COMMON CONTAINER STANDARDS:

PAST, PRESENT & FUTURE

Vincent Batts @vbatts

```
$> finger $(whoami)
Login: vbatts                    Name: Vincent Batt
Directory: /home/vbatts          Shell: /bin/bash
Such mail.
Plan:
OHMAN
```
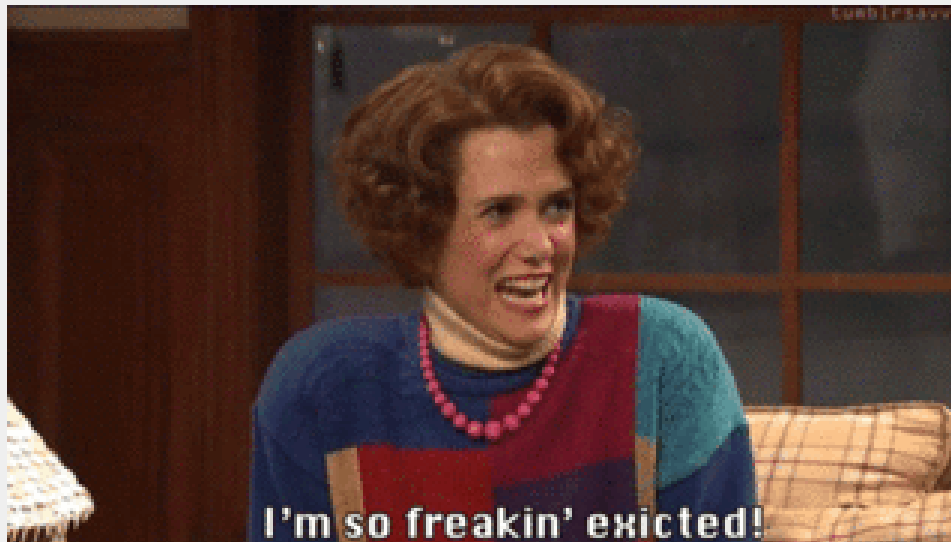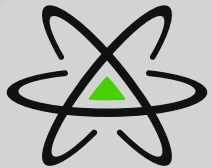

I'm so freakin' exicted!

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
        Desktop Applications
```

SO,
WHY, CONTAINERS?

```
Single Application
     Full System
          But Not a VM
               Except Maybe a VM
          Pods of applications
     Labels of services
Non-root
     Desktop Applications
          OMG AND CATS
```

SO,
WHY, CONTAINERS?

Single Application
    Full System
        But Not a VM
            Except Maybe a VM
        Pods of applications
    Labels of services
Non-root
    Desktop Applications
        OMG AND CATS

https://www.flickr.com/photos/27549668@N03/

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

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

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

STANDARDS!

STANDARDS!

Standard

/ˈstandəd/

noun

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

STANDARDS!

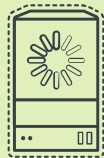Areas to Standardize:

STANDARDS!

Areas to Standardize:

- Packaging
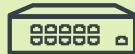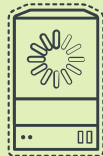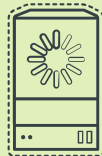
STANDARDS!

Areas to Standardize:

- Packaging
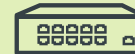
- Runtime

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

tar archives

zip 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
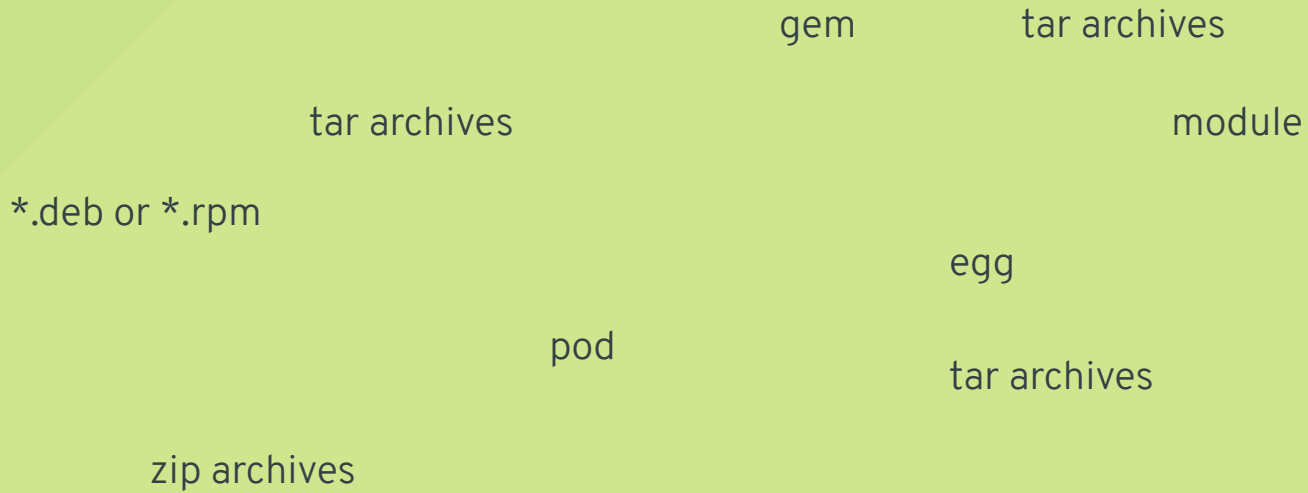
gem   tar archives

tar archives

*.deb or *.rpm

egg

pod

tar archives

zip archives

PAST
PACKAGES

gem   tar archives

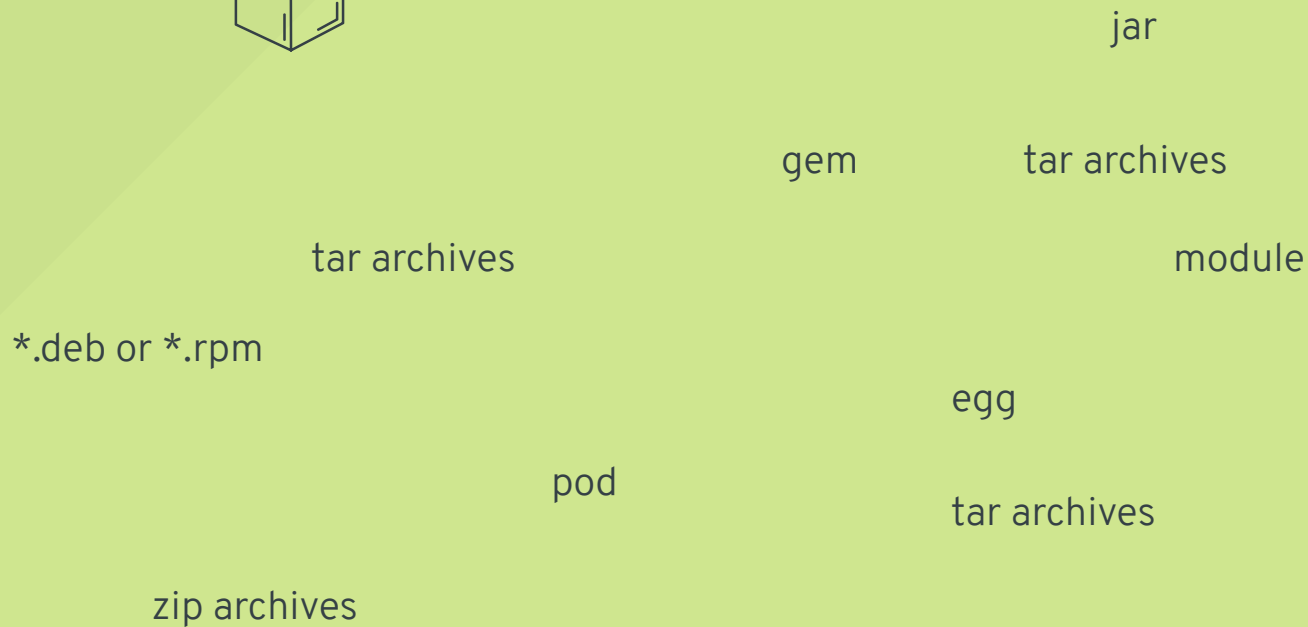tar archives   module

*.deb or *.rpm

egg

pod

tar archives

zip archives

PAST
PACKAGES

jar

gem   tar archives

tar archives   module

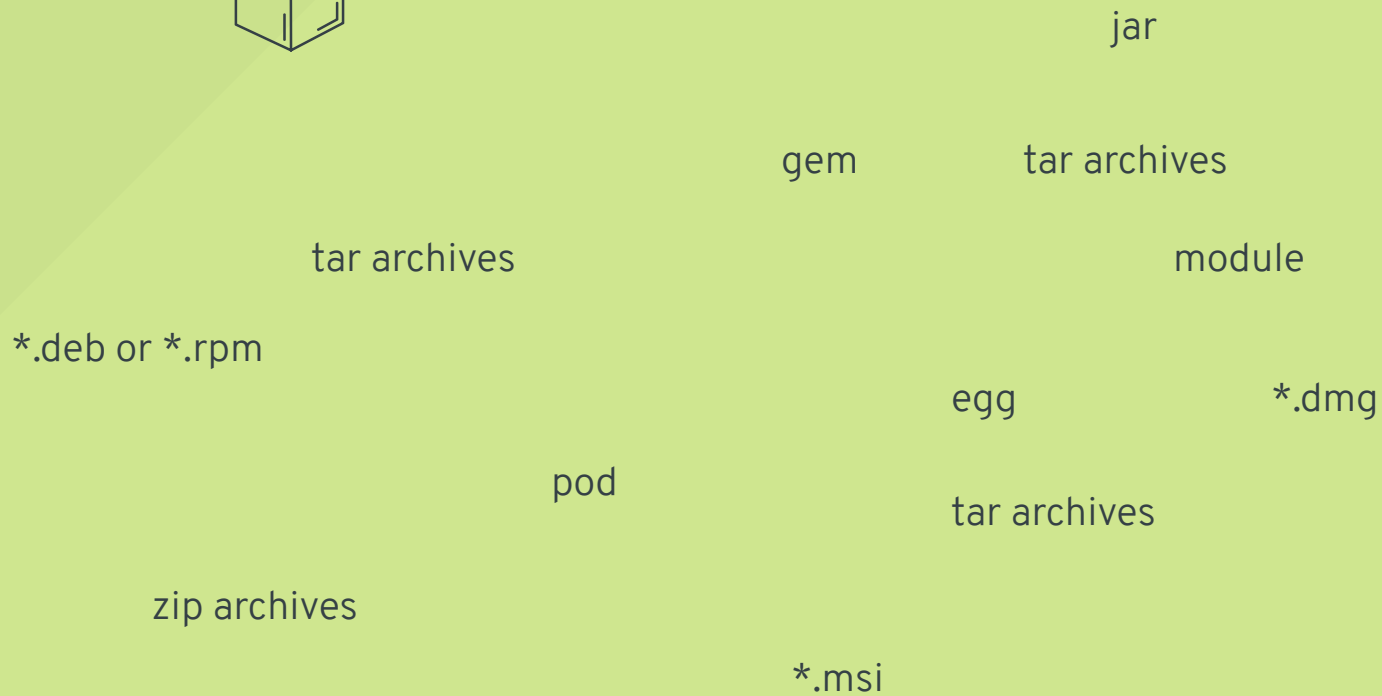*.deb or *.rpm

egg

pod

tar archives

zip archives

PAST
PACKAGES

jar

gem   tar archives

tar archives   module

*.deb or *.rpm

egg   *.dmg

pod   tar archives

zip archives

*.msi

PAST
PACKAGES

tar archives

jar

gem   tar archives

tar archives   module

*.deb or *.rpm

egg   *.dmg

pod

tar archives
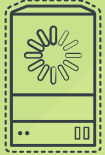
zip archives

*.msi

PAST

RUNTIME

PAST
RUNTIME

binaries?

PAST

RUNTIME

binaries?

ELF binaries?

PAST
RUNTIME

binaries?

ELF 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

custom

Hardware

shell scripts + telnet

PAST
NETWORK

custom

Hardware

SDN

shell scripts + telnet

PAST
CLOUD

PAST
CLOUD

REST

PAST
CLOUD

SOAP

REST

PAST
CLOUD

SOAP

APIs of APIs

REST

PRESENT
CLOUD

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

Kubernetes offered as seed tecnology

PRESENT
NETWORK

Container Networking Interface

(CNI - github.com/containernetworking/cni)

PRESENT
NETWORK

Container Networking Interface

(CNI - github.com/containernetworking/cni)

- December 2014

PRESENT
NETWORK

Container Networking Interface

(CNI - github.com/containernetworking/cni)

- December 2014
- Specification

PRESENT
NETWORK

Container Networking Interface

(CNI - github.com/containernetworking/cni)

- December 2014
- Specification
- Simple to integrate with a process based workflow

PRESENT
NETWORK

Container Networking Interface

(CNI - github.com/containernetworking/cni)

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

PRESENT
NETWORK

Container Network Model

(CNM - Docker libnetwork)
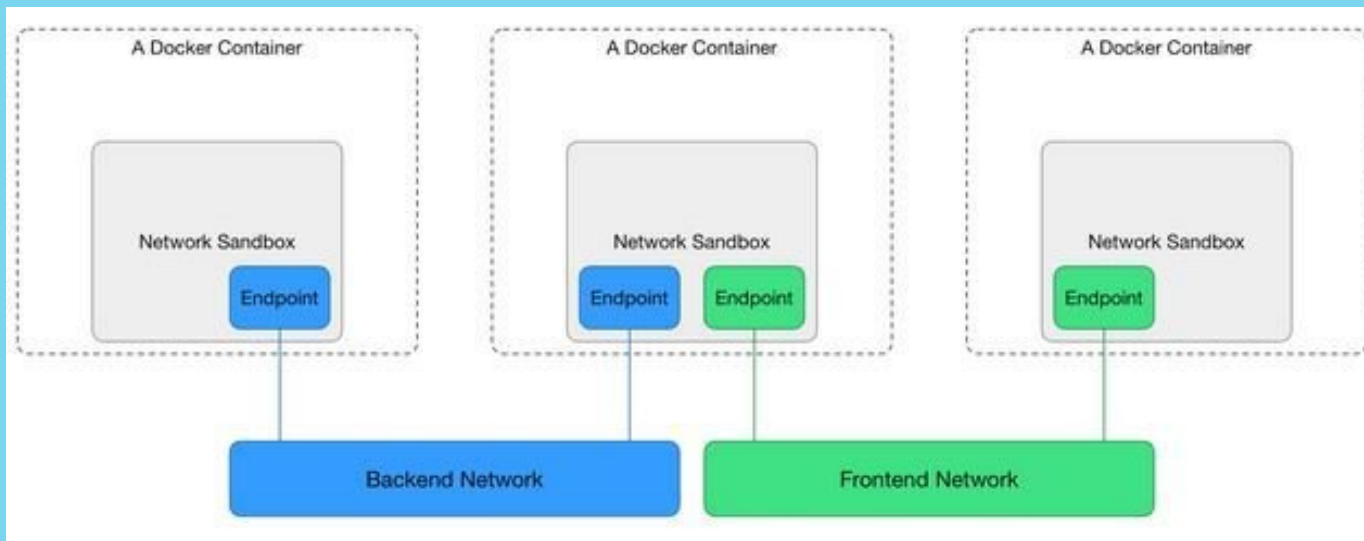
PRESENT
NETWORK

Container Network Model

(CNM - Docker libnetwork)

- April 2015

PRESENT
NETWORK

Container Network Model
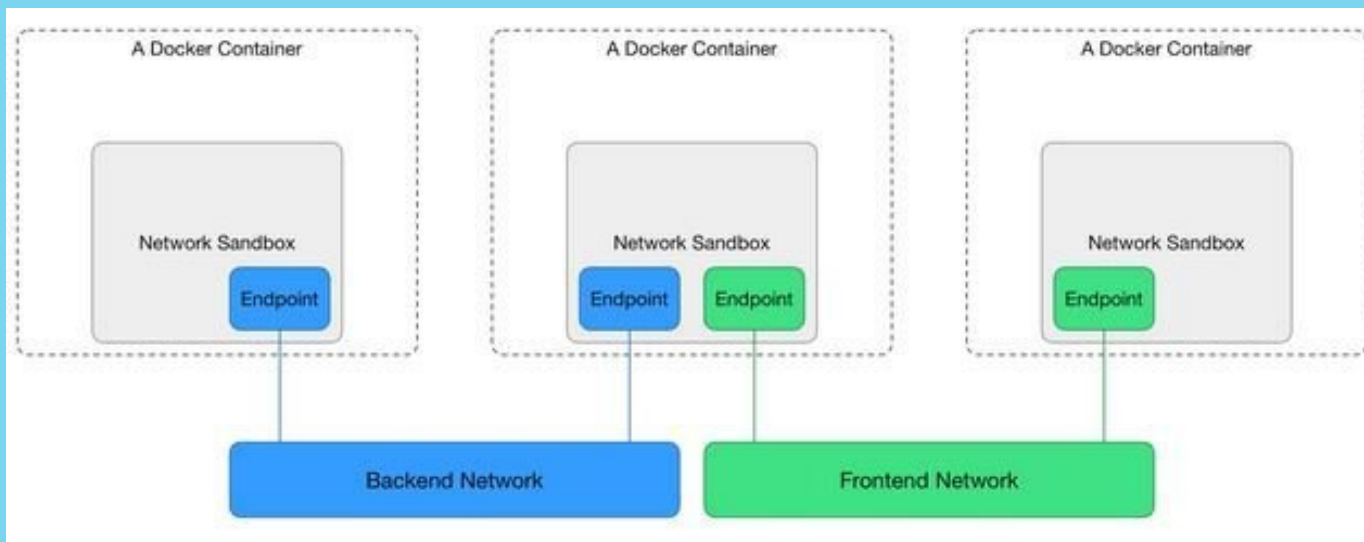
(CNM - Docker libnetwork)

- April 2015
- Hard to integrate into existing system

PRESENT
NETWORK

Container Network Model

(CNM - Docker libnetwork)

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

PRESENT
RUNTIME

PRESENT
RUNTIME

LXC

Docker

PRESENT
RUNTIME

LXC

- 2008

Docker

PRESENT
RUNTIME

LXC

- 2008
- lxc specifc config

Docker

PRESENT
RUNTIME

LXC
- 2008
- lxc specifc config

Docker
- 2013

PRESENT
RUNTIME

LXC
- 2008
- lxc specifc config

Docker
- 2013
- Docker specifc config and APIs

PRESENT
RUNTIME

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

PRESENT
RUNTIME

Application Container Spec (github.com/appc/spec)
- December 2014

PRESENT
RUNTIME

Application Container Spec (github.com/appc/spec)
- December 2014

- App Container Executor (ACE)

PRESENT
RUNTIME

Application Container Spec (github.com/appc/spec)
- December 2014

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

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)

PRESENT
RUNTIME

OpenContainer Runtime-Spec
(github.com/opencontainers/runtime-spec)
- June 2015

PRESENT
RUNTIME

OpenContainer Runtime-Spec
(github.com/opencontainers/runtime-spec)
- June 2015

- Several Implementations, with runc as flagship

PRESENT
RUNTIME

OpenContainer Runtime-Spec
(github.com/opencontainers/runtime-spec)
- June 2015
- Specification
- Several Implementations, with runc as flagship

PRESENT
RUNTIME

OpenContainer Runtime-Spec
(github.com/opencontainers/runtime-spec)
- June 2015
- Specification
- Several Implementations, with runc as flagship
- planned v1.0 in June 2016

PRESENT
PACKAGING

Docker Image

PRESENT
PACKAGING

Docker Image

- Docker specific format

PRESENT
PACKAGING

Docker Image

- Docker specific format
- Tight coupling with daemon version

PRESENT
PACKAGING

Docker Image

- Docker specific format
- Tight coupling with daemon version
- Signing requires Docker notary integration

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)

PRESENT
PACKAGING

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

- December 2014

PRESENT
PACKAGING

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

- December 2014

- A number of independent tooling

PRESENT
PACKAGING

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

- December 2014
- App Container Image (ACI)
- A number of independent tooling

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 addressibility, and versioned schema

PRESENT
PACKAGING

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

PRESENT
PACKAGING

OpenContainer Image-Spec
(github.com/opencontainers/image-spec)
- April 2016

PRESENT
PACKAGING

OpenContainer Image-Spec
(github.com/opencontainers/image-spec)
- April 2016
- Pulled from Docker-1.10 and Registry v2 format

PRESENT
PACKAGING

OpenContainer Image-Spec
(github.com/opencontainers/image-spec)
- April 2016
- Pulled from Docker-1.10 and Registry v2 format
- Content addressibility

PRESENT
PACKAGING

OpenContainer Image-Spec
(github.com/opencontainers/image-spec)
- April 2016
- Pulled from Docker-1.10 and Registry v2 format
- Content addressibility
- Signable. Possibility to have naming and discovery.

PRESENT
PACKAGING

OpenContainer Image-Spec
(github.com/opencontainers/image-spec)
- April 2016
- Pulled from Docker-1.10 and Registry v2 format
- Content addressibility
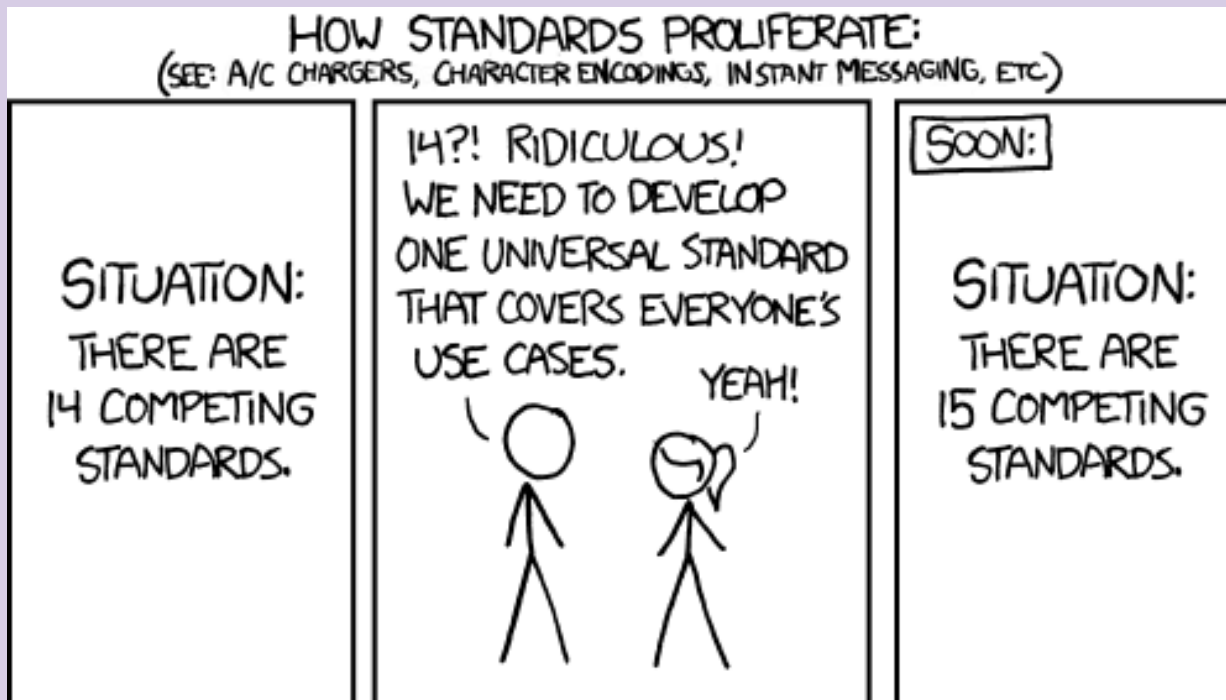- Signable. Possibility to have naming and discovery.
- planned v1.0 in June 2016

Why More Standards?!

Why More Standards?!

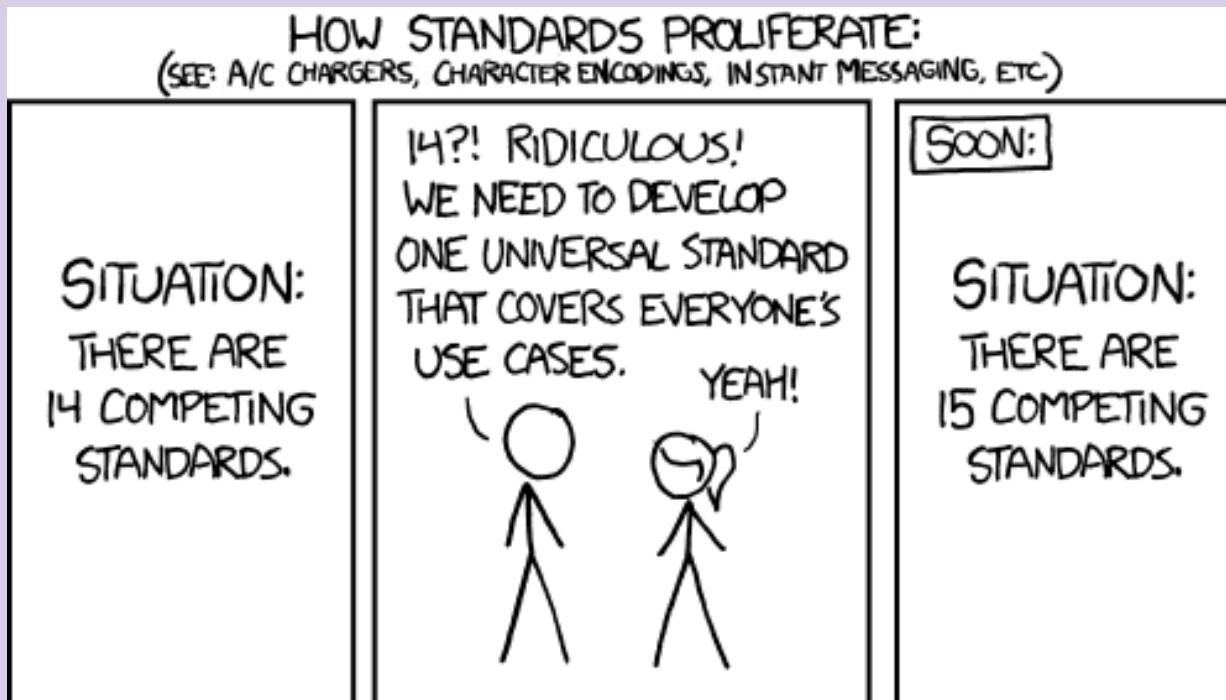Really great question. Thought you might ask ...

Why More Standards?!

Really great question. Thought you might ask ...

xkcd.com/927

Why More Standards?!

Really great question. Thought you might ask ...

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

xkcd.com/927

FUTURE

FUTURE

Broad consensus on v1 and forward

FUTURE

Broad consensus on v1 and forward

Portability of integrations

FUTURE

Broad consensus on v1 and forward

Portability of integrations

Perhaps, industry standards for CAS filesystems, and mapping to content publisher Fully-Qualified-Name

Call to Action!

Call to Action!

Define <u>your</u> use-cases **first**

Call to Action!

Define <u>your</u> use-cases **first**

Get involved in the conversations

Call to Action!

Define <u>your</u> 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 <u>your</u> 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 <u>your</u> 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!