Core

ubuntu core

Ubuntu Core Porting

Board enablement experience sharing

https://www.ubuntu.com/core

Woodrow Shen

woodrow.shen@canonical.com

Agenda

Core

- Prerequisites
- Snappy booting process
- Gadget snap creation
- Kernel snap creation
- Image creation
- Issues and Challenges
- Q&A

The official guide for porting used the github project I contributed ...

https://github.com/xapp-le/SnappyUbuntuCore

v.s

https://docs.ubuntu.com/core/en/guides/build-device/board-enablement

To build a Ubuntu Core image, three snaps are essential and fundamental

OS snap

Gadget snap

Kernel snap

Prerequisites

Core

Using roseapple-pi as example

- Install snappy tools and cross toolchains

```
$apt-get install -y build-essential u-boot-tools lzop
debootstrap gcc-arm-linux-gnueabihf device-tree-compiler
$apt-get install -y snapcraft snap ubuntu-image
```

- Fetch snappy builder - a script-based tool

```
$git clone https://github.com/xapp-le/SnappyUbuntuCore.git
```

- Get BSP from vendor
  - Check u-boot version
  - Check kernel version
  - Check image layout

Case study

For MT7623

Image layout

start	size	binary	usage
0	2K	bromhdr.bin	Rom used and Available for partition table etc.
2	318KB	preloader_iotg7683p1_emmc.bin	Mediatek preloader
320	512KB	u-boot.bin	u-boot
842	128KB		environment
970	54KB		reserved
1024	END		Free for partitions

Prerequisites

Core

Signature key for store upload

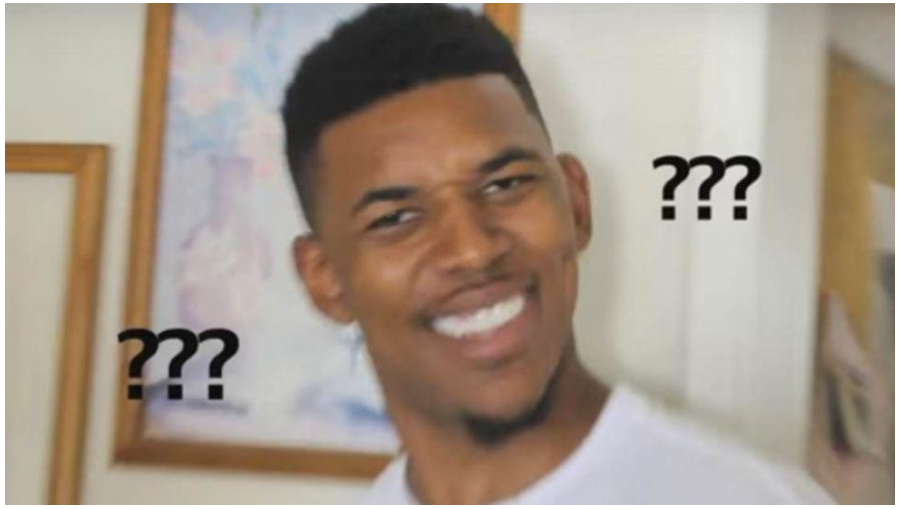
- Create a key

```
$snapcraft login
$snap create-key <key-name>
```

- Register a key to store

```
$snapcraft register-key
```

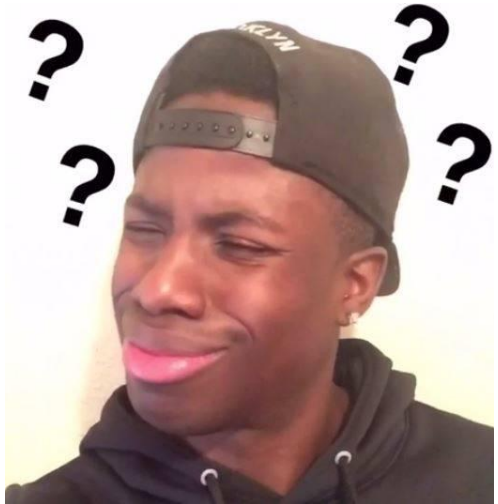
- For backup purposes, snap keys are stored under ~/.snap/gnupg



Core

Assertions

這三小？



A signed document to make sure our system can be trusted on the device according to snap policy

Assertions have different types - model/serial/account...

Model assertion

Core

To build an image, it is required you have a signed model assertion

- Create a json format

```
{
  "type": "model",
  "authority-id": "<your store account id>",
  "brand-id": "<your store account id>",
  "series": "16",
  "model": "roseapple",
  "architecture": "armhf",
  "gadget": "roseapple-pi",
  "kernel": "roseapple-pi-kernel",
  "timestamp": "<timestamp>"
}
```

Model assertion

Core

To build an image, it is required you have a signed model assertion

- Sign it

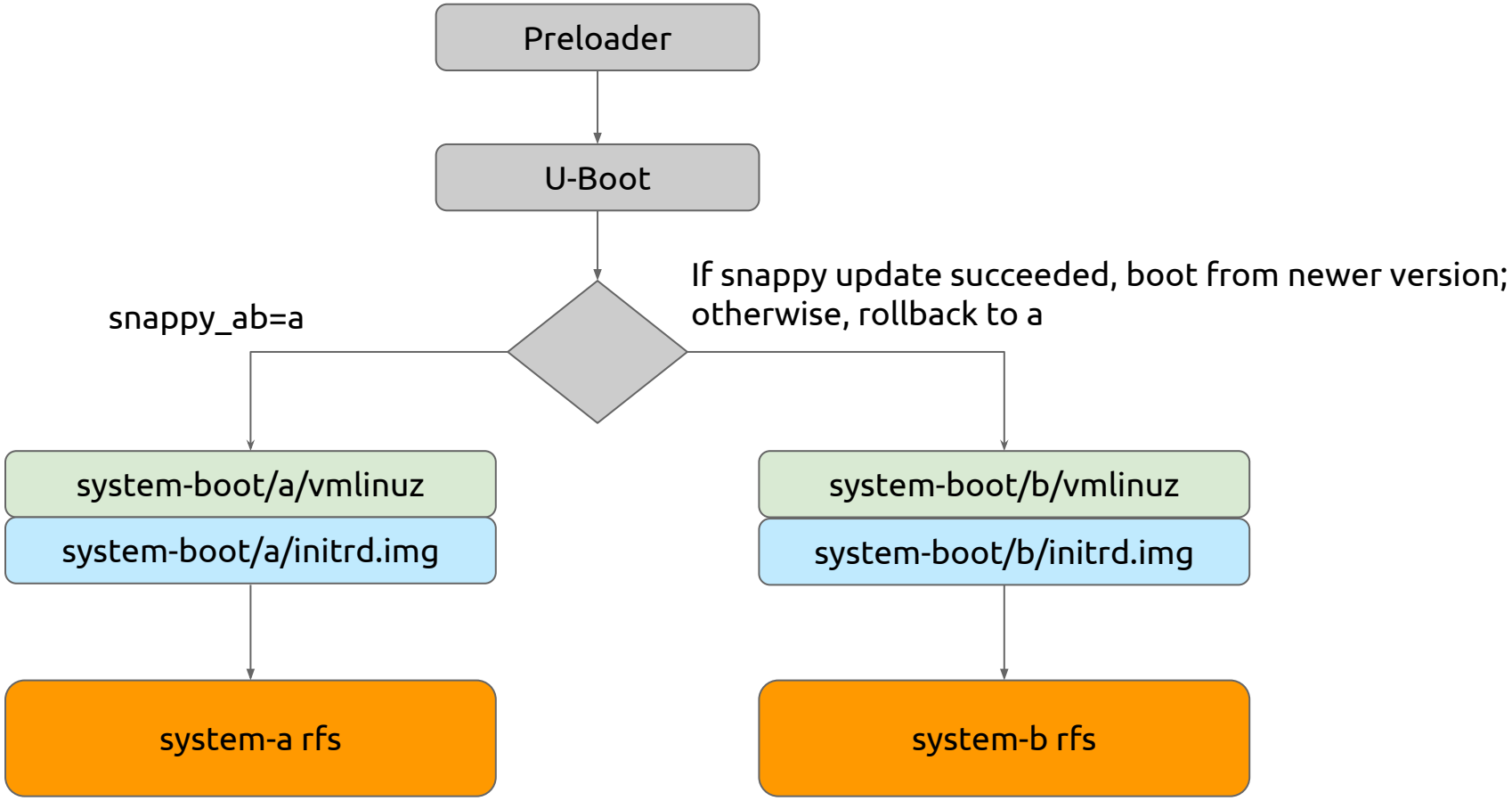
```
$cat roseapple-model.json | snap sign -k <key-name> &>
roseapple.model
```

Snappy Booting Process

Snappy booting logic

Core

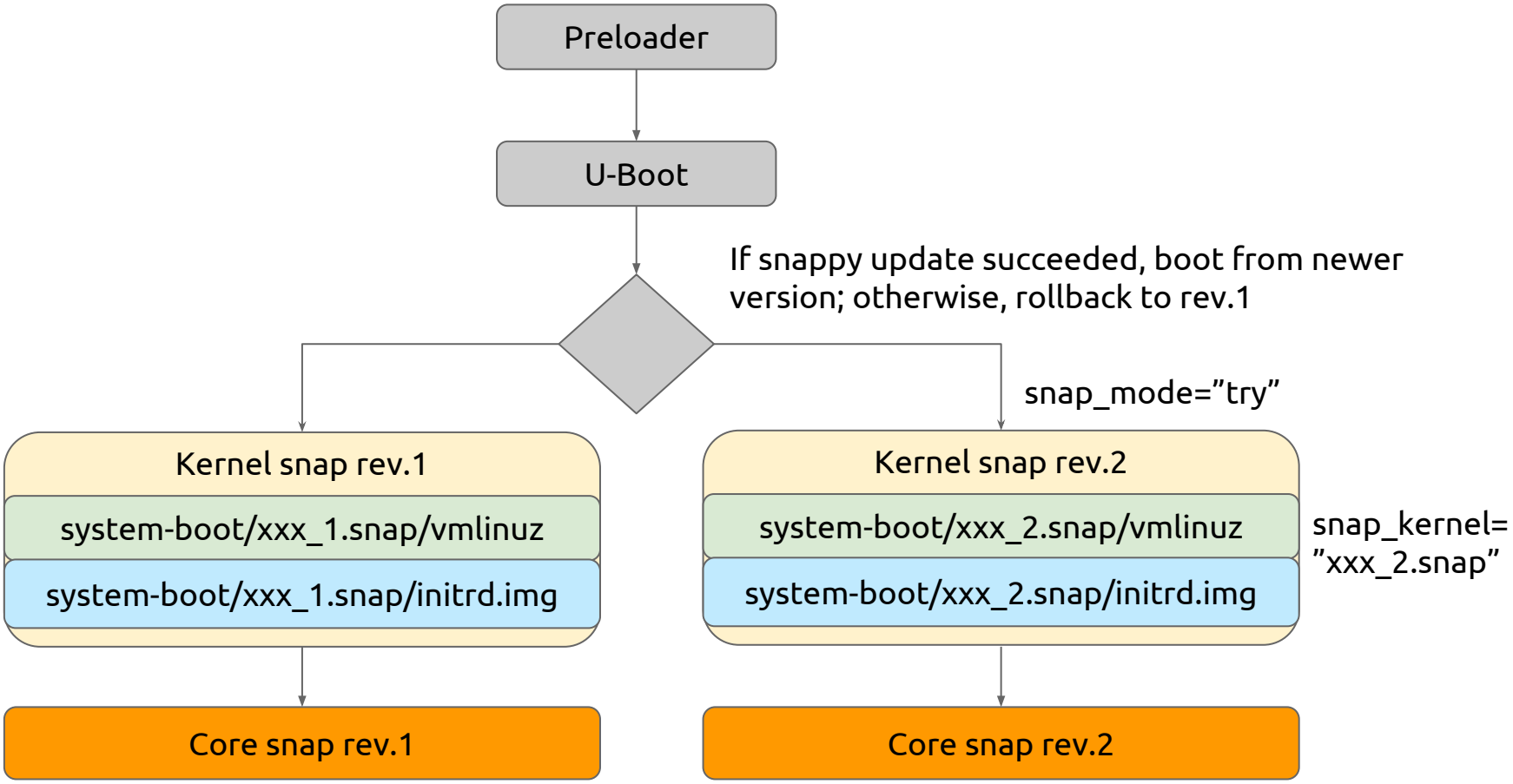
Recall 15.04 system-AB



Snappy booting logic

Core

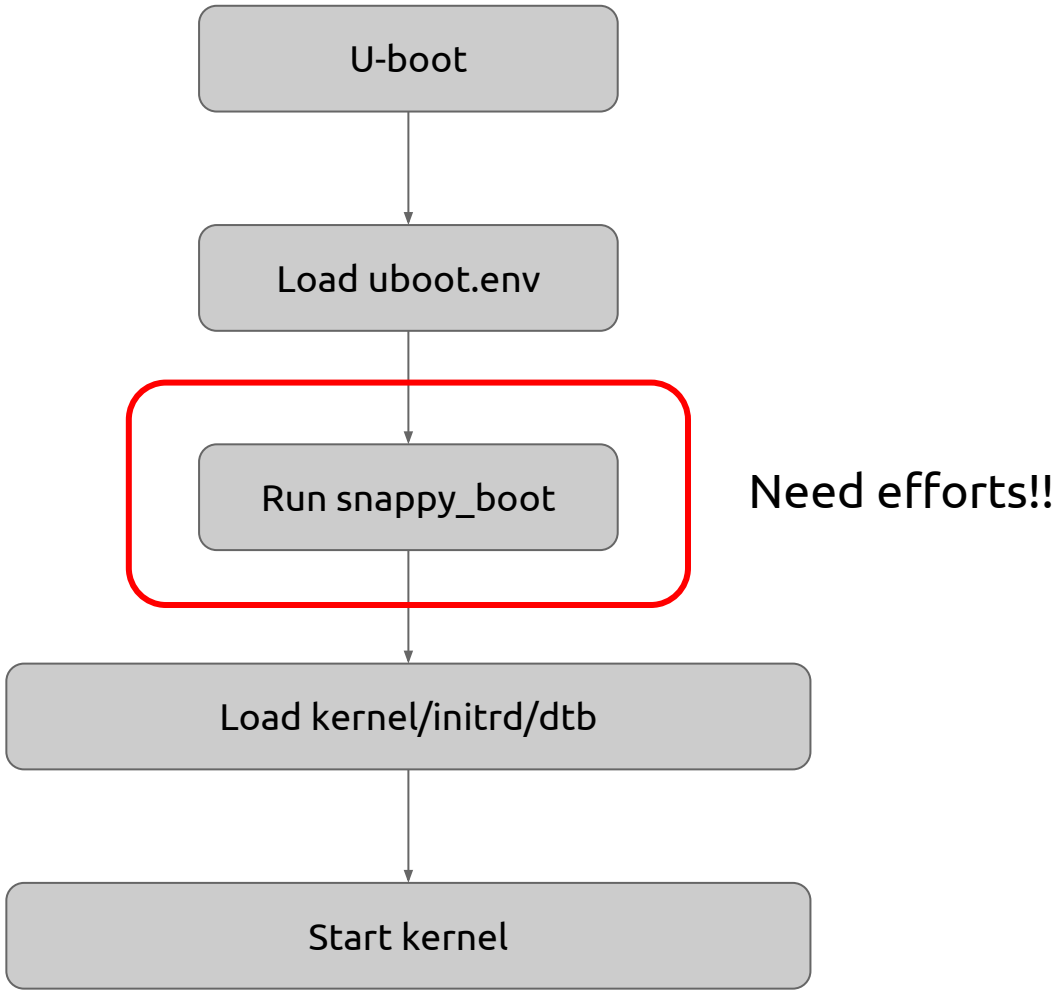
Ubuntu Core 16



Snappy booting logic: U-boot part

Core

All the things U-boot needs to do



Gadget snap creation

Create a gadget snap

Core

Pack boot parts together as snap

- Simplest way
  - Define a snapcraft.yaml
  - Define a gadget.yaml
  - Define a uboot.env.in

- Then use snapcraft cmd

```
$snapcraft --target-arch=armhf -d
```


[1] https://ograblog.wordpress.com/2017/05/30/building-u-boot-gadget-snap-packages-from-source/

[2] https://ograblog.wordpress.com/2017/06/13/patching-u-boot-for-use-in-an-ubuntu-core-gadget-snap/

[3] https://github.com/ogra1/sabrelite-gadget

Kernel snap creation

Create a kernel snap

Core

To fit all the features that snapd required

- Backport the snappy specific patches [2]
- Backport the latest apparmor asap [3]
- Use snapcraft.yaml to set all the configs the board needs
  - Also can apply additional patches and firmwares

- Then use snapcraft cmd

```
$snapcraft --target-arch=armhf -d
```

[1] https://github.com/xapp-le/kernel/tree/Ubuntu-Snappy-Core

[2] https://github.com/snapcore/sample-kernels

[3] http://kernel.ubuntu.com/git/jj/linux-apparmor-backports/refs/heads

Image creation

Build an image

Core

Use ubuntu-image tool

- “--extra-snaps” can fetch a snap from store or local path
- Instruction

```
$sudo ubuntu-image \
-c [edge|beta|candidate|stable] \
--image-size 4G \
--extra-snaps <gadget snap> \
--extra-snaps <kernel snap> \
--extra-snaps <preinstalled snaps> <model assertion file>
```

```
$sudo ubuntu-image \
-c stable \
--image-size 4G \
--extra-snaps roseapple-pi_16.04-1.5_armhf.snap \
--extra-snaps roseapple-pi-kernel_3.10.37_armhf.snap \
--extra-snaps snapweb roseapple.model
```

Issues and challenges

Core

Some tips

- 3.1x Kernel porting
  - Snapd needs some features introduced from new apparmor
  - It may need to backport from jj
    - v3.10.y-aa3.5-beta1
- Device nodes are managed by snapd and used through **snap interface**
  - Gadget snap can define slots of interface for any hardware
- Debugging
  - Boot partition
  - Kernel cmdline
  - syslog

[1] http://snapcraft.io/

[2] https://dashboard.snapcraft.io/

Issues and challenges

If you’re interested in any project on snapcore[1]

- snapd
  - https://github.com/snapcore/snapd
- snapcraft
  - https://github.com/snapcore/snapcraft
- snapweb
  - https://github.com/snapcore/snapweb

[1] https://github.com/snapcore


Q & A
Try to use Core as playground