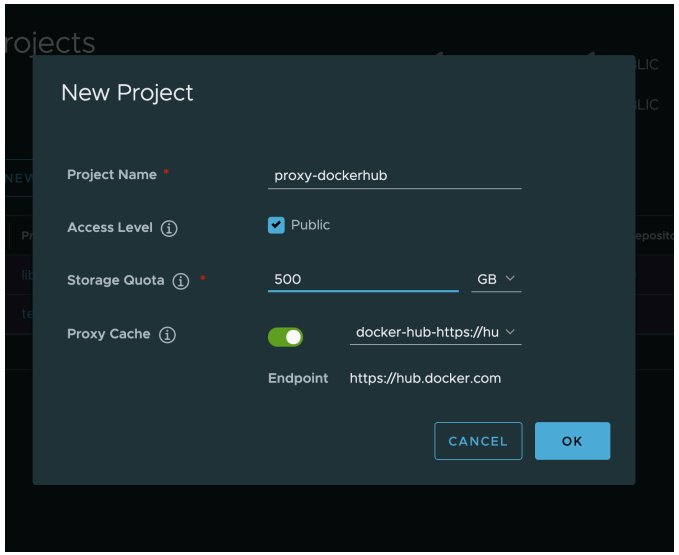
- Replicating artifacts across clouds
  - ✓ Multi-cloud & multi-cluster support
  - ✓ Multiple registries and services
  - ✓ Automatic and reliable



Proxy cache project

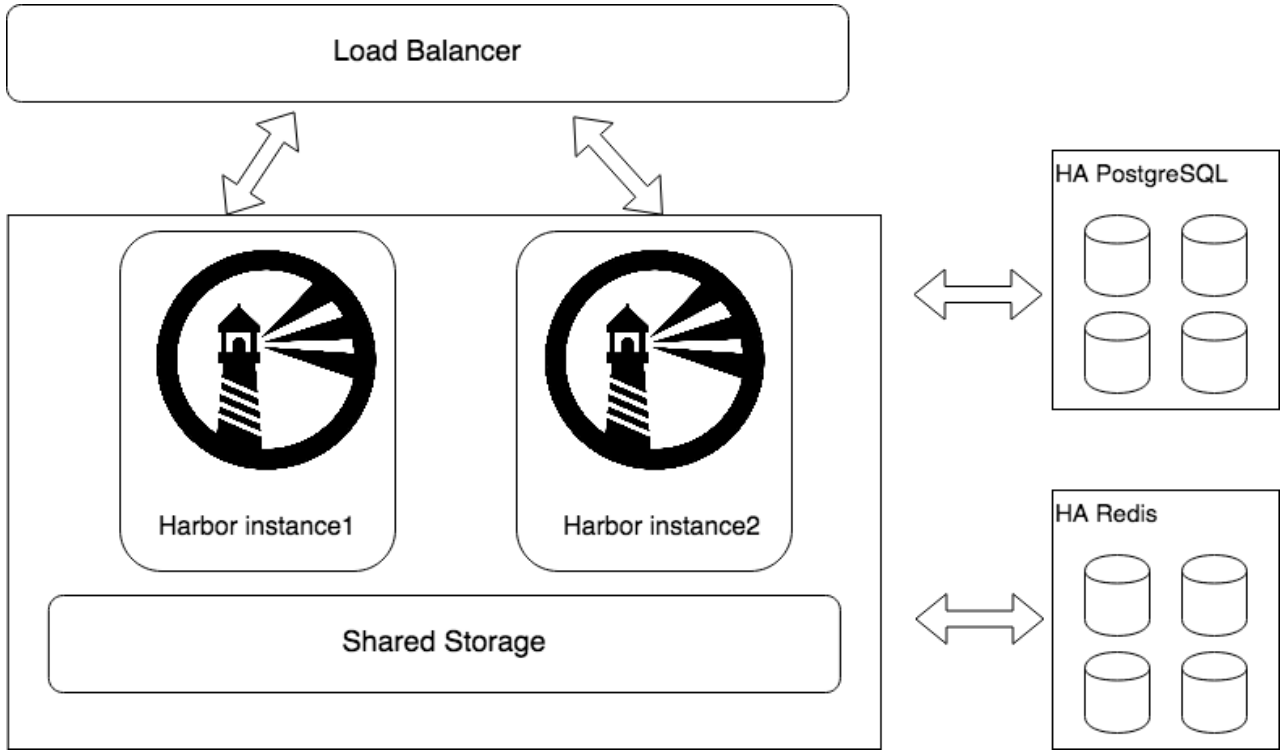
- Provide the “proxy” capability at project level
- Caching images and serving locally
- Faster distribution and minimize unneeded traversals over the network
- Management policies can be applied to proxied artifacts, eg. quota, scanning …





High Availability (HA) and Scalable Registry

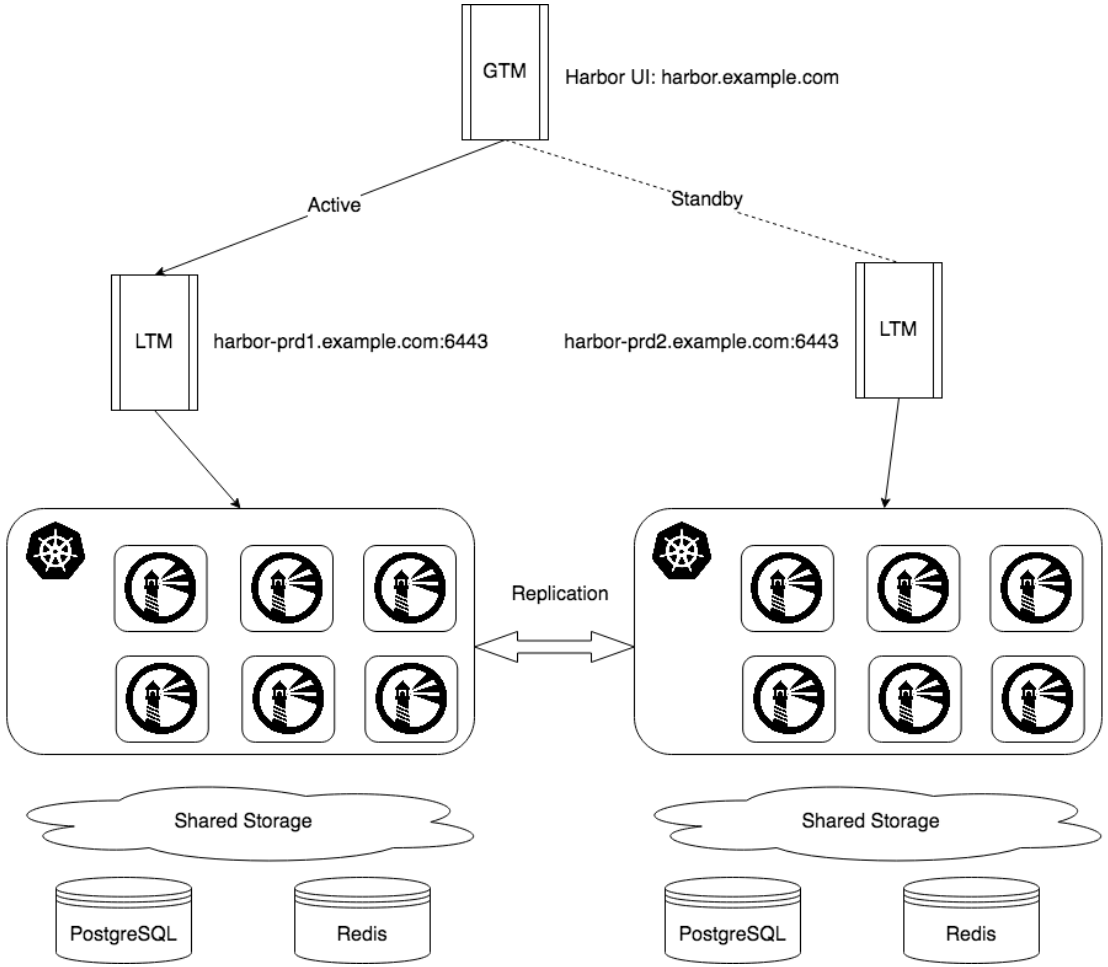
- Harbor core services are stateless
- Persistency:
  - PostgreSQL
  - Redis
  - Shared storage (mainly for artifacts, logs)



High Availability(HA) and Scalable Registry (2)

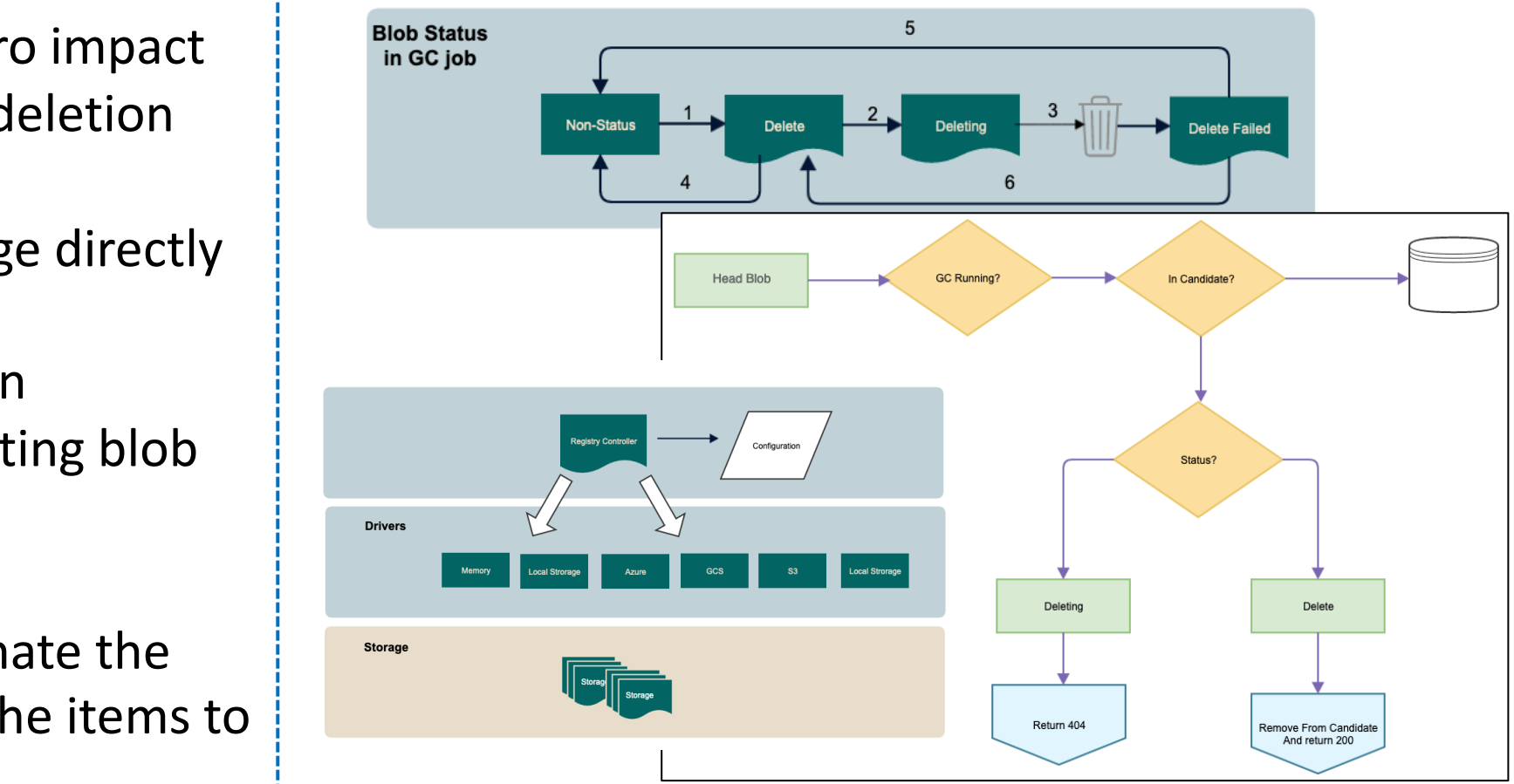
- Multi-datacenters and multi-cloud
- Artifacts replicated between datacenters
- Data in PostgreSQL and Redis need to be in sync
- Active –Standby

- Need to be careful of the propagation delay between two datacenters



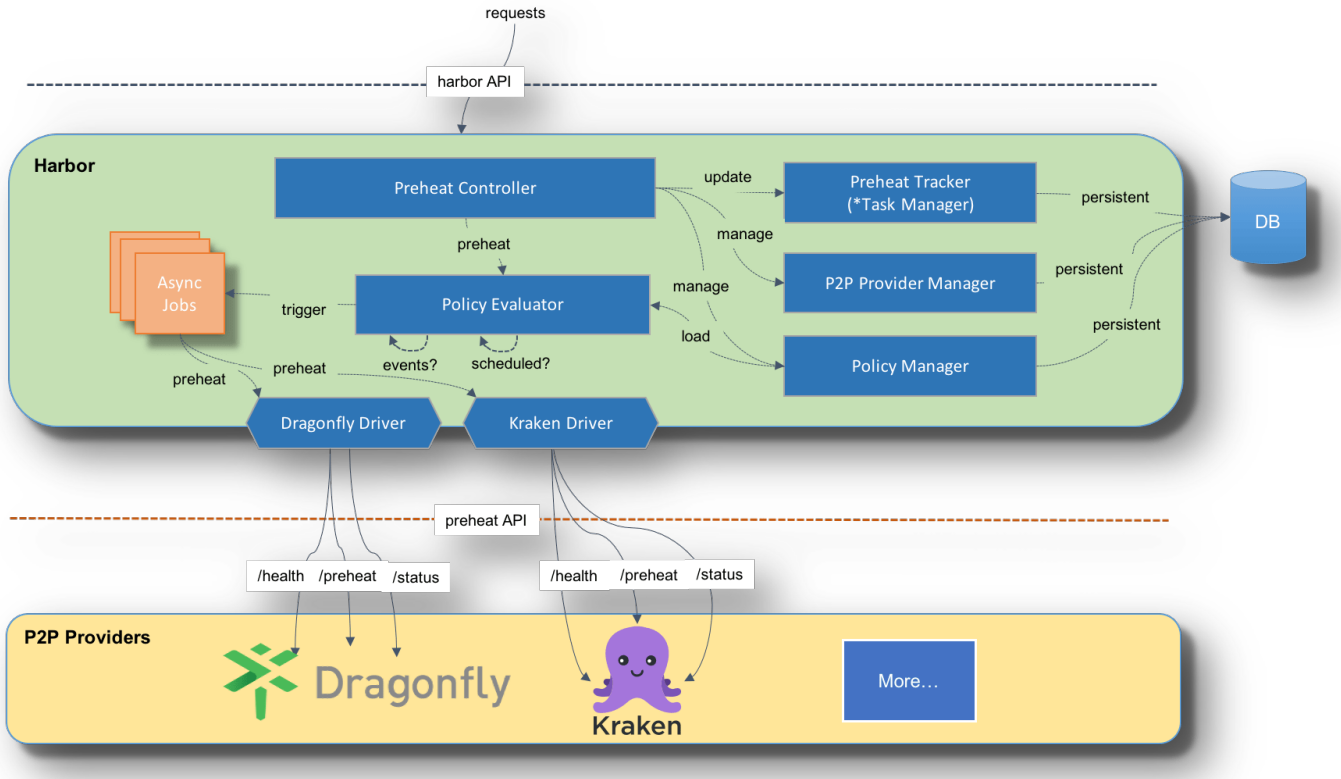
Non-Blocking Garbage Collection

- Garbage collection with zero impact to image pushing, pulling, deletion
- GC handler to access storage directly
- Remove the dependency on distribution binary for deleting blob data
- ”Dry-run” a GC job to estimate the released space and verify the items to be deleted



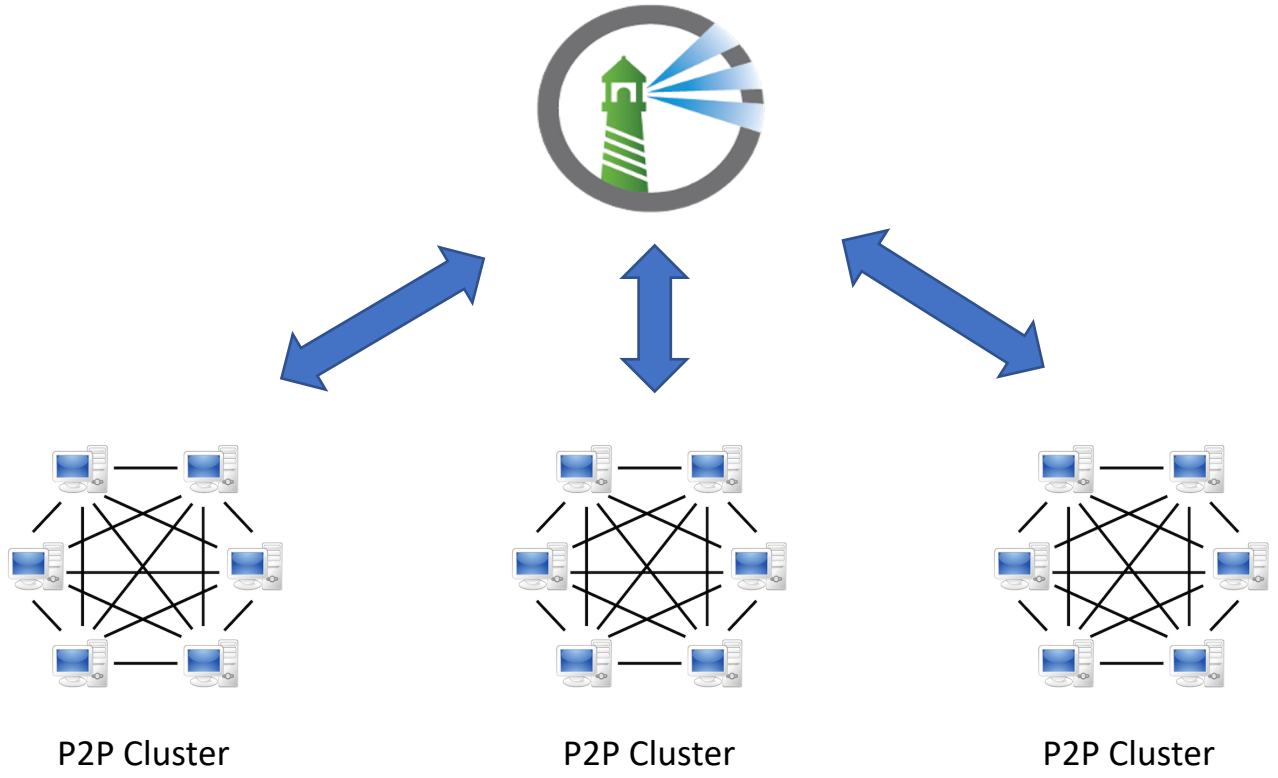
P2P Preheat

- Leverage the capabilities of P2P engines to accelerate the content distribution process (esp. for large scale deployments)
- Distribute the content to the P2P network in advance (warm up the P2P network)
- Policy-based control
- Supported engines
  - Dragonfly
  - Kraken



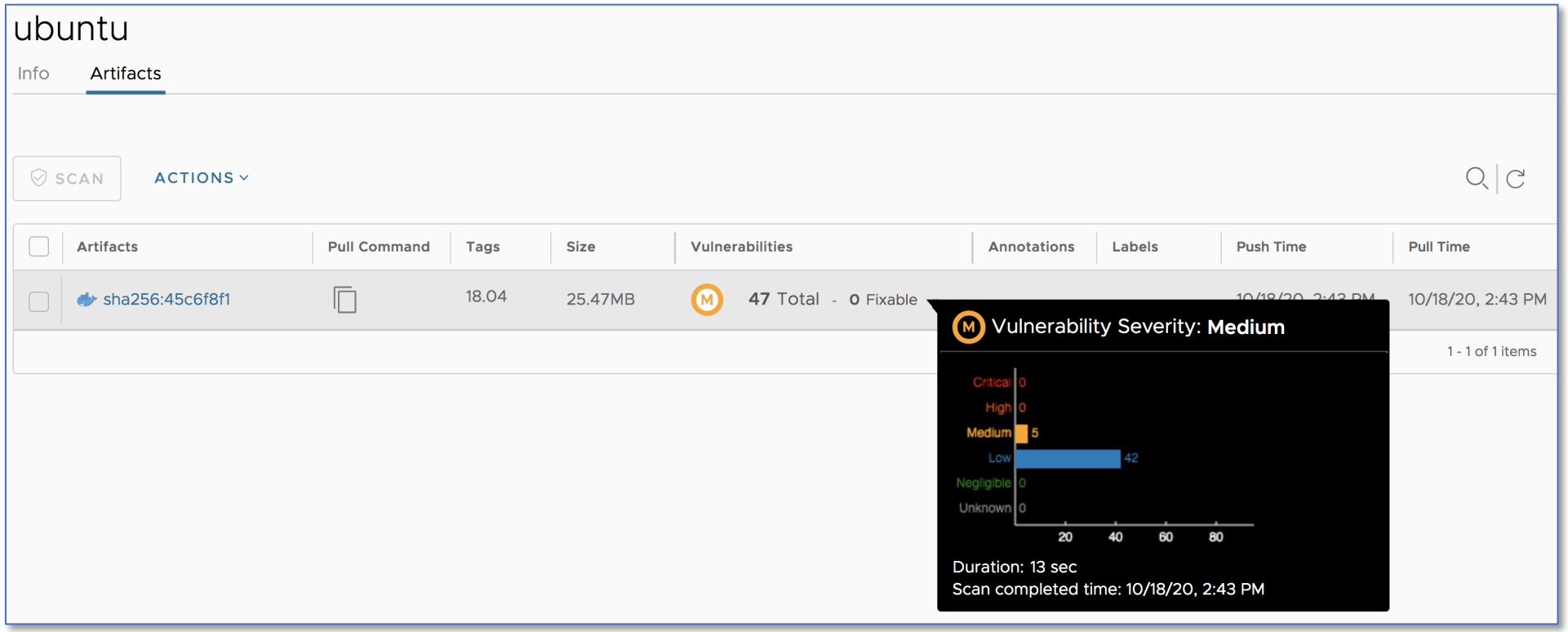
P2P Image Distribution

- Image preheating for multiple clouds and multiple clusters



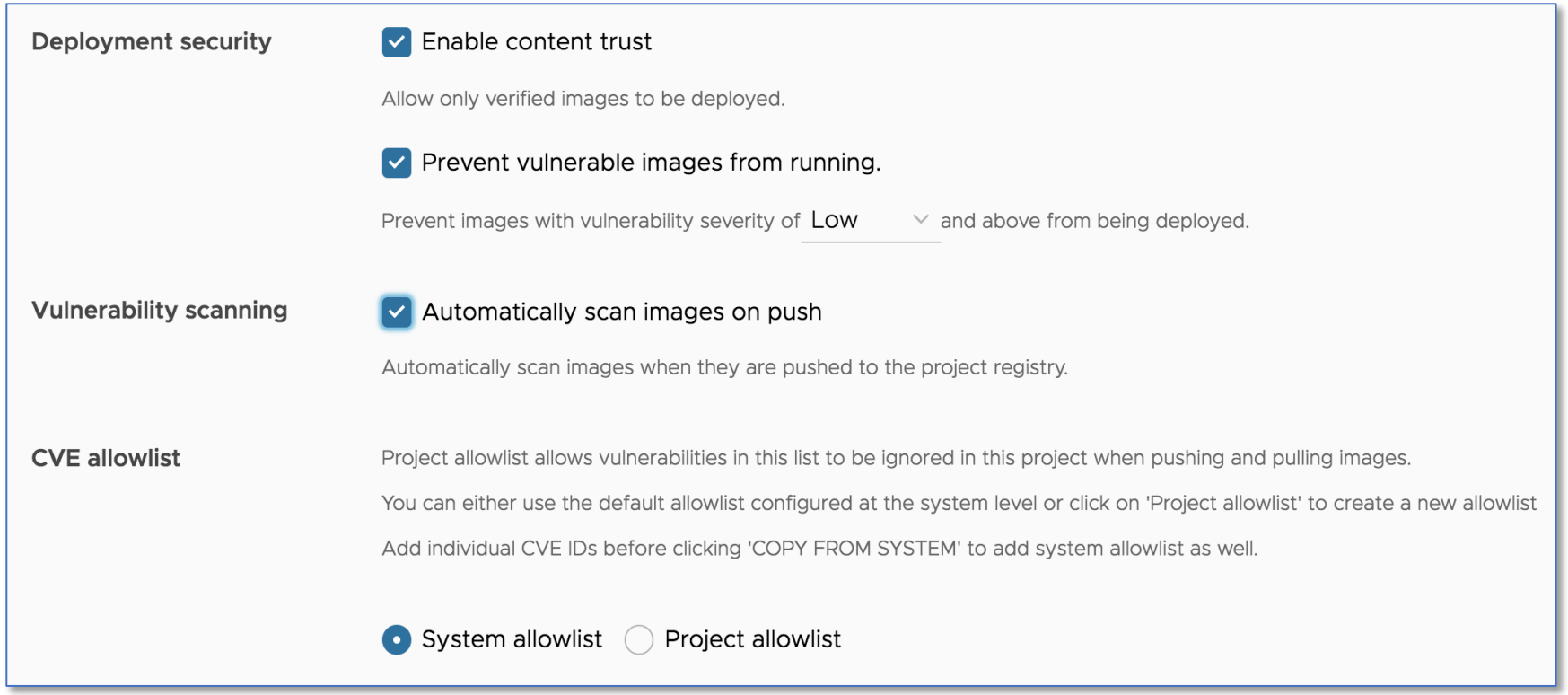
Vulnerability Scanning

- Static analysis of vulnerability based on CVE databases
- Could be used in CI pipelines and production



Security Considerations

- Severity threshold of vulnerability can block `pulls` if exceeded
- Content trust (image signing)
- Auto scan on push
- CVE allowlist



Enhanced Artifact Processor

OCI artifacts support:

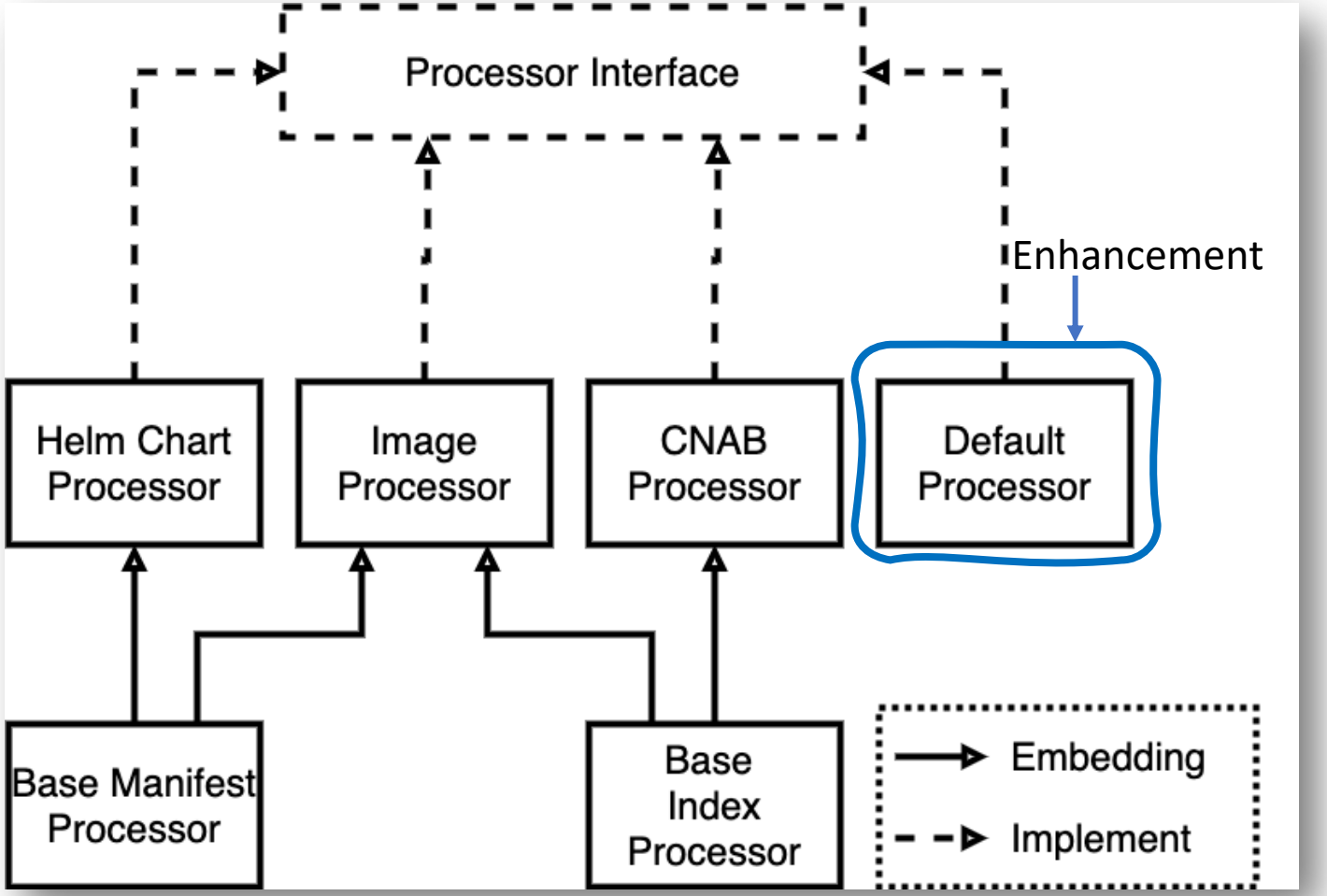
- Managing and distributing the artifacts that following the OCI spec from V2.0
- Extra metadata provided by Harbor via data processors

User-defined artifact type:

- Users can define their own bundle formats based on the OCI spec
- Processed by the extension of default processor

Use Case:

- ML models can be stored in Harbor

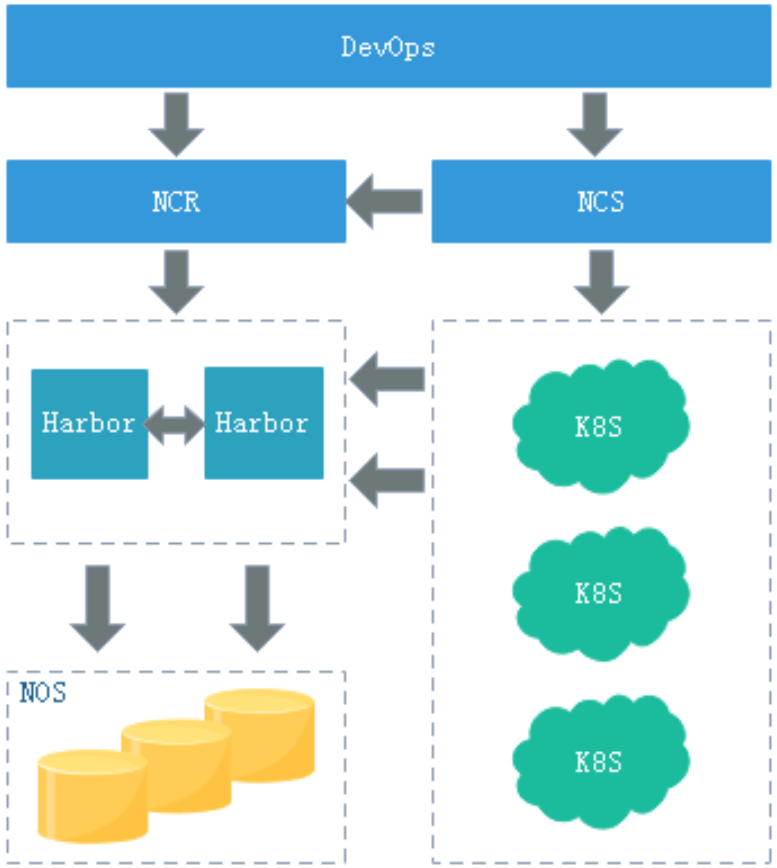


Scenarios in NetEase

- Container technology
- Container image, Helm chart, operator bundle as artifacts
- GitOps in use
- Harbor as the repository of cloud native artifacts
- Lots of Kubernetes clusters in NetEase and 5000+ nodes in large ones
- 20+ Harbor instances
- Largest Harbor instance manages about 100 thousand images

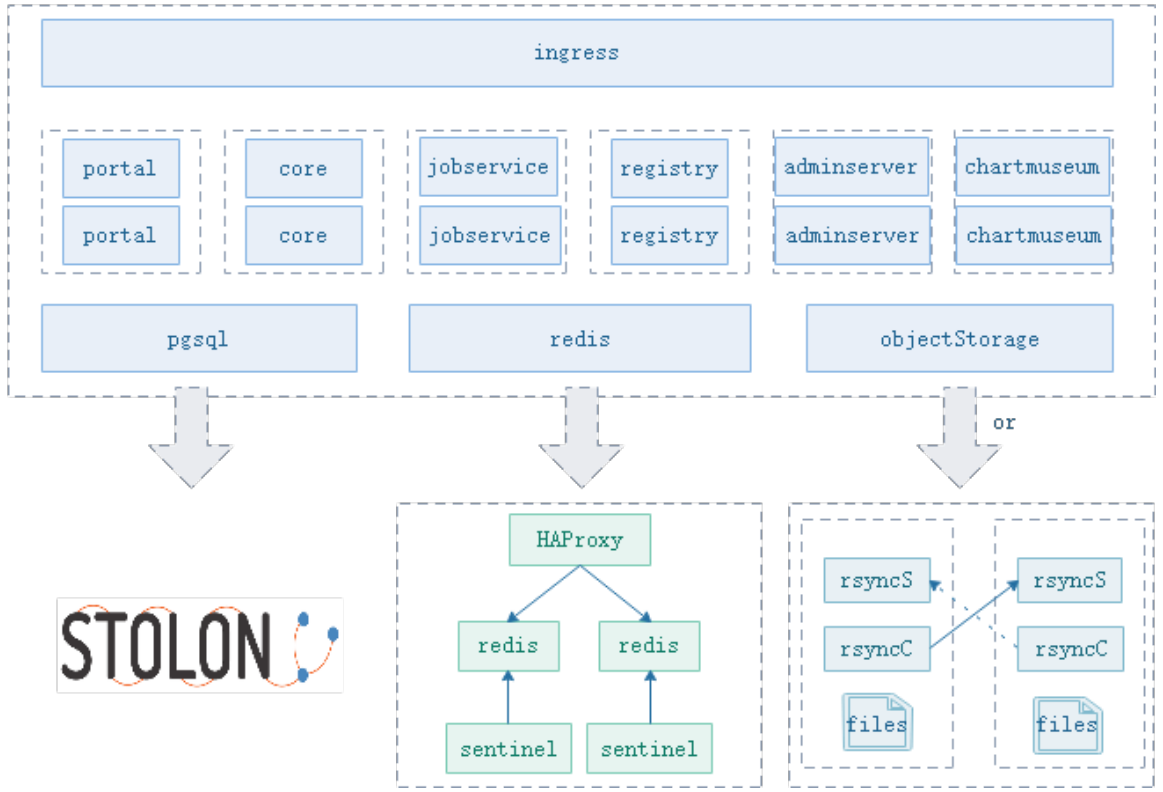
Our architecture

- NCR(NetEase Container Repository) uses Harbor instances as backend
- NCS(NetEase Container Service) uses Kubernetes as backend
- Flexible combination for K8S cluster and container repository
- Access using RBAC of NetEase Cloud Native Platform
- Integrated NOS(NetEase Object Storage) as backend for high availability and high performance



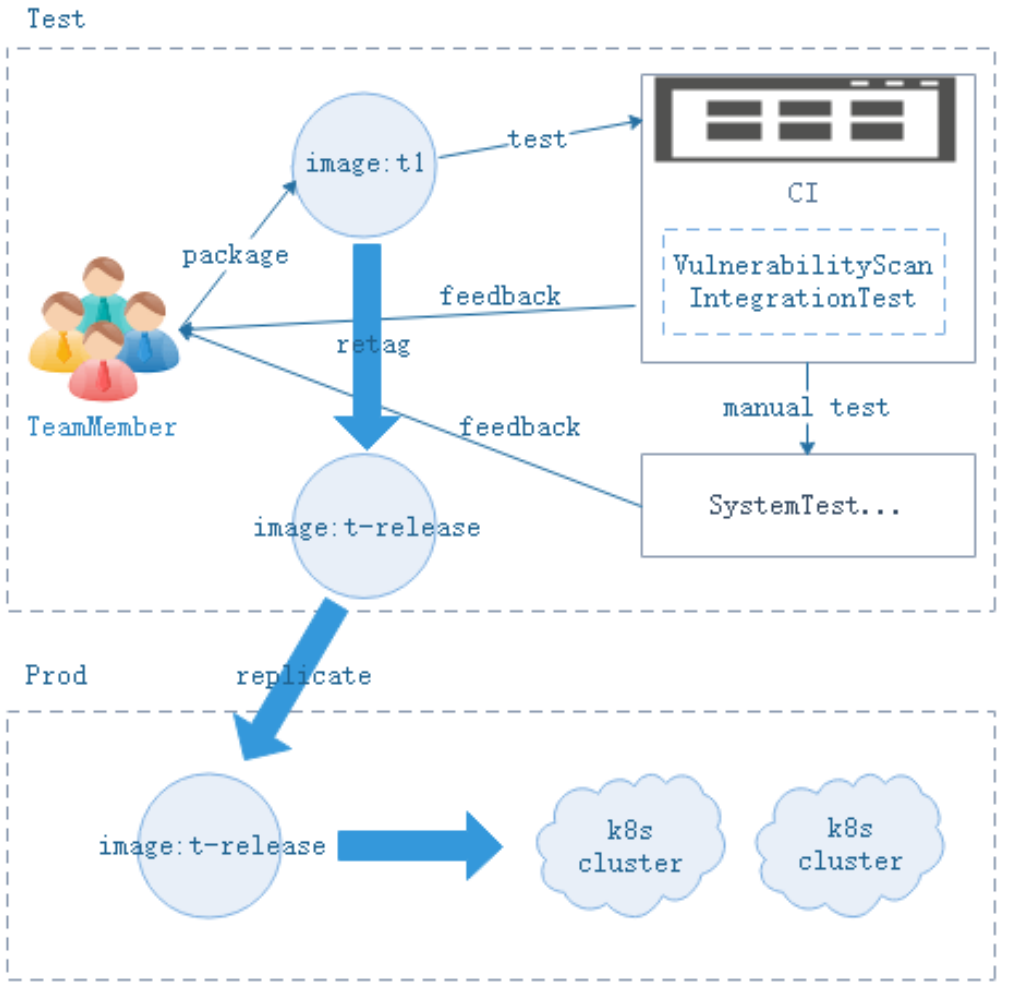
High availability

- Highly available Harbor
  - Object Storage as backend
  - Local filesystem and `rsync` for data sync
- External HA dependency
  - Stolon project for PostgreSql
  - HAProxy to provide HA of Redis
- Enhancement of monitoring
  - Health monitoring of Harbor service
  - Log monitoring and alarm
  - Scenario monitoring (base on Prometheus): replication failure, P2P dispatching failure etc.



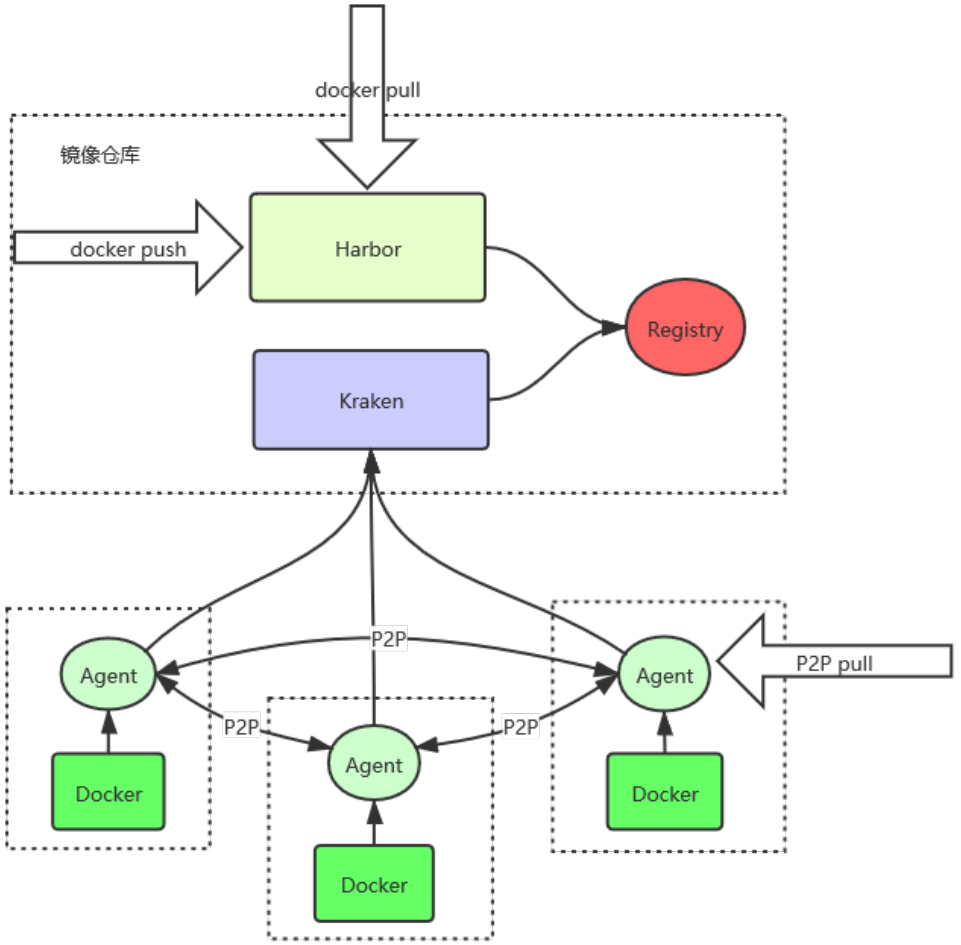
Used in multi-environments

- Multi-environments and isolated Network
- Images are packaged and tested in test environment
- Retag the image with `release` when all tests passed
- Trigger remote replication by tag



Distribution in large cluster

- Problems:
  - Throughput pressure in registry service
  - Network bandwidth pressure in backend storage
- Integrate Harbor with Kraken
  - Share the same registry as Harbor
  - Kraken preheat depends on registry push event
- Achieved results:
  - 5000+ concurrent pulls
  - Speed up the dispatching of images of 10GB+ in size



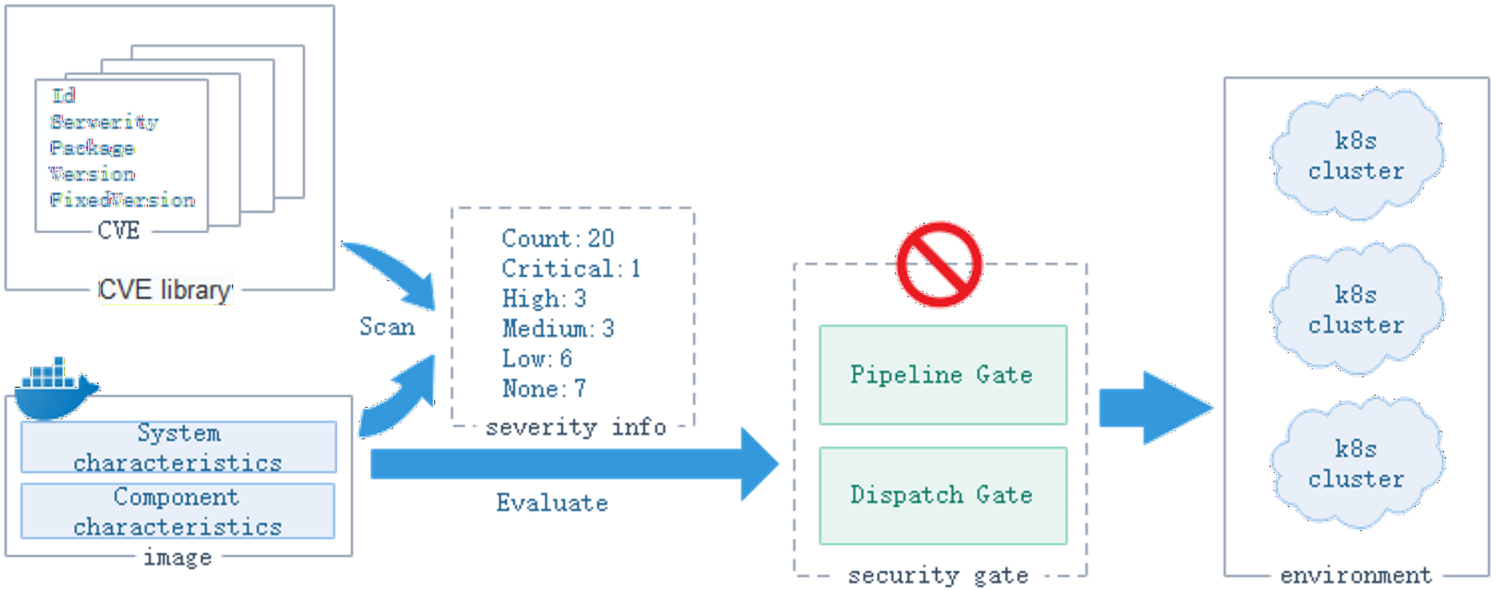
Performance in P2P distribution

- Network bandwidth limit is configurable
- More layers and smaller layer size cause lower utilization of bandwidth
- The number of p2p peers brings little impact to the distribution performance (max 15K peers supported officially)

Image size	Layer count	Pull concurrency	Network bandwidth limit(in , out)	Avg time(s)	Usage rate of Network bandwidth
420MB	14	1000	10MB/s, 8MB/s	4.9s	35%
420MB	14	1000	50MB/s, 40MB/s	7.3s	39%
1.58GB	11	800	50MB/s, 40MB/s	17.5s	53%
1.58GB	11	2000	100MB/s, 80MB/s	6.7s	69%
1.58GB	11	2000	200MB/s, 140MB/s	3.2s	72%
1.58GB	11	5000	200MB/s, 140MB/s	3.4s	70%

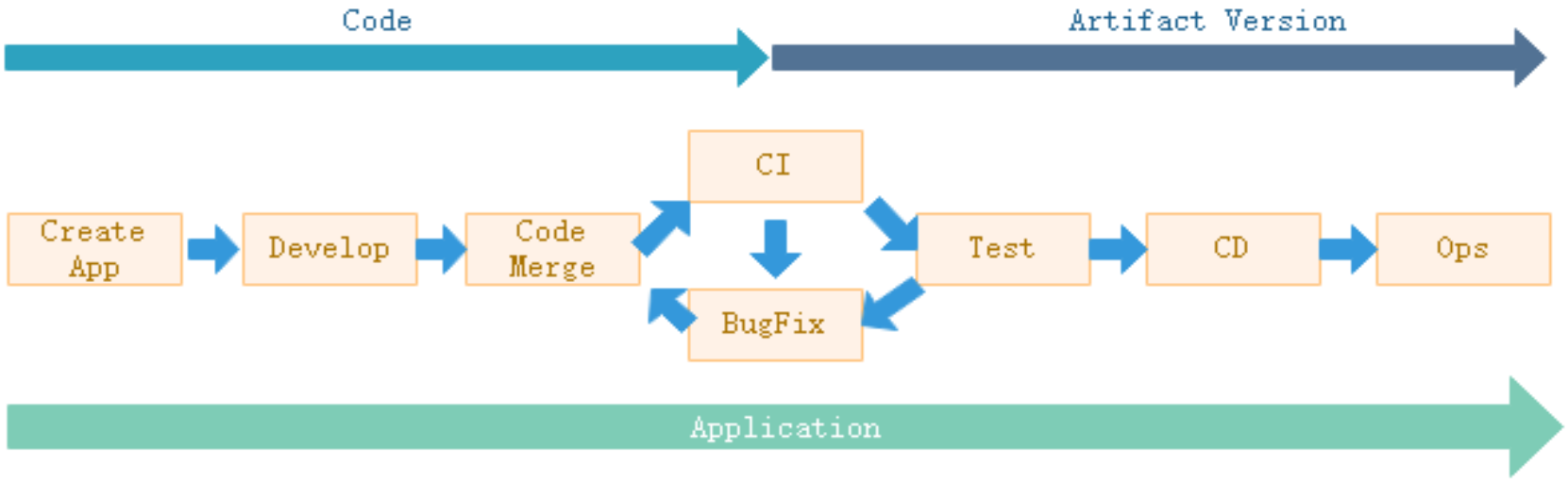
Artifact security

- Images are scanned and severity info generated
- Two kinds of security gate:
  - Pipeline Gate: Severity threshold of vulnerability can fail pipeline, if exceeded
  - Dispatch Gate: Specified severity level of vulnerability can block pulls
- Image signature by Notary to guarantee image's provenance



Integrated with CI/CD

- Application-centric CI/CD
- Code management in development period and followed by artifact management
- CI/CD: artifact version management, artifact security, CD triggered by image pushing webhook
- The CI/CD process builds around artifacts

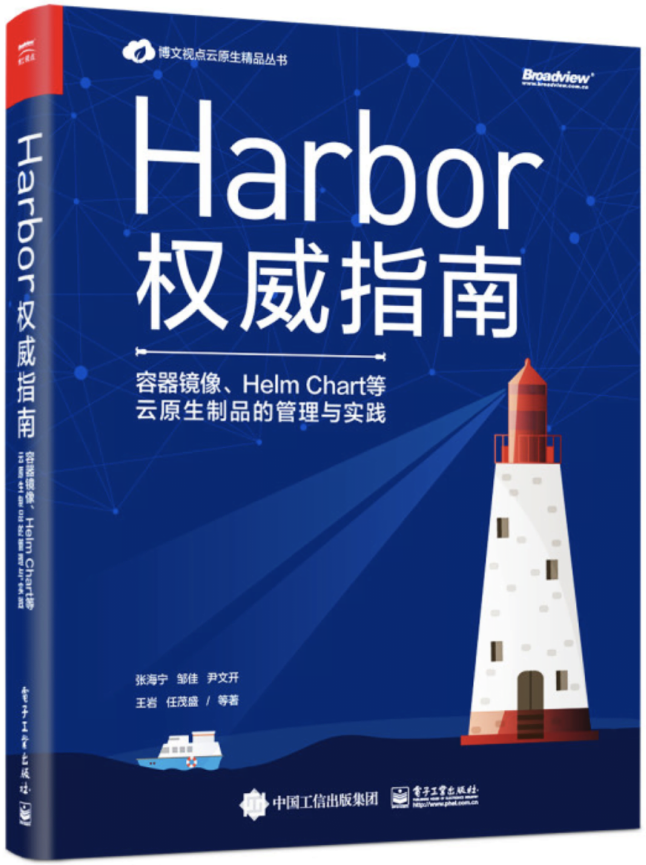


Summary

- Artifact management is an important aspect in operations of cloud native environment
- Registry is a good place to perform management of artifacts
- Harbor is a powerful tool for managing artifact
  - High availability and scalability
  - Artifact replication
  - Proxy caching and non-blocking GC
  - Vulnerability scanning and content trust
  - Integrated P2P cluster preheat
  - Customized artifact types
- Case study of Netease for large scale Kubernetes clusters

The Harbor Authoritative Guide

- The first book about Harbor (in Chinese)
- Authored by Harbor maintainers
- Get insights into the architecture, mechanics, configuration and customization of Harbor
- Learn about Harbor success stories through Case Studies
- Learn how to operate Harbor - For cloud native engineers, architects, and contributors
- Released in October 2020




KEEP CLOUD NATIVE
EVERYWHERE
KubeCon
CloudNativeCon
North America 2020
Virtual
HELM
KV