



OCI Image Spec & Distribution

Akihiro Suda (@_AkihiroSuda_)
NTT Software Innovation Center

Open Containers Initiative Specifications



- **OCI Runtime Spec**

- How to create container from config JSON and rootfs dir
- Based on Docker libcontainer (now runc)

- **OCI Image Spec**

- How to represent image layers for OCI runtimes
- Based on Docker Image Manifest V2, Schema 2

- **OCI Distribution Spec**

- How to distribute OCI images
- Based on Docker Registry HTTP API

Image layout

- **Merkle DAG structure ensures reproducibility of**
`docker pull foo@sha256:e692418e...`

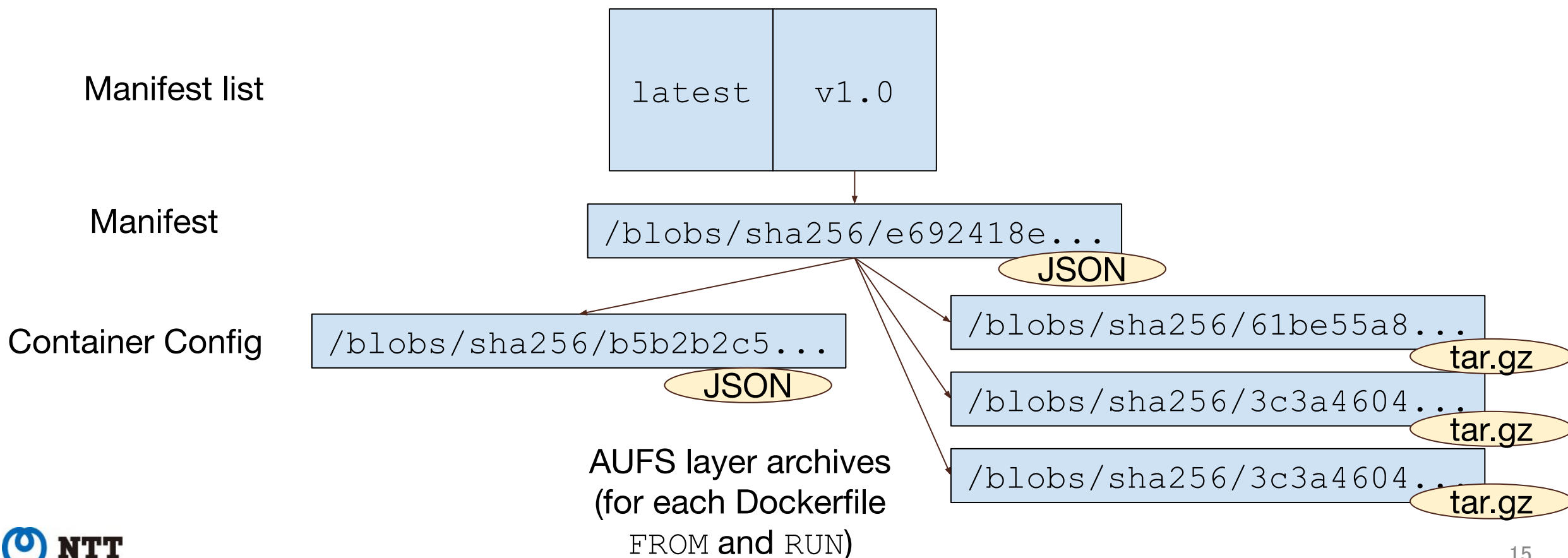


Image layout



- Supports multi-arch (use BuildKit to build)

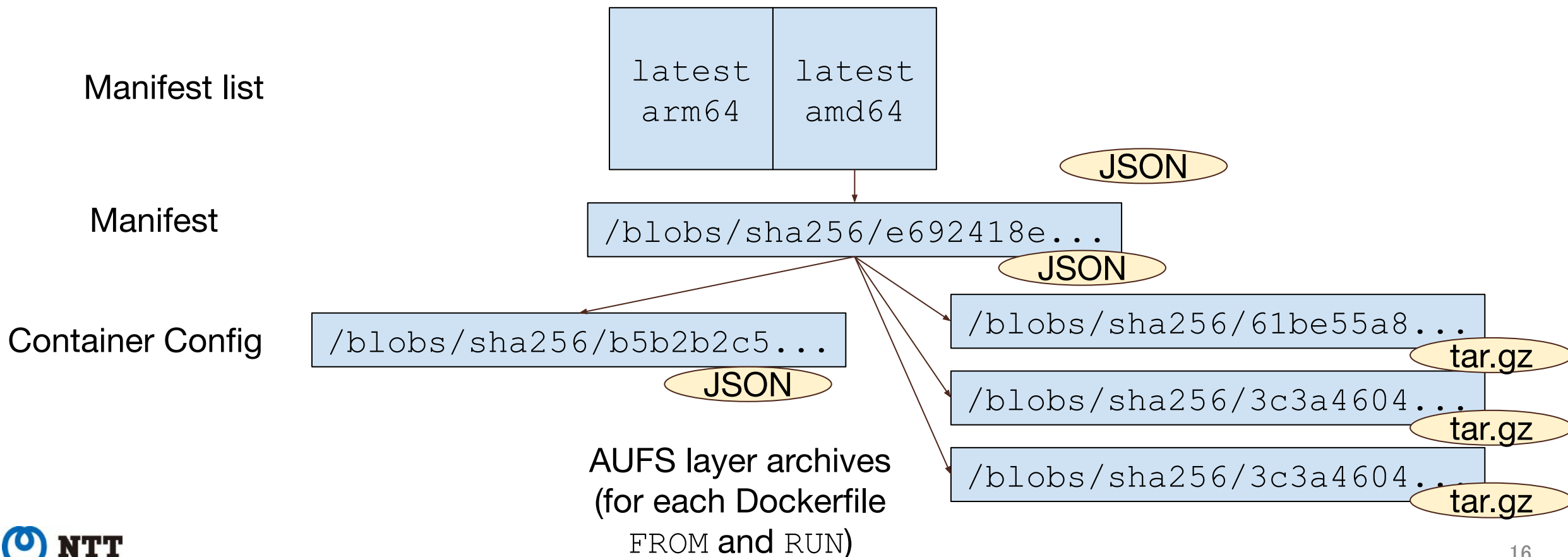
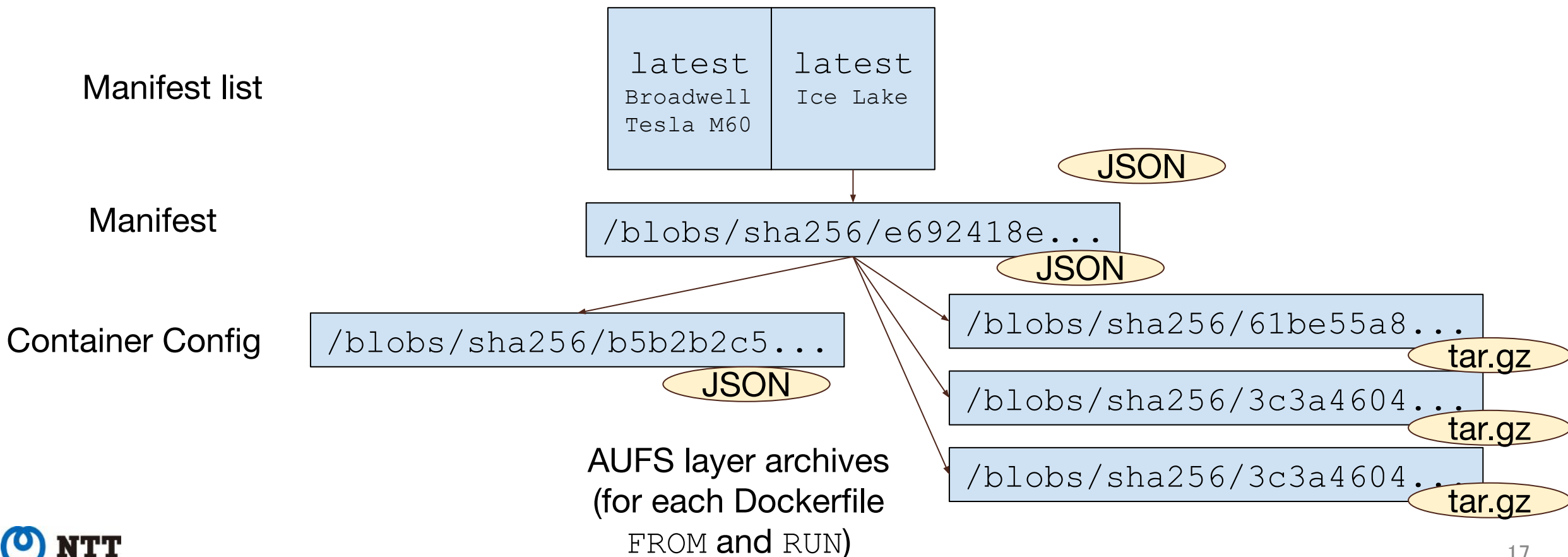


Image layout

- And even multi-microarchitectures via [qnib/metahub](https://metahub.qnib.org)
- <https://metahub.qnib.org>



Post-OCI image format?

- **Issues of current OCI v1**
 - Too coarse deduplication granularity
 - Containers cannot be started until the entire image is pulled
- **An alternative: CernVM-FS**
 - Supports file-level deduplication rather than layer-level
 - Files are lazy-pulled on demand using FUSE
- **Integrating CernVM-FS to containerd is under discussion**

<https://github.com/containerd/containerd/issues/2943>

Post-OCI image format?

- **"OCI v2"** <https://github.com/openSUSE/umoci/issues/256>
 - Much finer deduplication granularity
 - No implementation yet
- **Container Registry Filesystem** <https://github.com/google/crfs>
 - Focus on lazy-pulling CI images
- **IPCS** <https://github.com/hinshun/ipcs>
 - IPFS integration for containerd