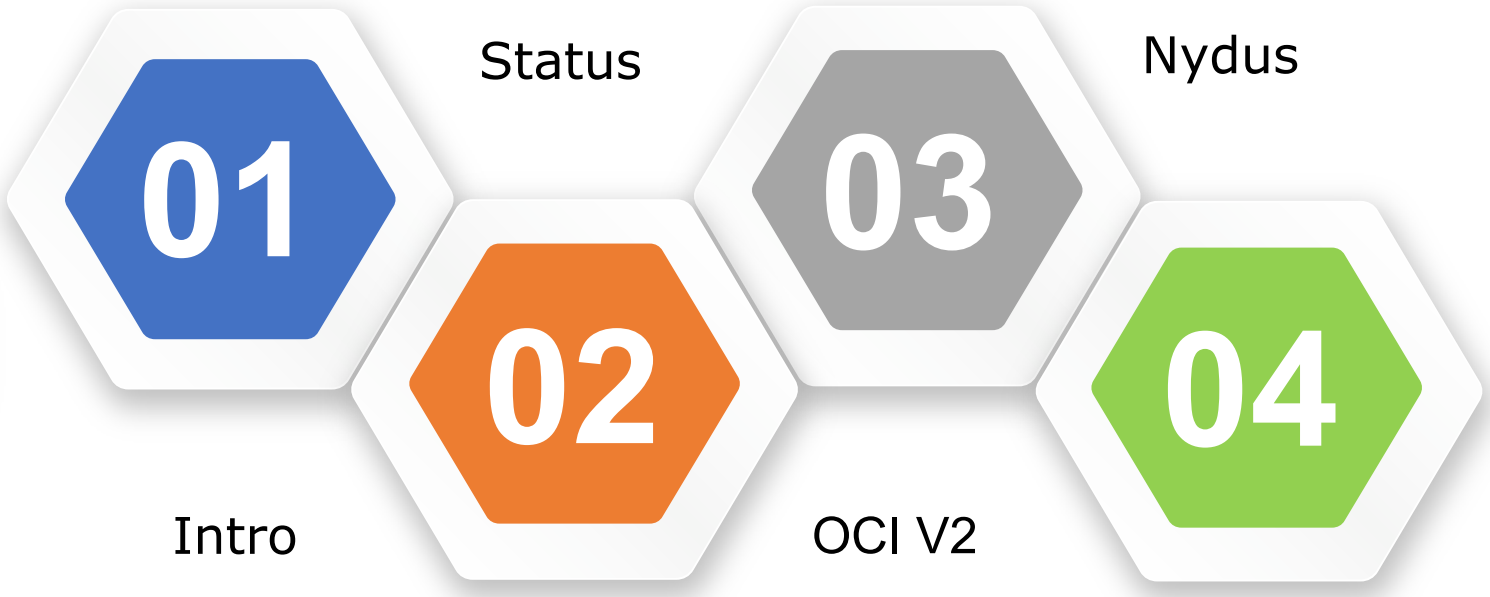
CLOUD NATIVE + OPEN SOURCE

Virtual Summit China 2020

Dragonfly - Make Image Distribution Efficiently and Safely

Jieyue Ma, Ant Group
Yuxing Liu, Alibaba Cloud

Agenda

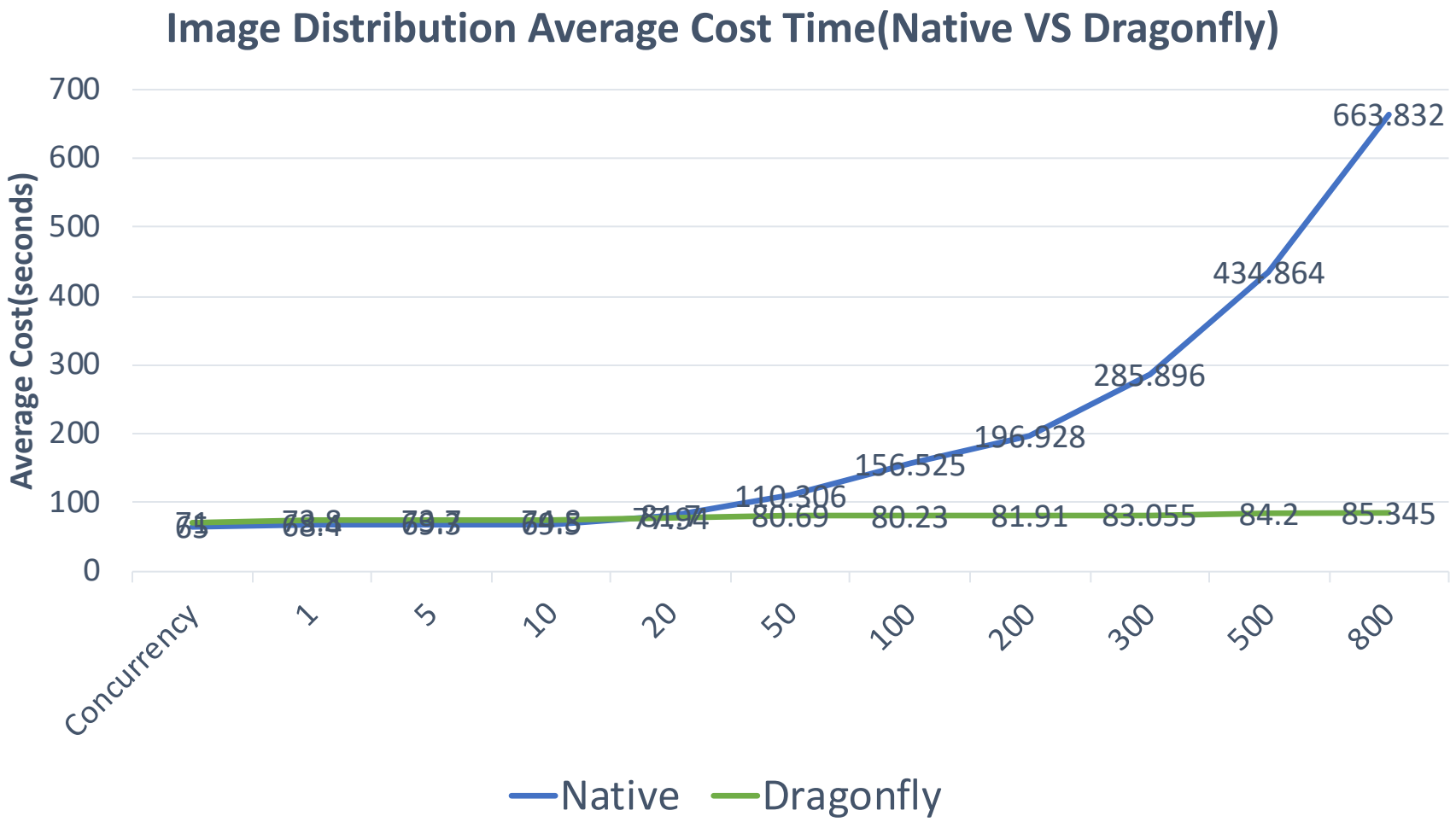


01

PART ONE
Intro

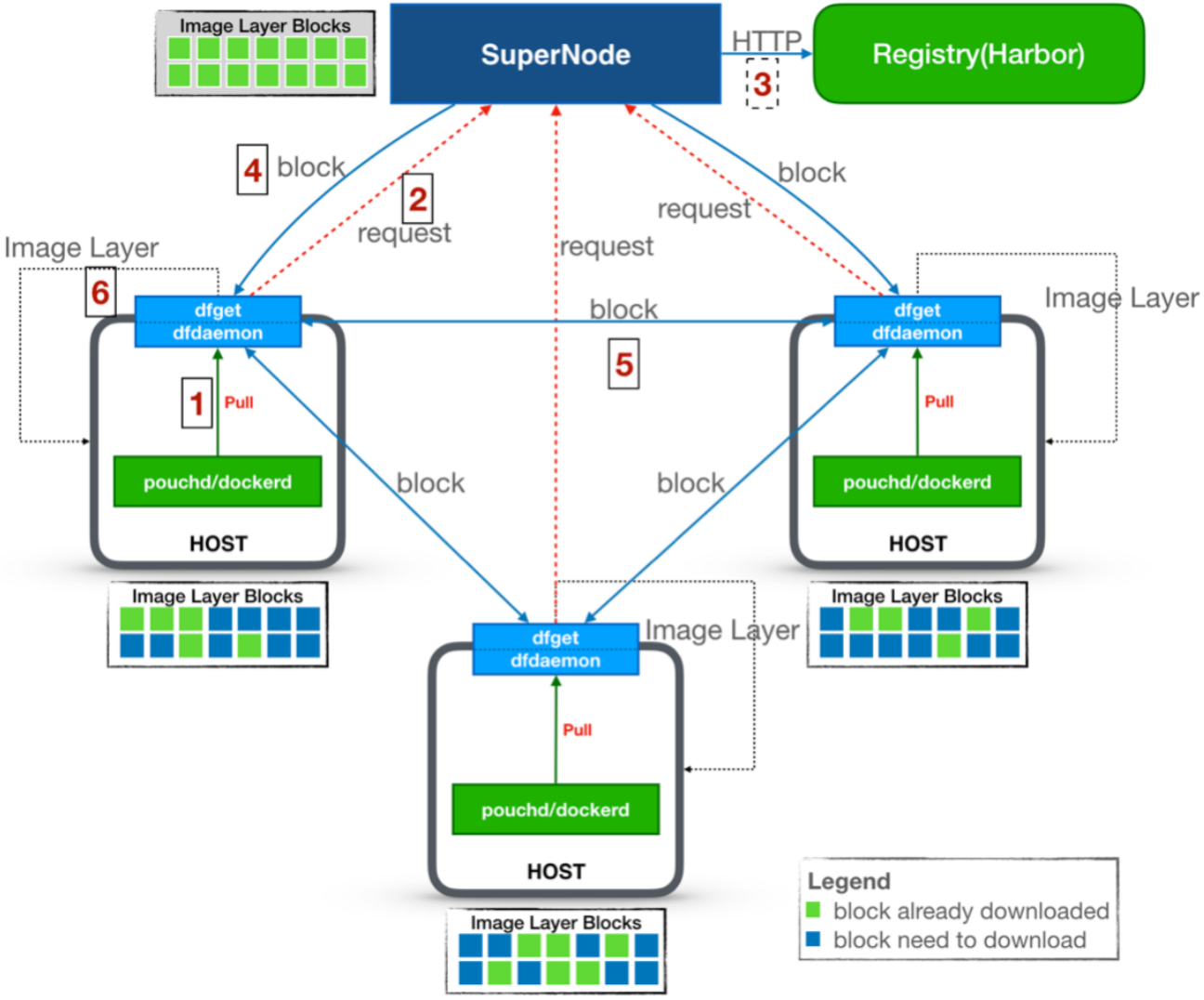
What & Why

- P2P based, high reliable, efficient, image distribution infrastructure
- Bottleneck in pulling images at large scale



P2P Mechanism

1. Pull image from node proxy
2. Send pulling requests to SuperNode
3. Cache image from Registry if non-exist
4. Reply to node peers which have blocks
5. Transport blocks among all peers
6. Finish whole pulling when all blocks downloaded



How to Use

1. Deploy SuperNode
2. Deploy dfclient on every client
3. Config Container Engine Daemon
4. Pull images with Dragonfly

- Docker Registry Mirrors

```
--registry-mirror=https://<your awesome mirror>
```

- Containerd CRI Registry Mirrors

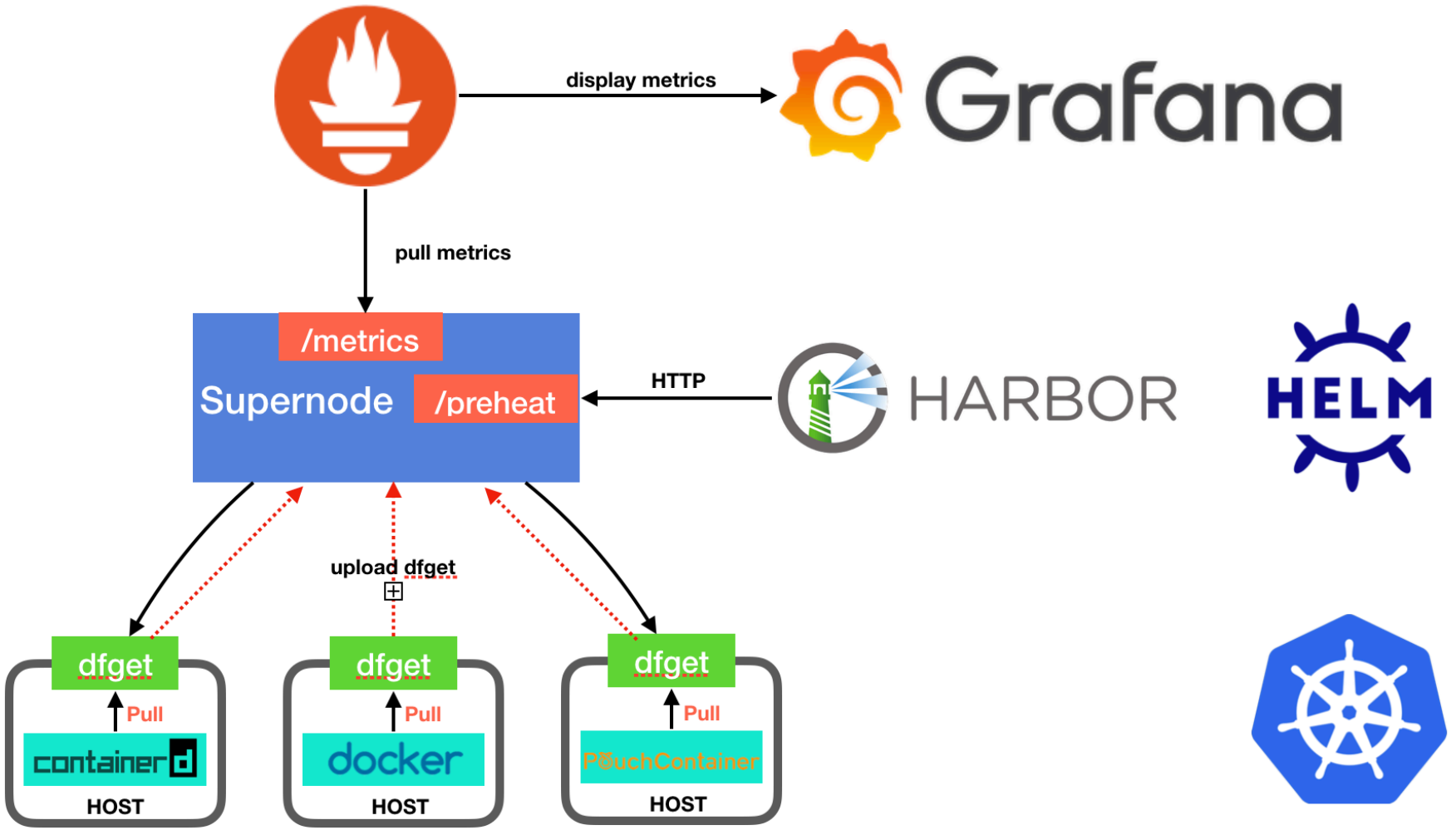
```
[plugins.cri.registry.mirrors]
  [plugins.cri.registry.mirrors."docker.io"]
    endpoint = ["https://registry-1.docker.io"]
  [plugins.cri.registry.mirrors."test.secure-registry.io"]
    endpoint = ["https://HostIP1:Port1"]
  [plugins.cri.registry.mirrors."test.insecure-registry.io"]
    endpoint = ["http://HostIP2:Port2"]
```

- Add redirection in your registry

02

PART TWO
Status

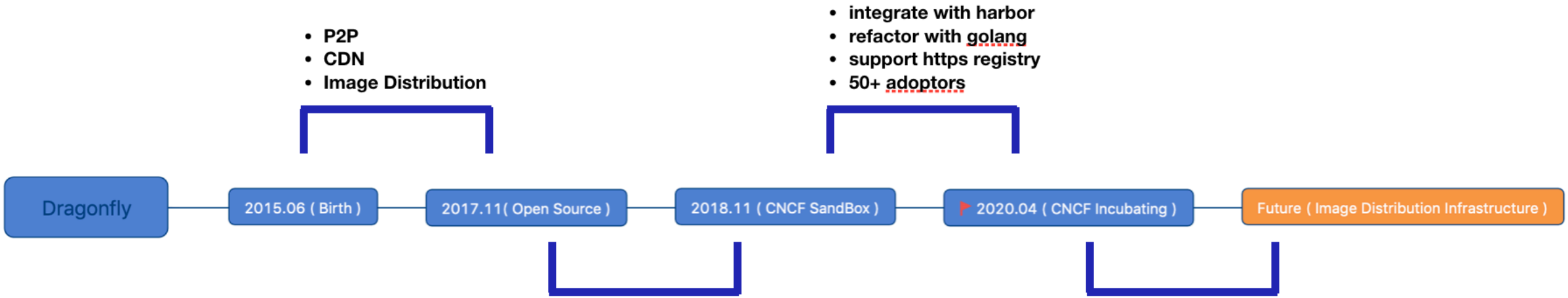
Ecology


display metrics
Grafana
pull metrics
/metrics
Supernode
/preheat
HTTP
HARBOR
HELM
upload dfget
dfget
Pull
containerd
HOST
dfget
Pull
docker
HOST
dfget
Pull
PouchContainer
HOST

CLOUD NATIVE + OPEN SOURCE
Virtual Summit China 2020

Community Situation

Milestone



- P2P
- CDN
- Image Distribution

- integrate with harbor
- refactor with golang
- support https registry
- 50+ adoptors

- preheat
- memory file system
- dynamic compression
- cover the whole "Alibaba Group"

- Security
  - Different encryption algorithm
- Openness
  - Highly user-customized modules
- Efficiency
  - Intelligent scheduling algorithm
  - Dynamic downloading rate limiting
  - Streaming-P2P Pattern
- CNCF Ecosystem
  - Integration with Harbor deeply
  - ★ OCI V2
- Scenarios
  - P2P tranmission on cloud disks
  - Support different repositories(mvn/svn/git/...)
- ……

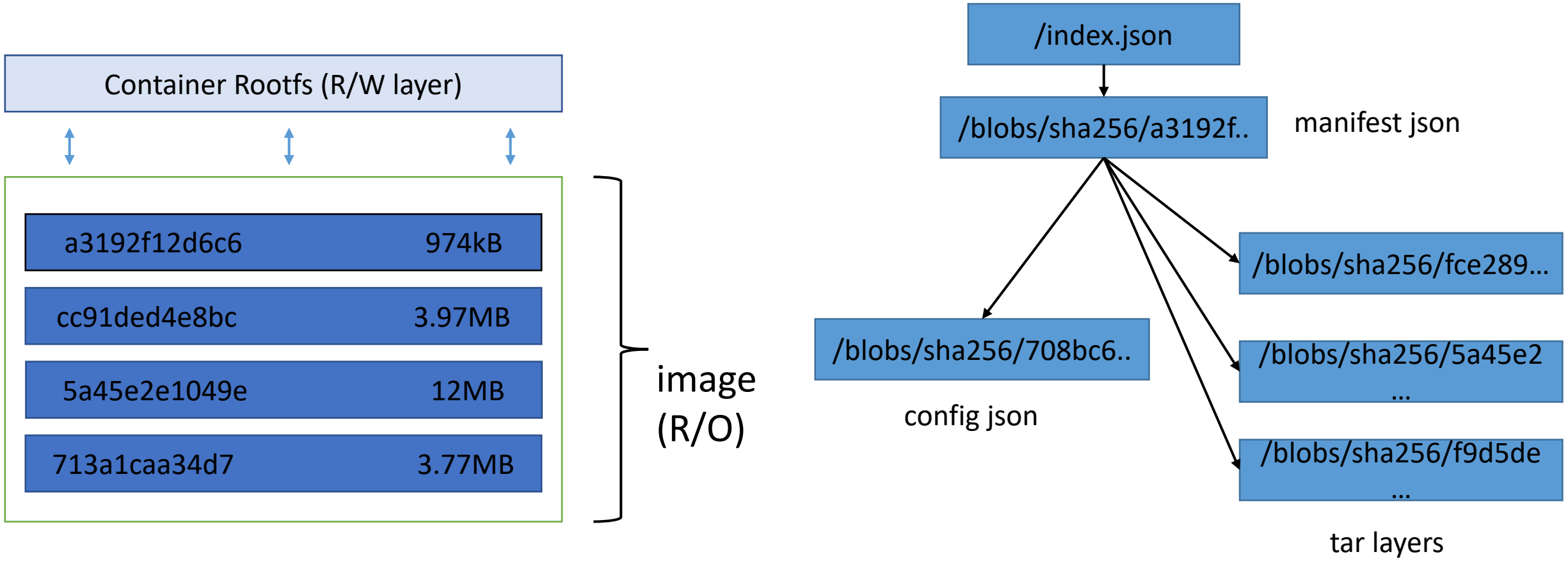
03

PART THREE

OCI V2

Container Image Background

Container image is a lightweight, standalone, executable package of software including everything needed to run an application, e.g. code, system tools and libraries



Why need OCIv2 format spec

The OCI organization is now discussing the new OCIv2 format, which focuses on these issues in OCIv1 format

- Lazyload

While only a small fraction of image might be used by application, we waste a lot of disk space, image pulling time, network bandwidth and cpu cycles for the entire image downloading & decompressing procedure

- Deduplication

Metadata or small part of modified data will lead to a total file copy in the upper layer, while deleted files still downloaded in the lower layers.

- Security

After decompressed, image files are not verifiable, and the targz lacks of signature verification, vulnerable to man-in-the-middle attacks. How to verify data in the lazyload mode is still a problem.

Community Prior Work

- P2P based image distribution: Dragonfly, Kraken etc.
- Leveraging and rely on storage level snapshot capability: e.g. Slacker, Ceph rbd solutions
- Layered userspace filesystem: CRFS, but it needs a new stargz file format, which is not a standard
- Leveragin file level blob and local file cache: NTT-filegrain, CERN-vmfs, umoci, but these images package too many blobs, and umoci also rely on local filesystem reflink capability

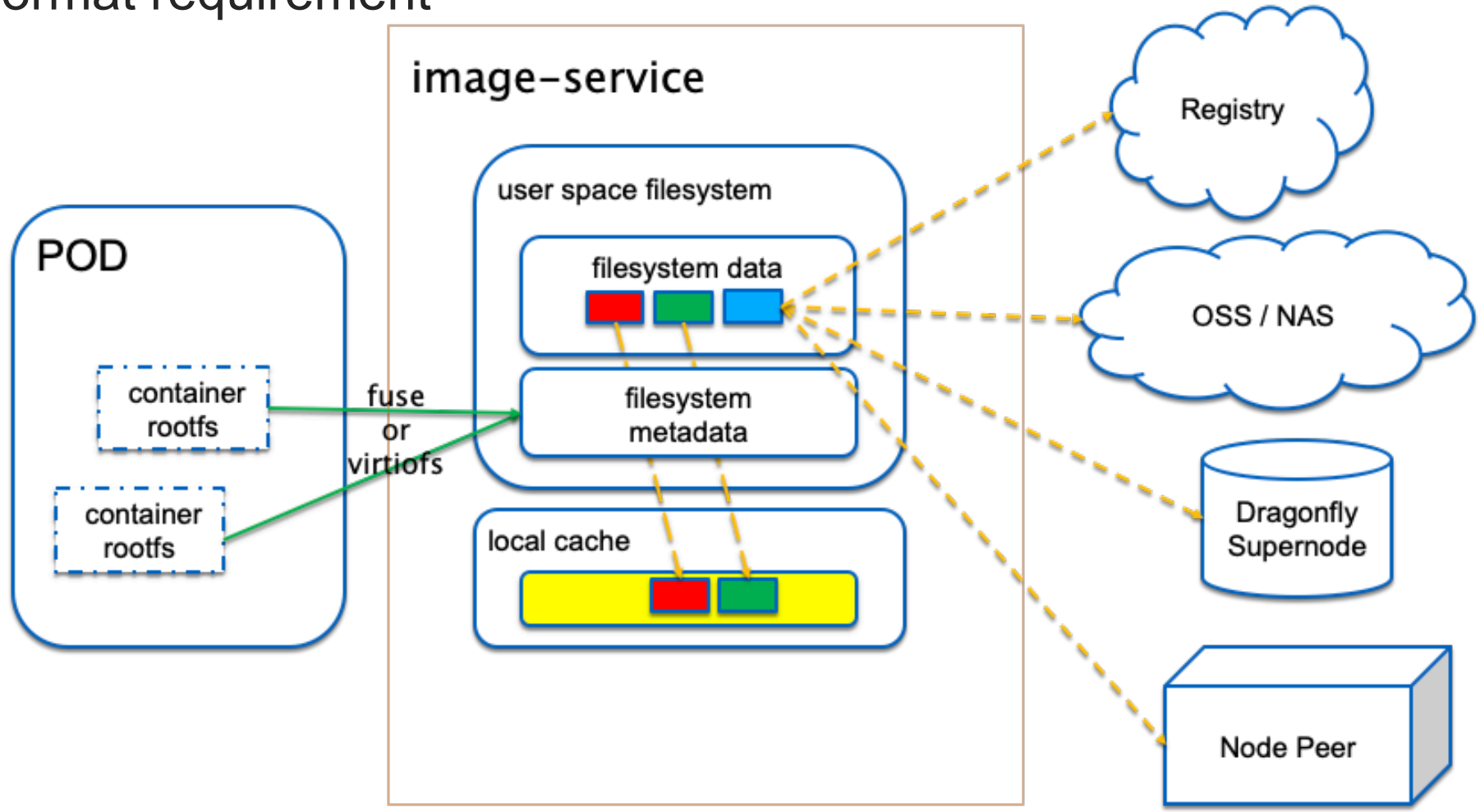
All these solutions lacks of data chunk level on-demand loading(except CRFS), data chunk in-flight verification, or real digital signature preventing from man-in-the-middle attacks.

04

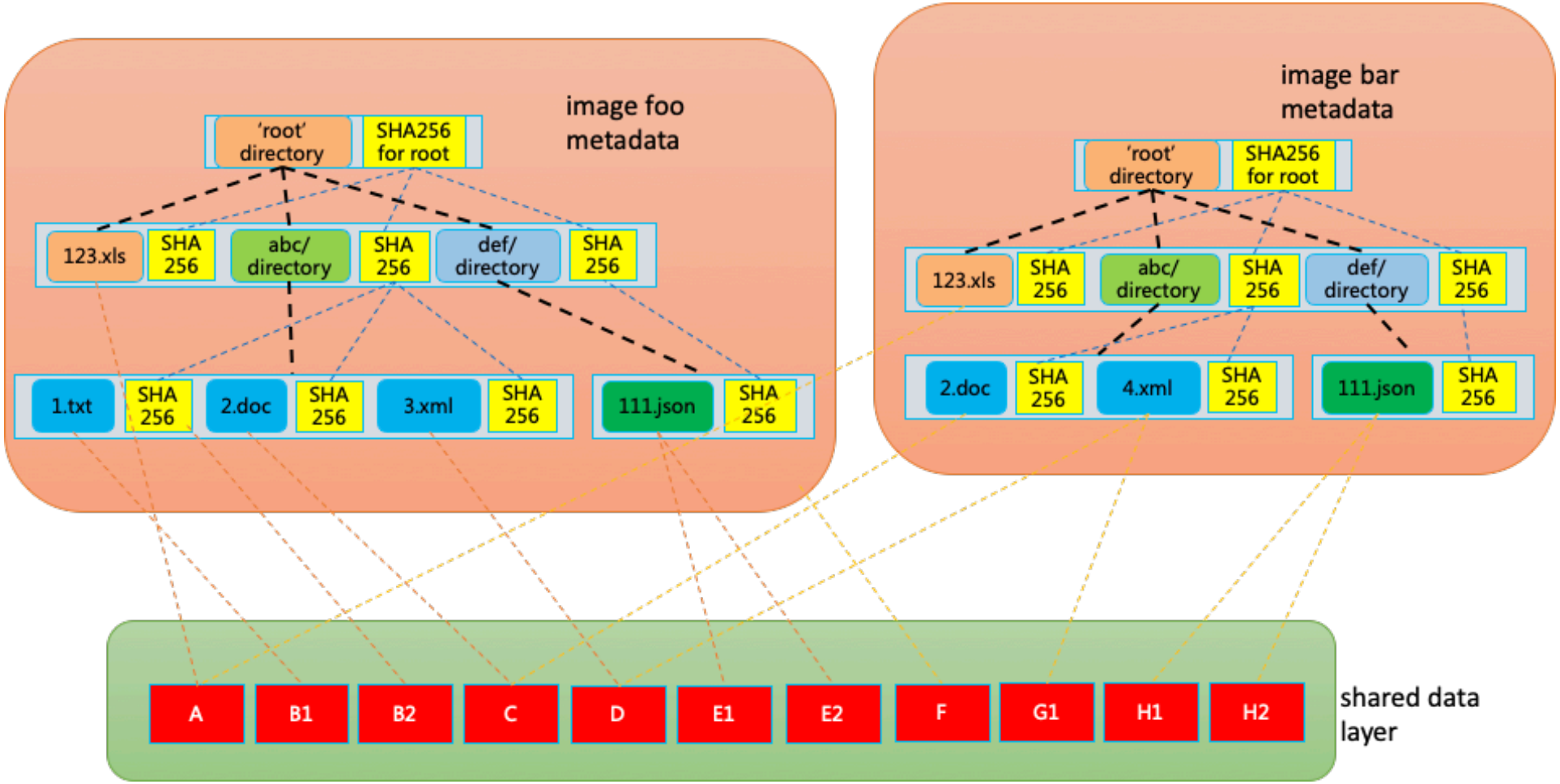
PART FOUR
Nydus

Nydus: The Dragonfly Image Service

We introduces Nydus, a new container image storage and distribution solution, which targets on current image problems and highly meets the ongoing OCIv2 image format requirement

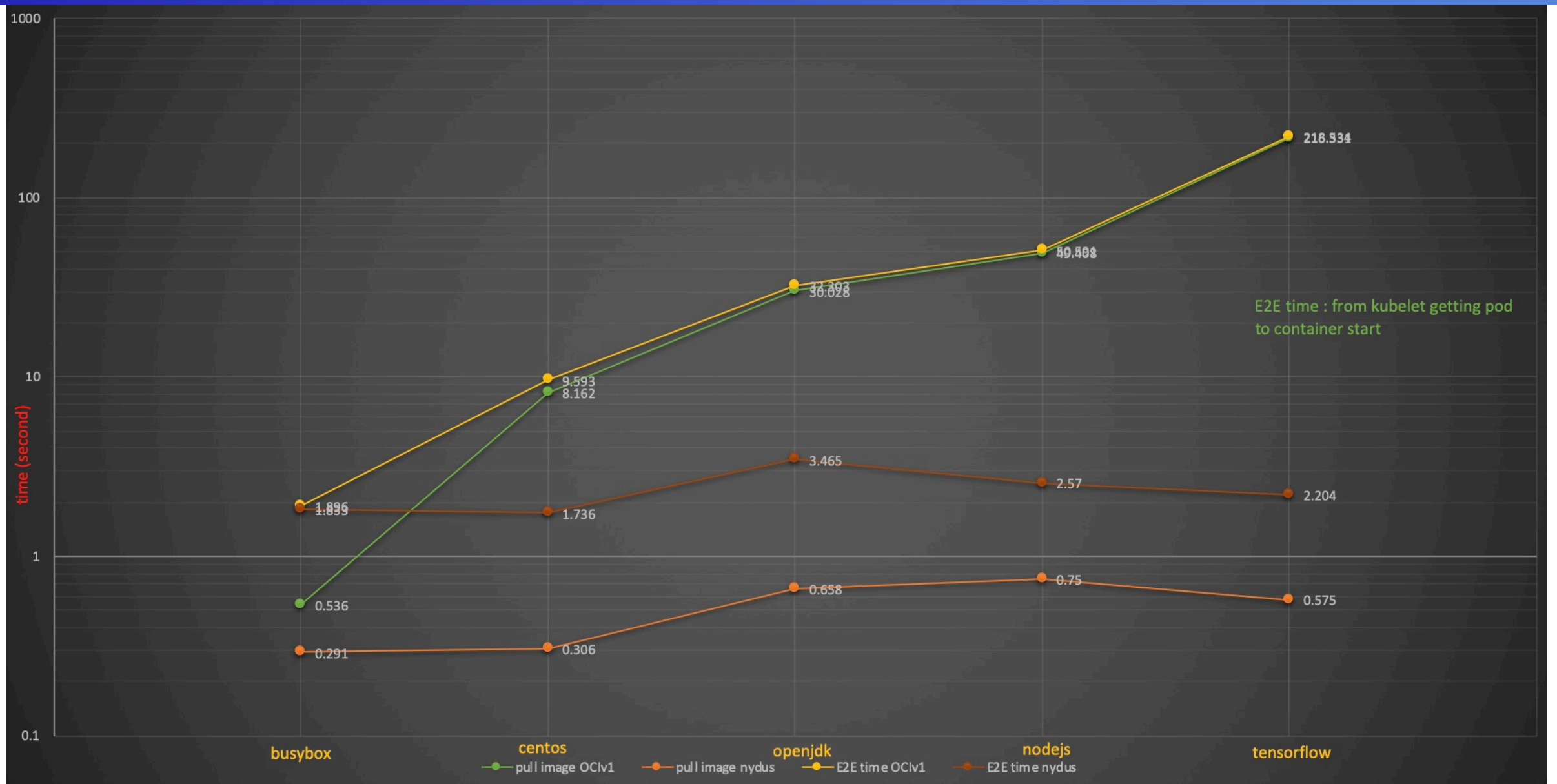


Nydus Image Format



CLOUD NATIVE + OPEN SOURCE
Virtual Summit China 2020

Nydus Benchmark



Future Work

- Open source as a reference implementation for incoming OCIv2 image specification
- Contribute to a new dragonfly p2p distribution service mechanism for image fast-speed lazyloading
- Support more compression/decompression algo, flexible data chunk size and data dedup levels
- Be compatible with different CRI runtimes

CLOUD NATIVE + OPEN SOURCE

Virtual Summit China 2020