

Developing Secure Mobile Architectures: The COTS Challenge

Adam Wick | HCSS 2011 | May 6th, 2011

What I'm Going To Tell You

- This is a talk about an idea.
 - I'm not going to tell you about an existing new product from Galois, Inc.
 - I'm not even sure if this technology will work in the end!
- Galois has been investigating some of the concepts I'm going to describe in this talk.
- ***My goal:*** Foster discussion about the need for COTS in the secure mobile space, and gather feedback about a particular solution that we are pursuing.

“Mobile Phone”



Use: Phone



Use: Phone & MMS



Use: Phone & MMS

“Mobile Phone”



Use: Phone

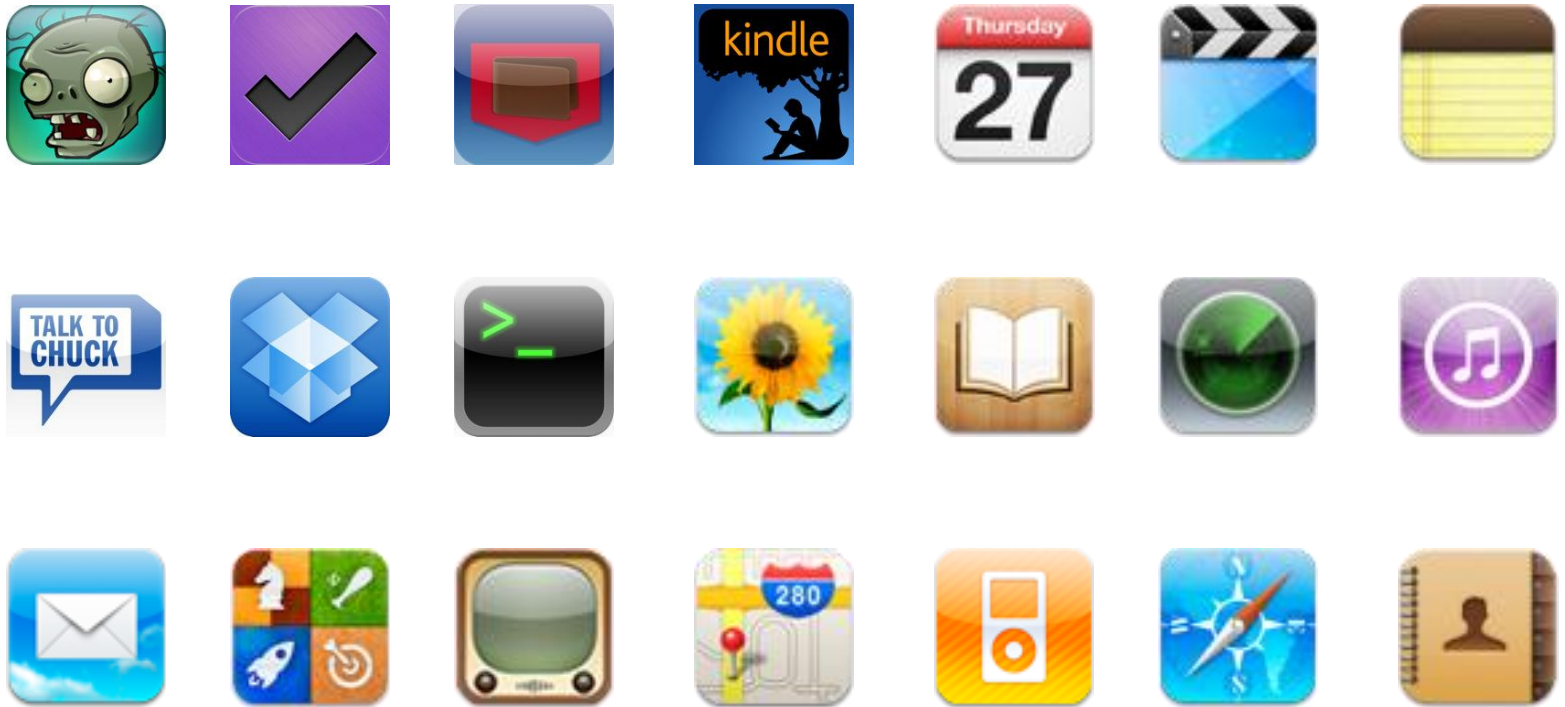


Use: Phone & MMS







Use: Phone & MMS
& *calendar, email,
Web browser, ToDo
List, games, books,
bank / budget tool, ...*

In Fact, Applications Are Endless!



And they are all potentially network connected!

How Secure Do You Want Your Apps?

- Well, I don't care if my game scores are intercepted. 
- I might be a little concerned about information about my pictures, movies, books, or music. 
- I really don't want people reading my email, or reading my calendar, or knowing my to-dos! 
 - I could lose my job!
- And I really, really don't want people in my bank accounts. 
 - Identity theft is really scary!

Levels of Security

- There are many threats we could be guarding against:
 - Shoulder surfing
 - Network sniffing
 - Left it on my desk
 - Left it on a bus
 - Left it in a war zone
- We are currently looking at a threat level of “innocent but potentially misguided user.”

- GOTS isn't a good idea
 - It's going to be very hard to develop secure hardware at the same pace as the feature-rich hardware of the commercial world.
 - In other words, if you give a user this, they're going to revolt:



- But even if you succeed, users want a phone and the apps they're used to, not just a smart phone. Will your secure phone run Android™ apps?

- Even if you could build it, watch out for the tail!
 - Who is going to maintain this? How much does that cost?
 - Who is going to port it to new hardware? How much does that cost?
- If you can answer those questions, how about one more:
 - How are you going to maintain the pace of technology update and innovation, so that we don't revisit this issue when company X builds the next great thing?

- Start with commodity phone hardware
- Create a security layer that:
 - Does not interfere with running a commodity operating system.
 - Does not interfere with running commodity applications.
 - Does provide additional layers of security for ensuring information security.
- Oh, and maybe we could run two versions of Android™ at once? One High and one Low?

Talk Outline

- Introduction
- **Our Solution**
- COTS Technologies That Work
- Security Arguments For A Simple Example
- Conclusion

Mission: Extended

- Start with use cases: How do we imagine people using their phone in the future?
 - As a personal use device for phone, web, email, etc.
 - As a unified device for work and personal use, where work data is sequestered from personal data.
 - As a tactical device at the front edges of the battlefield.
 - As the previous, but also as a component of a dynamic, ever-shifting MANET.
 - Just as a phone, because my Dad can't comprehend anything beyond that.
- Too many use cases!

Our Solution (Part #1)

- CAMA: Configuration-Assured Mobile Architecture
- Critical insight: Don't build a static system that tries to be everything to everyone, build a set of building blocks that people can build their ideal phone out of.
- Critical technologies:
 - Virtualization to allow commodity software to run.
 - Intercessor blocks to perform key security functions.
 - Off-the-shelf COTS OSes so the user gets the experience they want

CAMA: building blocks

Device Drivers



User-Facing OS
(Potentially several versions)

Direct Connections



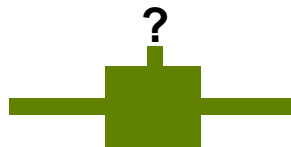
Device Multiplexing



Inline Cryptography



Guards / Flow Control



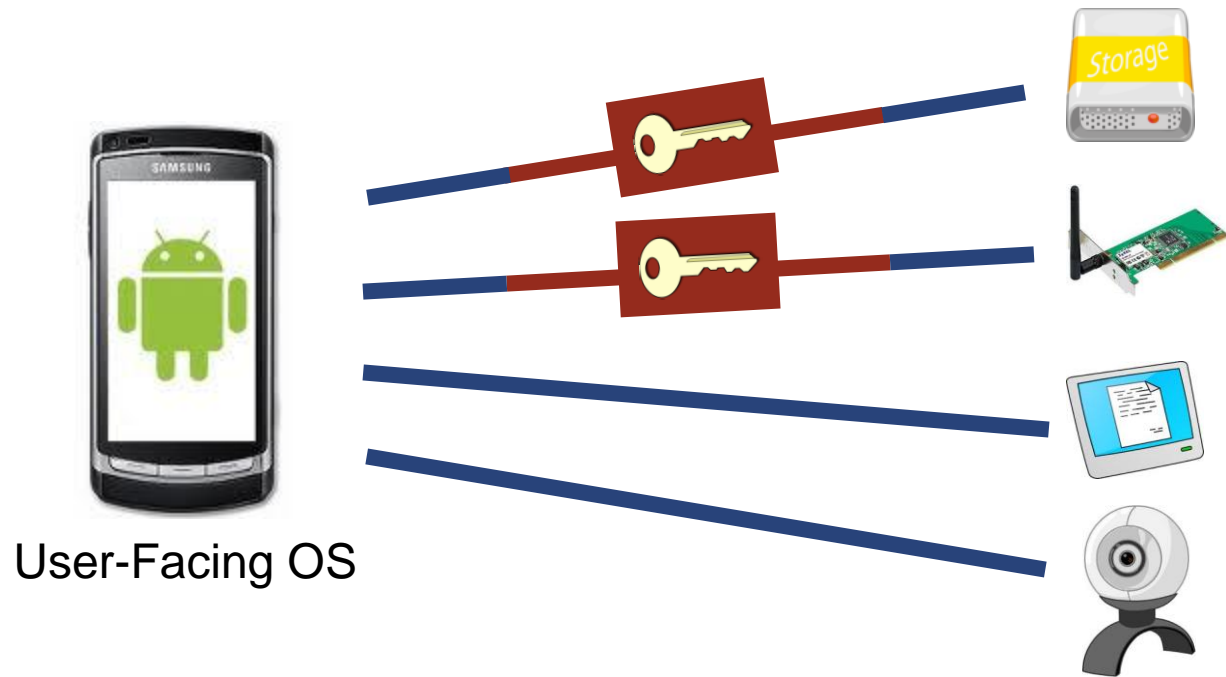
Conditional Connections



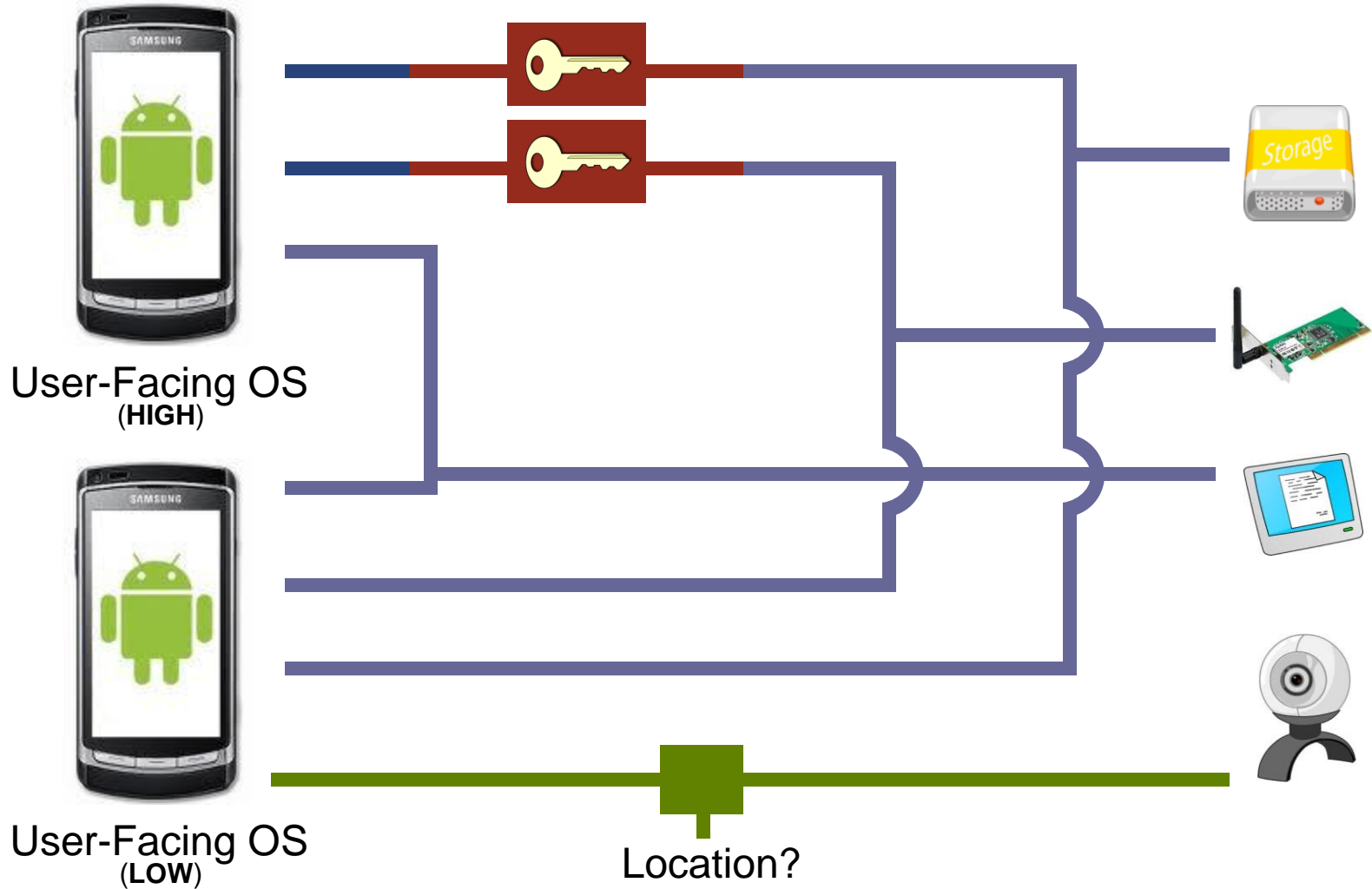
Multi-Level Components



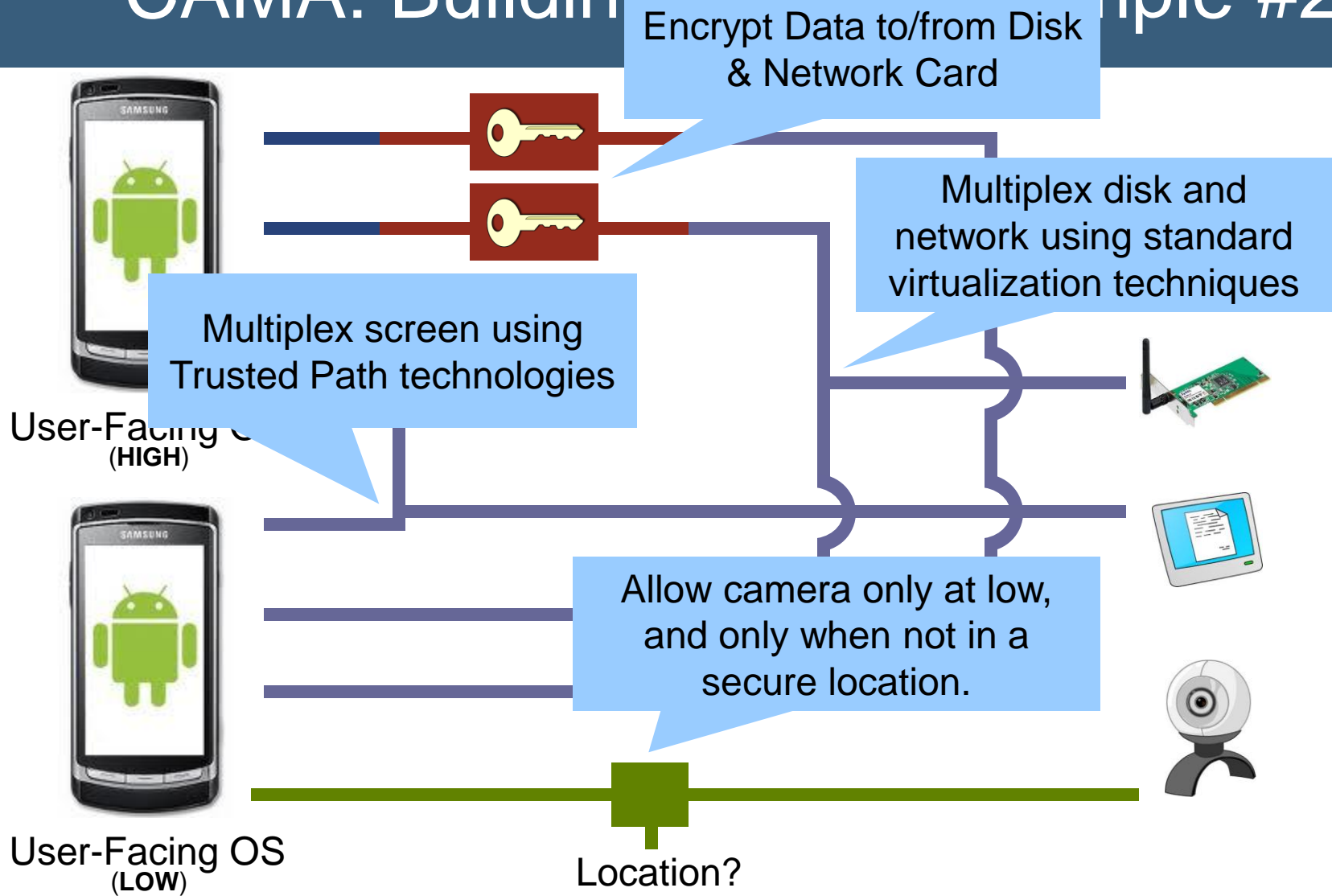
CAMA: Building Blocks Example #1



CAMA: Building Blocks Example #2



CAMA: Building a Platform Example #2



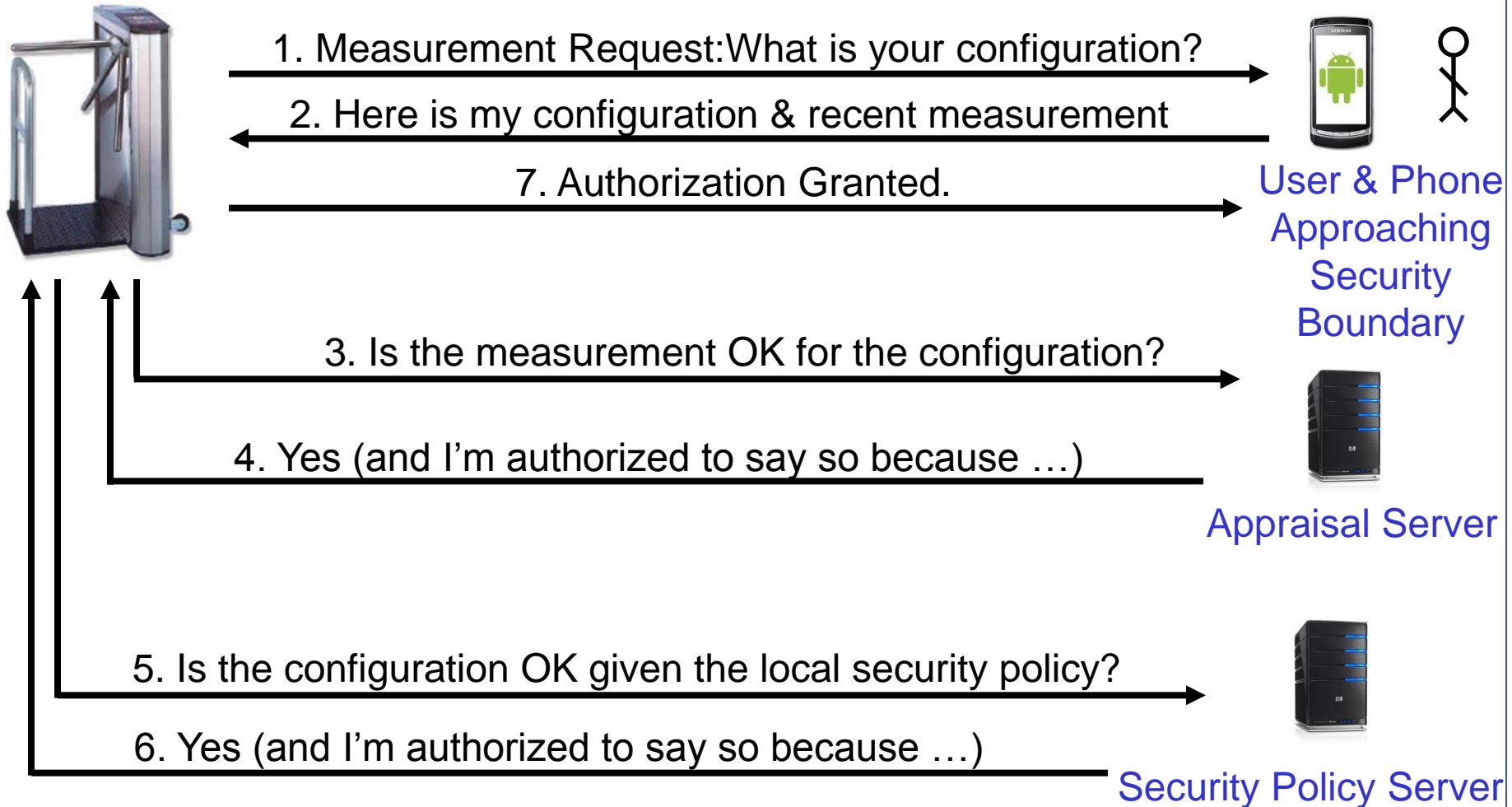
Our Solution (Part #2)

- Our CAMA approach does not just include the phone. It includes:
 - A configuration language for describing component layouts.
 - A security policy language for describing the security policies of an institution or location.
 - A phone capable of running a given configuration *AND* providing evidence that it is correctly running that configuration.
 - Tools for validating that a given configuration is acceptable for a given security policy.

But Why Do I Trust This?

- Separation kernel approach isolates key components.
- Small security components open up the possibility of formal verification.
- Use (re-)measurement techniques to prove appropriate components booted and running.
- Map measurements to configurations that describe the phone.
- Create automatic tools to check configurations against security policies.

CAMA: Access Point Check Example



Talk Outline

- Introduction
- Our Solution
- **COTS Technologies That Work**
- Security Arguments For A Simple Example
- Conclusion

- Our entire strategy is predicated on the belief that we must allow the use of COTS operating systems and apps.
- Android™ provides an ideal experimental platform:
 - It is open source, allowing us to modify it to debug it more easily.
 - It runs on a wide variety of hardware platforms.
- Significant existing market penetration makes it more than just a toy experimental platform.

- Virtualization is a core technology
 - It provides the platform on which we can run commercial operating systems.
 - It provides the separation required for our security components.
- Our choice: OKL4
 - L4 variant commercialized by Open Kernel Labs.
 - In use in existing devices.
 - Path to higher assurance via future offerings from Open Kernel Labs.

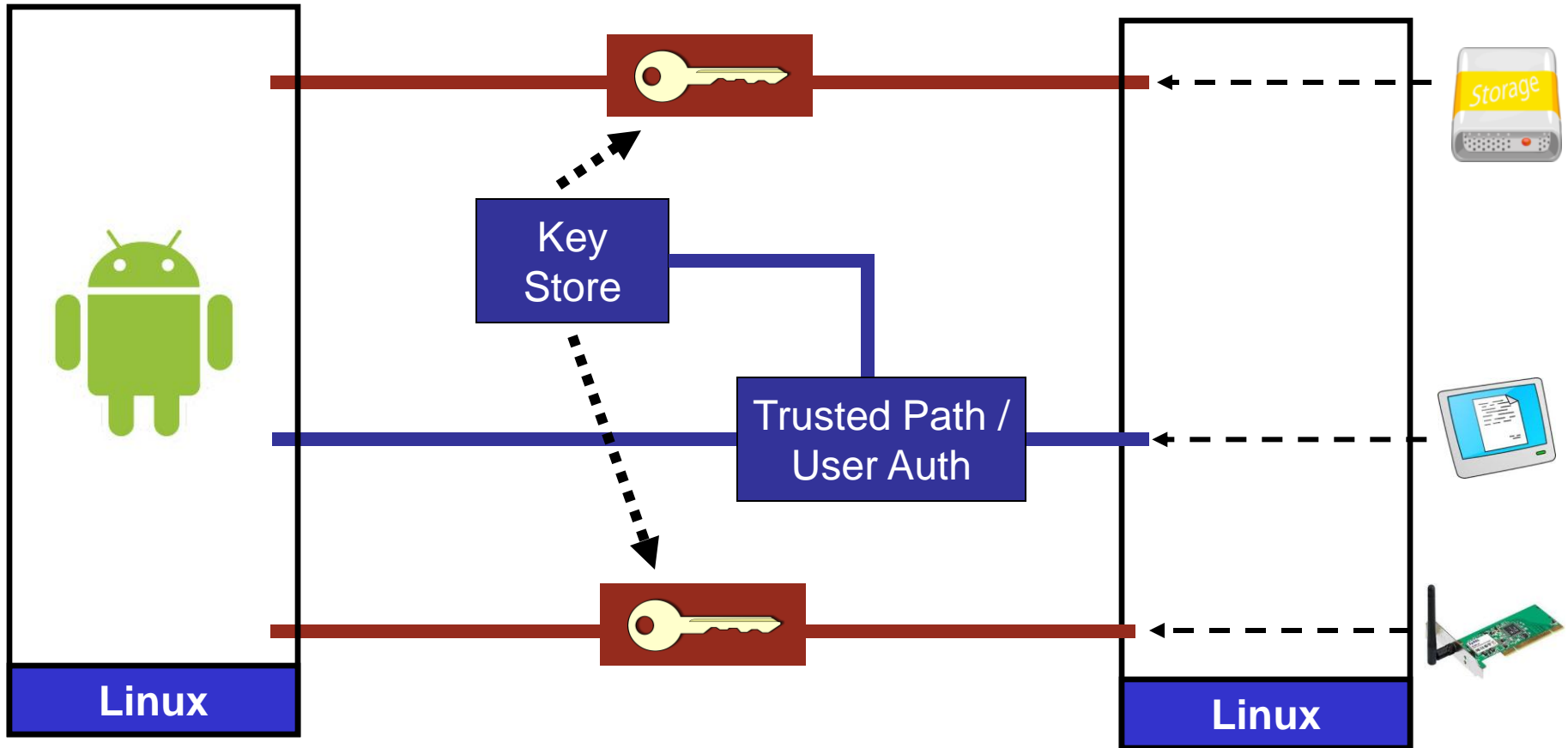
Talk Outline

- Introduction
- Our Solution
- COTS Technologies That Work
- **Security Arguments For A Simple Example**
- Conclusion

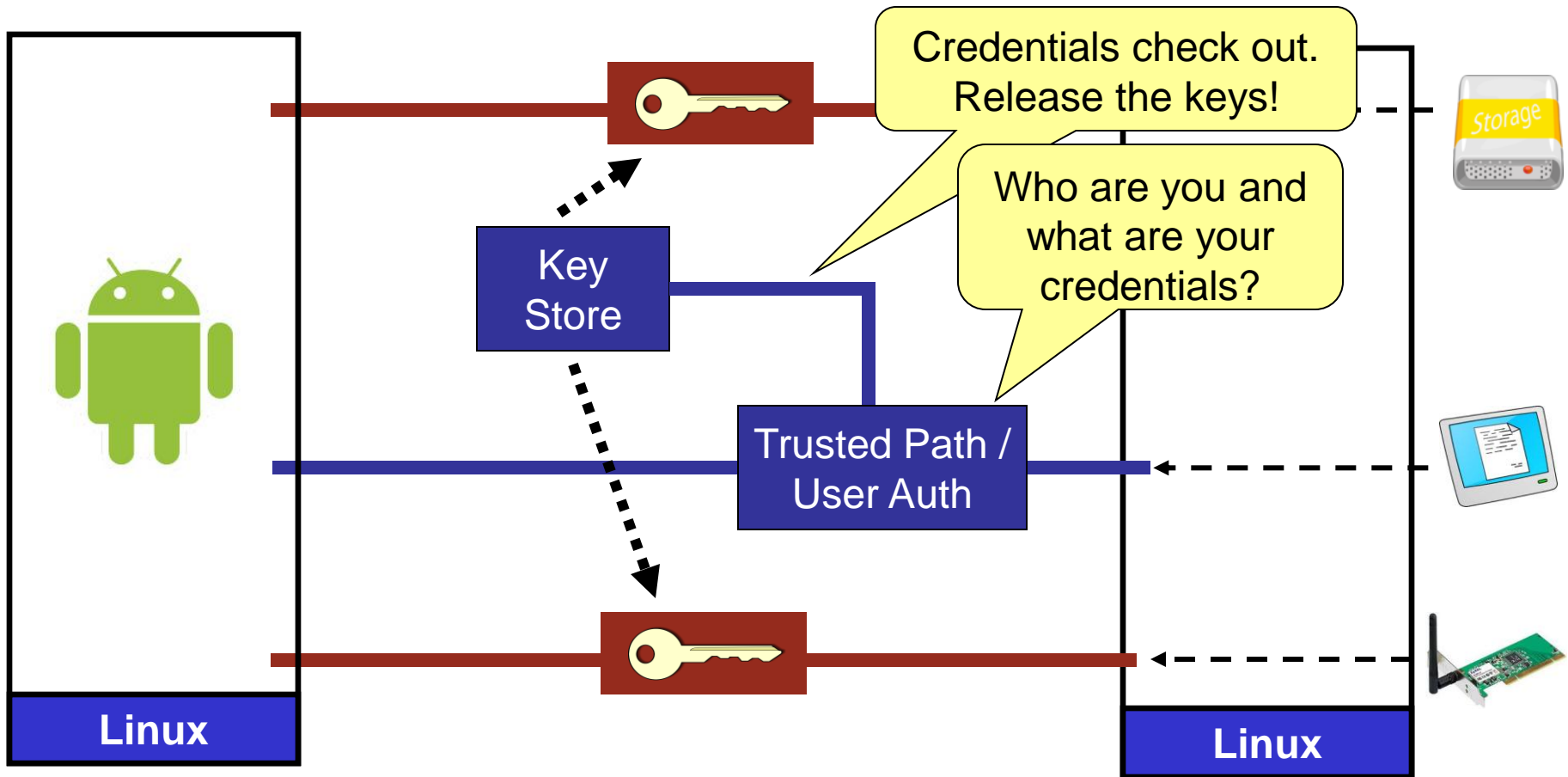
Threats We (Currently) Care About

- There are any number of bars you can set regarding the security of a mobile device, up to providing it directly to a well-funded adversary.
- We are currently stopping somewhere in the middle with our prototype:
 - Our users are not malicious, although they can do stupid things.
 - We care about information loss, but not necessarily about denial of service.
 - Our goal is to avoid a single point of failure.

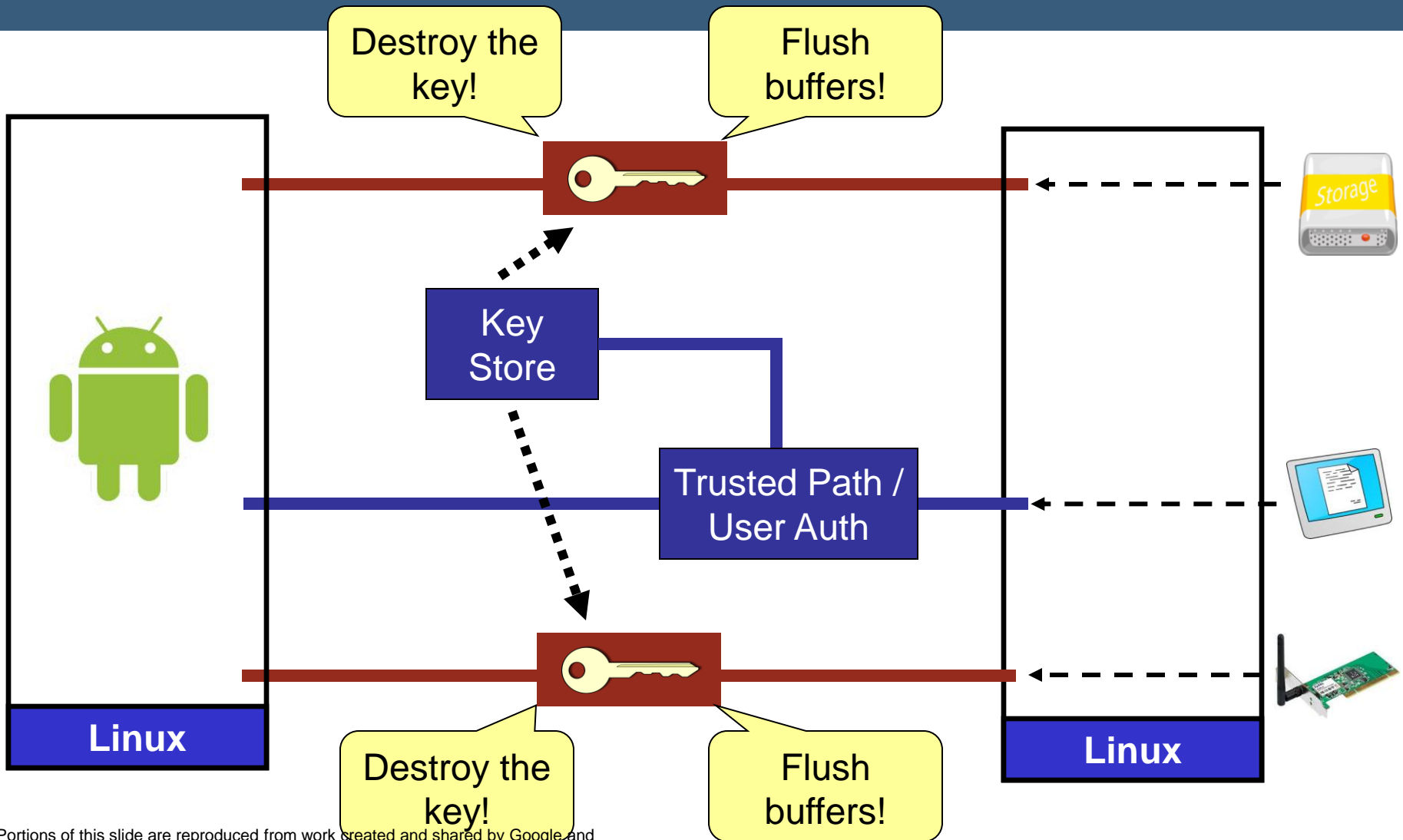
Our Prototype System



Boot Up / Wake Up Sequence



Sleep Sequence



Talk Outline

- Introduction
- Our Solution
- COTS Technologies That Work
- Security Arguments For A Simple Example
- **Conclusion**

- Security for mobile devices is increasingly important.
- It is too late to start from scratch.
- Microvisor technology allows us to utilize existing systems while adding security features “underneath”.
- CAMA provides a flexible system architecture for providing security for a variety of use cases.
- Galois is starting down this path, and is interested in comments, suggestions, collaboration opportunities, etc.
- Comments or questions?