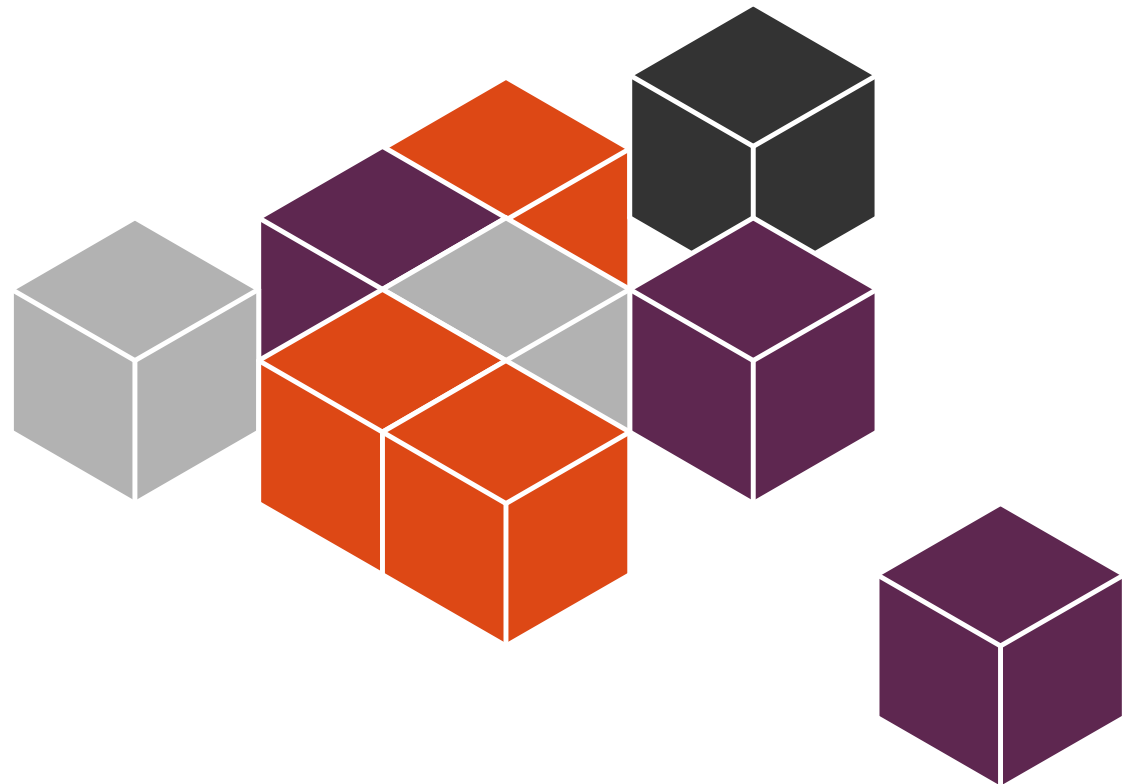


# Snap your App!



**Ted Gould**  
ted@canonical.com  
@tedjgould  
Texas Linux Fest  
21 Aug 2015

# The Plan

- Give instructions on getting Snappy installed
- Talk about Snappy
- Give instructions on downloading demo
- Talk about building apps for Snappy
- Build an app for Snappy

# QEMU (Virtual Machine Manager)



```
$ sudo apt-get install  
qemu-kvm ubuntu-virt-  
mgmt python-spice-  
client-gtk
```

```
$ kvm-ok
```

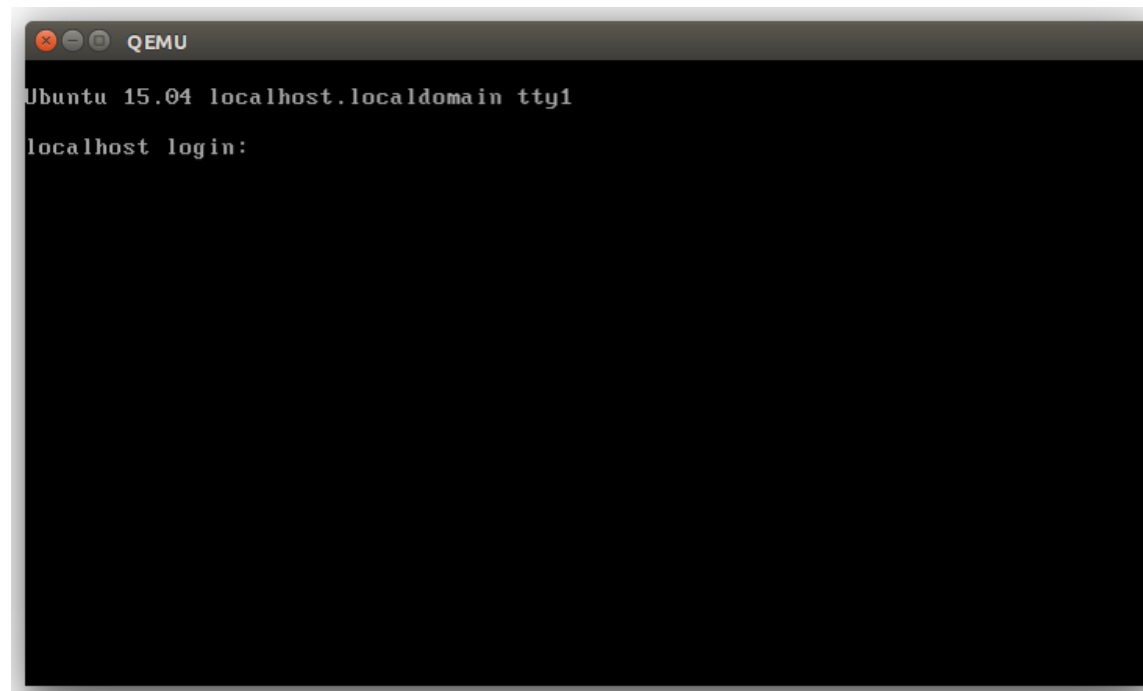
# Snappy Image

```
$ wget  
http://releases.ubuntu.com/15.04/ubuntu-  
15.04-snappy-amd64-generic.img.xz  
  
$ unxz ubuntu-15.04-snappy-amd64-  
generic.img.xz
```



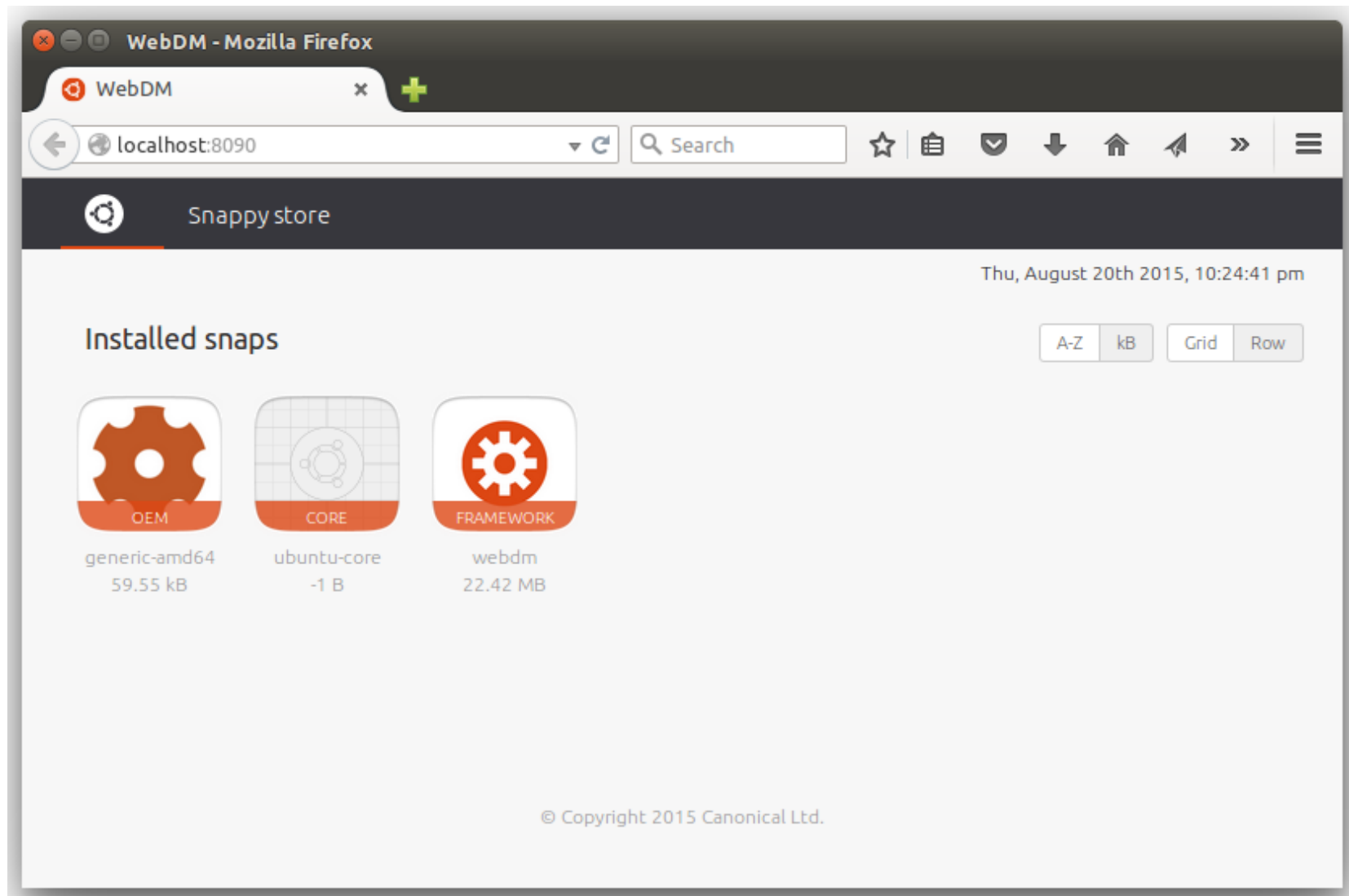
# Test It!

```
$ kvm -m 512 -redir :8090::4200  
-redir :8022::22 ubuntu-15.04-snappy-  
amd64-generic.img  
  
$ ssh -p 8022 ubuntu@localhost
```



```
QEMU  
Ubuntu 15.04 localhost.localdomain tty1  
localhost login:
```

# Play with it!

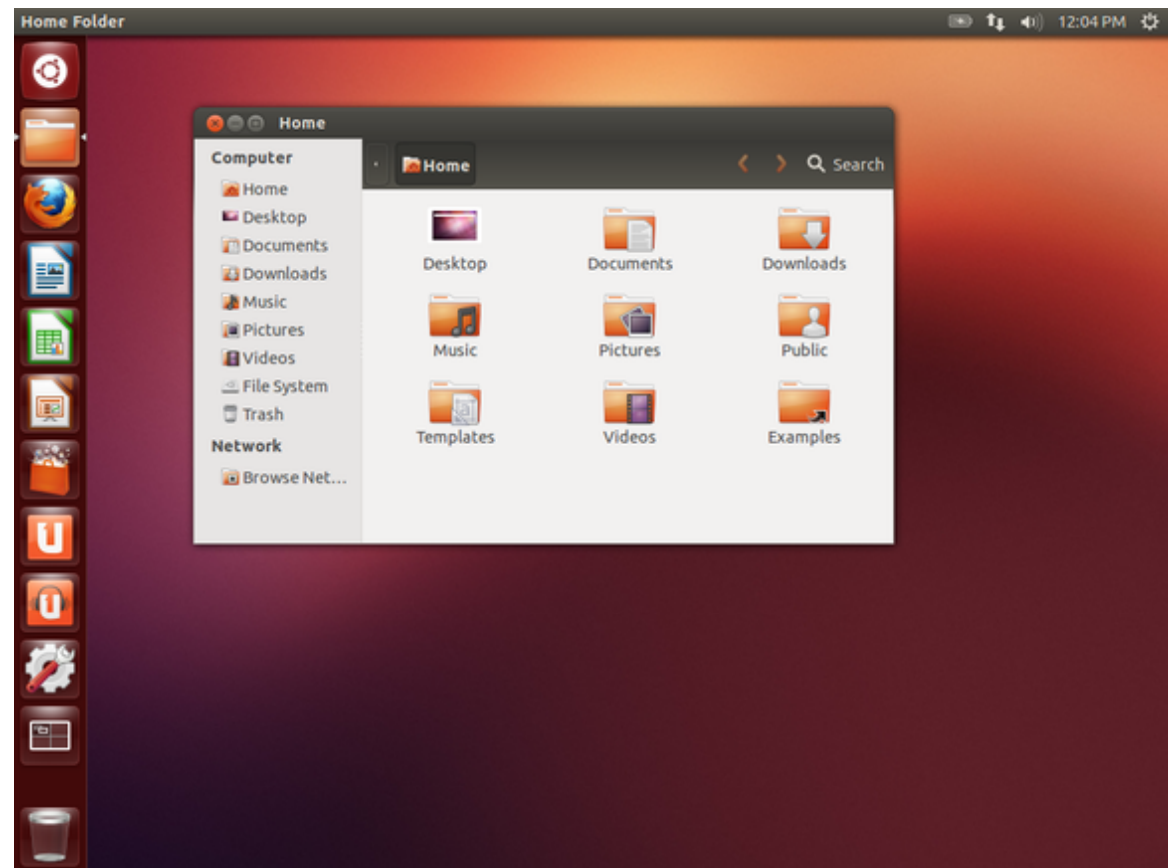


“ Ubuntu Core provides transactional updates with rigorous application isolation. This is the smallest, safest Ubuntu ever, on devices and on the cloud. We’re excited to unleash a new wave of developer innovation with snappy Ubuntu Core! ”

Mark Shuttleworth, founder of Ubuntu and Canonical.

# Ubuntu

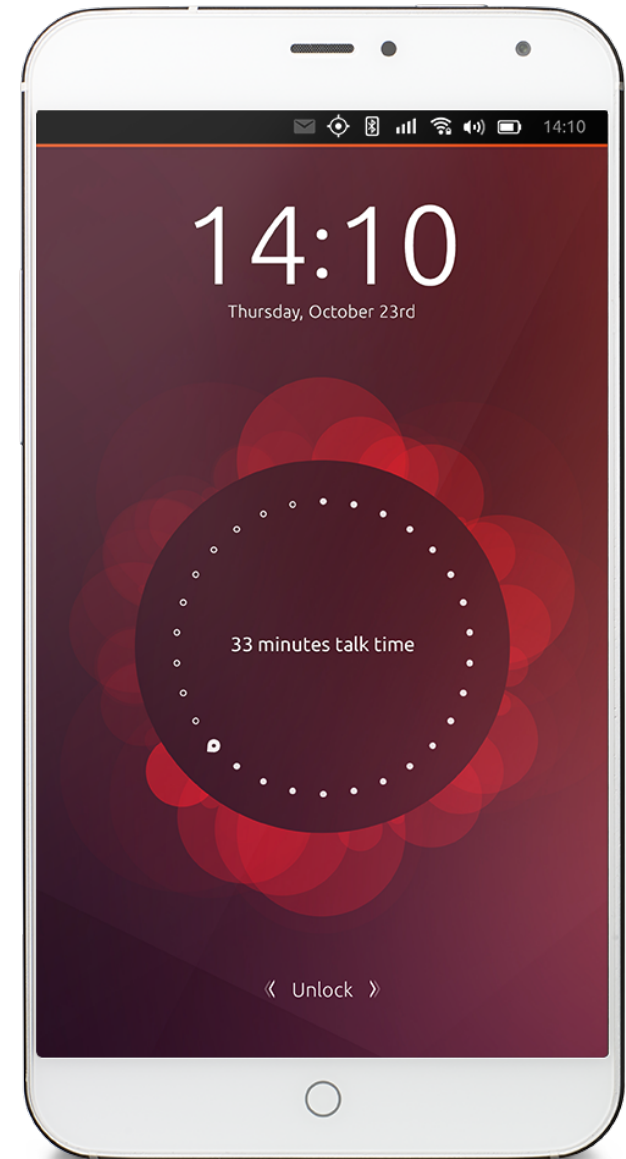
- Debian based (package format and policies)
- Regular Releases
- Updated via Apt



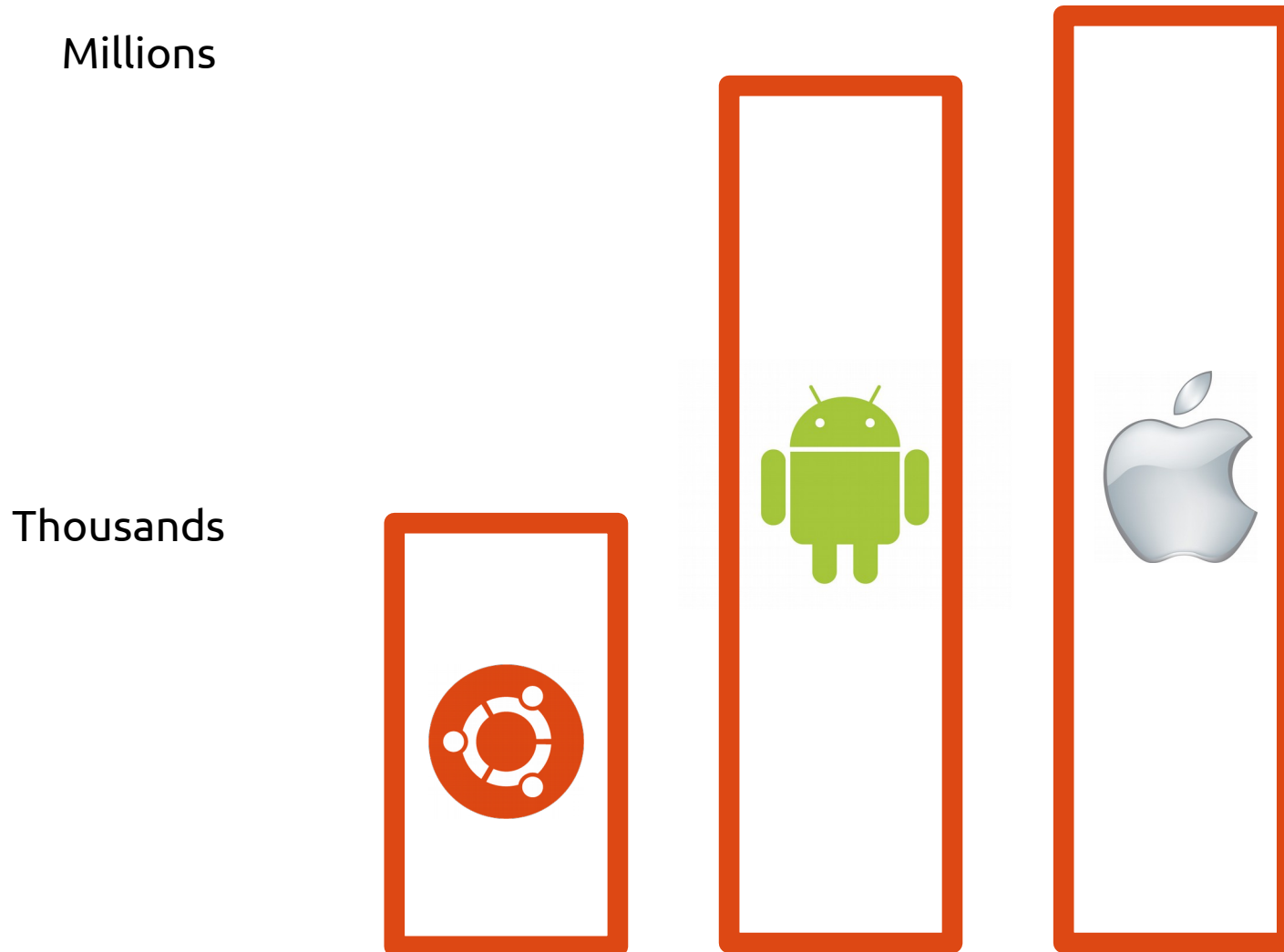


# Ubuntu Phone

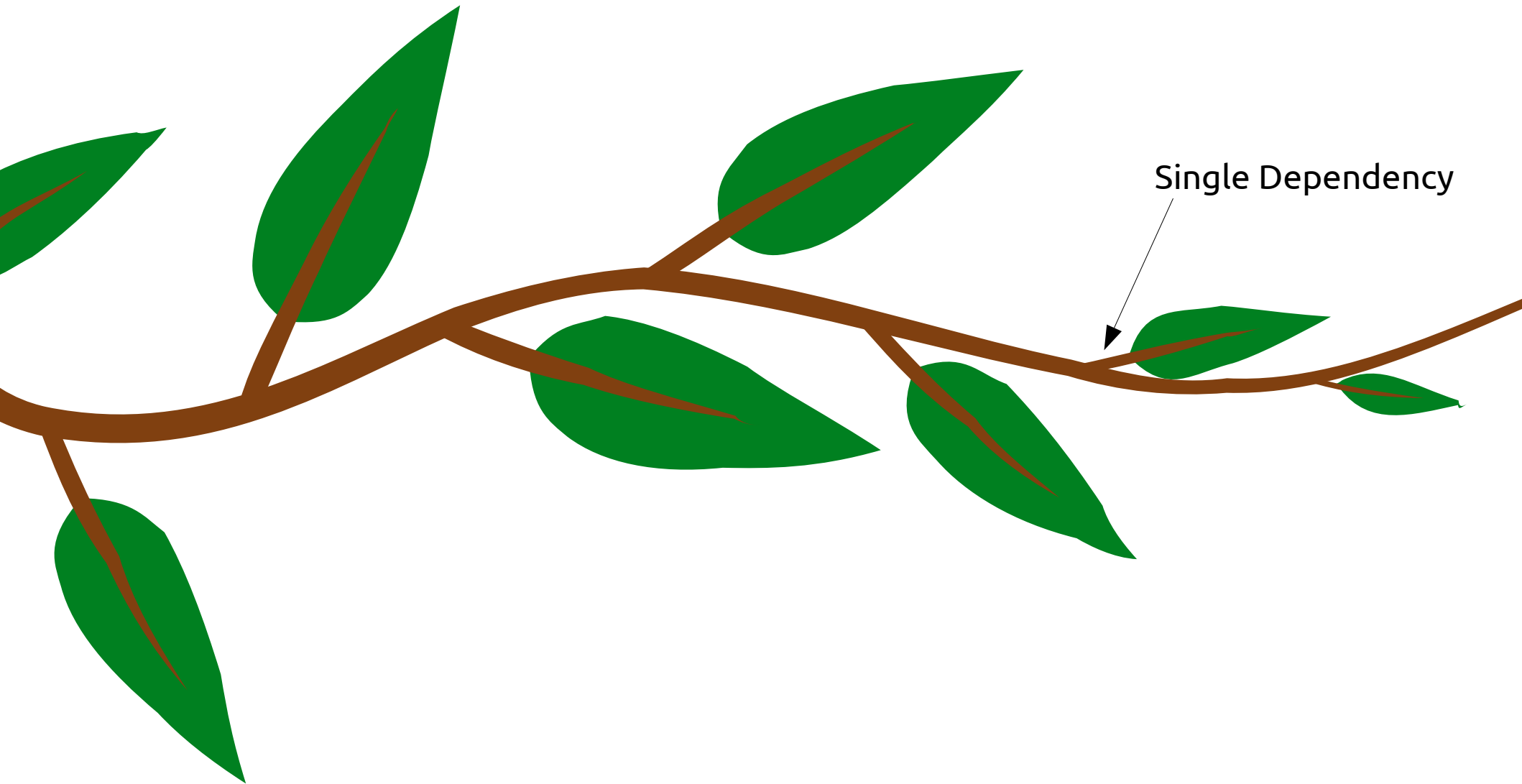
- Click Applications
- System Image Updates

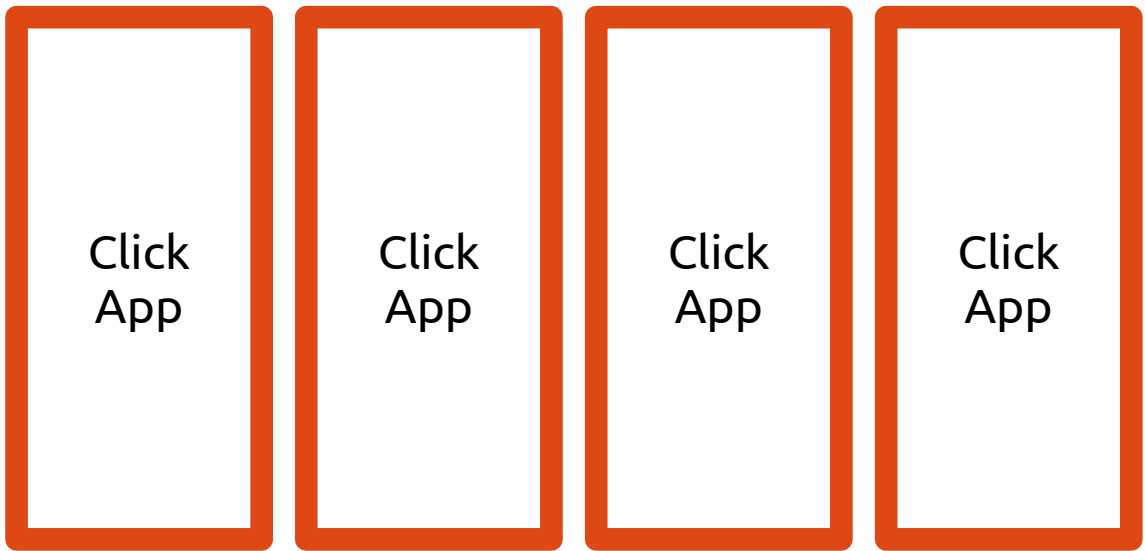


# Why Click?



# Leaves are Simpler





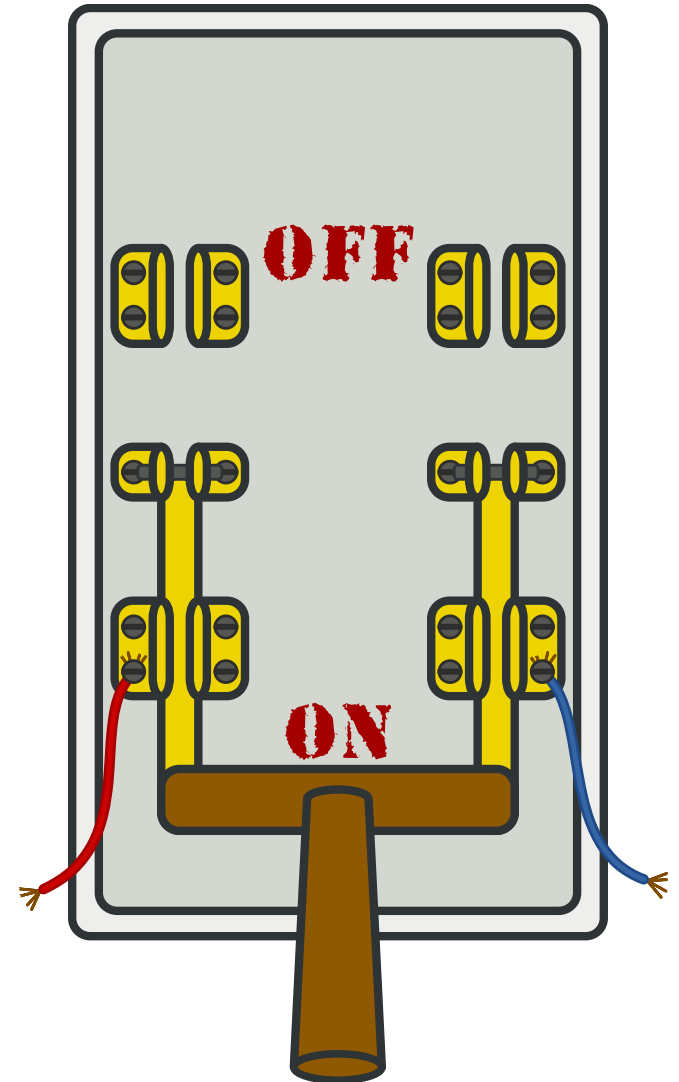
Ubuntu Touch

# Reviewers Don't Scale



# Why Image Based Updates?

- Binary: Works or it doesn't
- Provides opportunity to rollback
- Simple testable states



“Let's Take Ubuntu Phone  
all the way to 11!”

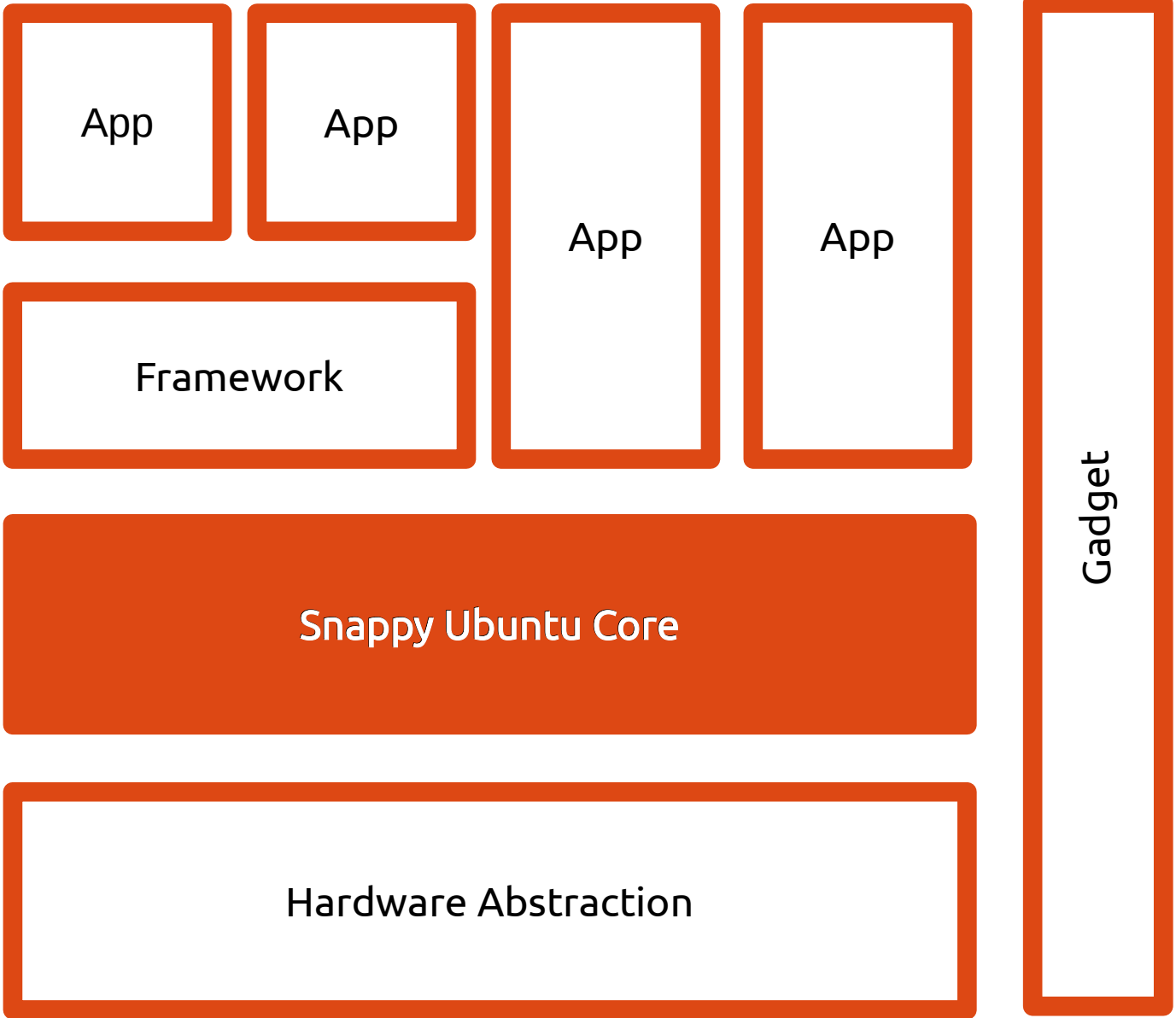
— Us, acting cooler than  
we actually are

# Snappy Ubuntu Core

- Transactional Updates
- Snap based Apps
- Snap frameworks
- OS snap







App

App

App

App

Framework

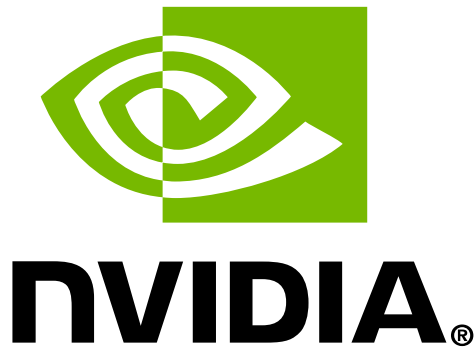
Snappy Ubuntu Core

Hardware Abstraction

Gadget

# Hardware Abstraction

- Provided by board vendors
- Allows for custom drivers and config

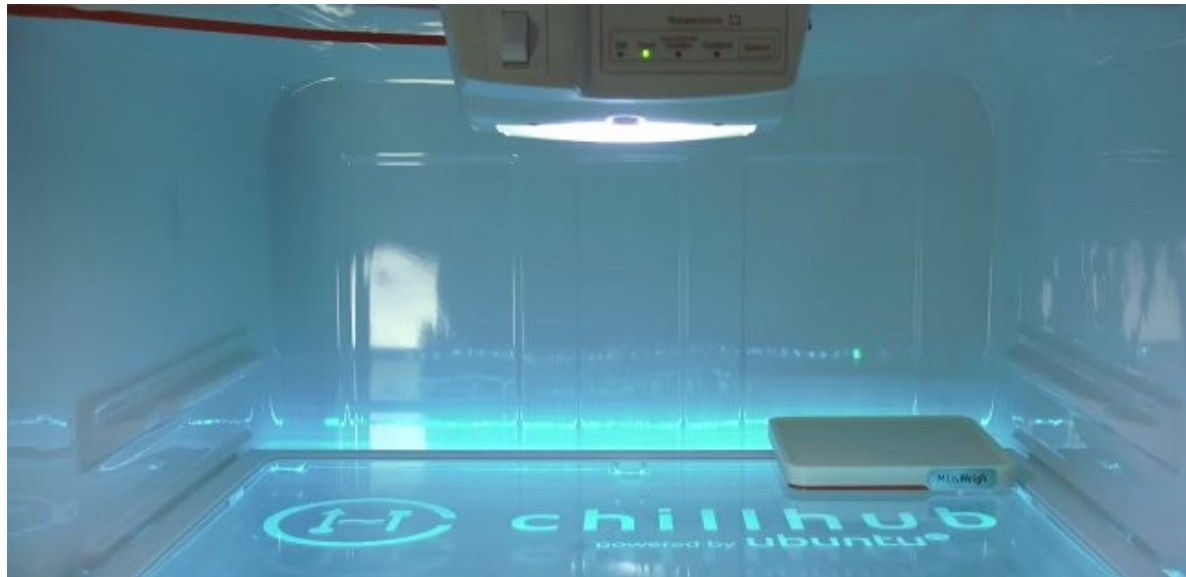


**ARM**

The ARM logo is the word "ARM" in a bold, blue, sans-serif font.

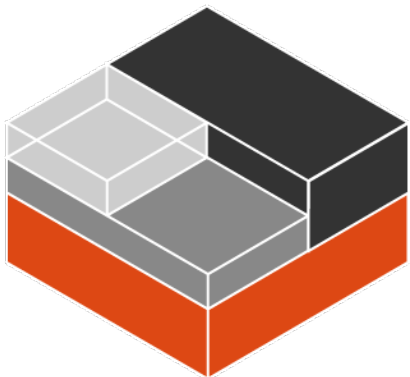
# “Gadget” Snap

- Per-device configuration
- Snaps to install
- Permissions
- Branding



# Framework Snaps

- Provide shared services
- Mediate resources
- IPC to Apps

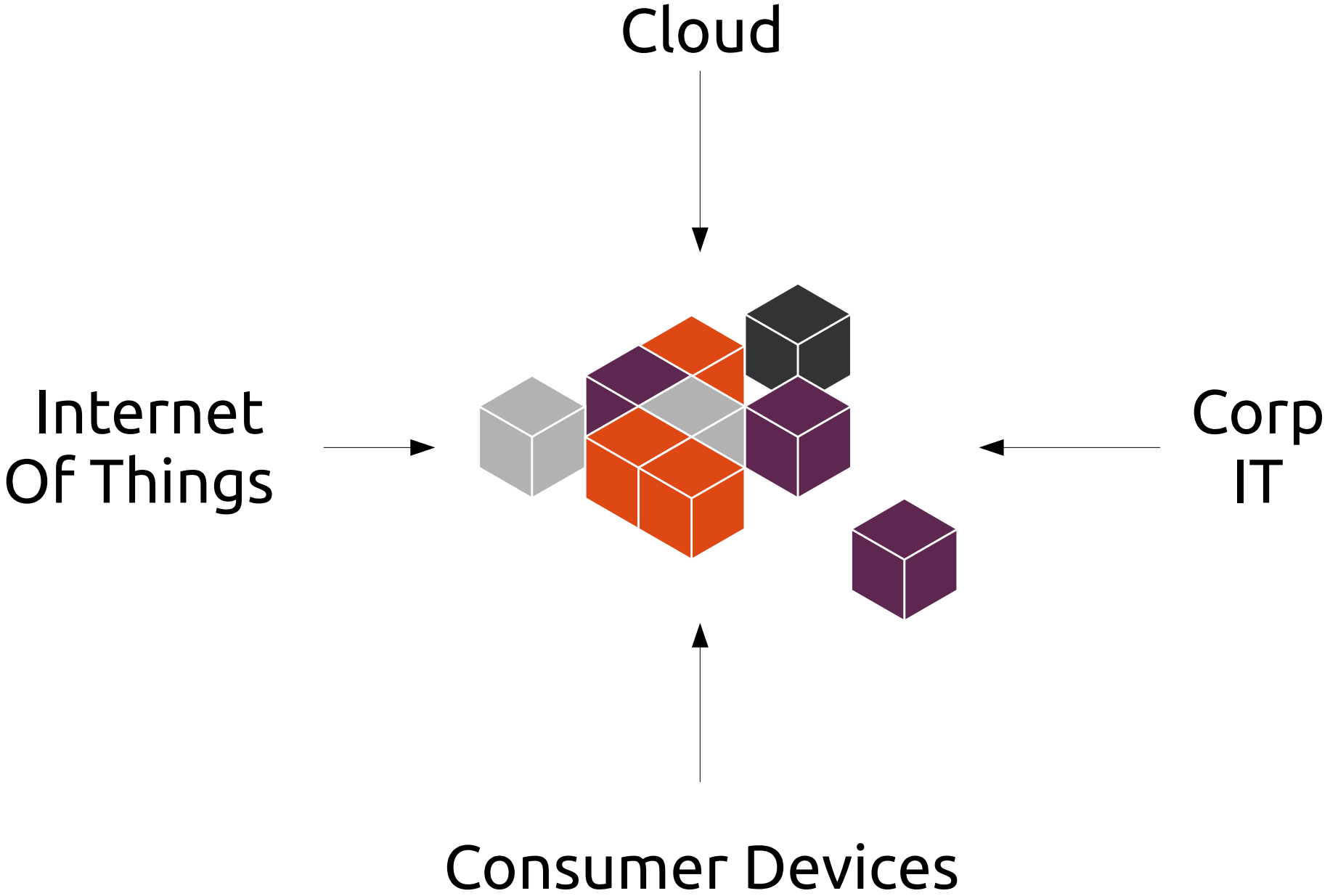


Mir System  
Compositor



# Why snap? (not click)

- Click v2.0
- Click only for leaf nodes
- Support for OS and framework snaps
- Lower level components



# Trend: Internet of Things

- Smaller computers
- Internet connectivity
- Full OS resources
- Must be reliable



# Trend: Consumer Devices

- More complex interactions
- Bullet-proof experience
- Complex security situations





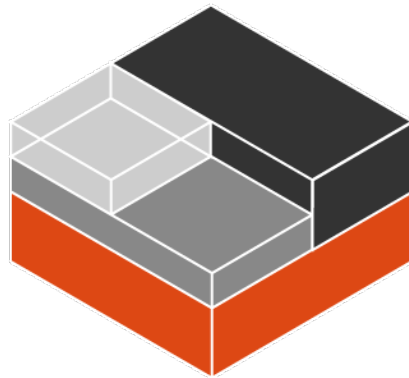
# Trend: Corporate/Education IT

- Verifiable Images
- Upgrades don't cause downtime
- Limited apps and app permissions

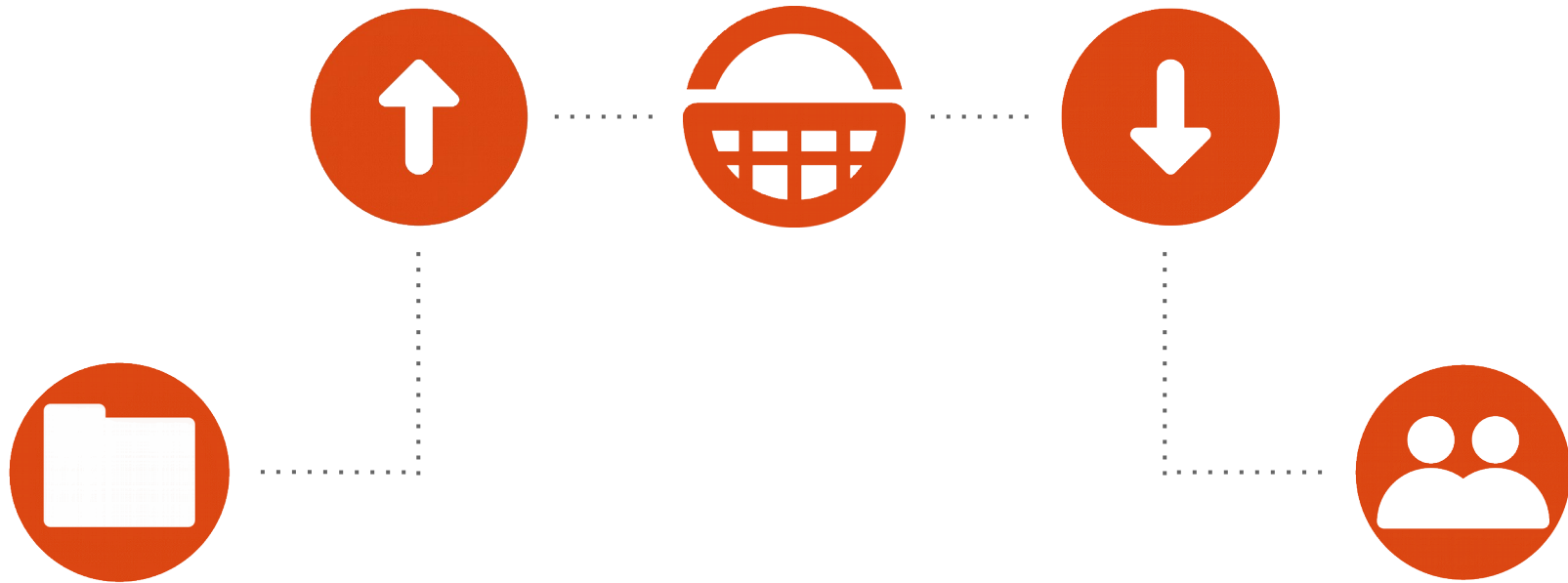


# Trend: Cloud and Containers

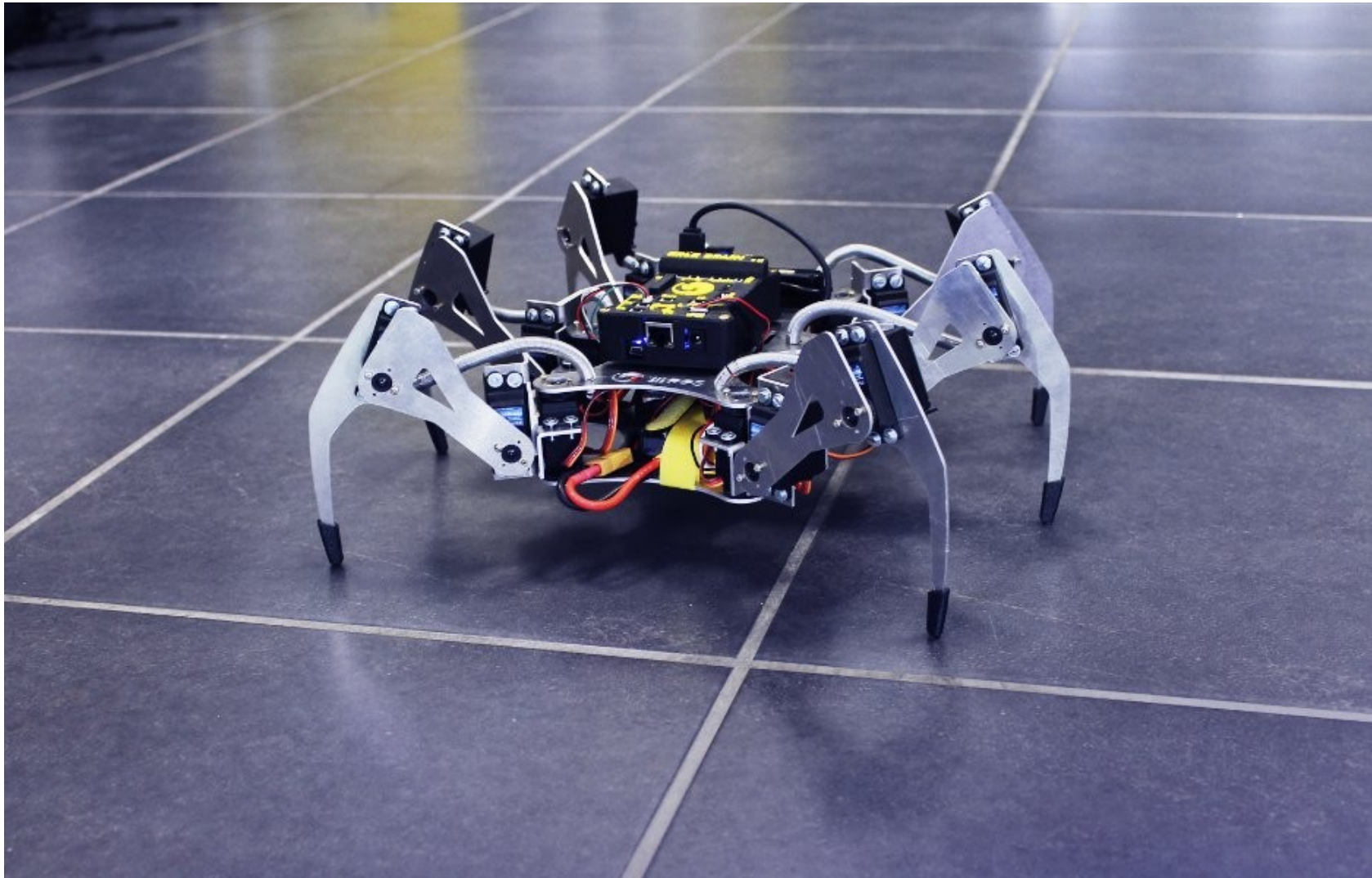
- Small base OS (JeOS)
- Unit of specialized code
- Tested as a unit



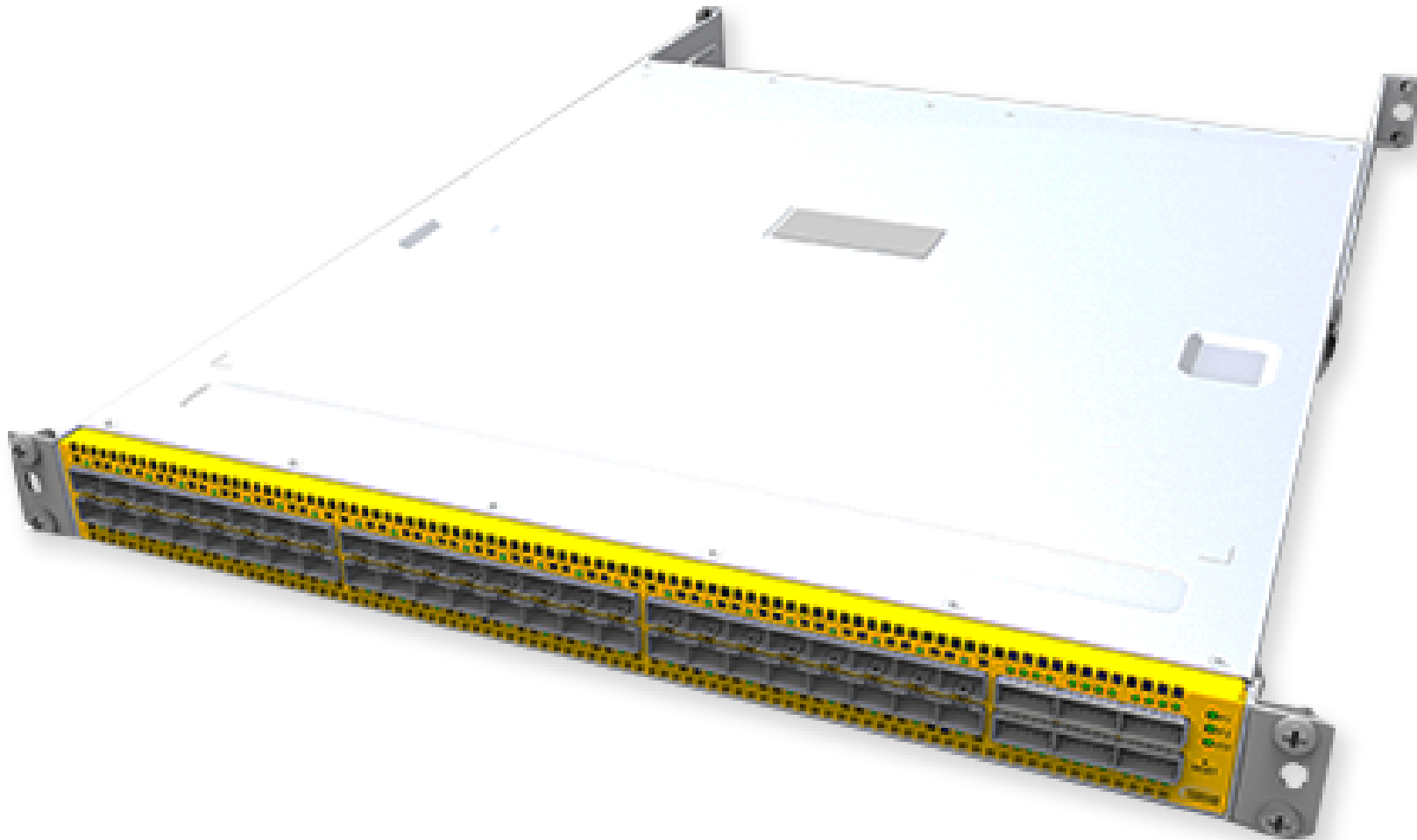
# Snappy Store



# Store Story: Erle Robotics



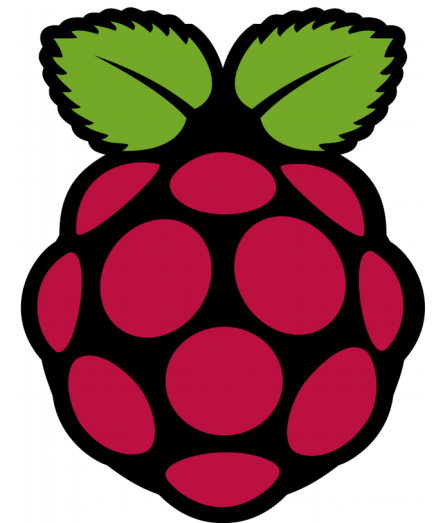
# Store Store: Networking



# Try it today!

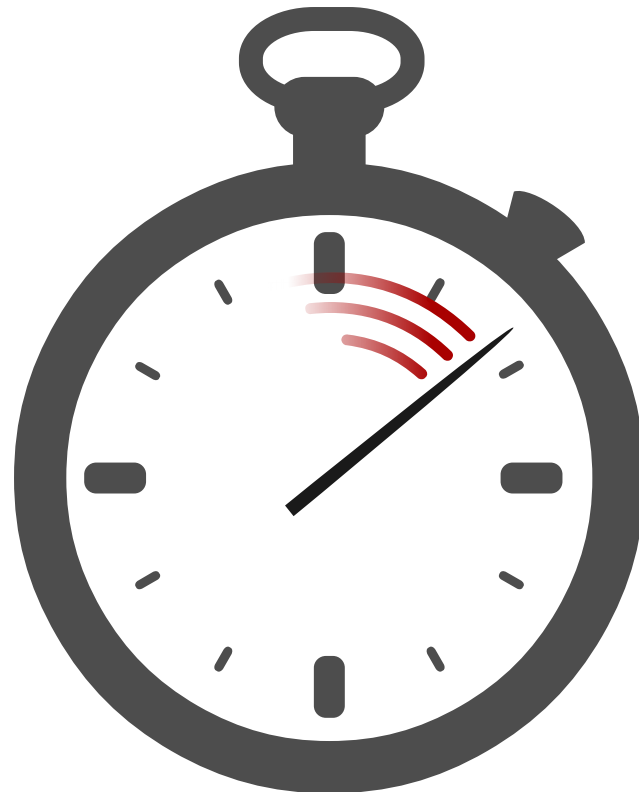


Microsoft Azure

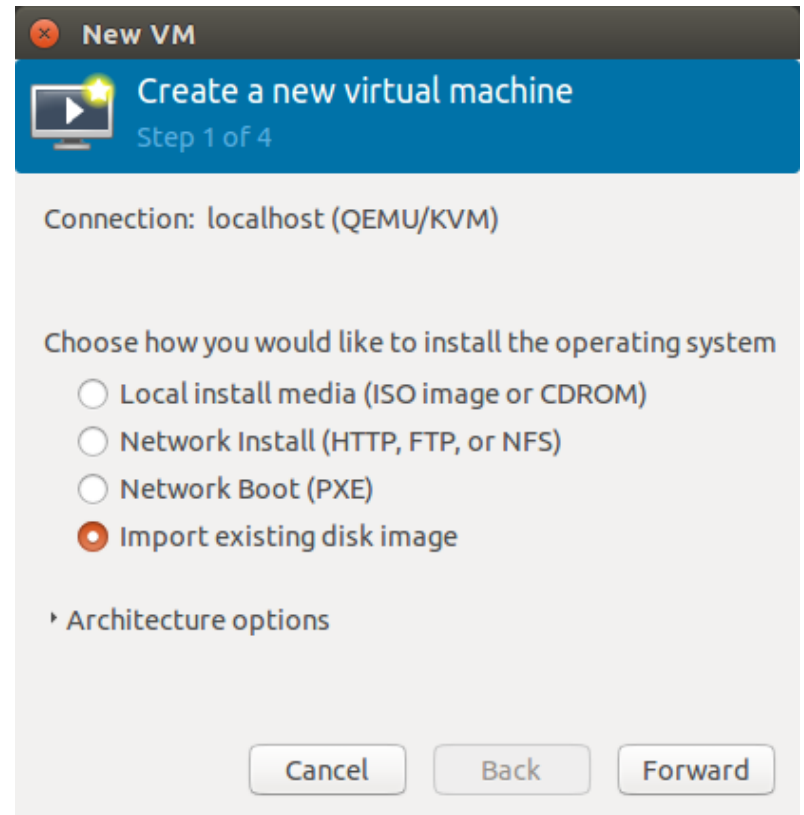
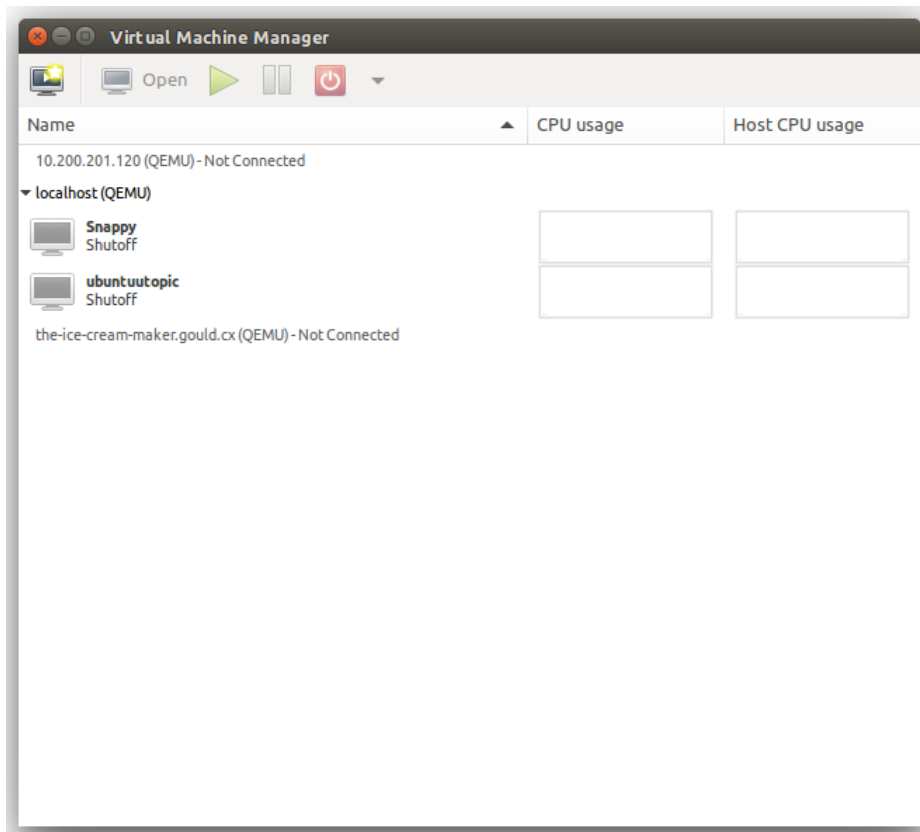


<http://ubuntu.com/snappy>

# Progress Check-In

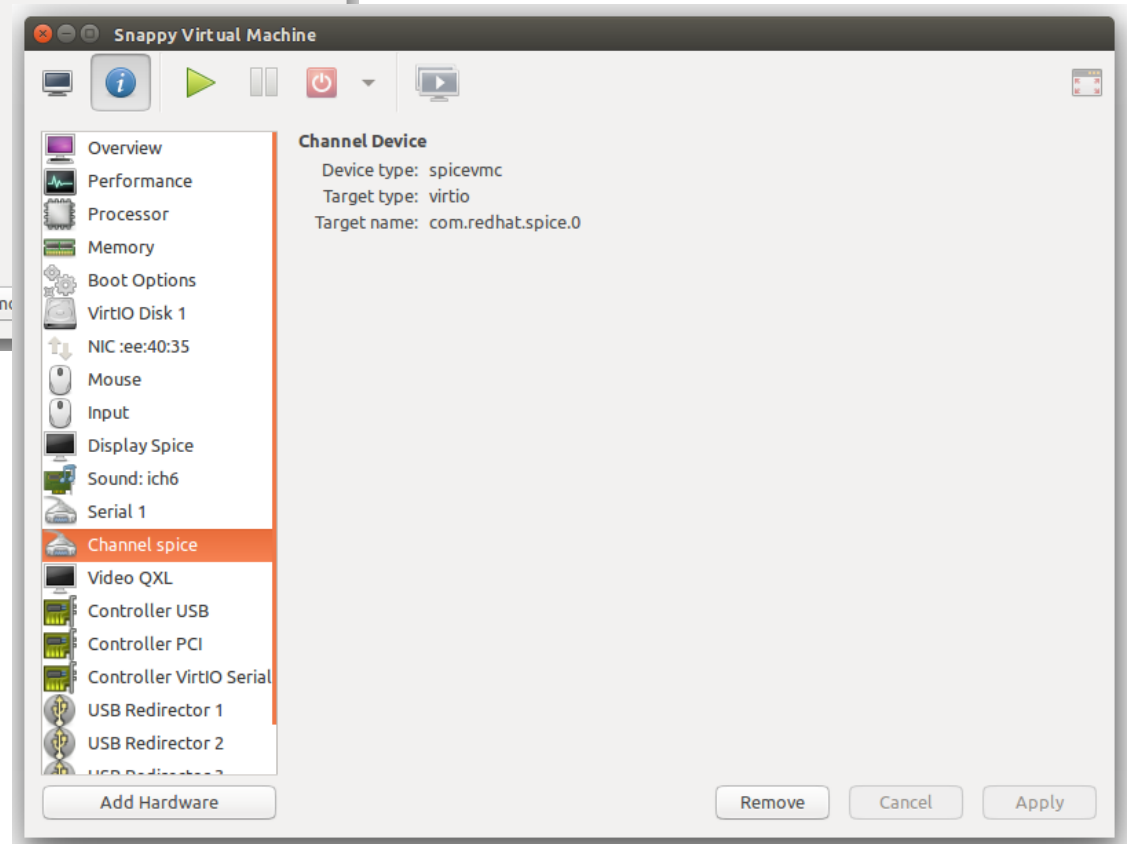
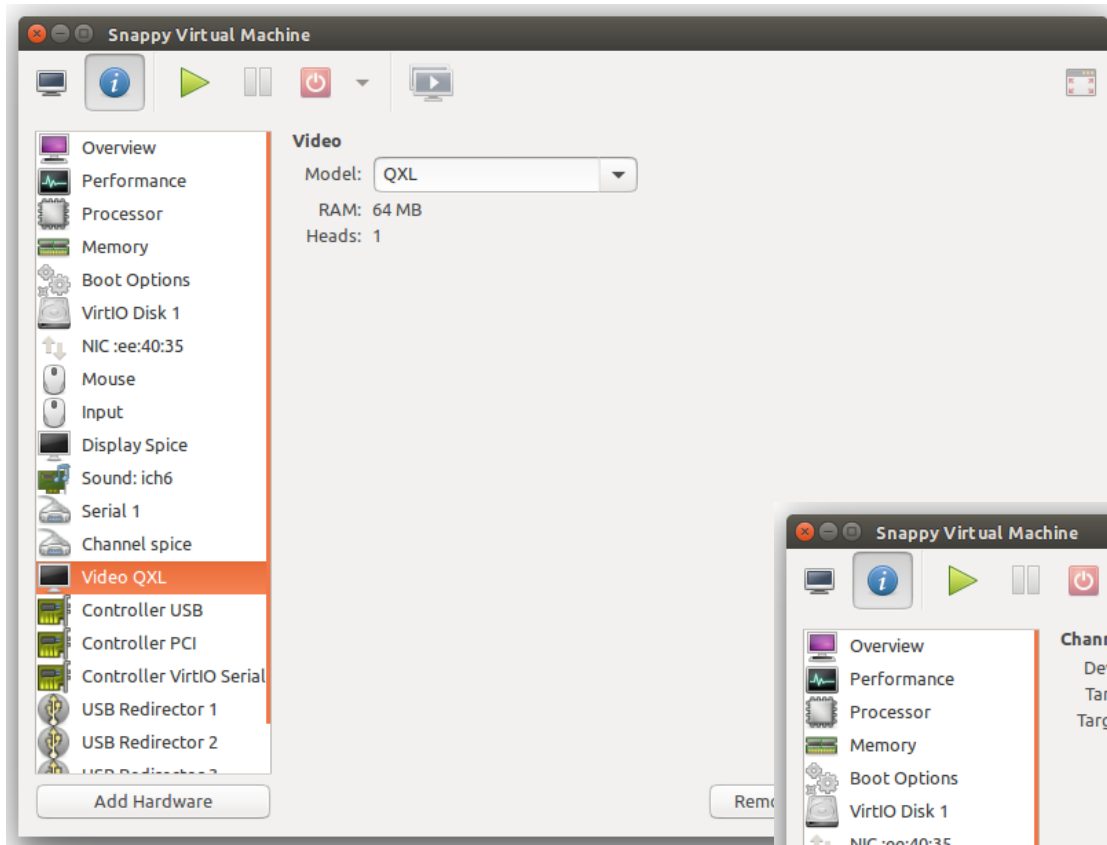


# Import into VMM



```
$ sudo cp ubuntu-15.04-snappy-amd64-generic.img  
/var/lib/libvirt/images/  
  
$ sudo chown libvirt-qemu.kvm  
/var/lib/libvirt/images/ubuntu-15.04-snappy-amd64-  
generic.img
```



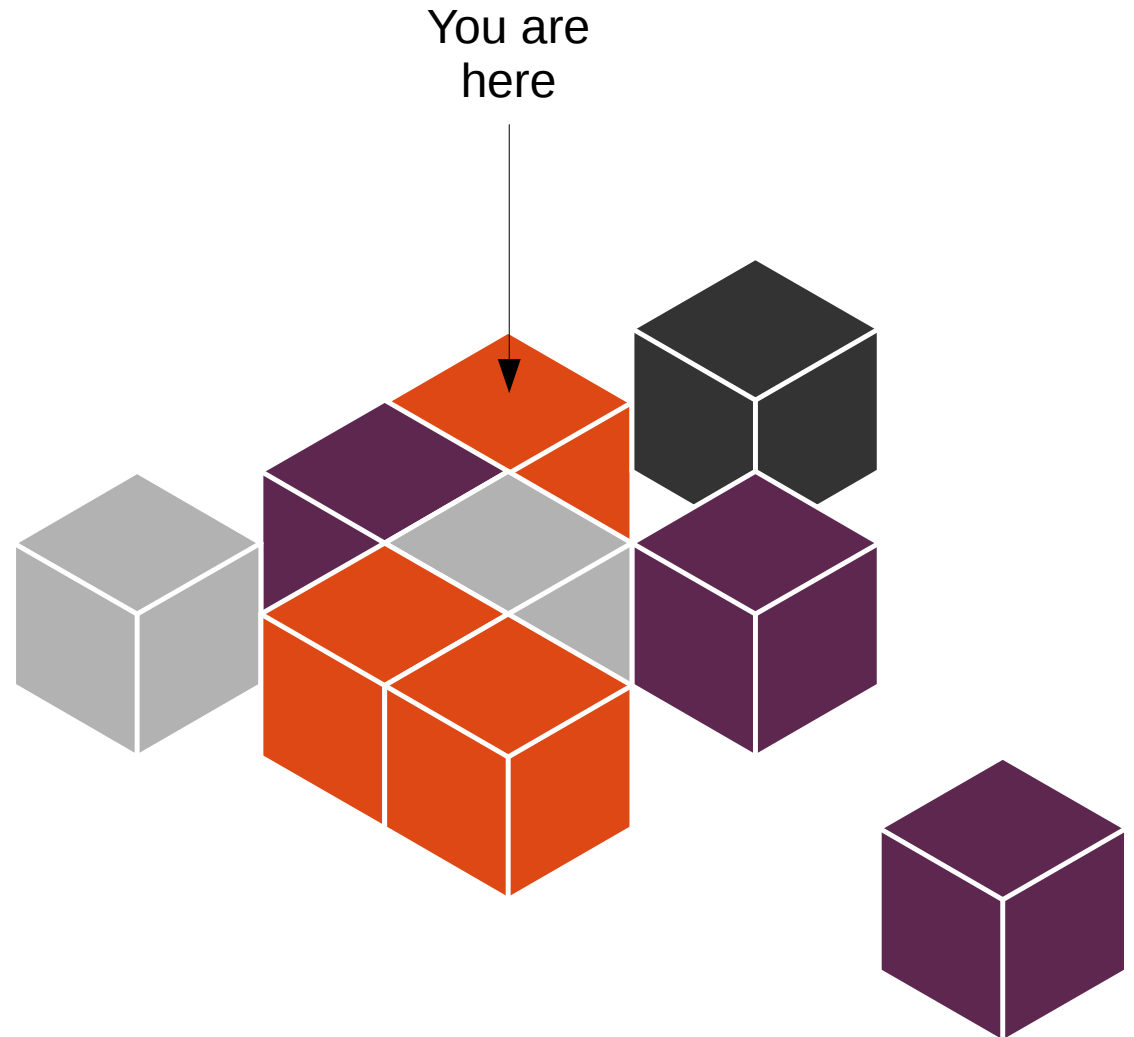


# Install Snapcraft

```
$ ifconfig  
$ sudo snappy install docker  
$ sudo snappy install snapcraft-daily.ted  
$ snapcraft-daily.snapcraft help
```

# What is a snap?

- A self-contained bundle
- Includes dependencies
- Works with frameworks

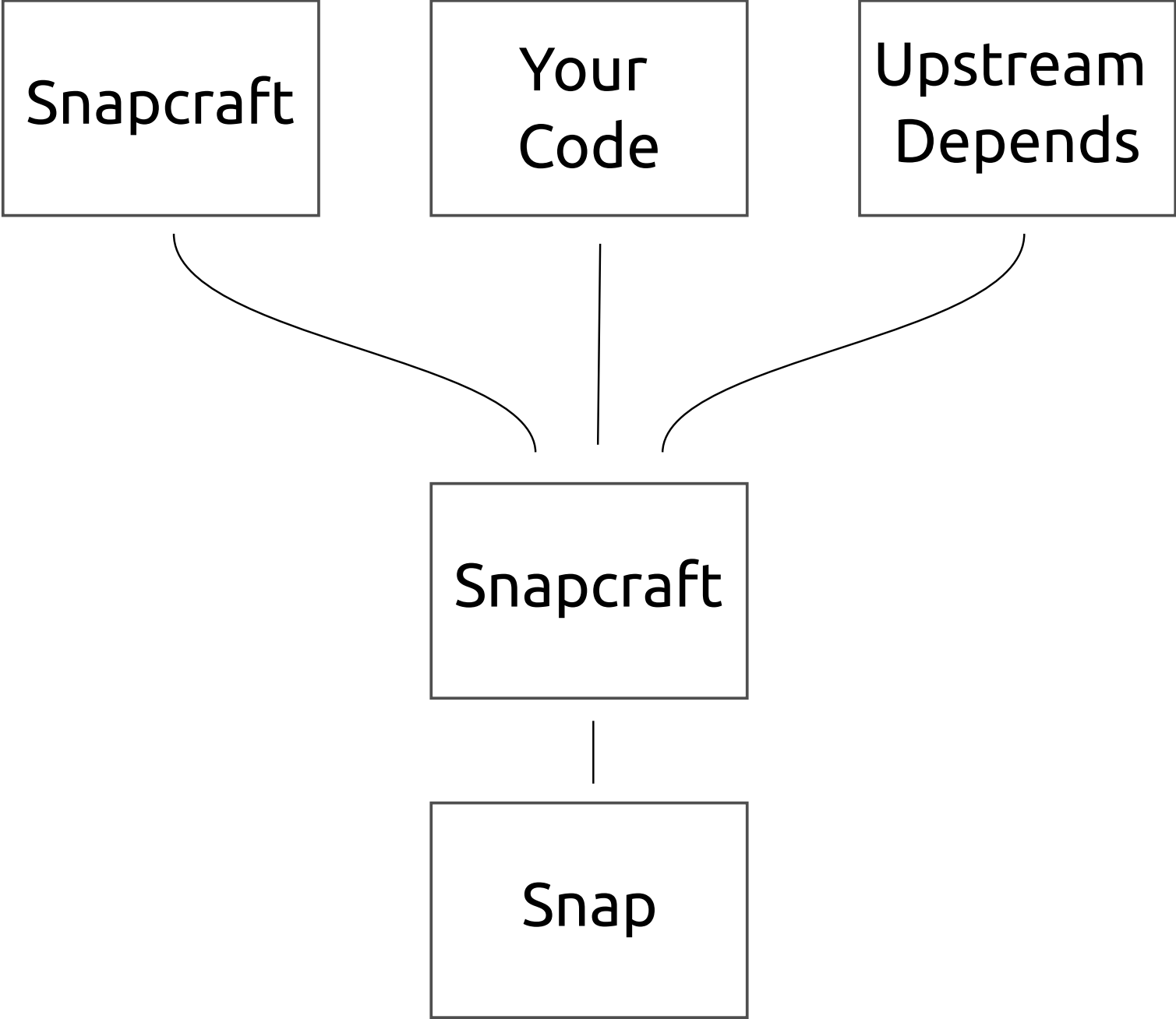


# Snap Definition (package.yaml)

```
name: photoviewer
version: 0.1
vendor: Ted Gould <ted@canonical.com>
architecture: amd64
frameworks: [mir]
binaries:
  - name: photoviewer
    exec: qmlscene main.qml --
    caps:
      - mir_client
      - network-client
```

# Snap Directories

Directory	Writable?	Description
<code>/apps/&lt;app-name&gt;/&lt;version&gt;/</code>	No	Read-only files, libraries, resource files, and other binary data shipped with the app.
<code>/var/lib/apps/&lt;app-name&gt;/&lt;version&gt;/</code>	Yes	Writable files, configuration or other data that is not specific to any user. This directory needs to be created by the application right now and will be created by the snappy tool soon.
<code>/var/lib/apps/&lt;app-name&gt;/&lt;old-versions&gt;</code>	No	Read-only for the apps, backup purpose.
<code>/home/user/apps/&lt;app-name&gt;/&lt;version&gt;/</code>	Yes	Writable, configuration or other data specific to the calling user. The app needs to create them right now. They will be created by the snappy tool soon.
<code>/home/user/apps/&lt;app-name&gt;/&lt;old-versions&gt;/</code>	No	Read-only for the configuration or other specific data for the calling user, backup purpose. This is not created by snappy right now but will be soon.



# Block Inception



# Snapcraft.yaml

```
parts:  
  qml:  
    plugin: qml  
  photoviewer:  
    plugin: copy  
    files:  
      main.qml: main.qml  
      PhotoViewerCore: PhotoViewerCore  
snappy-metadata: meta
```



Demo