

Mobile Information Systems

Lecture 11: Case Studies

© 2015-23 Dr. Florian Echtler
Bauhaus-Universität Weimar
Aalborg University

Case studies

- Inside the Amazon Dash button
- Tweaks without root access
- COVID-19 Contact Tracing
- Apple AirTags & “Find My”
- Mobile Card Emulation
- Computational Photography

Inside the Amazon Dash Button (1)

Image source (FU): <https://hackaday.com/2015/05/12/inside-the-amazon-dash-button/>

- Amazon Dash Button: prototypical IoT device
- Single button for ordering a single product

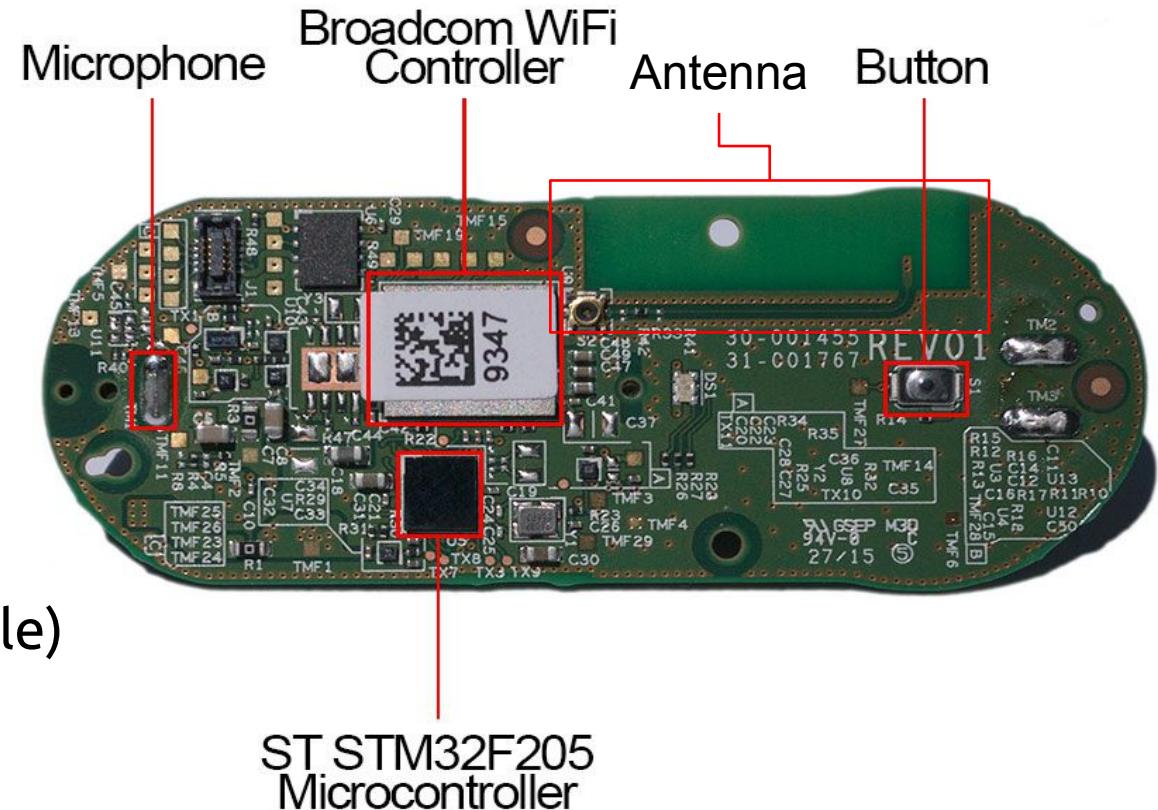


Inside the Amazon Dash Button (2)

Image source (FU): <https://www.cnet.com/news/appliance-science-how-the-amazon-dash-button-works/>

- **Contents:**

- Microcontroller (reprogrammable)
- Board antenna
- WiFi module
- Button (duh)
- Microphone (why?)
- Battery (non-rechargeable)



Inside the Amazon Dash Button (3)

- Chicken-and-egg problem for IoT devices:
 - Configuration via WiFi
 - No regular UI available

→ how to setup WiFi (network, password etc.)?
- Possible solutions:
 - Temporary local hotspot
 - Side-channel configuration

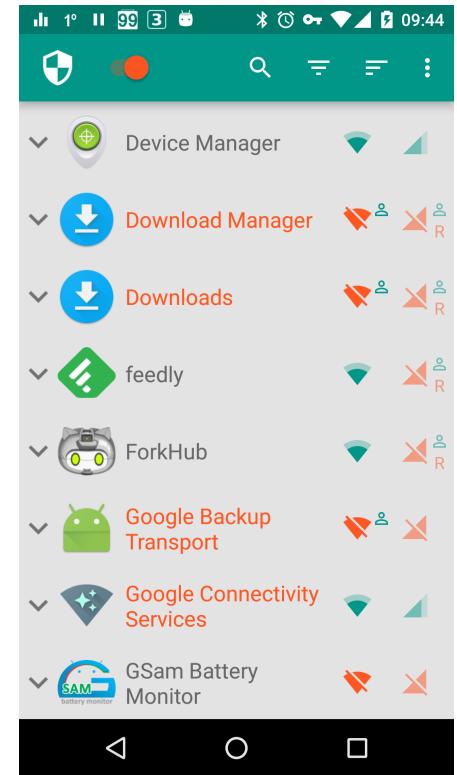
Inside the Amazon Dash Button (4)

- Approach 1: local hotspot (Android)
 - Dash creates own network hotspot
 - Mobile device connects, configures WiFi
 - Dash closes hotspot, connects to WiFi as client
- Approach 2: side-channel (iOS)
 - iOS app sends configuration via ultrasound
 - Dash button listens for specific sound patterns
 - Extracts configuration, connects to WiFi as client
- [Approach 3: misuse of WiFi layer 2 frames, “ProbMe”]

Tweaks without root access (1)

Image source (FU): <https://www.netguard.me/>

- Netguard: User-Level Network Filter
 - Android blocks root-level access to network (unless device is rooted)
 - Loophole: VPN apps allowed at user level
 - Can be (ab-)used for network filtering
 - Helpful for blocking ads/trackers



Tweaks without root access (2)

Image source (FU): <https://play.google.com/store/apps/details?id=com.urysoft.pixelfilter>

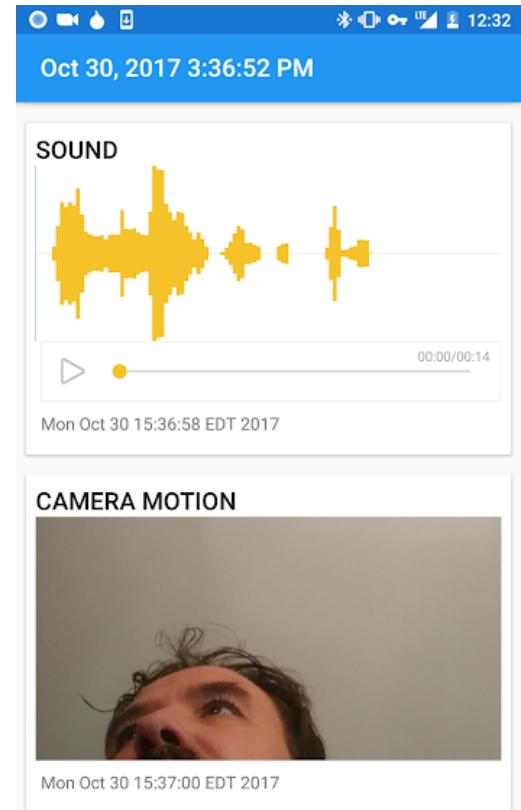
- PixOff: AMOLED Battery Saver
 - Selectively turn off individual pixels
 - Doesn't work for LCD screens (why?)
 - Various patterns, depending on user pref.
 - Currently not available in Play Store?



Tweaks without root access (3)

Image source (FU): <https://play.google.com/store/apps/details?id=org.havenapp.main&hl=en>

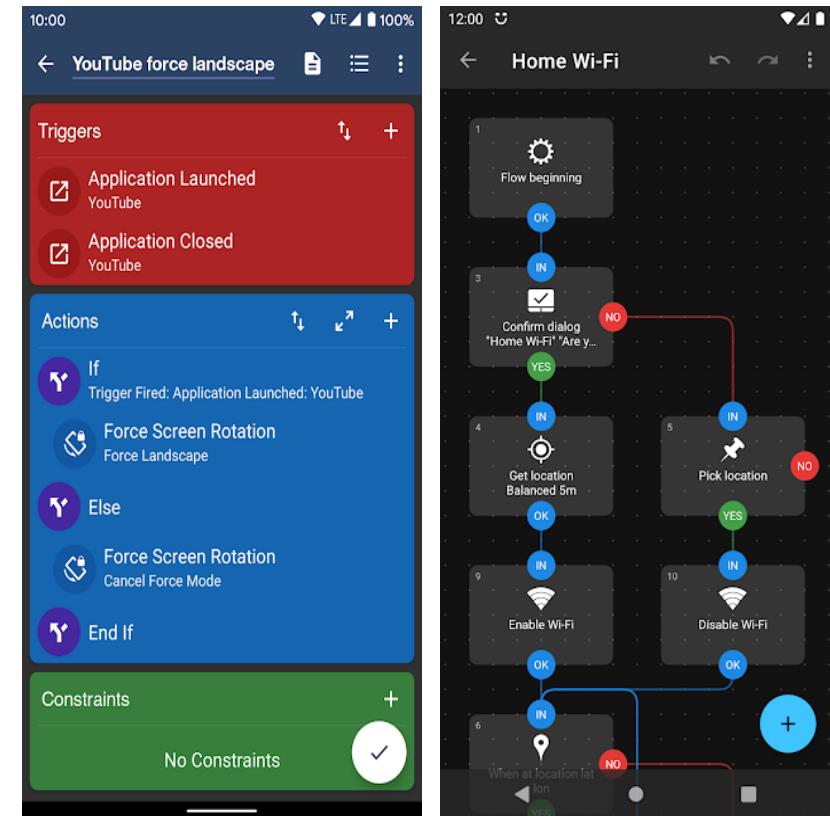
- Secondary use for old devices
 - Kiosk mode apps
 - Lock device to single app, e.g. music player, web browser
 - Useful to create “appliances”
 - Room surveillance
 - Use camera, microphone, IMU to watch for intruders
 - Example: Haven (open-source) →



Tweaks without root access (4)

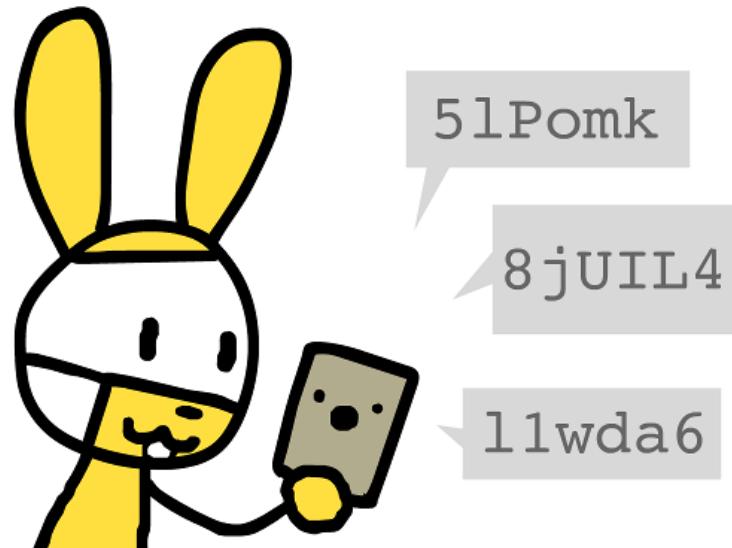
Image source (FU): <https://play.google.com/...com.arlosoft.macrodroid>, <https://play.google.com/...com.llamalab.automate>

- Macro/automation tools
 - e.g. MacroDroid (l), Automate (r)
 - Sometimes approaching full-blown development environment
 - Usability issues?
 - Possible to e.g. accidentally lock yourself out of your own device



COVID-19 Contact Tracing (1)

Image source (PD): <https://ncase.me/contact-tracing/>



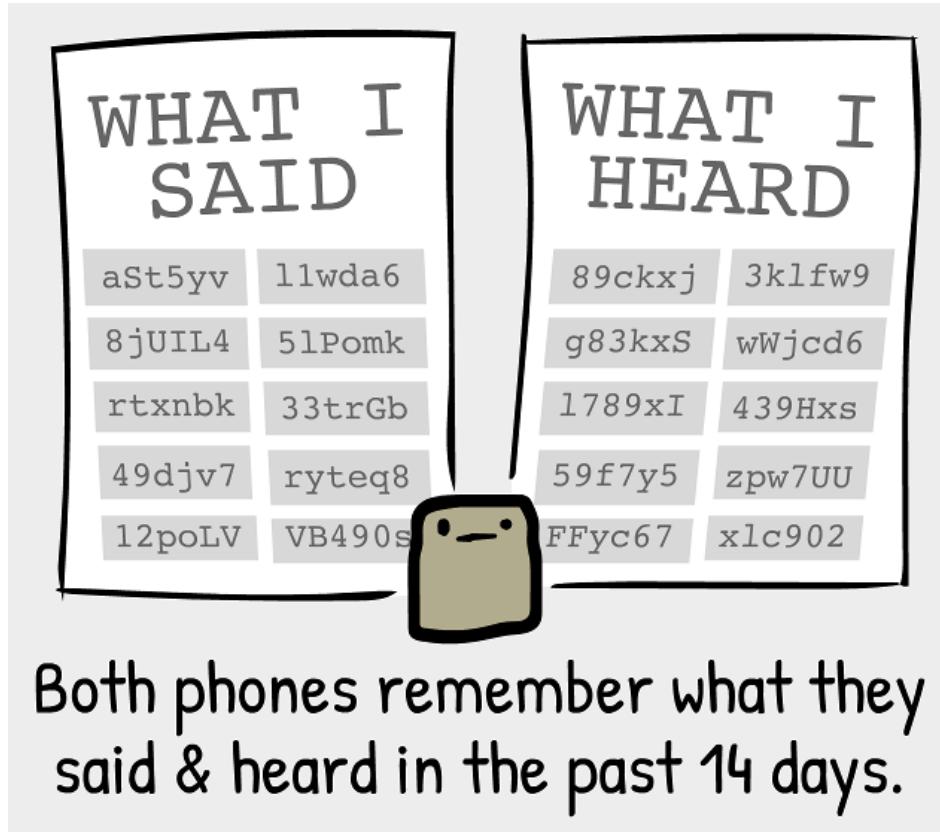
Alice's phone broadcasts a random message every few minutes.



Alice sits next to Bob. Their phones exchange messages.

COVID-19 Contact Tracing (2)

Image source (PD): <https://ncase.me/contact-tracing/>



If Alice gets Covid-19, she sends her messages to a hospital.

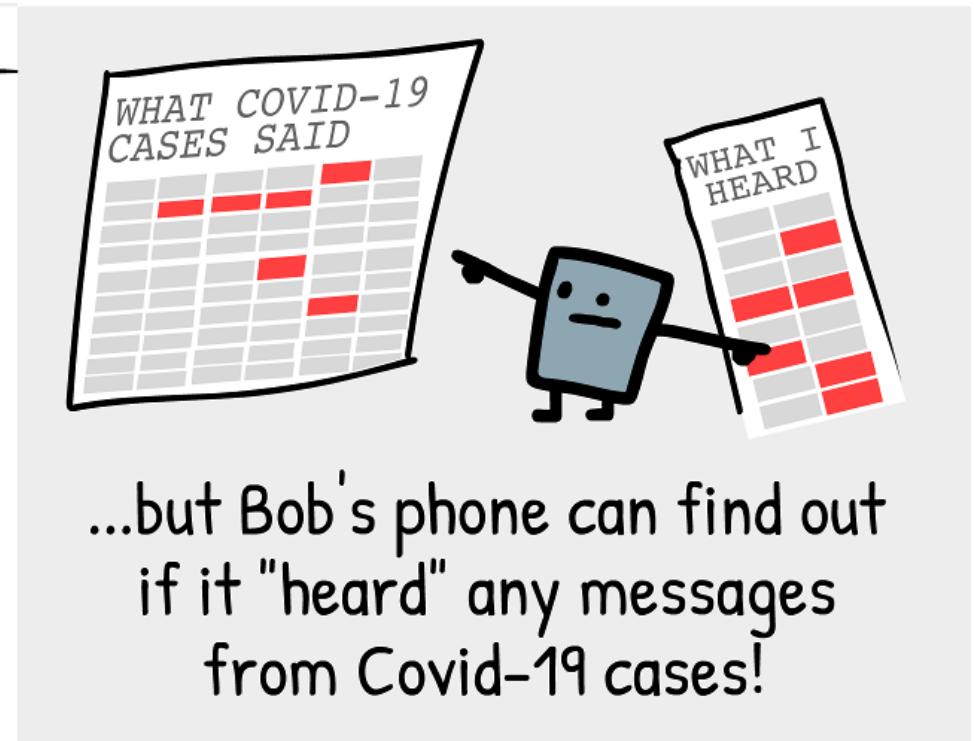
COVID-19 Contact Tracing (3)

Image source (PD): <https://ncase.me/contact-tracing/>

WHAT COVID-19 CASES SAID

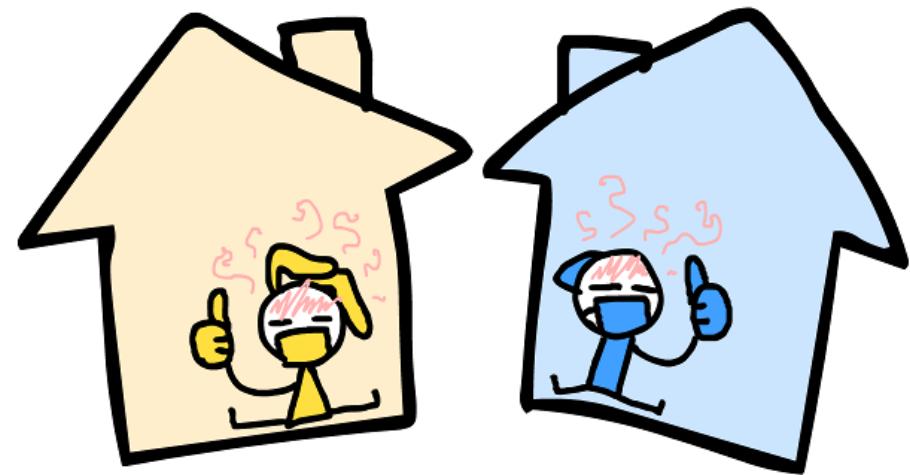
Because the messages are random, no info's revealed to the hospital...

A large grid labeled "WHAT COVID-19 CASES SAID" is shown. A small blue phone icon with a speech bubble saying "WHAT I SAID" has an arrow pointing to the top-left corner of the grid.



COVID-19 Contact Tracing (4)

Image source (PD): <https://ncase.me/contact-tracing/>

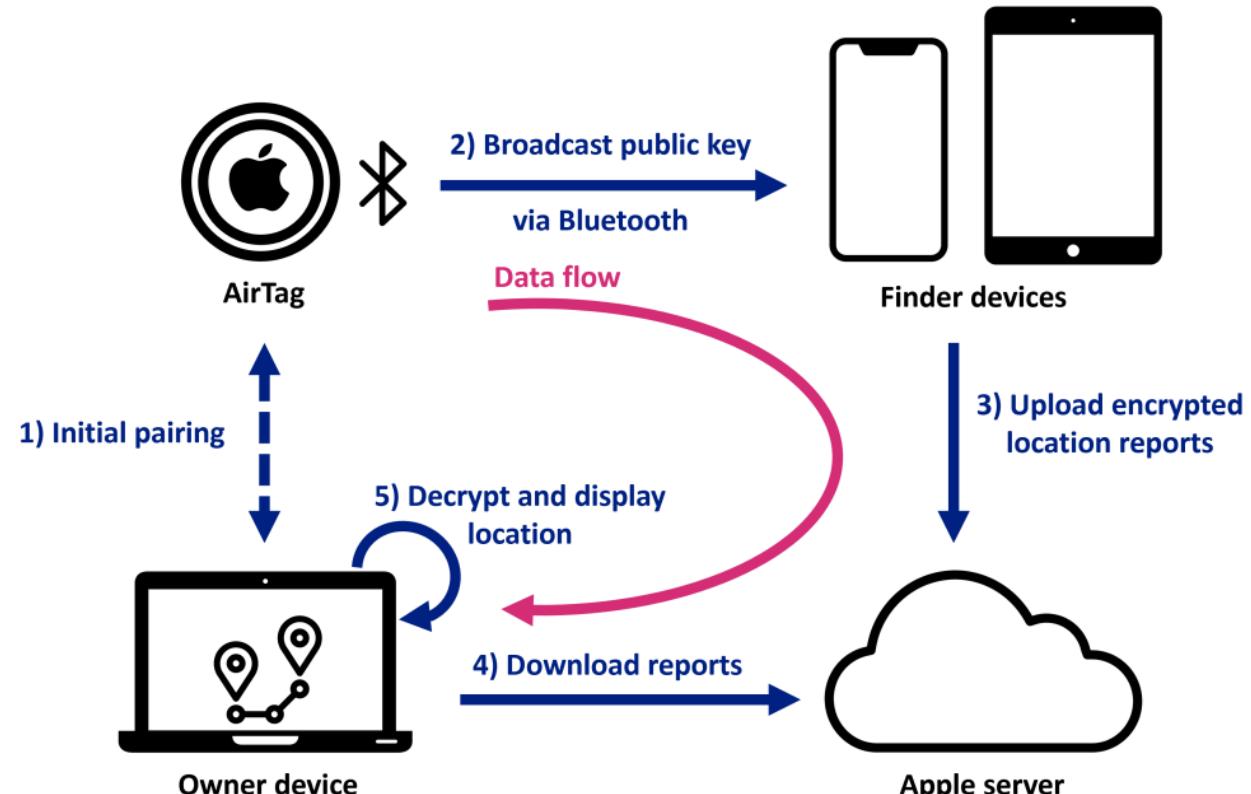


*And that's how contact tracing can protect our health *and* privacy!*

by Nicky Case (ncase.me). CC0/public domain, feel free to re-post anywhere!

Apple AirTags & “Find My”

Image source (FU): <https://makezine.com/projects/find-my-diy-airtag-tracker/>



Apple AirTags & “Find My” (2)

- Standard process using Bluetooth Low Energy
 - Initial setup of public-private keypair
 - AirTag broadcasts *public* key
 - Other devices: upload own location, encrypted with public key
 - *Private* key can be used to download and decrypt GPS location
- iPhone 11+: Ultra Wide Band (UWB) for last-meter localization
- Android devices: can use NFC to identify AirTag on contact

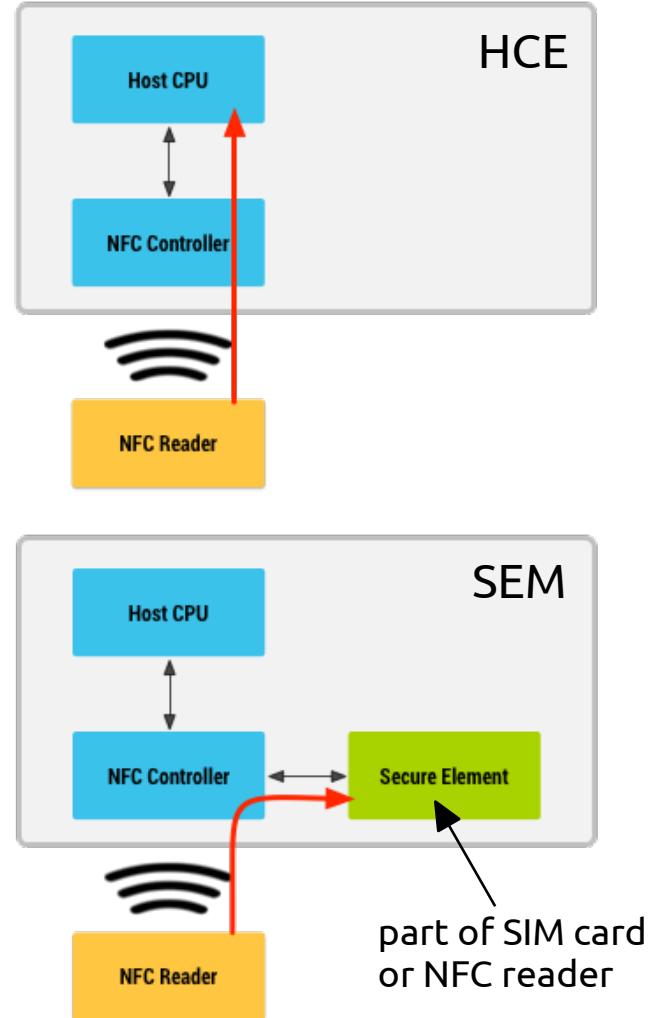
Apple AirTags & “Find My” (3)

- 3 different wireless technologies in one device:
 - BTLE for broadcasting public key (& audio notifications)
 - NFC for direct-touch identification & pairing
 - UWB for last-meter localization
- Issues with AirTags:
 - *Not* a large issue: owner privacy (thanks to public-key crypto)
 - *Big* issue: stalking & theft
 - iPhones (try to) detect unknown AirTags moving with you
 - Android needs custom app

Mobile Card Emulation (1)

Image source (FU): <https://developer.android.com/guide/.../hce>

- Two modes: Host Card Emulation (top), Secure Element Mode (bottom)
- HCE: less secure, as malware can interfere with payment process
- SEM: harder to implement, as extra security chip required (e.g. Apple Pay)
- Google Pay: issues with carrier support (restricted SIM card access)



Mobile Card Emulation (2)

Image source (CC): [https://commons.wikimedia.org/...](https://commons.wikimedia.org/), [https://commons.wikimedia.org/...](https://commons.wikimedia.org/)

- Usage scenarios
 - Mobile payment: credit card
 - Identification: national ID card
- Pass-through to websites possible
 - Multiple components involved: website, card issuer, certificate authority, browser, secure element, helper app, ...
 - Difficult to debug & secure properly



Computational Photography

Image sources: (CC) [Marco Verch](#), (FU) [Florian Kainz](#)

- Improve camera performance beyond physical limits, e.g.
 - artificial “bokeh” effect → (usually only feasible with DSLR & large aperture)
 - “Night Sight” on Pixel 4 → (usually only feasible with tripod & long exposure)



Computational Photography (2)

Image source (FU): <https://ai.googleblog.com/2018/10/see-better-and-further-with-super-res.html>

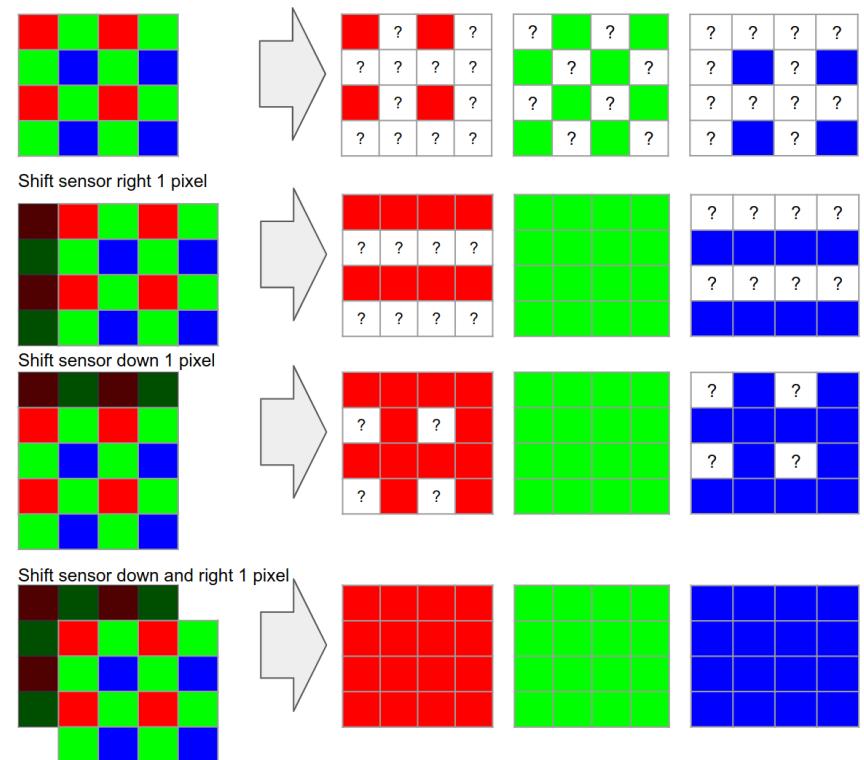
- Example: Superresolution/
Superzoom
 - Take burst of several images
in quick succession (< 1 s)
 - Merge into final image with
2-3x resolution/zoom of sensor



Computational Photography (3)

Image source (FU): <https://ai.googleblog.com/2018/10/see-better-and-further-with-super-res.html>

- Approach:
 - Use minimal shifts between images to improve details
 - E.g. hand tremor, measured with IMU (or artificial motion of OIS)
 - Takes adv. of Bayer pattern
- Challenges
 - Moving objects
 - Image sensor noise
 - Unpredictable motion



The End

