

CONTAINER STANDARDS

THE PROBLEM

With no standard, there is no way to automate. Each box is a different size, has different specifications. No ecosystem of tools can form.

Image: Boxes manually loaded on trains and ships in 1921



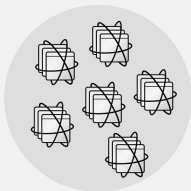
WHY STANDARDS MATTER TO YOU

Click to add subtitle



Protect customer investment

The world of containers is moving very quickly. Protect your investment in training, software, and building infrastructure.



Enable ecosystems of products and tools to form

Cloud providers, software providers, communities and individual contributors can all build tools.



Allow communities with competing interests to work together

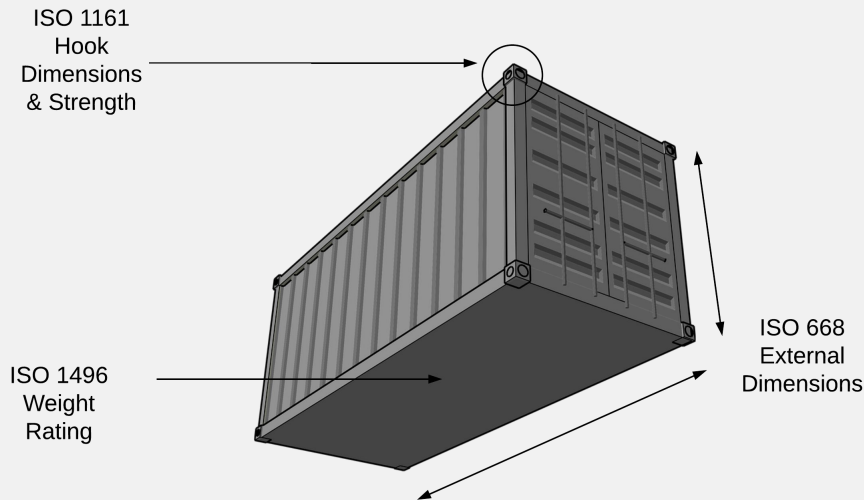
There are many competing interests, but as a community we have common goals.

SIMILAR TO REAL SHIPPING CONTAINERS

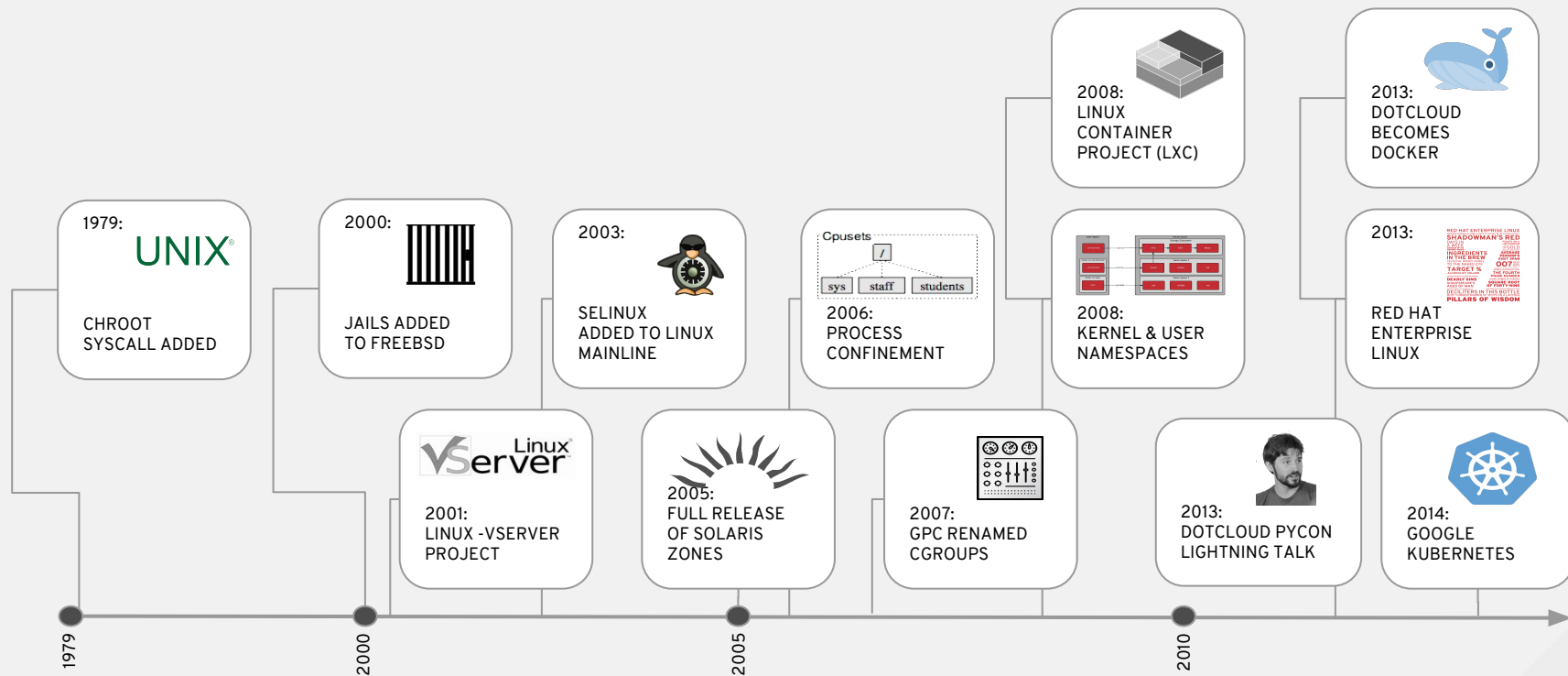
Standards in different places achieve different goals

The analogy is strikingly good. The importance of standards is critical:

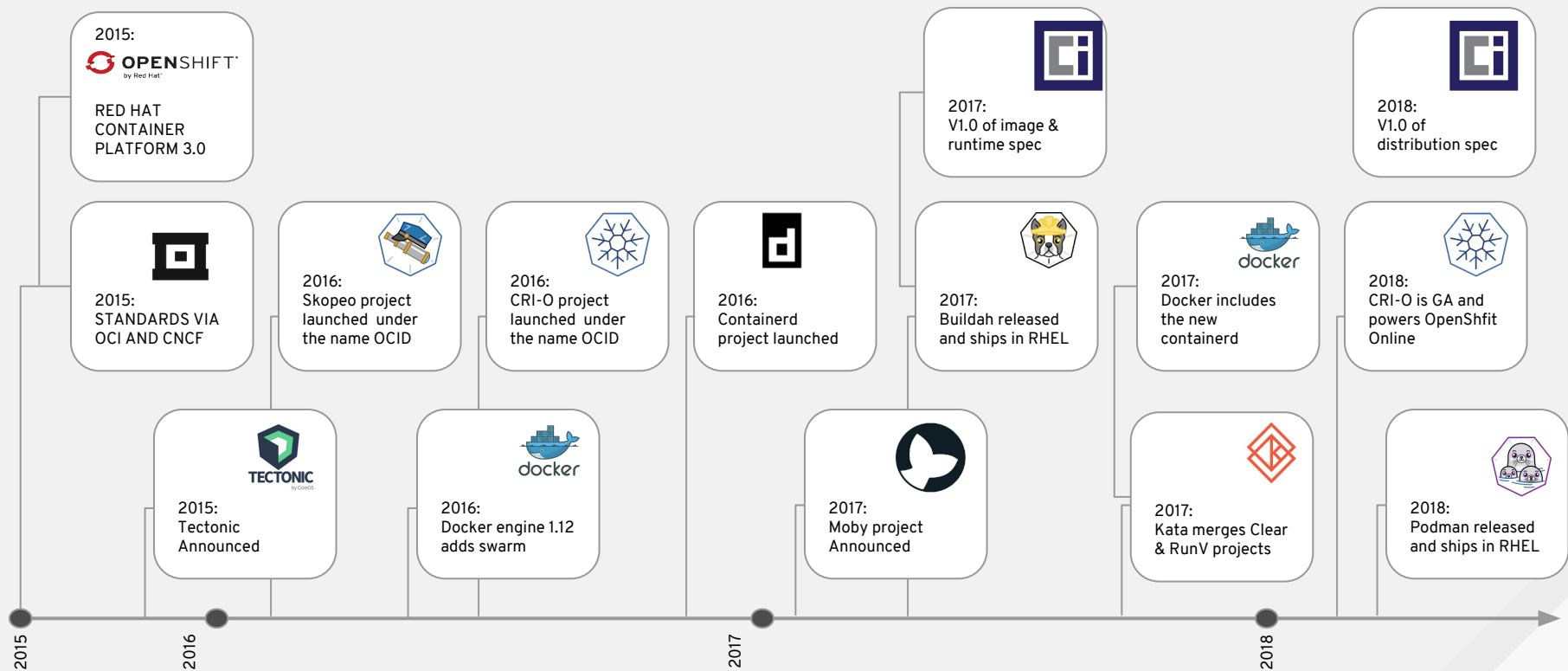
- Failures are catastrophic in a fully automated environments, such as port in Shanghai (think CI/CD)
- Something so simple, requires precise specification for interoperability (Files & Metadata)
- Only way to protect investment in equipment & infrastructure (container orchestration & build processes)



How did we get here?



Where are we going?

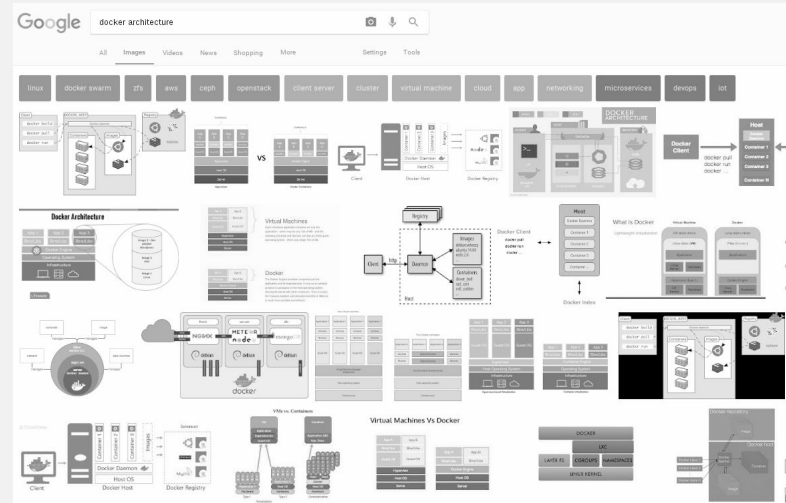


ARCHITECTURE

The Internet is WRONG :-)

Important corrections

- Containers do not run ON docker. Containers are processes - they run on the Linux kernel. Containers are Linux.
- The docker daemon is one of the many user space tools/libraries that talks to the kernel to set up containers



Containers Are Open



Established in June 2015 by Docker and other leaders in the container industry, the OCI currently contains three specifications which govern, building, running, and moving containers.

Standards Are Well Governed



- Governed by The Linux Foundation
- Ecosystem includes:
 - Vendors
 - Cloud Providers
 - Open Source Projects

OVERVIEW OF THE DIFFERENT STANDARDS

Vendor, Community, and Standards Body driven



kubernetes



CNI

Open Containers Initiative (OCI)
Image Specification

Open Containers Initiative (OCI)
Distribution Specification

Open Containers Initiative (OCI)
Runtime Specification

Container Runtime Interface
(CRI)

Container Network Interface
(CNI)

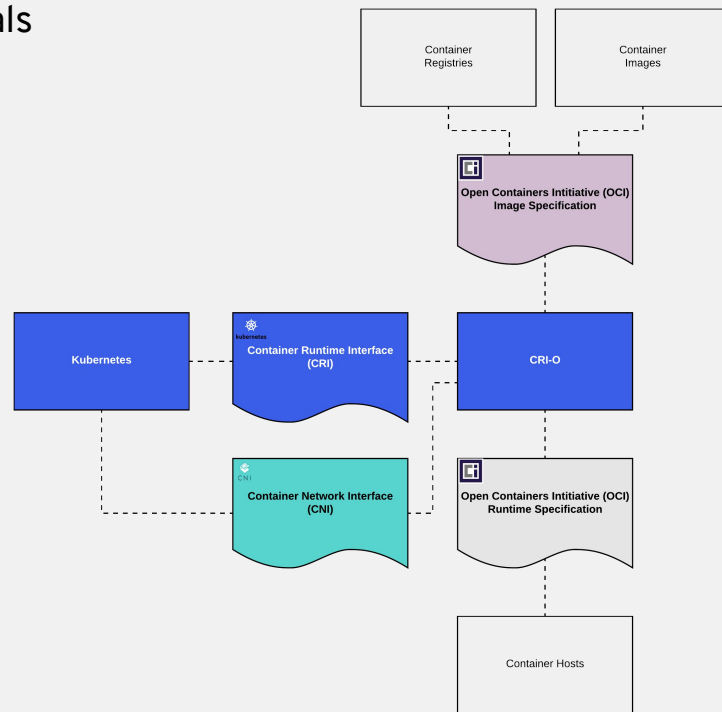
Many different standards

WORKING TOGETHER

Standards in different places achieve different goals

Different standards are focused on different parts of the stack.

- Container Images & Registries
- Container Runtimes
- Container Networking



WHAT ARE CONTAINERS ANYWAY?

Data and metadata

Container images need to express user's intent when built and run.

- How to run
- What to run
- Where to run

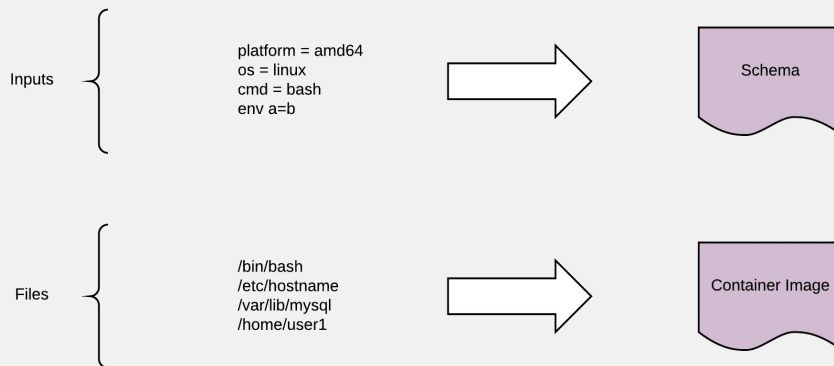
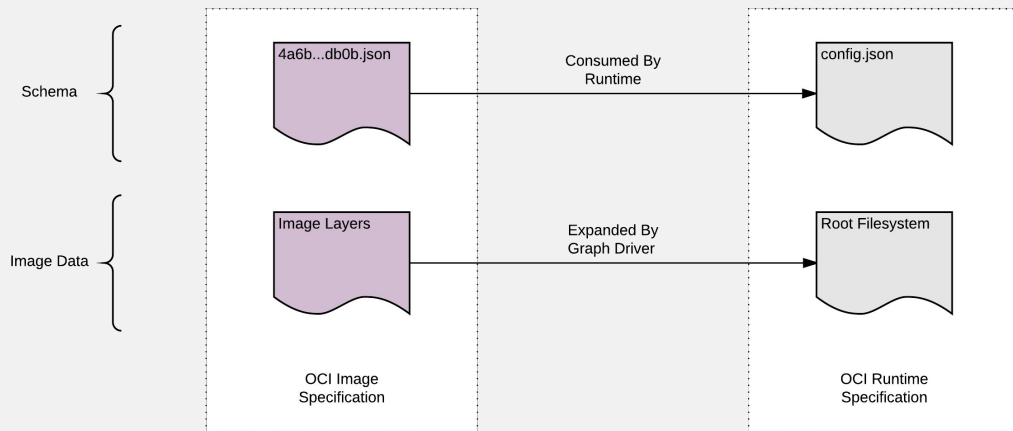


IMAGE AND RUNTIME SPECIFICATIONS

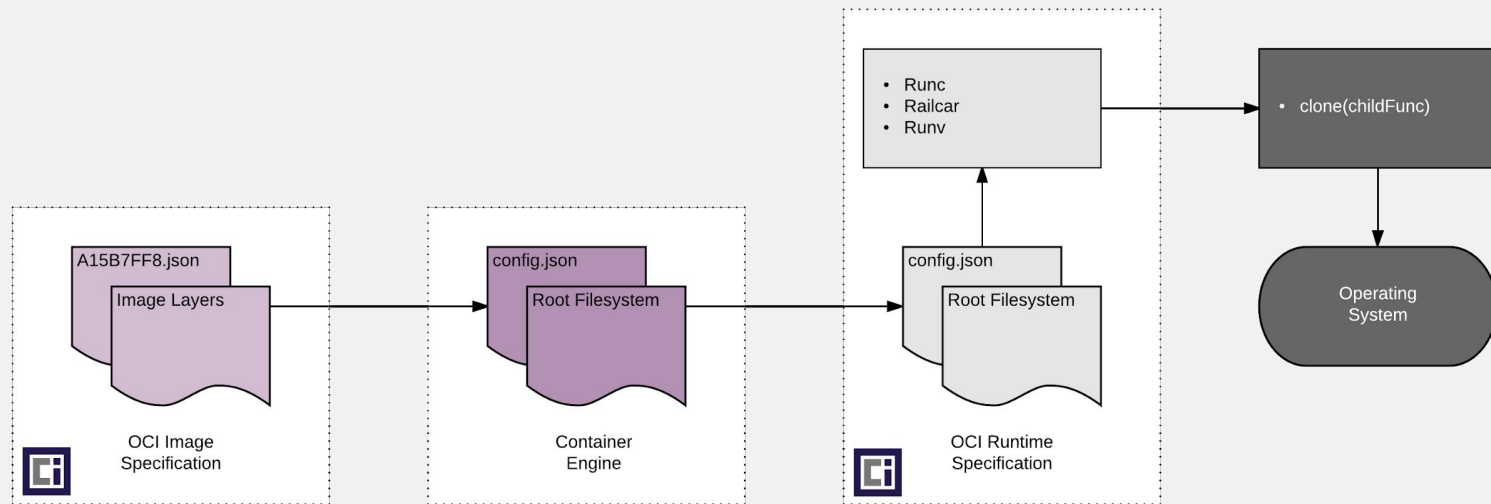
Powerful standards which enable communities and companies to build best of breed tools



Fancy files and fancy processes

WORKFLOW OF CONTAINERS

The building blocks of how a container goes from image to running process



Allows users to build container images with any tool they choose. Different tools are good for different use cases.

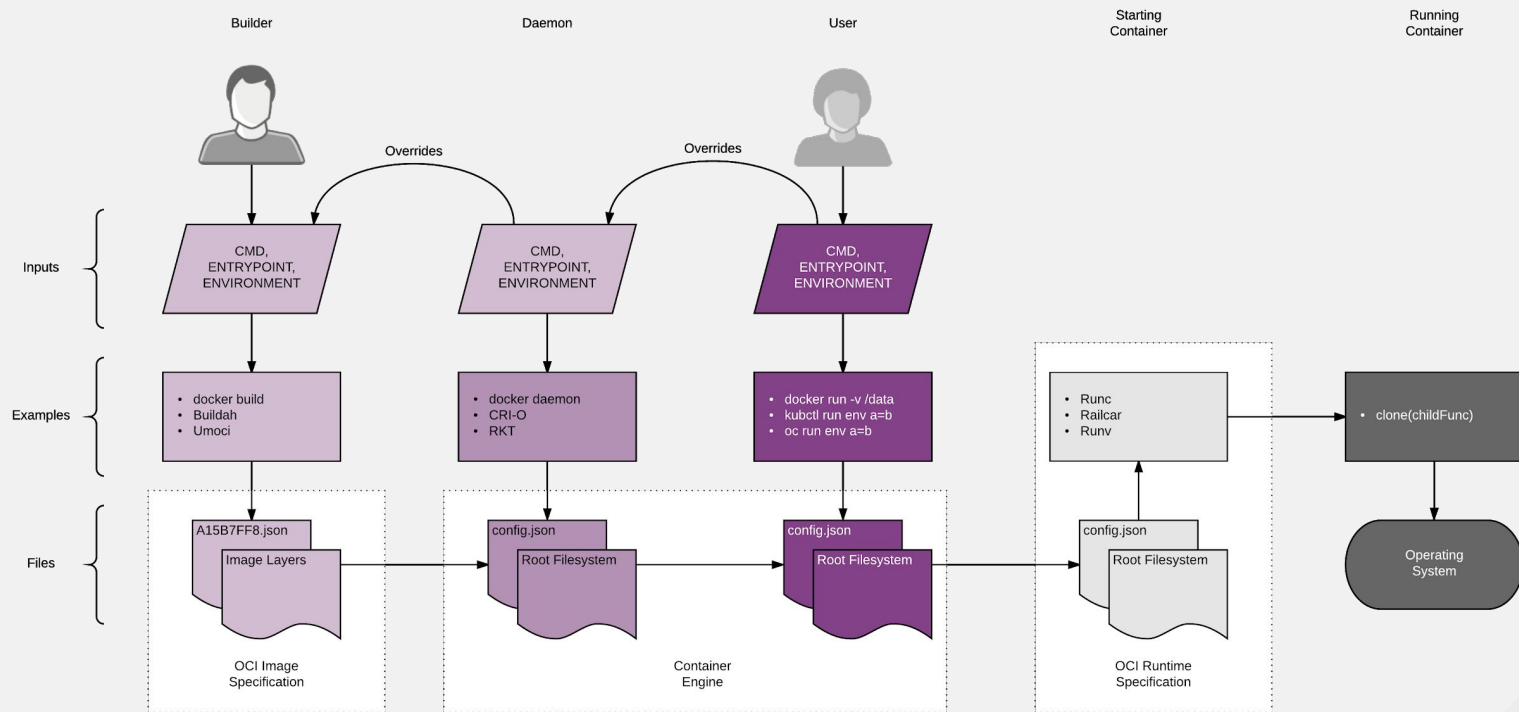
The container engine is responsible for creating the config.json file and unpacking images into a root file system.

OCI compliant runtimes can consume the config.json and root filesystem, and tell the kernel to create a container.

OCI compliant runtimes can be built for multiple operating systems including Linux, Windows, and Solaris

TYING IT ALL TOGETHER

These standards are extremely powerful

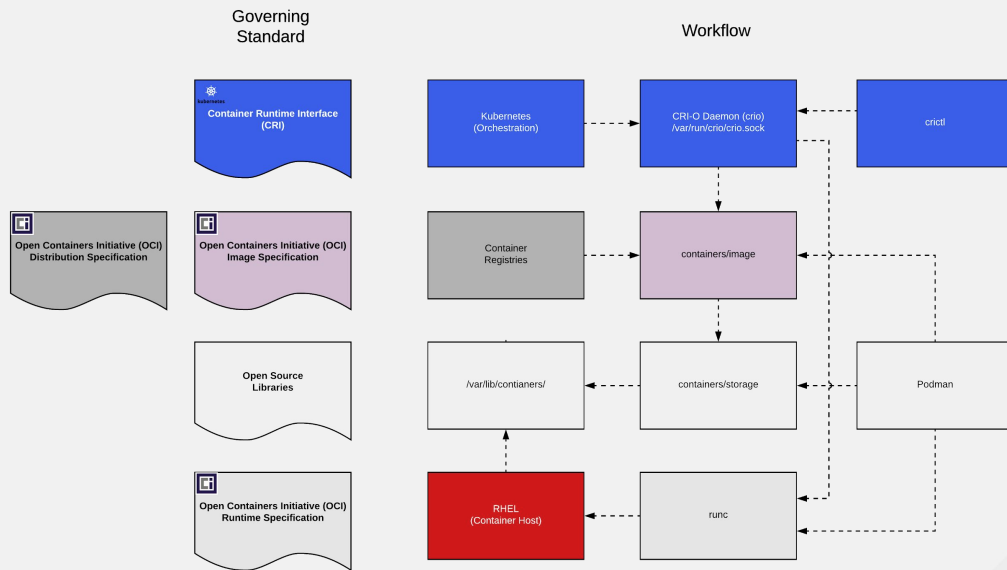


WORKING TOGETHER

Technical example

Different standards are focused on different parts of the stack.

- Tools like crictl use the CRI standard
- Tools like Podman use standard libraries
- Tools like runc are widely used



THE COMMUNITY LANDSCAPE

Open Source, Leadership & Standards

The landscape is made up of committees, standards bodies, and open source projects:

- Docker/Moby
- Kubernetes/OpenShift
- OCI Specifications
- Cloud Native Technical Leadership



Distribution of Kubernetes
containing many different
open source projects

