Snappy Ubuntu Core

A snappy Platform for Embedded, IoT and 96boards

Alexander Sack <asac@canonical.com>
Director, Ubuntu Core

Ricardo Mendoza <ricardo.mendoza@canonical.com>
Lead Architect, Ubuntu Core
CANONICAL
Overview
We are the company behind Ubuntu.
Canonical and Ubuntu | Best of both worlds

CANONICAL
Commercial Backing for the #1 general purpose Linux OS: Ubuntu
Global Services, Support, Certification!

Ubuntu
#1 Linux Desktop
#1 Cloud OS
Now also for phones, tablets & IoT devices
The Devices Revolution!
The ecosystem is in a race to try new frontiers of connected compute scenarios around IoT!
CONNECTED DEVICES IN 2020

26 BILLION

9 BILLION MOBILE PHONES

10 BILLION CONSUMER ELECTRONICS

7 BILLION M2M DEVICES
WE ARE AT THE BEGINNING OF SOMETHING BIG

Smart Cities  Agriculture & Environment  Process Industries  Retail Banking  Automotive Transport

Consumer Electronics  Utilities  Wellness  Infrastructures

$1.9 TRILLION
Predicted value-add of IoT across sectors in 2020

CONNECTIVITY
It has never been easier to make a custom hardware appliance...
Building an IoT Appliance

1. Take a 96board
2. Add IoT components
3. Design a cool chassis/box
4. Make the software
5. Call an ODM and ship it!
It has never been easier to solve the challenges for making smart devices using open source components!
Building a Smart IoT M2M Device

1. Take a 96board reference stack
2. Integrate existing open-source software
3. Glue and write new!
Good News

The Open Source community has these days solutions for almost everything you can imagine!
But why...
... do many device innovations run out of funding before they can get tested in the market?
... is it hard for device manufacturers to make enough money to provide the "luxury" of timely and automated security updates?
... is it challenging for SoC providers to offer product builders supported software that can be directly used in products at scale?
... do devices we have at home for a while not have the latest software features found in current gen devices?
IT SEEMS IT IS ...
... too complex to make a product!
... too hard to add features to products!
... too pricey to provide security updates!
... too limited collaboration and reuse
What if...
A device builder can focus on investing in just the device experience instead of having to solve everything that it takes to make, ship and maintain a Linux-based product?
A device builder roll software updates to devices in the field with ease and confidence as often as needed?

... just like the Apples, Googles and Ubuntu's of this world?
A device builder *add features to your devices* already in the field at any time without having to convince the user to flash a firmware?
A device builder could rely on others taking care of enabling the hardware platform and keeping the kernel and OS secure?
There was a ready to use factory process for your software stack!
You could collaborate and reuse in an ecosystem of hardware manufacturers, software and service providers?
What if...

There would be an **OS platform** that is optimized for building, shipping and maintaining device products that would solve those problems and more for you?
Snappy Ubuntu Core
Amazing developer experience with snapcraft to build from source Ubuntu Core with snappy transactional updates. Applications confined by Canonical’s AppArmor kernel security system. Safe, reliable, worry free updates with tests and rollback. Amazing developer experience with snapcraft to build from source. Easily extensible with frameworks. Perfect for the smartest IoT.
Snappy System makes updates easy and safe

Current: Updates to a system is a high risk operation that you rather want to avoid

Snappy: Updates to your devices in the field is convenient and can be done with confidence just like done by the Apples, Googles and Ubuntu's

Snappy Platform: enables device builders to implement a modern update strategy: **update through store, canary testing in field; phased updates; auto rollback**
classic device

any package can write to any file

filesystem

snappy devices

app writable area

app writable area

snap

snap

os writable files

writable spaces per snap

read-only snaps

os snap

kernel config

kernel snap
The end of 'one size fits all' 

Leverage a single HW platform for multiple purposes

**Ubuntu Core**

**Certified HW Support**

**Snappy Ubuntu**

**Browsing Appliance**

Browser Snap

Display Framework

Ubuntu Core

Certified HW Support

**Snappy Ubuntu**

**Digital Signage Appliance**

Digital Signage Agent Snap

Browser Snap

Display Framework

Ubuntu Core

Certified HW Support

**Snappy Ubuntu**

**Home Router Appliance**

Device Mgmt Snap

Router Admin UI

OpenWRT Framework

Ubuntu Core

Certified HW Support

**Browser Snap**

**Display Framework**

**Ubuntu Core**

**Certified HW Support**

**Digital Signage Agent Snap**

**Display Framework**

**Ubuntu Core**

**Certified HW Support**

**Device Mgmt Snap**

**Router Admin UI**

**OpenWRT Framework**

**Ubuntu Core**

**Certified HW Support**
The end of the 'early device death'

Add value to devices already in the field

**Snappy Ubuntu Home Router Appliance**
- Device Mgmt
- Snap
- OpenWRT
- Framework
- Ubuntu Core
- Certified HW Support

**Snappy Ubuntu Home Router Appliance w/ Store**
- Device Mgmt
- Snap
- OpenWRT
- Framework
- Parental Control
- Store
- WebDM
- Ubuntu Core
- Certified HW Support

**Snappy Ubuntu Home Router Appliance w/ IoT**
- Device Mgmt
- Snap
- OpenWRT
- Framework
- Parental Control
- Store
- WebDM
- IoTivity
- Ubuntu Core
- Certified HW Support
Snappy Factory Story

Another Problem you don't want to solve...
Snappy Store

Long Lasting Value and Revenue out of "Device Real Estate"
Snappy enables all systems to have large scale app ecosystem without compromising on security.

For that all snaps run in their very own contained & isolated and sandbox.

Snappy provides a framework that surfaces apparmor, seccomp and container technology in a very easy to manage manner.
Snappy surfaces sandboxing features available in Linux to the app developer and system builder as interfaces.

Interfaces allow snap to export and consume features and hw-resources that would otherwise be unreachable in other snaps.

Interfaces get assigned by system builder, by user interactively or automatically.

Interfaces help find suitable software in the store!
Snappy Ubuntu Core and 96boards
Kernel snap as the **core to build upon**

Gadget snap **differentiates** your product

One **unified** application ecosystem
Kernel Snap

- db410c-gadget
- hikey-gadget
- bb gum-gadget

snappy
Mix and match ecosystem: kernels, gadgets and applications

Single tool for image creation, generic instructions to deploy

Abstract away the platform details (ptable, bootloaders, blobs)

Concentrate on your software solution

Collaboration in the developer community
How does this look on the Dragonboard?
Demo and Q/A
Getting Started

http://developer.ubuntu.com/en/snappy/start

Docs:
http://www.ubuntu.com/things
http://developer.ubuntu.com/snappy

Github:
http://www.github.com/ubuntu-core

Mailing Lists:
snappy-devel@lists.ubuntu.com
snappy-app-devel@lists.ubuntu.com

IRC:
#snappy on irc.freenode.net