

COMMON CONTAINER STANDARDS:

PAST, PRESENT & FUTURE

Vincent Batts @vbatts

```
$> finger $(whoami)
```

```
Login: vbatts
```

```
Name: Vincent Batts
```

```
Directory: /home/vbatts
```

```
Shell: /bin/bash
```

```
Such mail.
```

```
Plan:
```

```
OHMAN
```

```
$> id -Gn
```

```
devel opencontainers docker appc redhat golang slackware
```



SO,
WHY, CONTAINERS?



SO, WHY, CONTAINERS?

Single Application



SO, WHY, CONTAINERS?

Single Application
Full System



SO, WHY, CONTAINERS?

Single Application

Full System

But Not a VM



SO, WHY, CONTAINERS?

Single Application

Full System

But Not a VM

Except Maybe a VM



SO, WHY, CONTAINERS?

Single Application

Full System

But Not a VM

Except Maybe a VM

Pods of applications



SO, WHY, CONTAINERS?

Single Application

Full System

But Not a VM

Except Maybe a VM

Pods of applications

Labels of services



SO, WHY, CONTAINERS?

Single Application

Full System

But Not a VM

Except Maybe a VM

Pods of applications

Labels of services

Non-root



SO, WHY, CONTAINERS?

Single Application

Full System

But Not a VM

Except Maybe a VM

Pods of applications

Labels of services

Non-root

Non-root Full systems?



SO, WHY, CONTAINERS?

Single Application

Full System

But Not a VM

Except Maybe a VM

Pods of applications

Labels of services

Non-root

Non-root Full systems?

Desktop Applications



But Wait,

What does "container" mean to you?

But Wait,
What does "container" mean to you?



STANDARDS!

STANDARDS!

Standard

/ˈstændəd/

noun

something used as a measure, norm, or model in comparative evaluations

STANDARDS!

Areas to Standardize:

STANDARDS!

Areas to Standardize:

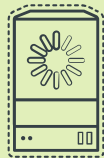
- Packaging



STANDARDS!

Areas to Standardize:

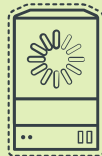
- Packaging
- Runtime



STANDARDS!

Areas to Standardize:

- Packaging
- Runtime
- Networking



STANDARDS!

Areas to Standardize:

- Packaging
- Runtime
- Networking
- Cloud



PAST

PAST PACKAGES



PAST PACKAGES



tar archives

PAST PACKAGES



tar archives

*.deb or *.rpm

zip archives

PAST PACKAGES



tar archives

*.deb or *.rpm

zip archives

tar archives

PAST PACKAGES



gem

tar archives

*.deb or *.rpm

tar archives

zip archives

PAST PACKAGES



gem

tar archives

*.deb or *.rpm

pod

tar archives

zip archives

PAST PACKAGES



gem

tar archives

*.deb or *.rpm

egg

pod

tar archives

zip archives

PAST PACKAGES



gem

tar archives

tar archives

*.deb or *.rpm

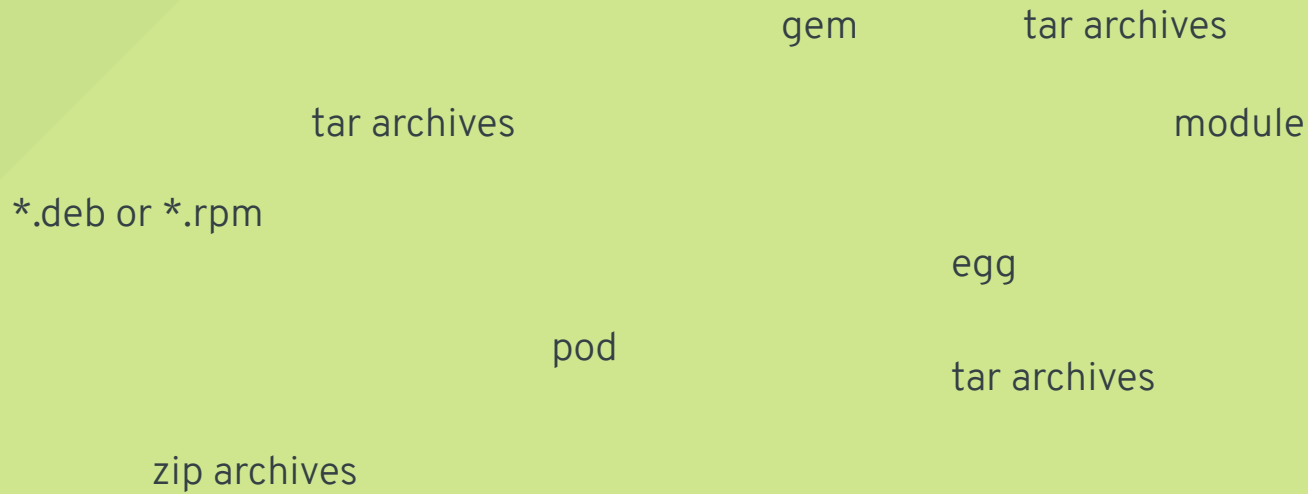
egg

pod

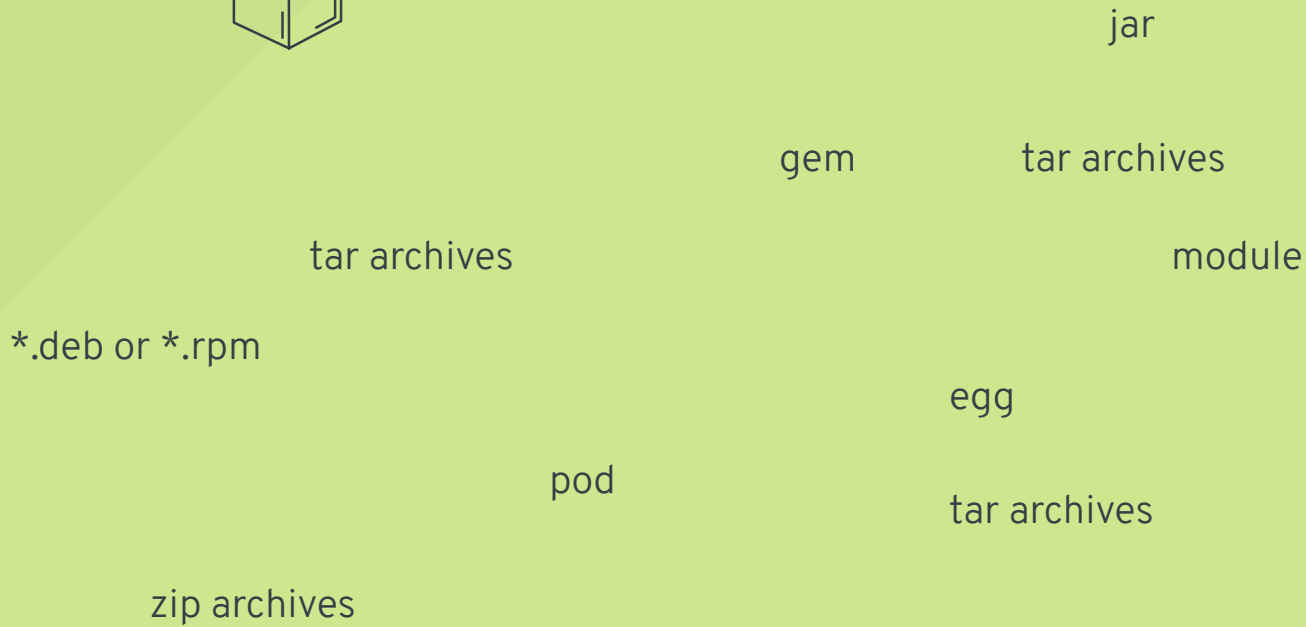
tar archives

zip archives

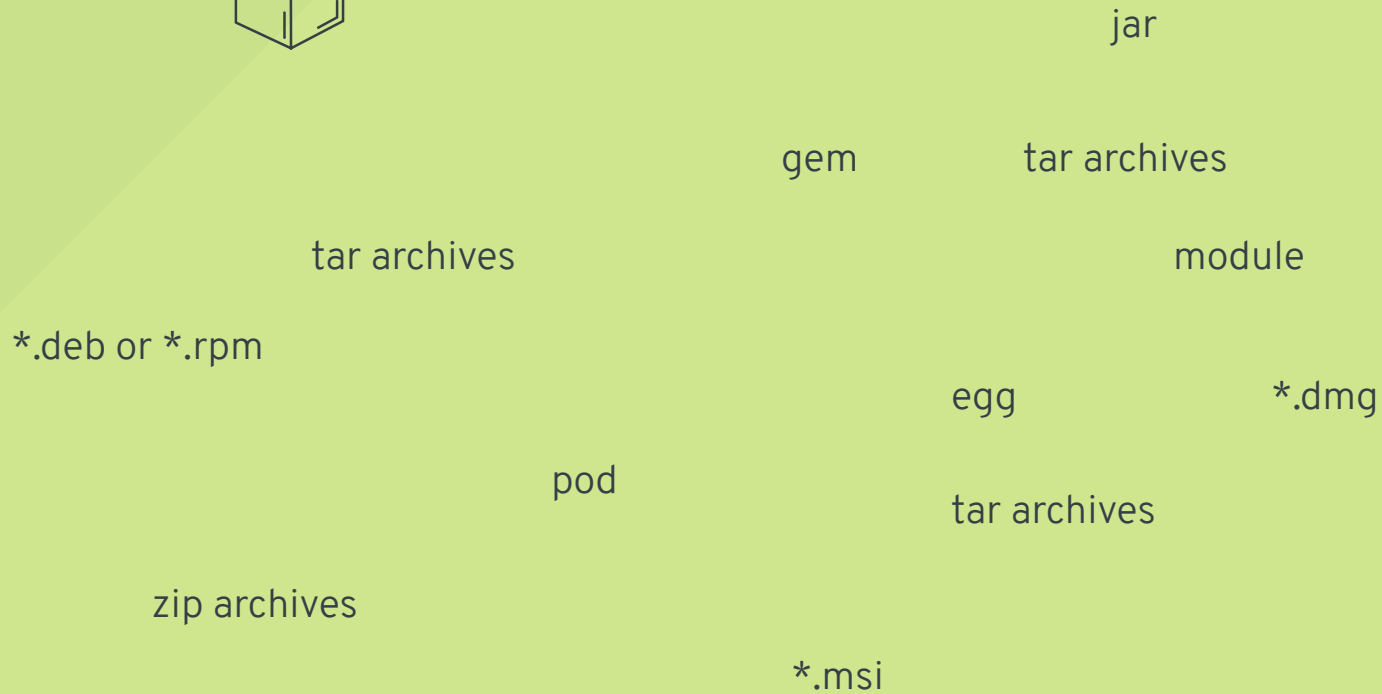
PAST PACKAGES



PAST PACKAGES



PAST PACKAGES



PAST PACKAGES



tar archives

jar

gem

tar archives

module

tar archives

*.deb or *.rpm

egg

*.dmg

pod

tar archives

zip archives

*.msi

PAST RUNTIME



PAST RUNTIME



binaries?

PAST RUNTIME



ELF binaries?

binaries?

PAST RUNTIME

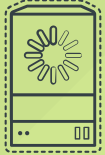


ELF binaries?

binaries?

shell scripts

PAST RUNTIME



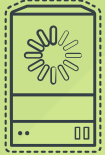
binaries?

ELF binaries?

SysVinit

shell scripts

PAST RUNTIME



binaries?

ELF binaries?

WAR files

SysVinit

shell scripts

PAST RUNTIME



binaries?

so many shell scripts

ELF binaries?

WAR files

SysVinit

shell scripts

PAST NETWORK



PAST NETWORK



Hardware

PAST NETWORK



Hardware

shell scripts + telnet

PAST NETWORK



Hardware

shell scripts + telnet

custom

PAST NETWORK



Hardware

custom

SDN

shell scripts + telnet

PAST CLOUD



PAST CLOUD



REST

PAST CLOUD



SOAP

REST

SOA

PAST CLOUD



SOAP

APIs of APIs

REST

SOA

PRESENT RUNTIME

LXC

- 2008
- lxc specific config

Docker

- 2013
- Docker specific config and APIs

PRESENT

RUNTIME

Application Container Spec (github.com/appc/spec)

- December 2014
- Specification
- App Container Executor (ACE)
- Several implementations, with rkt as the flagship

PRESENT

RUNTIME

OpenContainer Runtime-Spec
(github.com/opencontainers/runtime-spec)

- June 2015
- Specification
- Several Implementations, with runc as flagship
- Currently v1.0.0-rc2

PRESENT NETWORK

Container Networking Interface

([CNI - github.com/container networking/cni](https://github.com/container networking/cni))

- December 2014
- Specification and Library
- Simple to integrate with a process based workflow
- Used by RKT, kubernetes, OpenShift, Kurma, Cloud Foundry, RancherOS, usable with runC, and more

PRESENT NETWORK

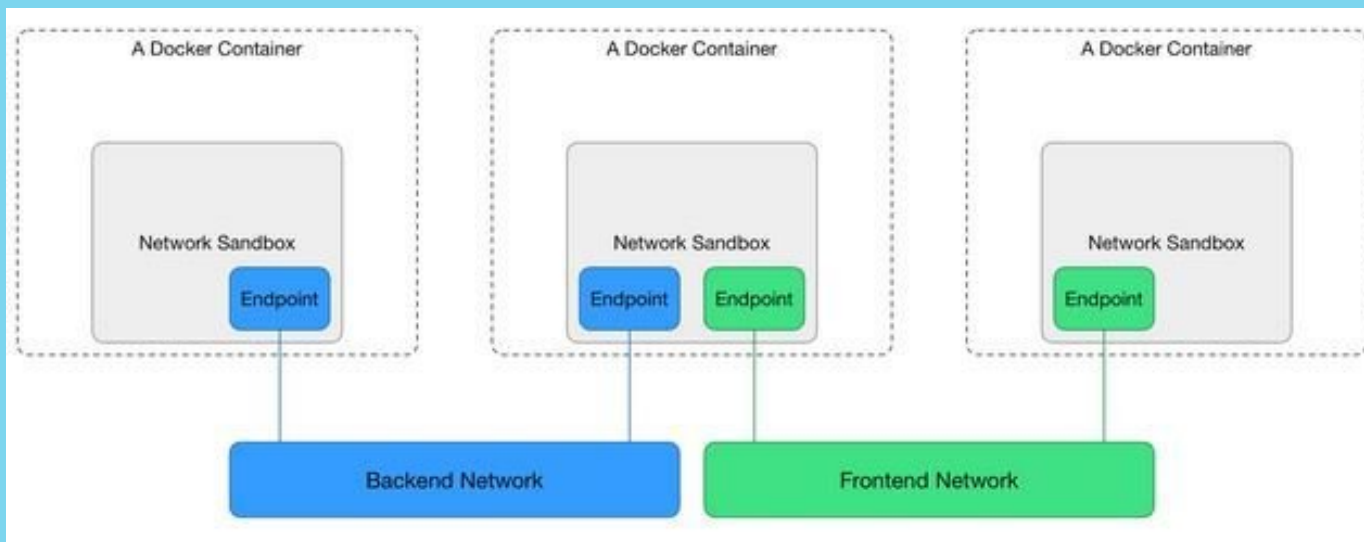
Container Network Model
(CNM - Docker libnetwork)

- April 2015
- Hard to integrate into existing system
- Used by Docker Engine

PRESENT NETWORK

Container Network Model (CNM - Docker libnetwork)

- April 2015
- Hard to integrate into existing system
- Used by Docker Engine



PRESENT PACKAGING

Docker Image

- Docker specific format
- Tight coupling with daemon version
- Signing requires Docker notary integration
- Image naming is Docker specific and bound to registries

PRESENT PACKAGING

Application Container Spec (github.com/appc/spec)

- December 2014
- App Container Image (ACI)
- A number of independent tooling
- Addresses Fully-Qualified-Naming, image discovery, signing, content addressability, and versioned schema

PRESENT PACKAGING

OpenContainer Image-Spec
(github.com/opencontainers/image-spec)

- April 2016
- Pulled from Docker-1.10 and Registry v2 format
- Content addressability
- Signable. Possibility to have naming and discovery.
- Currently releasing v1.0.0-rc1
- Gaining support from rkt, flatpak, skopeo, cri-o and more

PRESENT CLOUD

Cloud Native Computing Foundation
(<https://cncf.io>)

- Kubernetes orchestration donated by Google
- Prometheus monitoring donated

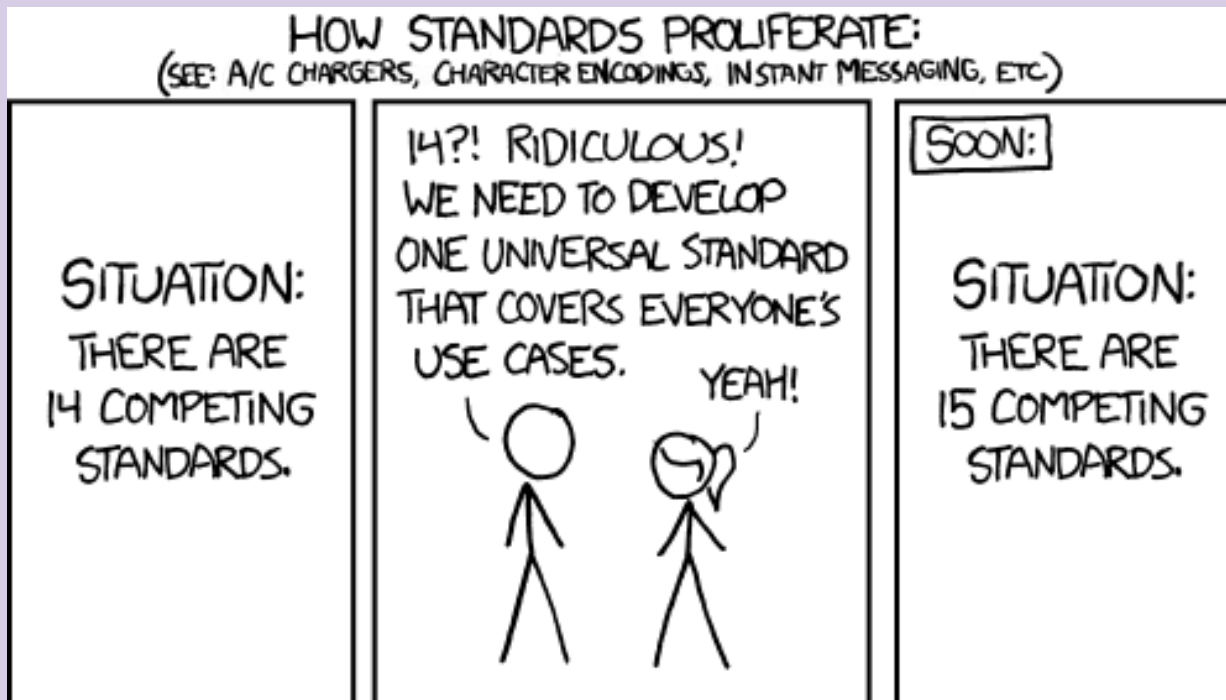
Why More Standards?!

Why More Standards?!

Really great question. Thought you might ask ...

Why More Standards?!

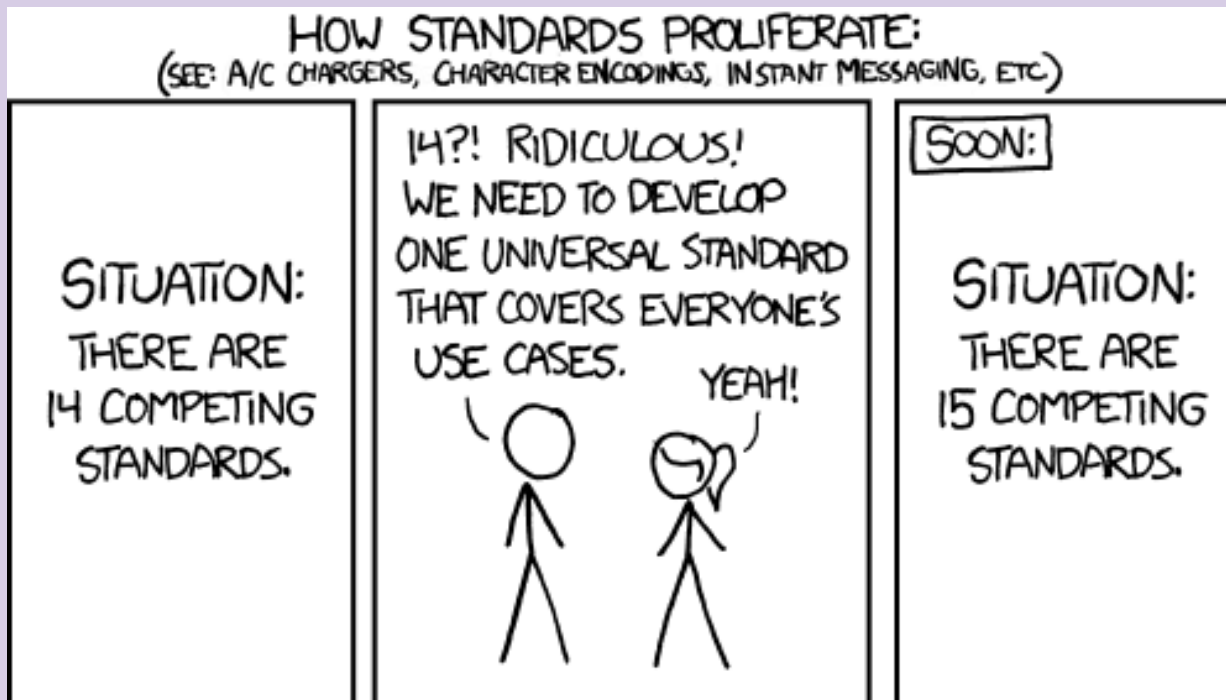
Really great question. Thought you might ask ...



Why More Standards?!

Really great question. Thought you might ask ...

The package wars of deb vs rpm set back the broad adoption of Linux



FUTURE

FUTURE

Verification and certification of integrations/implementations

FUTURE

Continued adoption

Verification and certification of integrations/implementations

FUTURE

Continued adoption

Verification and certification of integrations/implementations

Tooling to further distribution and discovery

Call to Action!

Call to Action!

Define your use-cases **first**

Call to Action!

Define your use-cases **first**

Get involved in the conversations

Call to Action!

Define your use-cases **first**

Get involved in the conversations

Ensure your container integration touchpoint stay generic,
to avoid lock-in to a particular platform.

Call to Action!

Define your use-cases **first**

Get involved in the conversations

Ensure your container integration touchpoint stay generic,
to avoid lock-in to a particular platform.

PoC tooling for your integration

Call to Action!

Define your use-cases **first**

Get involved in the conversations

Ensure your container integration touchpoint stay generic,
to avoid lock-in to a particular platform.

PoC tooling for your integration



VINCENT BATTS

@VBATTS | VBATTS@REDHAT.COM

THANKS!