

# **Convergent Qt applications for Ubuntu**

Andrea Bernabei, UX Engineer @ Canonical

# Before we begin

- This presentation is written in QML (it's basically an app!)
- If you're reading the PDF version, make sure you also check out the QML version
  - it includes a livecoding environment with code samples!
  - it has higher rendering quality
  - [https://github.com/faenil/DevelopingUbuntuConvergentApps\\_presentation](https://github.com/faenil/DevelopingUbuntuConvergentApps_presentation)

## **About me**

Let me tell you a story...

# About me: Andrea Bernabei, a.k.a faenil

- Joined Canonical as **UX Engineer** in March 2015
  - right after getting my Master's degree in Computer Science
  - a **developer**, member of Design Team!
  - C++ UI components and QML prototyping
- Passionate about software development
  - and communities...and tinkering! :)
- 1 year in Canonical
  - London HQ



# **Ubuntu for Devices**

A fully scalable OS



Ubuntu is an open source software platform that runs from the  
cloud, to the smartphone, to all your things

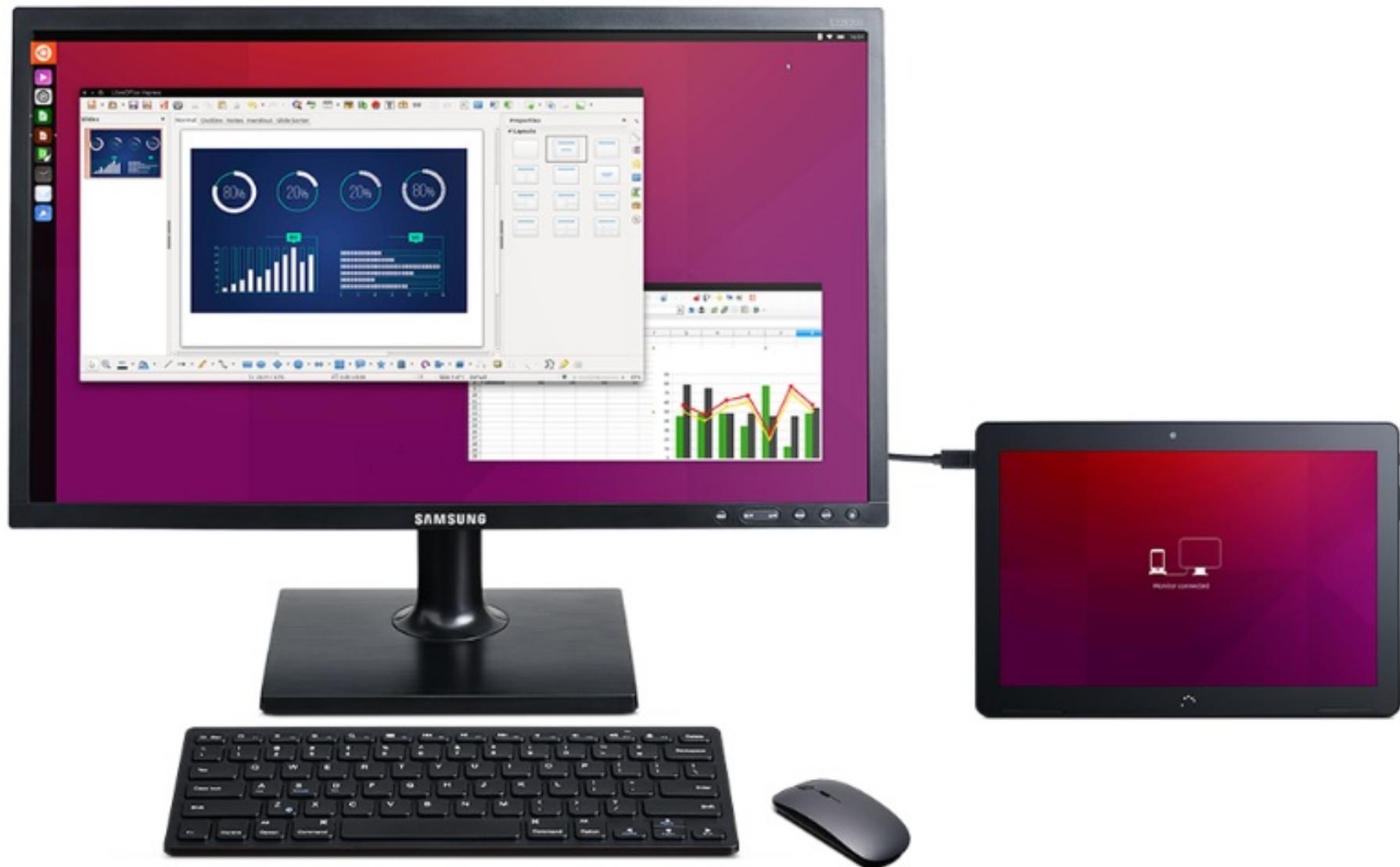
# **Road to convergence**

The future of personal computing

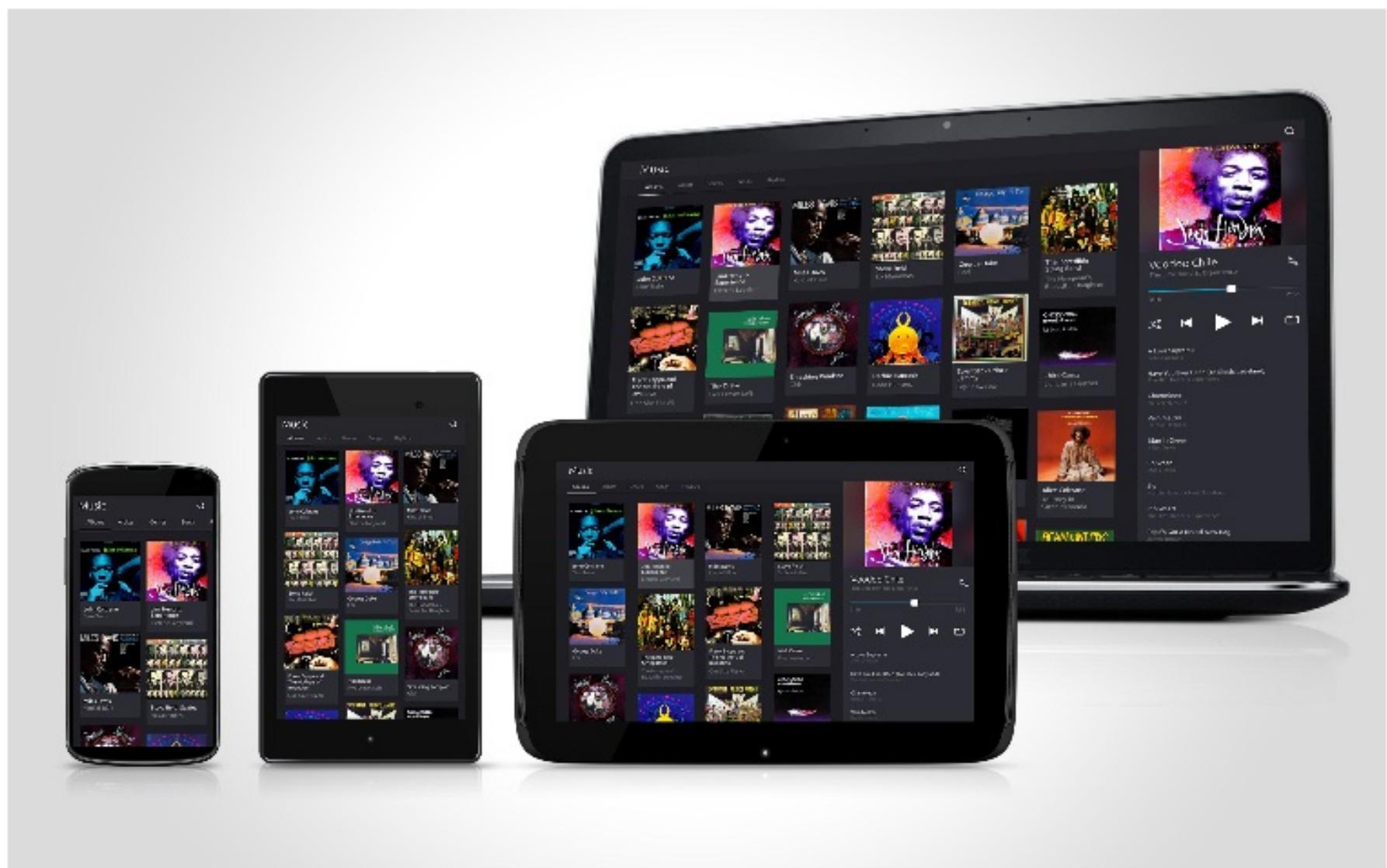
# Everything you need from a PC...in a tablet



# The converged experience



# The converged experience



# **Ubuntu SDK**

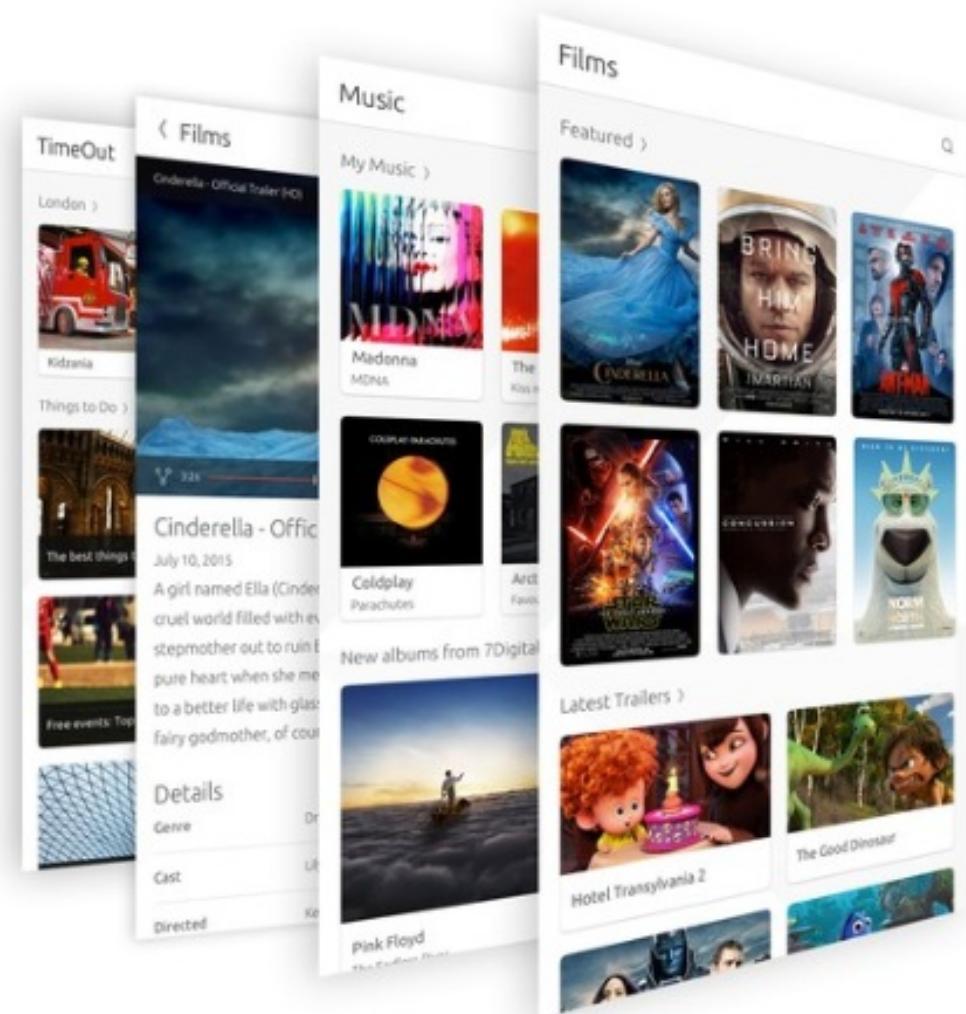
The heart of apps development

# Entry points for developers

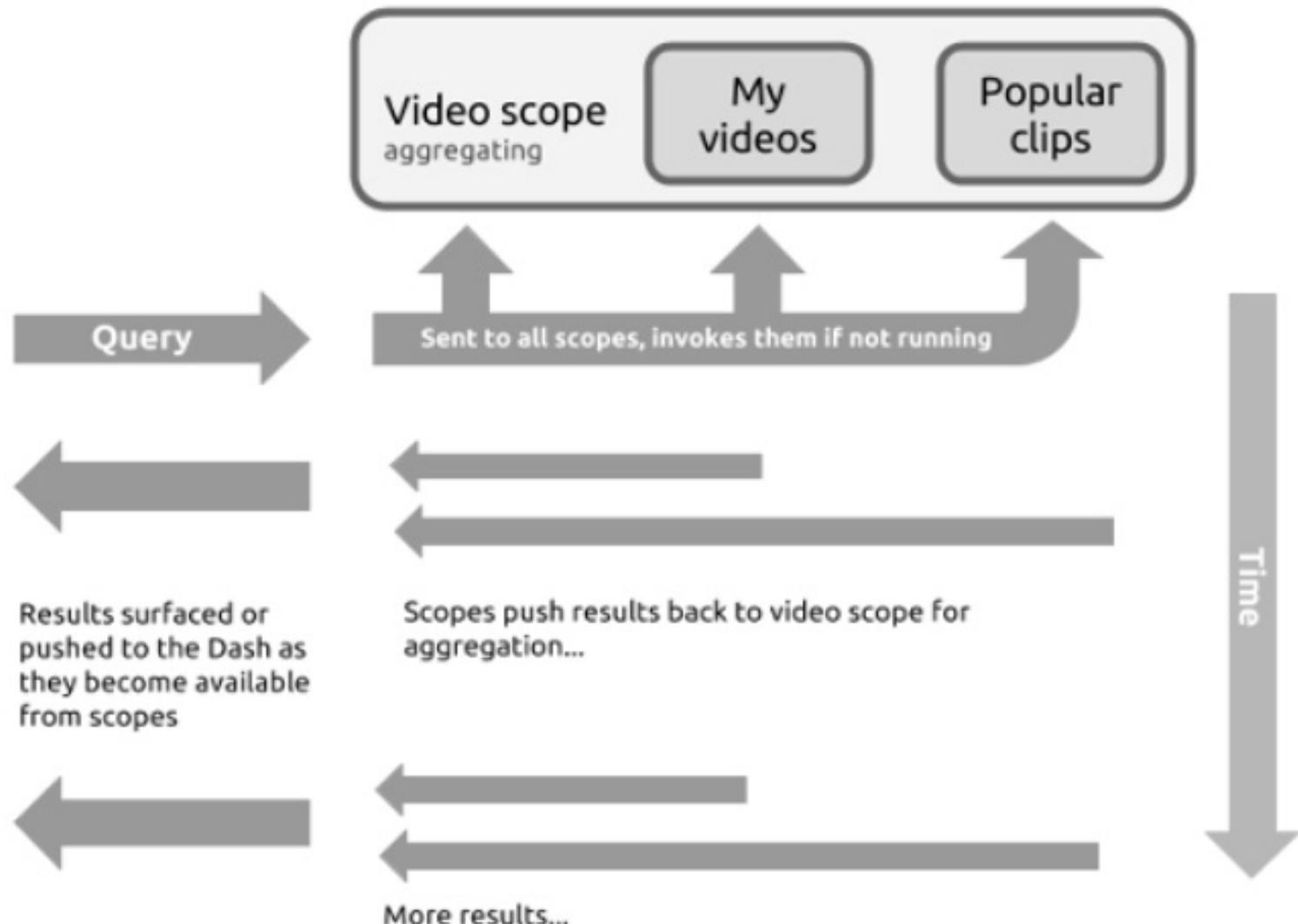
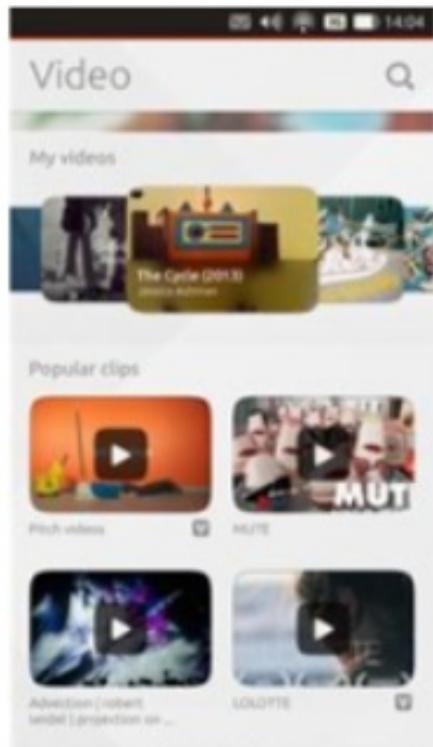
- Scopes
  - we'll give a quick overview...
- Webapps
  - a quick overview...
- **Qt/QML apps.** \o/
  - the real deal, the best and most future-proof framework to build convergent apps

# Scopes

- Shifting away from the apps grid
- Like individual homescreens
  - for different kind of contents
- Life at your fingertips
- Can be implemented in C++, JS, or Go!



# Scopes: data flow



# **HTML5 Apps**

using Cordova API

# HTML5 Apps: Cordova

- Apps run in web container
  - Chromium-powered (Oxide)
  - Isolated cookies/history
  - 509/555 score at [html5test.com](http://html5test.com)
- Ubuntu is an official Cordova platform!
- Familiar toolset for existing Cordova/Phonegap devs
  - More info on Cordova's website: see [goo.gl/27pgHY](http://goo.gl/27pgHY)



# **Native apps**

Powered by Qt/QML

## Native apps: Qt/QML

- Full native OpenGL
- The power and the simplicity of QML/JS
  - plus the efficiency of C++
- **Convergent** Ubuntu UI Toolkit available
- Extensive Platform API
- **Best framework to build first-class convergent apps!**



## **News about Ubuntu SDK**

No more chroot problems!

## Ubuntu SDK Kits: before

- "click chroot" creates a chroot env
- Uses **debootstrap** to create env
  - slow to create
  - requires download/install of all packages
  - breaks often on upgrades (Overlay PPA not checked)
- Uses **schroot** to handle env
  - starts/resumes a session whenever a cmd is run
  - sessions not always cleanly exited (bug?)
    - mountpoints never released, they accumulate over reboots!
  - running commands is slow, because of the sessions
  - session daemon not very reliable
- Only one chroot per architecture (and framework)

# **Ubuntu SDK Kits: now (or, soon!)**

- Now using LXD!
  - LXD: the Linux container hypervisor
  - developed by Canonical
  - LXC-based
    - i.e. kernel is shared with the host
  - super fast, much faster than a VM
    - no virtualization overhead!

## Ubuntu SDK Kits: now (or, soon!)

- LXD containers! \o/
  - image based, very fast container creation
- Can potentially have one container per app!
  - to avoid littering other containers with app specific dependencies
- No need to run apps on the host system anymore
  - you can run them from a desktop container instead!
    - using the X server from the host
    - supports system services and everything
- No need to install development tools on the host
  - the container will hold all what's needed

# **AdaptivePageLayout**

A convergent dual-page stack

# AdaptivePageLayout

- A multiple columns page stack
  - internally a RowLayout of items
  - by default shows 1 or 2 columns based on window size
  - very **flexible**
    - can host multiple column layouts, each with its own sizing properties (min/max size, etc)
    - can override the condition that triggers a layout
- The columns are user-resizable
- Automatic handling of "back" button in the headers, where needed
- Main methods: **addPageToCurrentColumn(sourcePage, newPage)**,  
or **addPageToNextColumn(sourcePage, newPage)**

# AdaptivePageLayout: an example

Contacts + ⌂ ⌂ Andrew Johnson ⌂ ⌂

All Favourites Google Skype

A

 Anthony Chang	 Andrew Johnson <span>★</span>
 Aga Sakota	Phone +44 3069 990451 <span>✉️ 📞</span> Mobile +44 4629 491099 <span>✉️ 📞</span> Work
 Andrew Johnson	Telegram +44 3069 990451 <span>↗️</span>
B	WeChat +44 3069 990451 <span>🔗</span>
 Becki Green	Email j.andrews@gmail.com <span>✉️</span>
 Bill Klein	Invite to: <span>&gt;</span>
 Brian	
C	
 Cecil Brown	
 Camilla	

# AdaptivePageLayout

Livecoding!

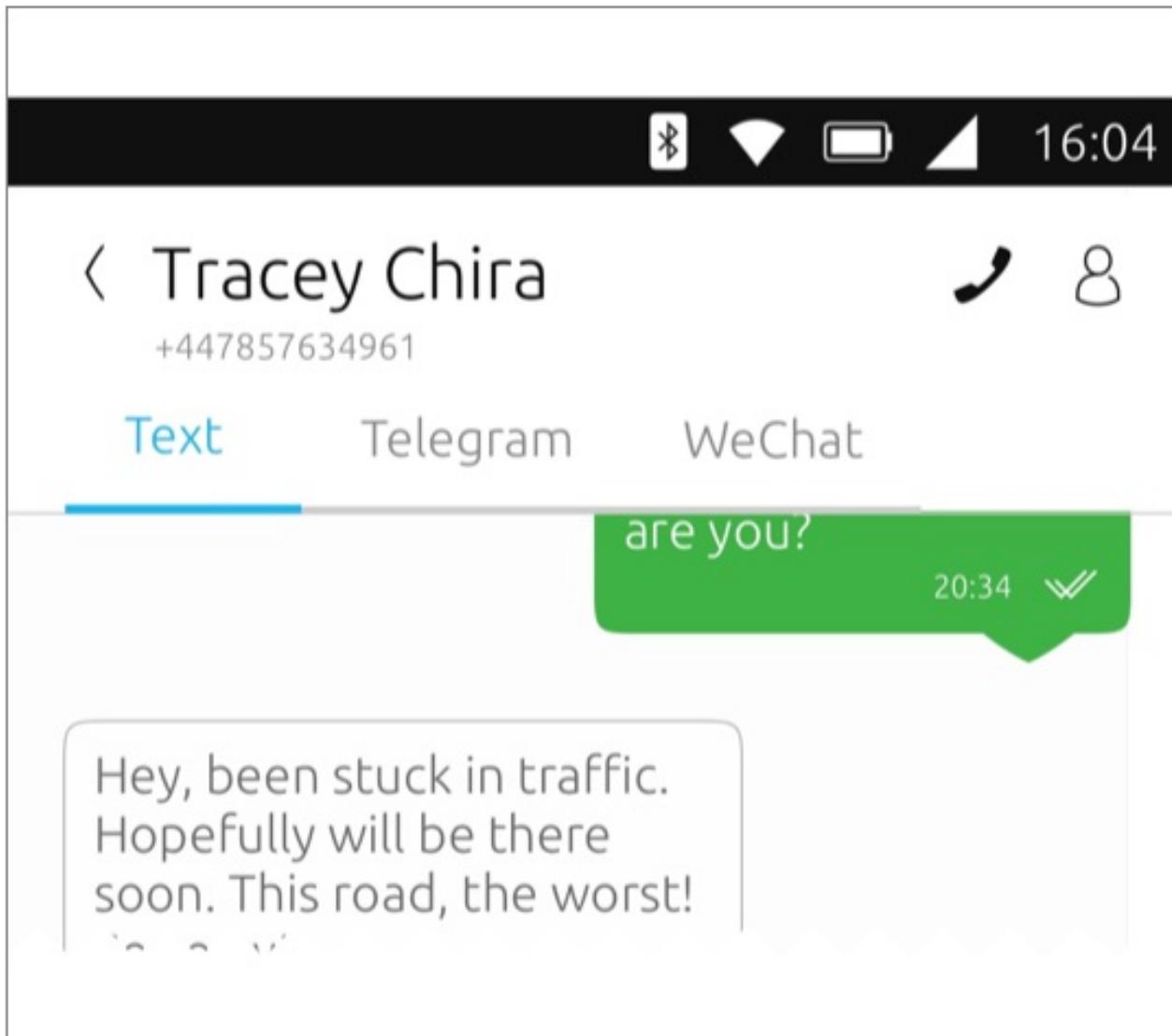
# **PageHeader**

The application header

# PageHeader

- Adapts its size to save precious screen estate
  - shrinks when in landscape on phones
- Allows adding an **extension**
- Dynamically hides/shows based on user interaction with the surface
- Automatic back button handling when used in PageStack/  
AdaptivePageLayout
  - handles multiple columns as well
- Supports **focus states** and **keyboard navigation**
- Time for a demo!

# PageHeader: example of a header with header sections (Text, Telegram, WeChat)



# PageHeader

Livecoding!

## **Showing content: lists**

ListItem, ListItemLayout

## Lists: ListItem

- C++ internals, **16x faster** instantiation than the old ListItems.Empty component!
- Provides swipable actions: leading and trailing
  - also via mouse right click
- Automatically handles divider
  - autohides in last element of a list
- **Drag&Drop** is super easy! See **ViewItems** attached property
- **Multiselection** is super easy! See **ViewItems** API
- Features **expansion** mode to accomodate additional content
- Supports **focus states** and **keyboard navigation**

## ListItem: an example of swipable actions

**Anthony Chang (2)**  
Want to grab lunch today?

13:33

 **Brian (1)**  
You can take me instead

12:55



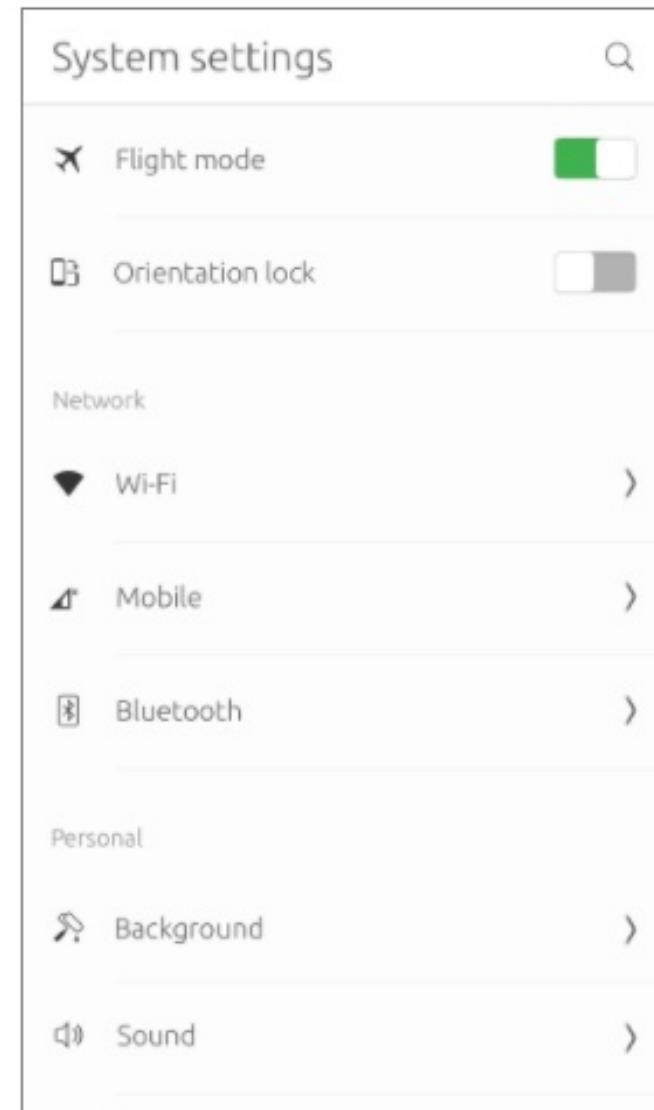
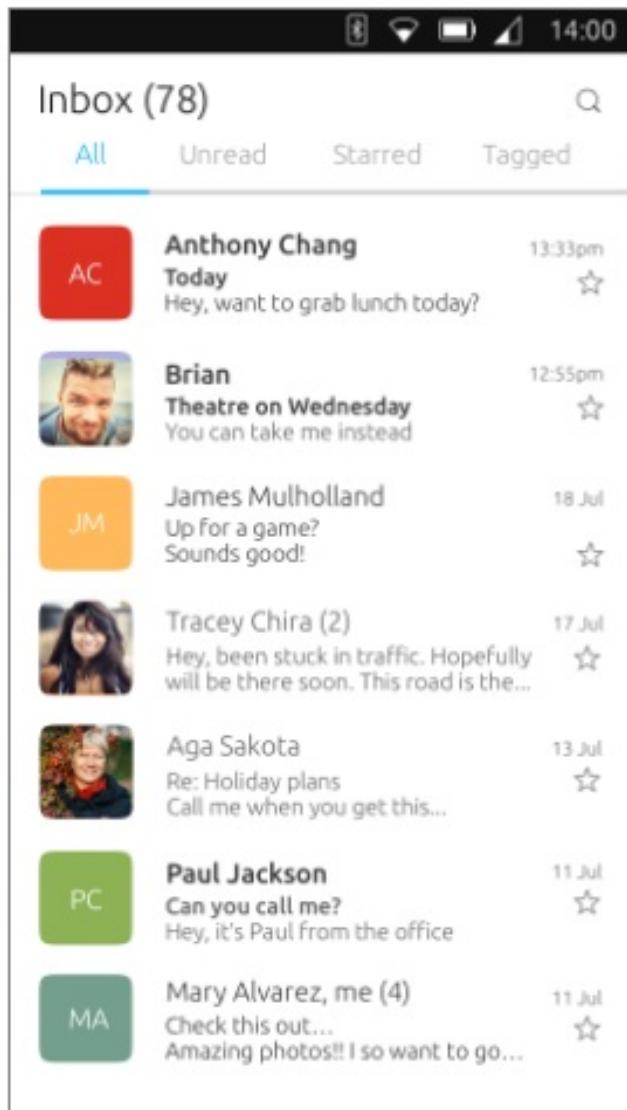
# ListItem

Livecoding!

## Lists: ListItemLayout

- C++ internals, QML API, perfect ListItem buddy (**ListItem** defines **swiping actions**, **ListItemLayout** defines the **main content!**)
- **ListItem with 4 swiping actions + ListItemLayout holding 3 labels and 3 Items are still 3x quicker to instantiate than old QML ListItems.Empty component (empty!)**
- Provides automatic padding and positioning for list-items content
  - following visual rules defined by Design Team
  - layout is driven by **mainSlot** (child)
- Provides up to 3 labels with predefined styling
- Flexible and customizable, covers most of the usecases
  - dev can tweak both per-child and global paddings
  - for the rest, see **SlotsLayout**

# ListItemLayout: examples



# ListItemLayout

Livecoding!

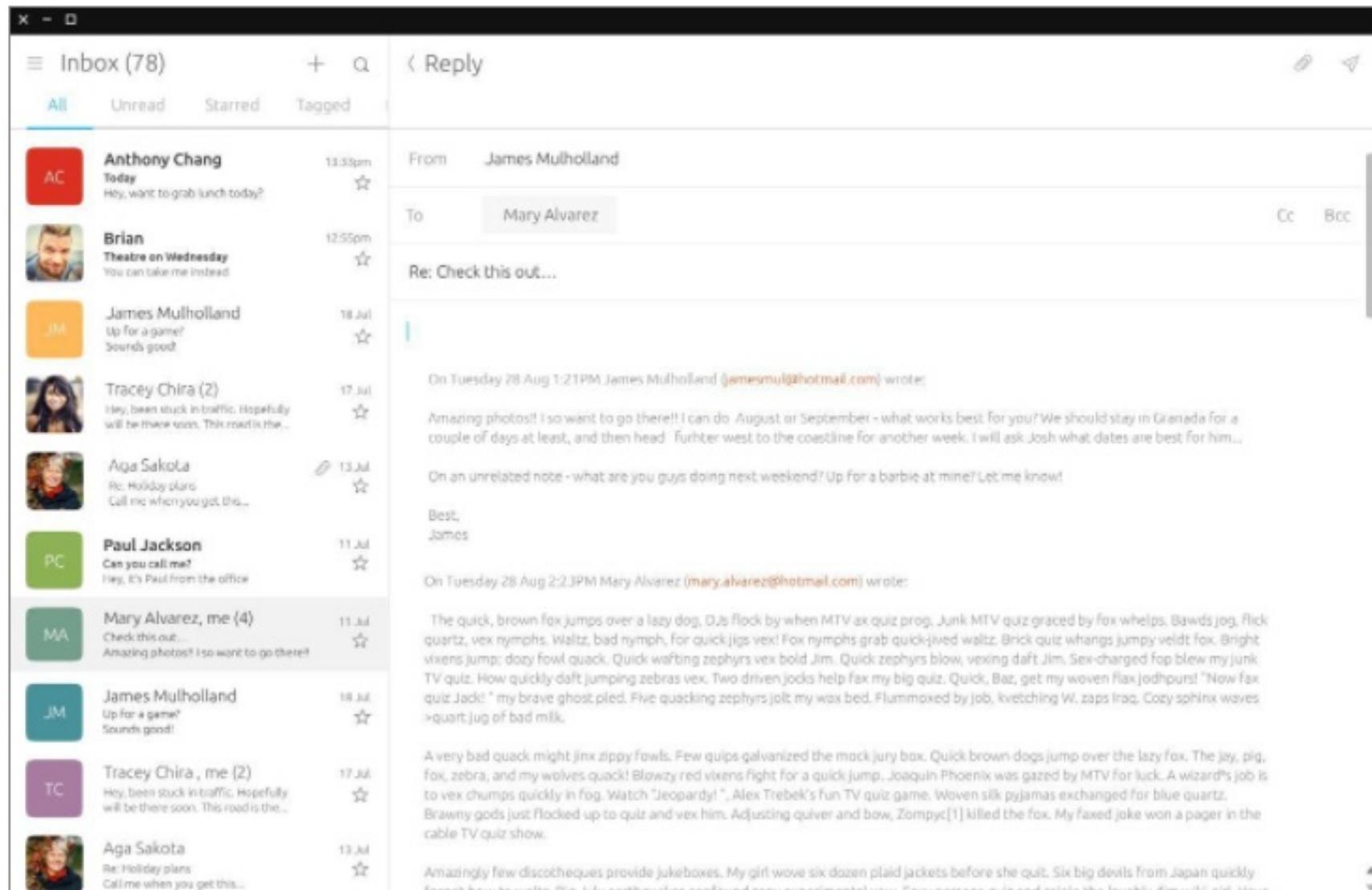
# **Handling scrollable surfaces**

ScrollView

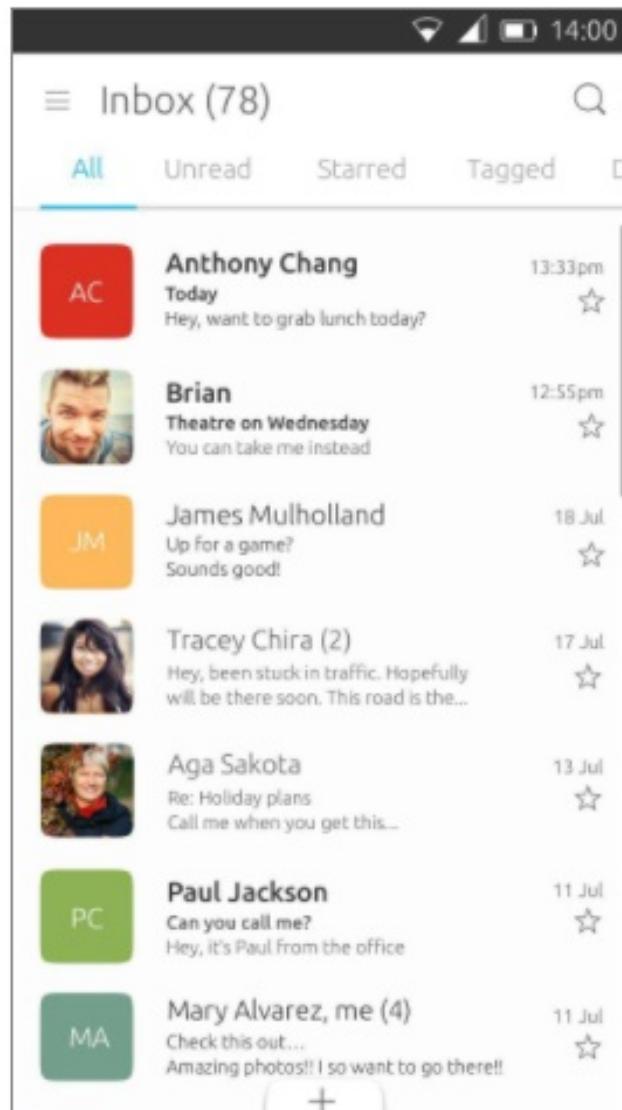
# ScrollView

- Super easy to use API, just wrap your scrollable item in a ScrollView{...}
- 3 states: indicator, thumb, steppers (desktop-y)
- Active state depends on the input device
- Scrolling indicator when using **touch**
- Desktop friendly thumb+steppers when using **mouse**
- Handles scrolling via **Keyboard** keys

# ScrollView convergence: dynamically adapts between mouse-friendly...



# ...and touch-friendly



# ScrollView

Livecoding!

## More cool stuff

- Advanced style customization
  - via **StyleHint** component
- Colour Palette with semantic names
- WIP: input devices detection from QML
  - undergoing security review
- Minimum **touch** area for components deriving from AbstractButton
  - mouse area not affected!
- **Mnemonics** and **shortcut keys** handling
  - via Action component

# **Verifying your app is convergent**

Checklist

## Verifying your app is convergent

- Make sure your UI scales across different screens and form factors
  - Use **units.gu(val)** for sizes, not pixels!
  - Use Label's textSize to make labels scale beautifully
- Handle focus states
- Handle keyboard UI navigation (where appropriate)
- Consider adding main action to PageHeader instead of using **BottomEdgeHint** when pointer device is detected

# Verifying your app is convergent

- Tailor UX to different input devices
  - e.g. don't mix touch and mouse
  - check out **Mouse.ignoreSynthesizedEvents** in Ubuntu Components!
- Tailor each UX journey to be comfortable to use with the input device that started it
  - e.g. menus opened via touch should have taller menu items than menus opened via mouse
  - NOTE: the menu should keep touch-friendly UI until it's closed and then reopened via another device

# Convergent UX: popover - touch

A screenshot of a web browser window. The title bar reads "Self-driving cars: safe, reliable - but a challenging sell for Google | Techology |". The address bar shows the URL "https://en.wikipedia.org/wiki/Ubuntu\_(operating\_system)". A context menu is open over the URL bar, containing "Cut", "Copy", and "Paste" options. The main content area displays the Wikipedia article for "Ubuntu (operating system)". The page title is "Ubuntu (operating system)" and it starts with "From Wikipedia, the free encyclopedia". The text describes Ubuntu as an operating system and distribution, mentioning Unity as its default desktop environment. The sidebar on the left contains links like "Main page", "Contents", "Featured content", etc.

Ubuntu (operating system)

From Wikipedia, the free encyclopedia

This article is about the operating system. For other uses, see [Ubuntu \(disambiguation\)](#).

**Ubuntu** (originally [/ubuːntu/](#) [uu-BOON-tuu](#), according to the company website [/ubuːntu/](#)) is a free and open-source Linux operating system and distribution, with [Unity](#) as its default desktop environment. It is developed by the [Ubuntu Project](#). Ubuntu also runs [network servers](#). It is based on [free software](#) and named after the [Ubuntu](#) concept, which often is translated as "humanity towards others" or "the belief in a universal".

# Convergent UX: popover - pointer

A screenshot of a web browser window. The title bar reads "Self-driving cars: safe, reliable - but a challenging sell for Google | Techology |". The address bar shows the URL "https://en.wikipedia.org/wiki/Ubuntu\_(operating\_system)". A light blue popover is positioned over the address bar, containing the text "Cut", "Copy", and "Paste" with a cursor hovering over "Copy". The main content area displays the Wikipedia article for "Ubuntu (operating system)". The page header says "Ubuntu (operating system)" and "From Wikipedia, the free encyclopedia". The text begins with "This article is about the operating system. For other uses, see Ubuntu (disambiguation)". Below the text, there is a summary of the Ubuntu operating system, mentioning its name origin, distribution, desktop environment, and network servers.

Ubuntu (operating system)

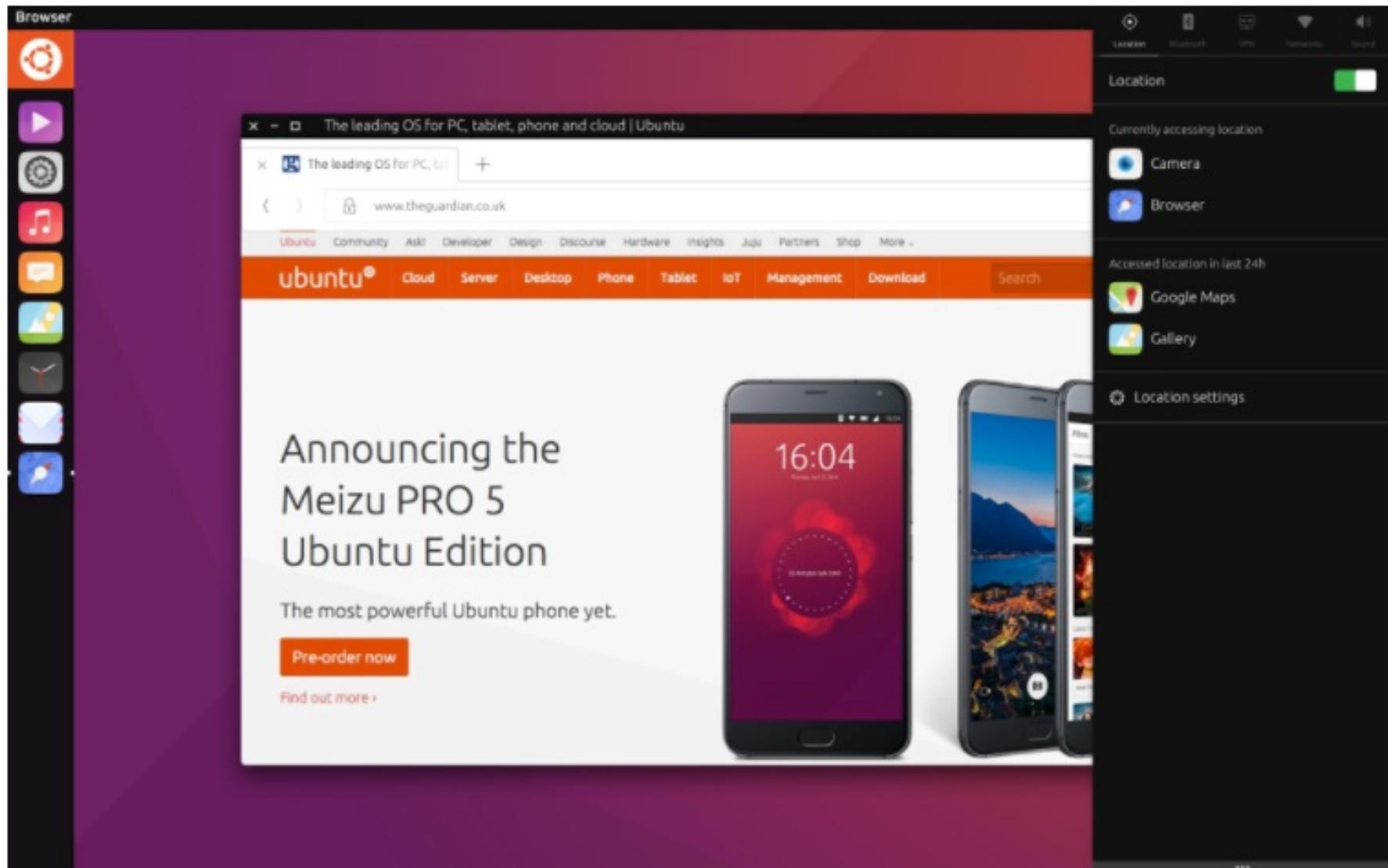
From Wikipedia, the free encyclopedia

This article is about the operating system. For other uses, see Ubuntu (disambiguation).

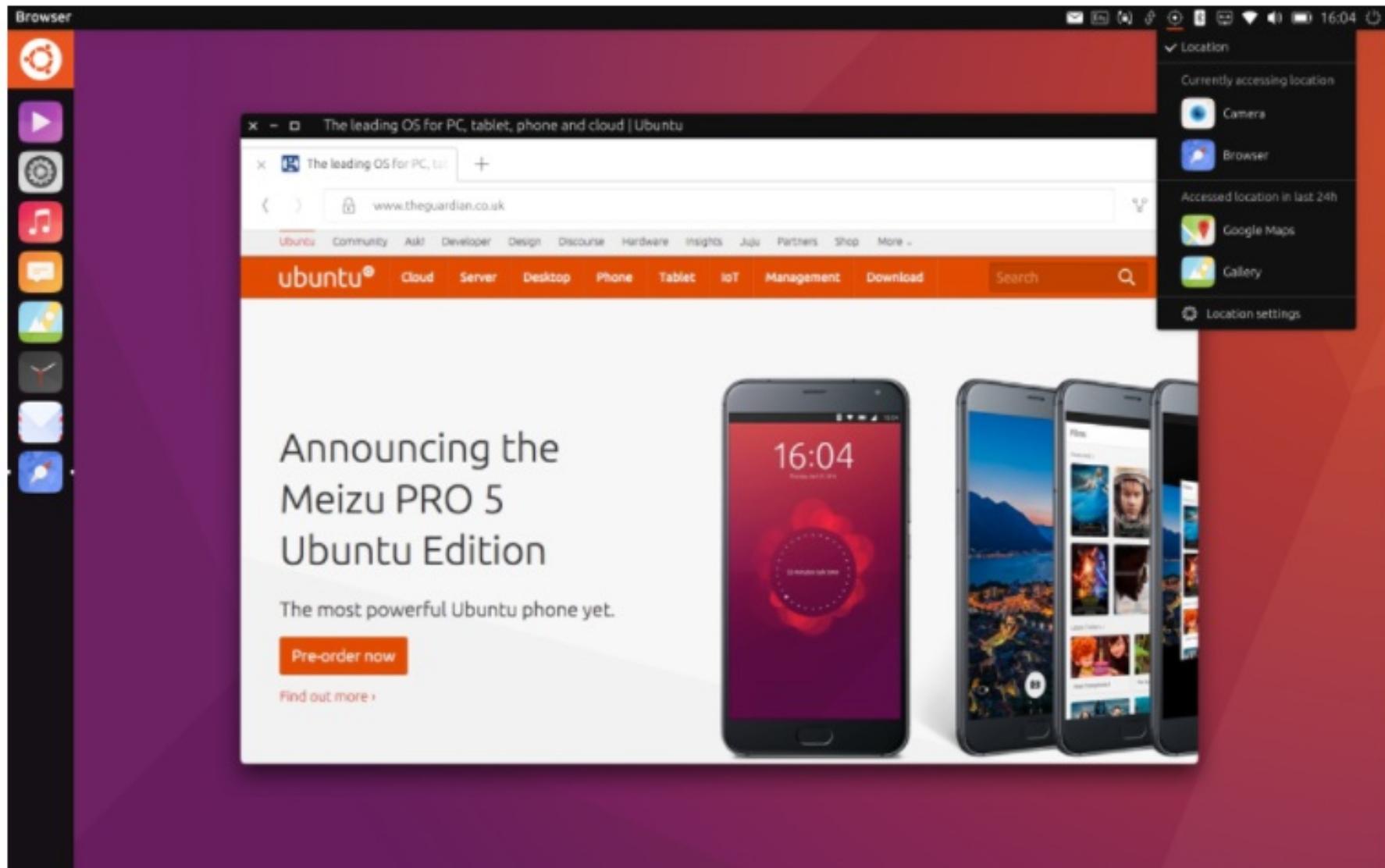
Ubuntu (originally /ʊˈbuːntu/ *uu-BOON-tuu*, according to the company website /ubu/) is a family of open-source Linux-based operating systems and distributions, with Unity as its default desktop environment for desktop computers. Ubuntu also runs network servers. It is based on free software and named after the Swahili word for "humanity towards others" or "the belief in a universal brotherhood of all humankind".

Main page  
Contents  
Featured content  
Current events  
Random article  
Donate to Wikipedia  
Wikipedia store

# Convergent UX: indicators - touch



# Convergent UX: indicators - pointer



# **The future of Ubuntu SDK**

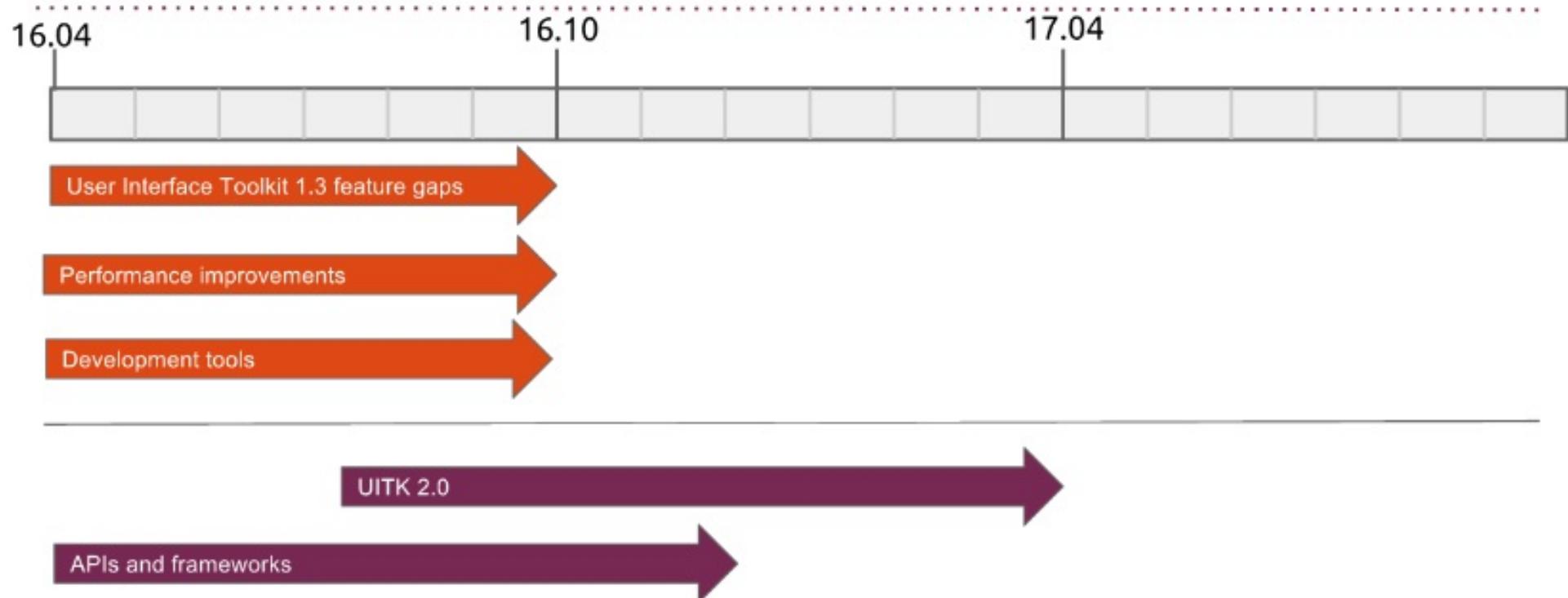
## Roadmap

## The future of Ubuntu SDK

- Priority to improvement of app startup time
  - by optimizing components logic
- Continue conversion of components logic to C++
  - faster than QML
- Move Click chroots to LXD
  - Finally! No more schroot issues!
  - Coming **soon!**
- Planning UITK 2.0
  - consider rebasing on QtQuickControls 2.0
- Keep adding convergent features to components
  - improve focus states handling
  - improve keyboard navigation

# The future of Ubuntu SDK

## Roadmap



# **Frameworks**

## Targeting APIs

# Frameworks

- A "contract" between app and platform
  - "My app needs this API"
- Version of Ubuntu on your laptop != version of the framework
  - 14.04 laptop can target 15.04 framework
    - thanks to SDK's Device Kits
- At the end of the day: mainly used as a way to filter apps at Store level.

# **Beyond UI Components...**

## Integrating with the platform

# **Security at its core**

## AppArmor

## AppArmor

- Apps run in a restricted sandbox
- Can only access its own data
  - and import content from other apps via ContentHub
- Cannot access Privileged APIs (Telepathy, etc)
- Can request user permission to access APIs such as Location and Online Accounts
  - IMPORTANT: user gets security prompt in-context (not at installation time)

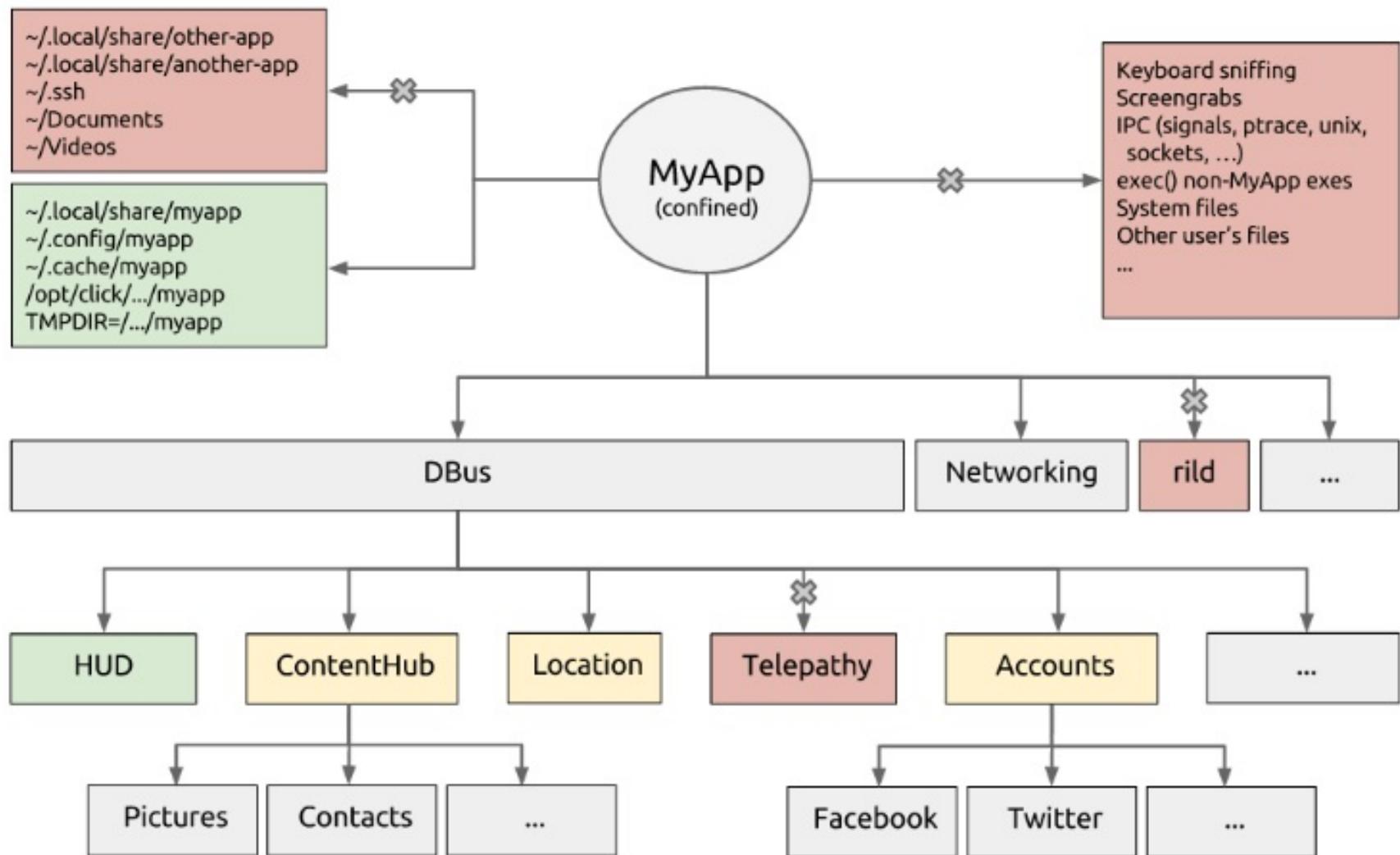
# AppArmor: API access

- Opt-in API:
  - requires user-permission
  - use app's manifest to specify what you need
  - examples: accounts, audio, mic, location, camera, sensors, content\_exchange...
- Reserved API:
  - requires manual review in Store
  - calendar, contacts, sms/calls history...
  - mostly used by Core apps

## AppArmor: filesystem access

- Apps have R/W access to:
  - XDG\_CACHE\_HOME/<APP\_PKGNAME>
  - XDG\_CONFIG\_HOME/<APP\_PKGNAME>
  - XDG\_DATA\_HOME/<APP\_PKGNAME>
  - XDG\_RUNTIME\_DIR/<APP\_PKGNAME>
  - XDG\_RUNTIME\_DIR/confined/<APP\_PKGNAME> (for TMPDIR)
- All paths exposed via QStandardPaths

# AppArmor: summary

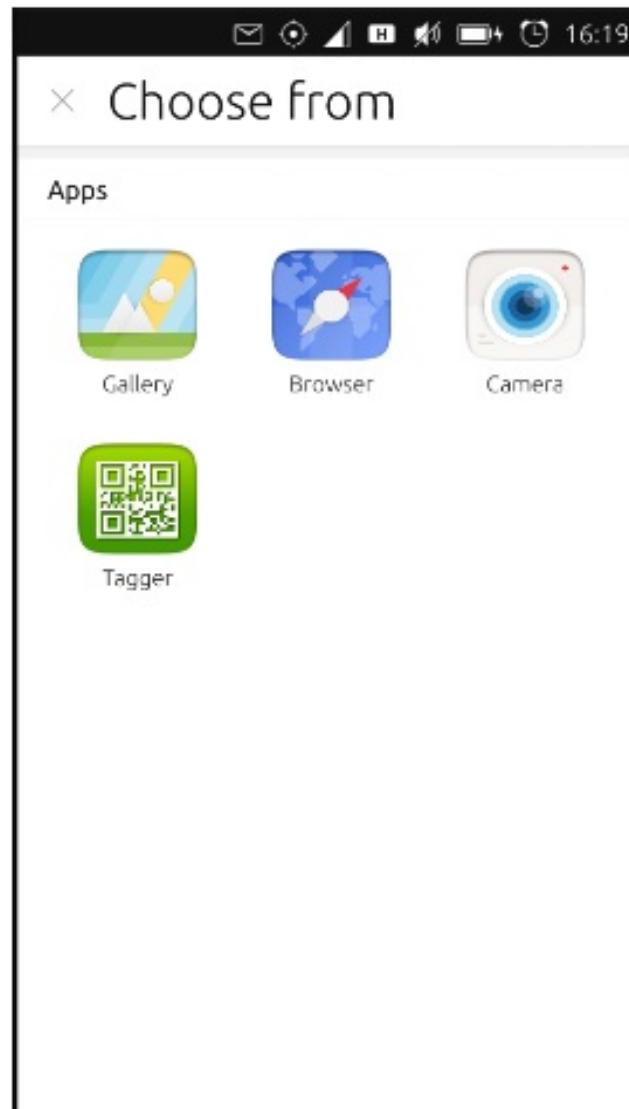


## **Sharing content between apps**

The Content Hub

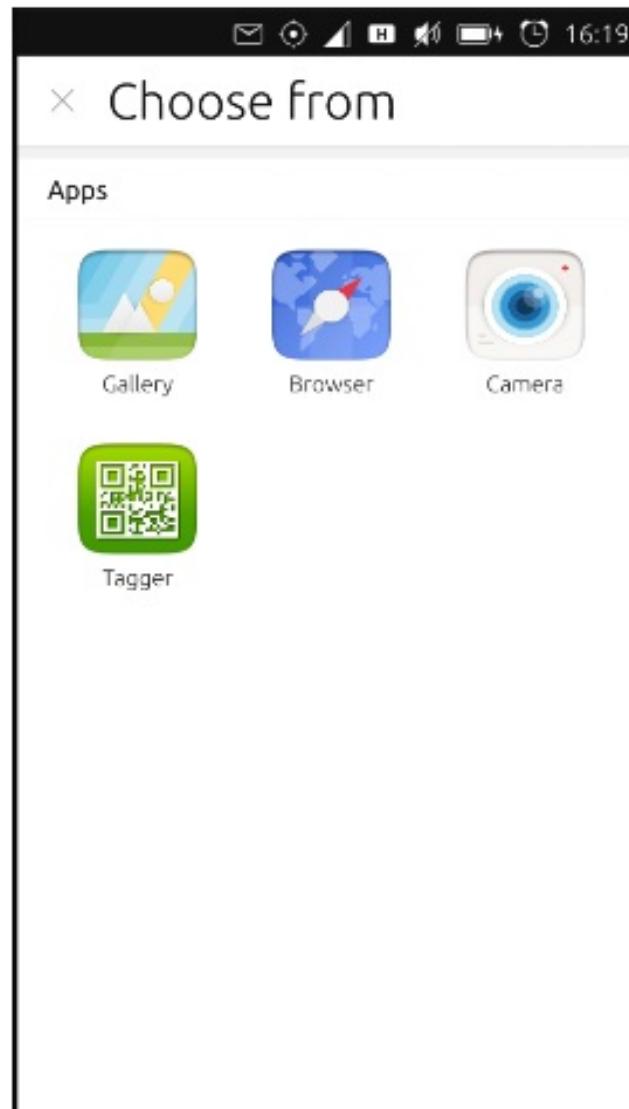
# Sharing content: the Content Hub

- Every piece of content is owned by an app
  - no single app owns everything
- Each content type has a default app
- Apps register as source or destination for a content type
  - using app's manifest



# Sharing content: the Content Hub

- Content Hub handles the UI and actual transfer of data
- Imported content is discarded after use by default
  - use ContentStore API to save it
- Don't forget to add AppArmor policy (`content_exchange`, `content_exchange_source`)



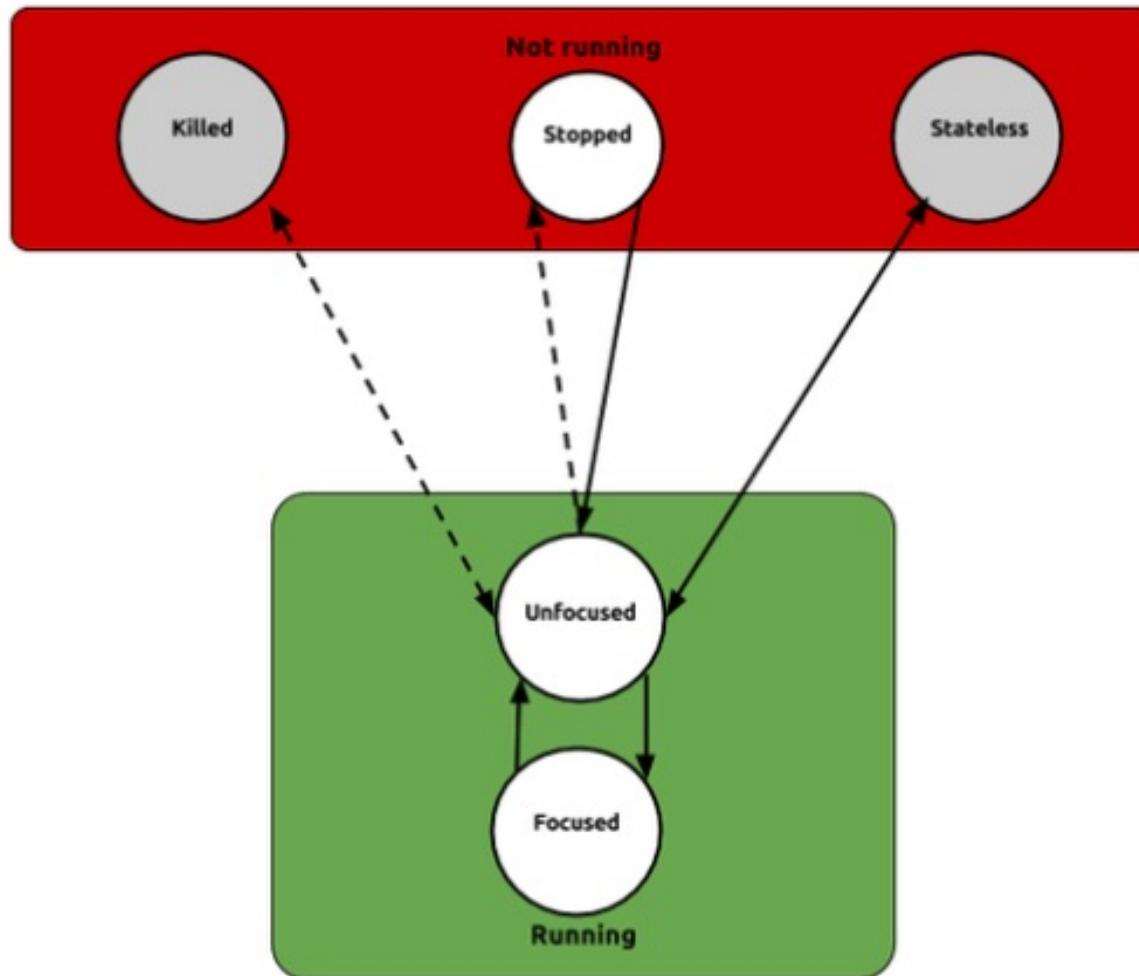
# Multitasking

Handling your application lifecycle

## Handling your application lifecycle

- Application started using their Upstart job to ensure:
  - correct env
  - confinement
  - single-instance
- App can be in non-running (Stopped, Stateless, Killed) or running (Focused, Unfocused) states
  - apps are notified on state transitions, so that they can store/restore state
  - see **StateSaver** SDK component!

# Application lifecycle



## **Deploying your app**

Let's test our creation!

## Deploying your app: easy!

- Tools->Options->Ubuntu->Create Click Target
- Choose armhf or amd64 target arch
- Configure your project to use the just created Kit
- Plug in and setup USB access to the device
- Press the green arrow (Run, CTRL+R)
- **Enjoy!**

## **Internationalization**

Make your app available in multiple languages

# Internationalization

- `i18n.tr("String %1").arg(firstVar)`
- SDK will create .pot file with all strings that need translating
- Prefix code comments with //TRANSLATORS:
  - to leave info about context of use
- See QML Locale documentation for useful methods
- .po files hold the translations, one file per language
- app only ships .mo binary file, built by SDK

## **Publishing your app**

Make it available to all users

# Publishing your app

- Register to Ubuntu MyApps
- Build the package using Ubuntu SDK
- Run automatic checks using SDK
  - Publish->"Build and Validate package"
- Upload .click package to MyApps
- **Publish!**

The screenshot shows the Ubuntu MyApps website. At the top, there is a navigation bar with links for "ubuntu® myapps", "Ubuntu Core", "Ubuntu Personal", "Desktop", and "Sign in or register". Below the navigation bar, a large heading says "Sign in to see your packages". A subtext explains: "The Ubuntu Store will help you submit your packages to Ubuntu devices and it will give you an overview of their status. After approval they are available within seconds." There is a red "New package" button. On the right, a box titled "Participate on AskUbuntu" describes it as a "collaboratively-edited question and answer site for Ubuntu users and developers. 100% free, no registration required." It includes a "Ask a question now!" button and a "Report a bug on this site" link. A sidebar on the left lists "Get help" resources: "Publishing an app", "Packaging click apps", "Choosing a license", and "Other forms of submitting apps". To the right of these are links: "Get started", "Application status", "Creating a good icon", and "Security policy for click packages".

# **Ubuntu's roadmap**

What does the future look like?

# Roadmap

**Ubuntu**

.deb based, Unity 7



**Ubuntu Personal**

snappy based, Unity 8



**Ubuntu Phone**

.click based, Unity 8



support

migration

.....

14.10

15.04

15.10

16.04

# **Ubuntu's Community**

The heart of the platform

## Ubuntu's Community

- Joyful and respectful community developed around Ubuntu Touch
- Many of the core apps are developed by community members!
  - Canonical provides Design support
- We wouldn't be here without you, **thank you!**
- Join the discussions on mailing lists, report bugs, give feedback!
  - **We do listen! :)**
- **Join Ubuntu Online Summit, 3-5 May 2016**
  - [summit.ubuntu.com](http://summit.ubuntu.com)

# Credits

# Credits

- SDK Team at Canonical
  - for their feedback on the presentation
- my friend Timur Kristóf
  - for the idea (and code) of using particle effects on cursor position change in the livecoding editor
  - check him out! [github.com/venemo](https://github.com/venemo)
- Gunnar Sletta
  - for the QML presentation system I used as a base
- Community Team at Canonical
  - for providing guidance on the presentation

**That's all, folks!  
Any questions?**

**[andrea.bernabei@canonical.com](mailto:andrea.bernabei@canonical.com)**

**@faenil on Twitter  
faenil on FreeNode IRC**

**Check out the API online!  
<https://developer.ubuntu.com/api/apps/qml/>**

**#ubuntu-app-devel IRC @ FreeNode**

**Mailing list: <https://lists.launchpad.net/ubuntu-phone/>**