End-to-End Security for the Internet of Things

Dan Boneh[†], Prabal Dutta[^], Dawson Engler[†], Björn Hartmann[º], Mark Horowitz[†], Philip Levis[†], Raluca Ada Popa[º], and Keith Winstein[†]

[†]Stanford University, [º]UC Berkeley, [^]University of Michigan

Intel/NSF CPS-Security
November 1, 2016

The Internet of Things (IoT)

A Security Disaster

The Economist | World politics | Business & finance | Economics | Science & technology | Culture

Cyber-security

The internet of things (to be hacked)

Hooking up gadgets to the web promises huge benefits. But security must not be an afterthought

Jul 12th 2014 | From the print edition   Timekeeper   Like 217   Tweet 594

How the Internet of Things Could Kill You

By Fahmida Y. Rashid JULY 18, 2014 7:30 AM - Source: Tom's Guide US | 5 COMMENTS

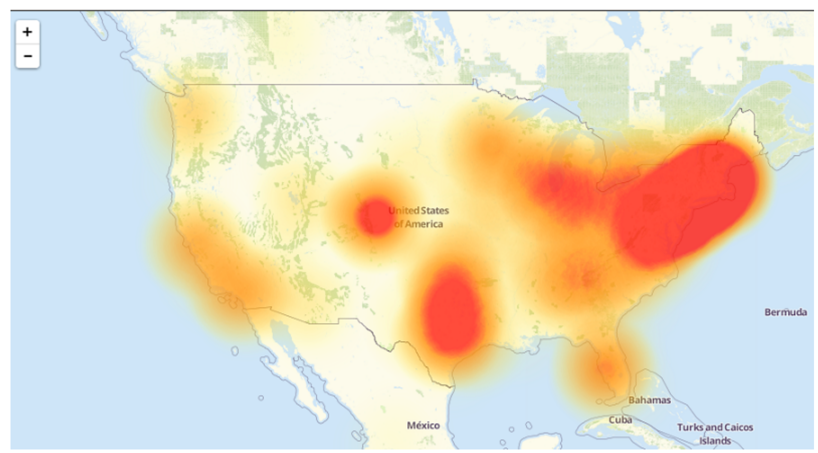
Hacking the Fridge: Internet of Things Has Security Vulnerabilities

JESS SCANLON | MORE ARTICLES
JUNE 28, 2014

Philips Hue LED smart lights hacked, home blacked out by security researcher

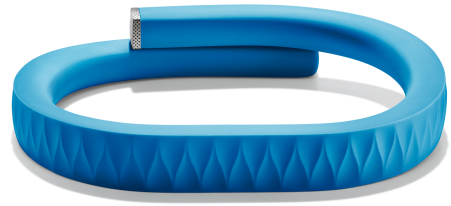
By Sal Cangeloso on August 15, 2013 at 11:45 am | 7 Comments

- HP conducted a security analysis of IoT devices[1]
  - 80% had privacy concerns
  - 80% had poor passwords
  - 70% lacked encryption
  - 60% had vulnerabilities in UI
  - 60% had insecure updates



[1] http://fortifyprotect.com/HP_IoT_Research_Study.pdf

Internet(s) of Things



Industrial Automation

Thousands/person
Controlled Environment
High reliability
Control networks
Industrial requirements

WirelessHART, 802.15.4
6tsch, RPL
IEEE/IIC/IETF

Home Area Networks

Hundreds/person
Uncontrolled Environment
Unlicensed spectrum
Convenience
Consumer requirements

ZigBee, Z-Wave
6lowpan, RPL
IETF/ZigBee/private

Personal Area Networks

Tens/person
Personal environment
Unlicensed spectrum
Instrumentation
Fashion vs. function

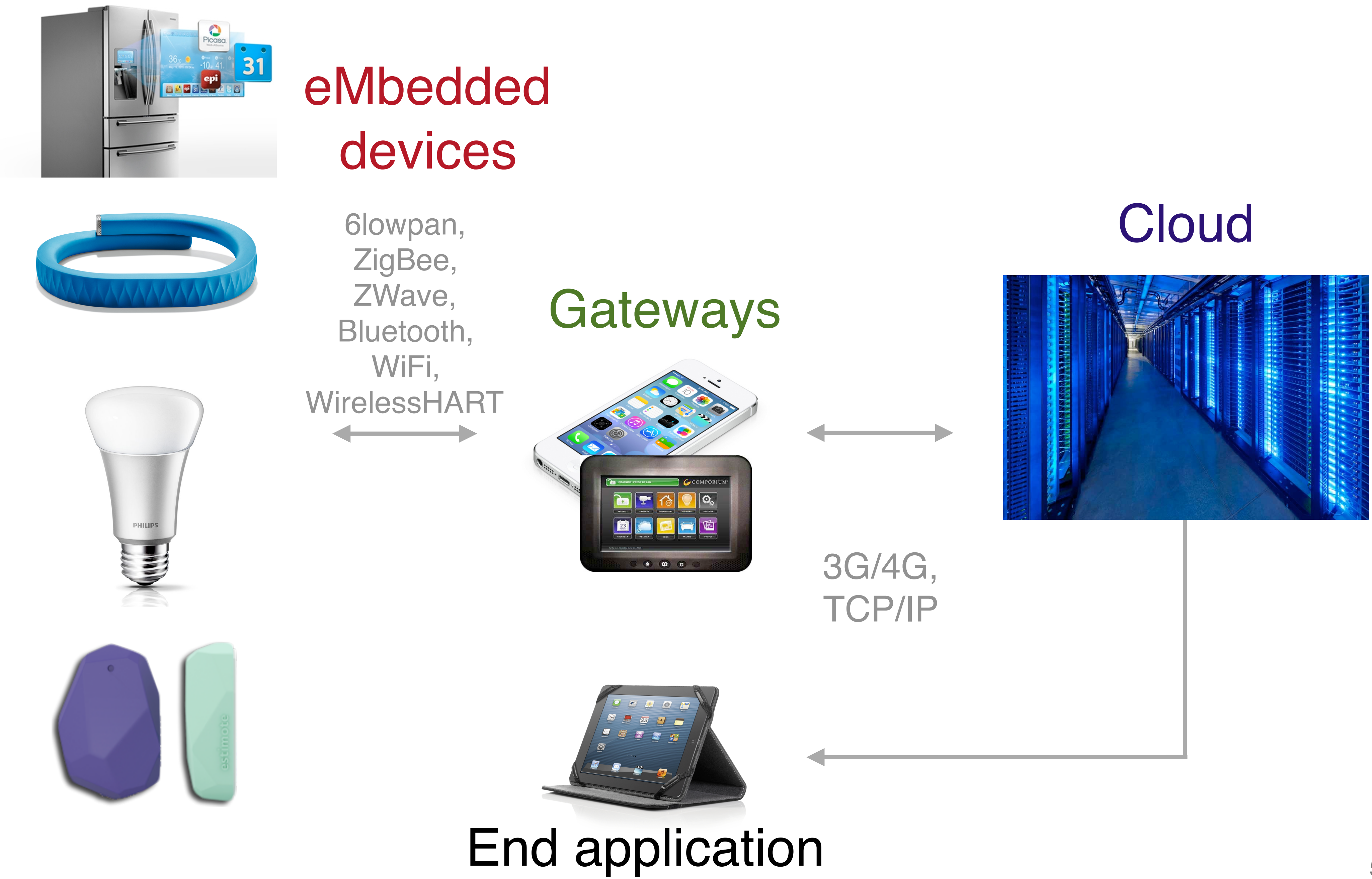
Bluetooth, BLE
3G/LTE
3GPP/IEEE

Networked Devices

Tens/person
Uncontrolled Environment
Unlicensed spectrum
Convenience
Powered

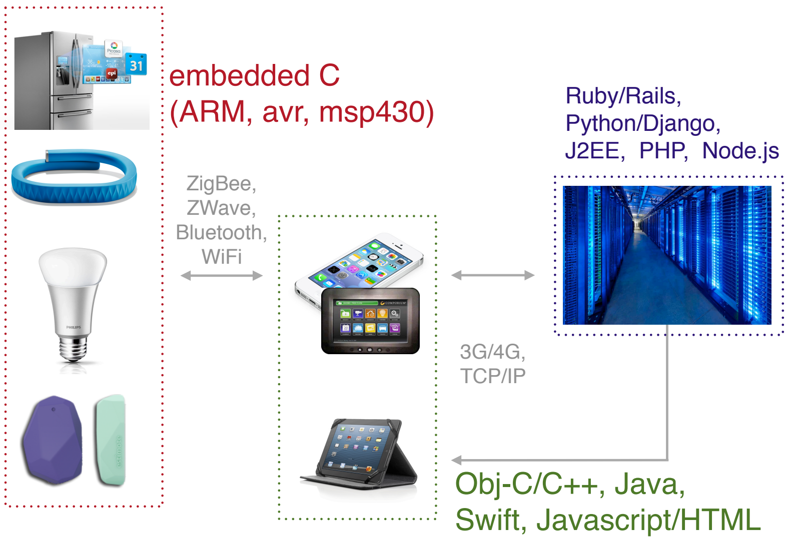
WiFi/802.11
TCP/IP
IEEE/IETF

IoT: MGC Architecture



IoT Security is Hard

- Complex, distributed systems
  - 10^3-10^6 differences in resources across tiers
  - Many languages, OSes, and networks
  - Specialized hardware
- Just *developing* applications is hard
- Securing them is even harder
  - Enormous attack surface
  - Reasoning across hardware, software, languages, devices, etc.
  - What are the threats and attack models?
- Valuable data: personal, location, presence



- Rush to development + hard ➔ avoid, deal later

Architectural Principles

- Longevity: these systems will last for up to 20 years and their security must too.
- Transparency: we must be able to observe what our devices are saying about us.
- End-to-end: consider security holistically, from data generation to end-user display.

Architectural Principles

- Longevity: these systems will last for up to 20 years and their security must too.

- Transparency: we must be able to observe what our devices are saying about us.

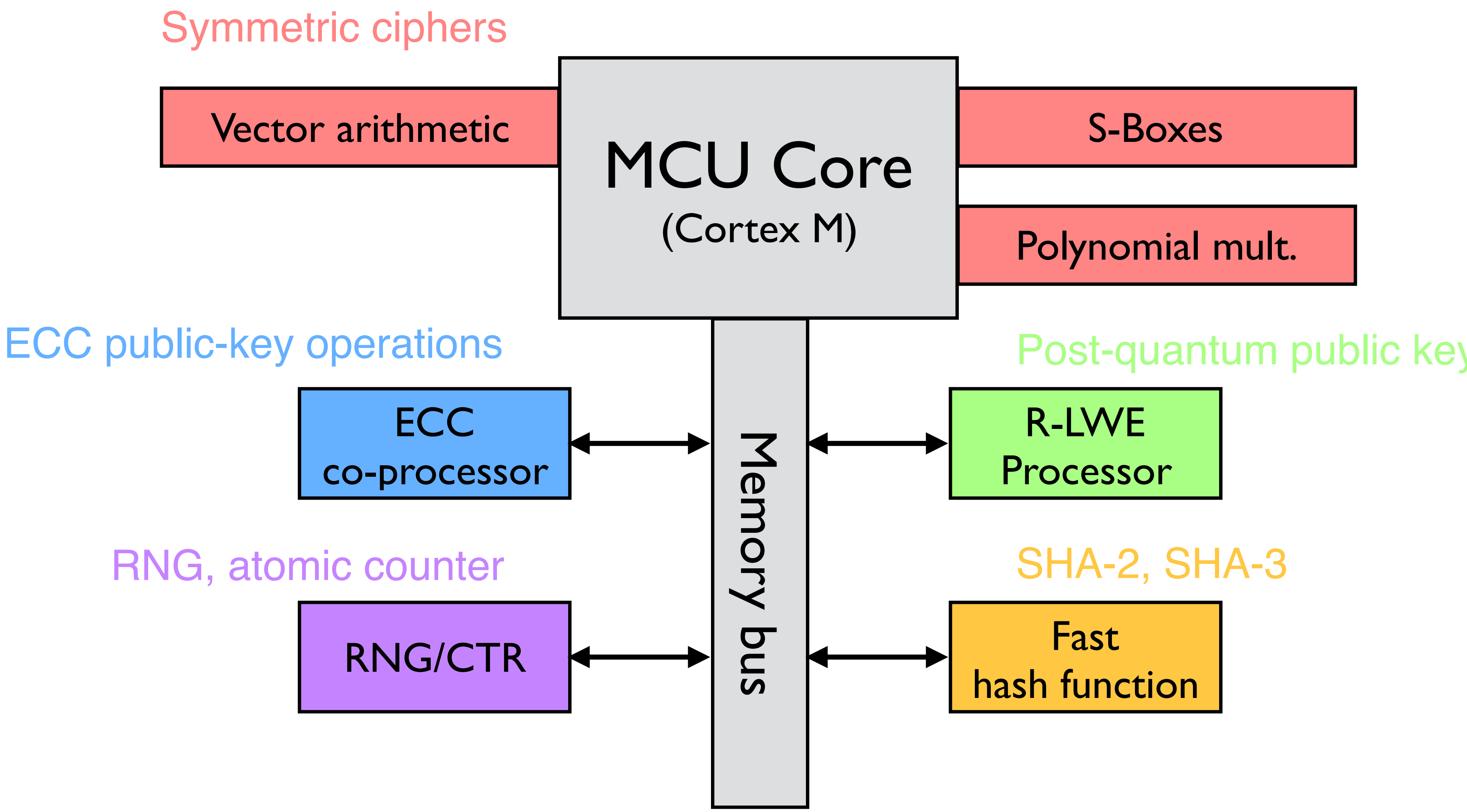
- End-to-end: consider security holistically, from data generation to end-user display.


Microsoft
Microsoft
Windows 95

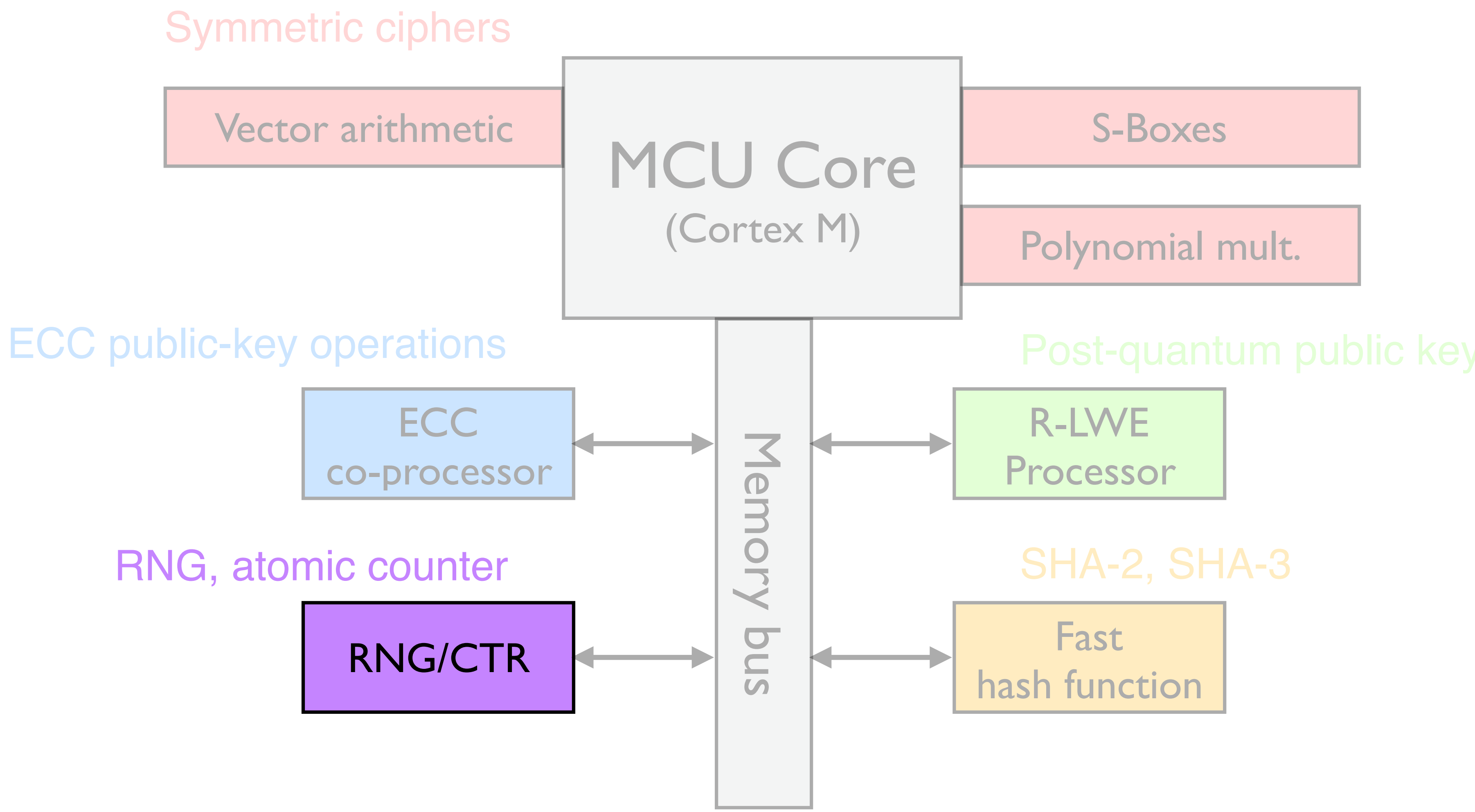
Flexible Crypto Hardware

- Devices need to be able to support ciphers that are used 20 years from now
- Add extensible cryptographic accelerator: silicon is cheap and BLE dominates the SoC
- Designing a 20-year crypto processor
  - Symmetric crypto: S-boxes and vectors, an instruction set
  - Public key crypto: several very different constructions
  - What if quantum computers are real in 20 years?
- There is often unused micro controller die area

CESEL


Symmetric ciphers
Vector arithmetic
MCU Core
(Cortex M)
S-Boxes
Polynomial mult.
ECC public-key operations
Post-quantum public key
ECC
co-processor
Memory bus
R-LWE
Processor
RNG, atomic counter
SHA-2, SHA-3
RNG/CTR
Fast
hash function

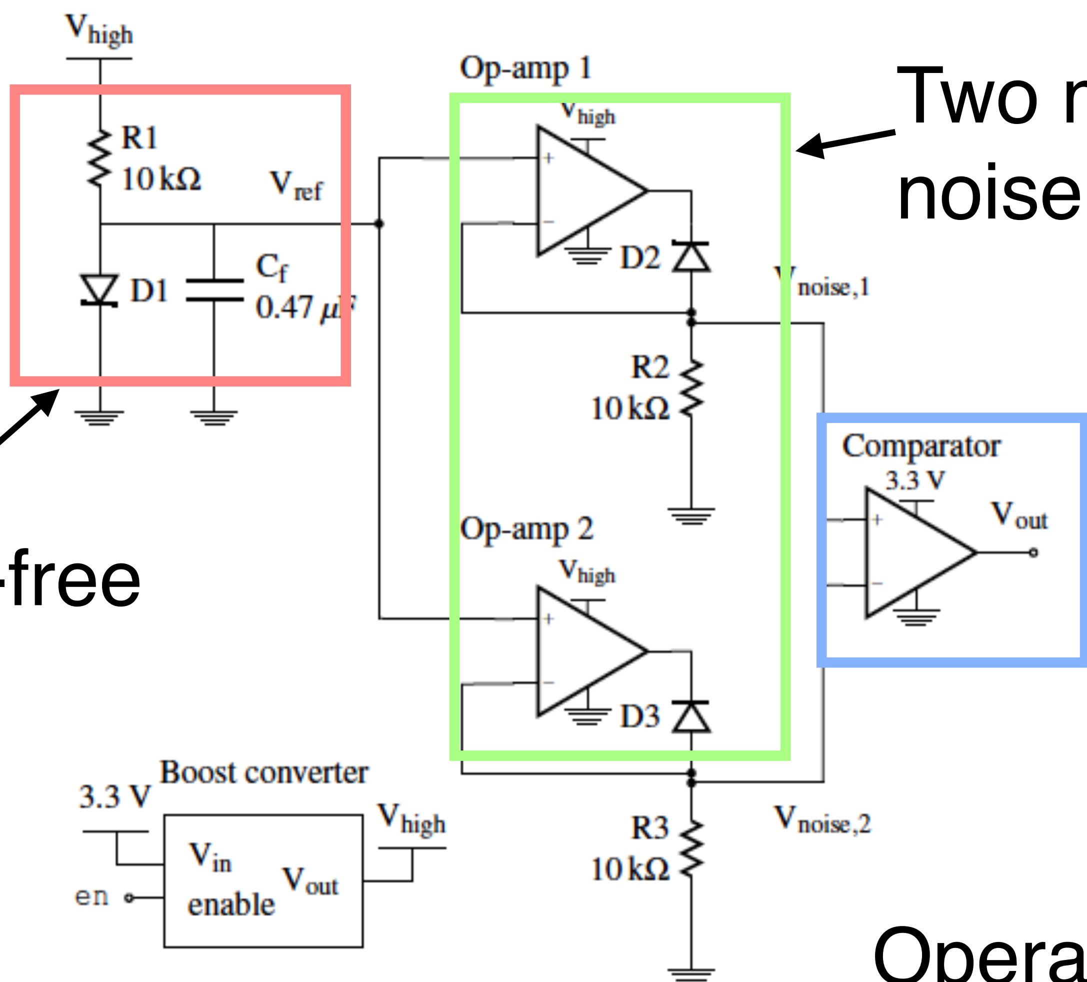
CESEL


Symmetric ciphers
Vector arithmetic
MCU Core
(Cortex M)
S-Boxes
Polynomial mult.
ECC public-key operations
Post-quantum public key
ECC
co-processor
Memory bus
R-LWE
Processor
RNG, atomic counter
SHA-2, SHA-3
RNG/CTR
Fast
hash function

Embedded IoT Security

- Random numbers are foundational to security
  - E.g., key generation, nonces, etc.
- A random number generator needs a seed of entropy — truly random bits
  - Can then expand this entropy seed into a huge stream of unguessable bits, assuming adversary cannot see seed
  - E.g., use entropy to generate AES-128 key, then run AES-128 in counter mode, encrypting 0, 1, 2, … , 2^{128-1}
- Can add *more* entropy, but not needed

Lampert Circuit



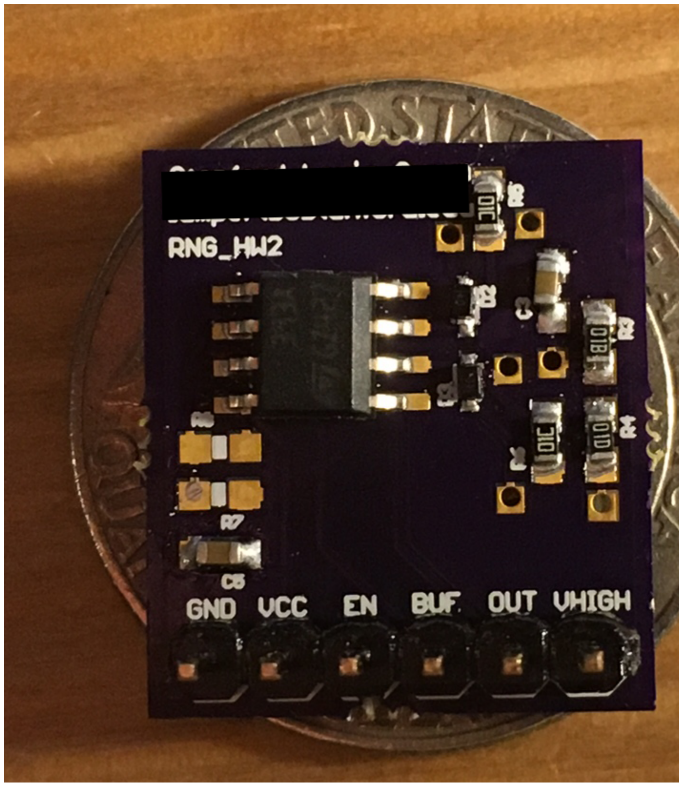
Two matched avalanche noise sources (no bias)

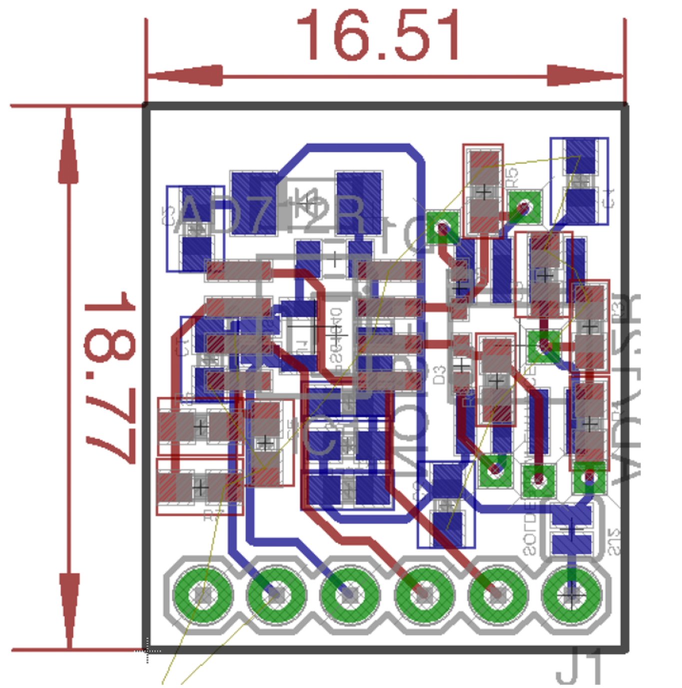
Noise-free power

Comparator

Operating mode: seed hardware AES-128

Lampert Circuit





Going Up the Stack

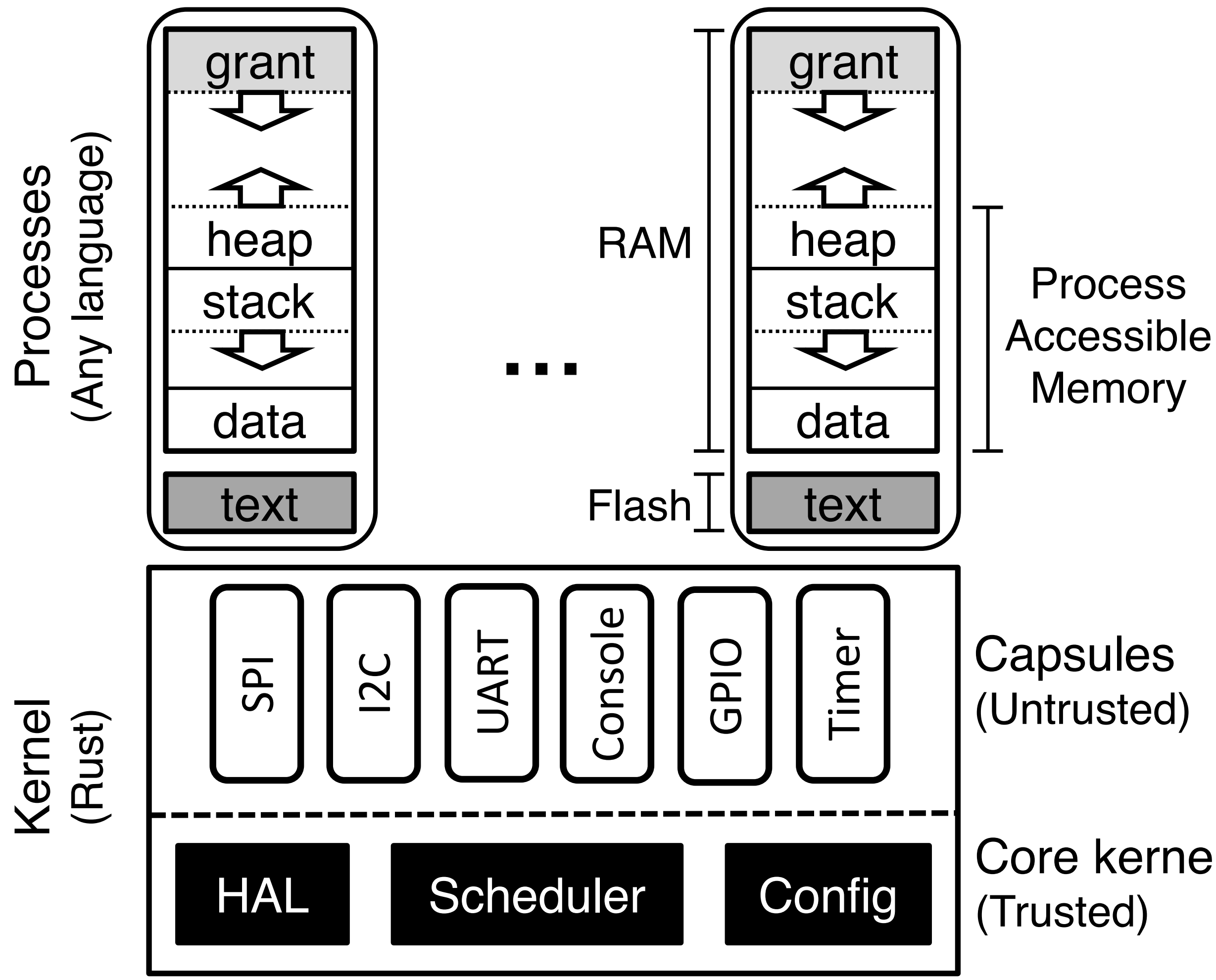
TockOS

CESEL

Tock Operating System

- Safe, multi-tasking operating system for memory-constrained devices
- Core kernel written in Rust, a safe systems language
  - Small amount of trusted code (can do unsafe things)
    - Rust bindings for memory-mapped I/O
    - Core scheduler, context switches
- Core kernel can be extended with *capsules*
  - Safe, written in Rust
  - Run inside kernel
- *Processes* can be written in any language (asm, C)
  - Leverage Cortex-M memory protection unit (MPU)
  - User-level, traps to kernel with system calls

Tock: Secure Embedded OS

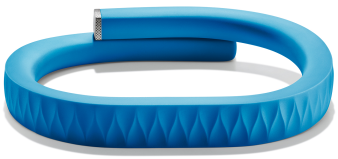


Architectural Principles

- Longevity: these systems will last for up to 20 years and their security must too.
- Transparency: we must be able to observe what our devices are saying about us.
- End-to-end: consider security holistically, from data generation to end-user display.

IoT: MGC Architecture

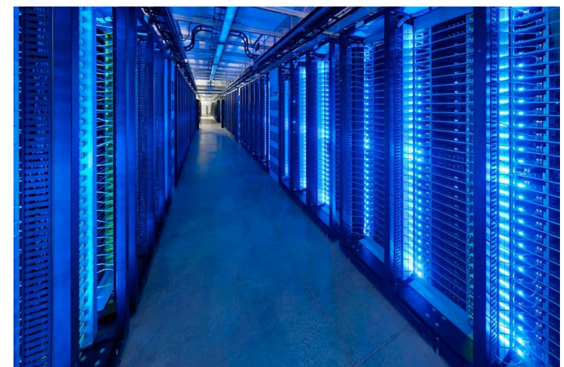
eMbedded devices



6lowpan, ZigBee, ZWave, Bluetooth, WiFi, WirelessHART

Gateways

Cloud

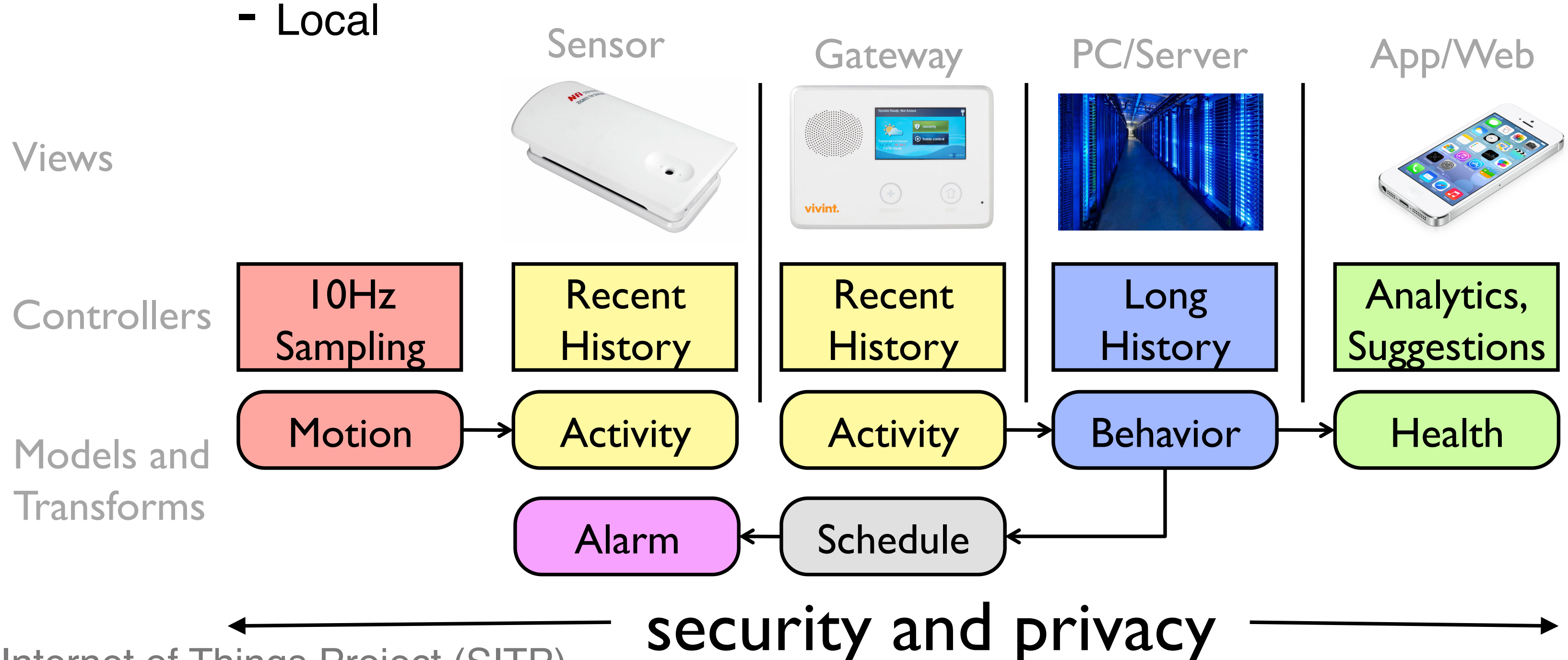


3G/4G, TCP/IP

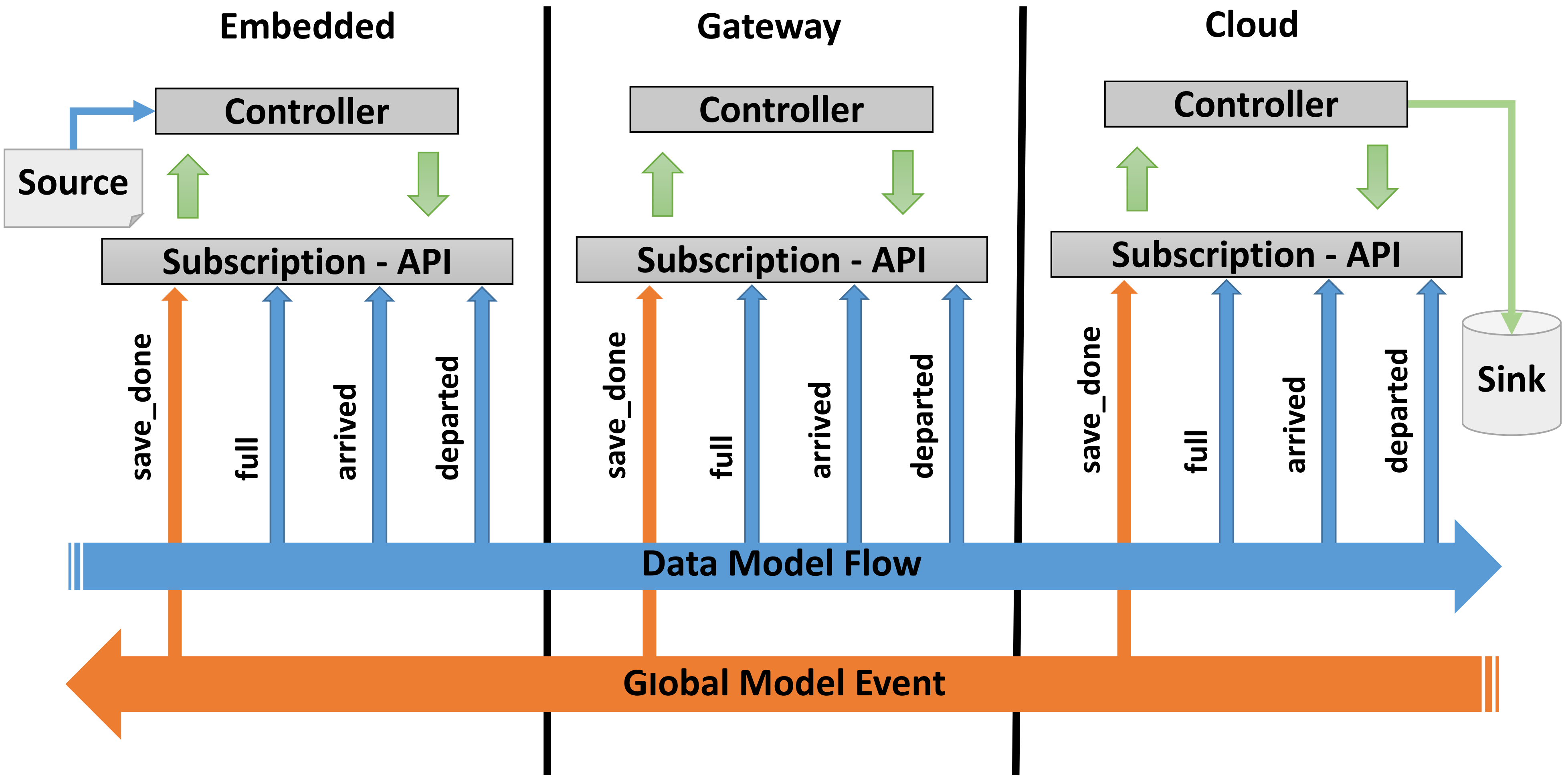
End application

Ravel Framework

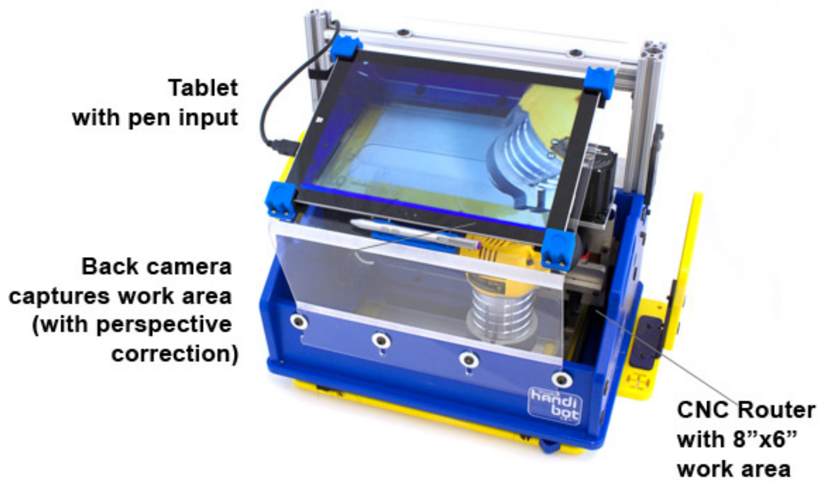
- Write a data processing pipeline
  - Consists of a set of Models, describing data as it is stored
  - Instances of Models are bound to Spaces
  - Three types of models
    - Replicated
    - Streaming
    - Local

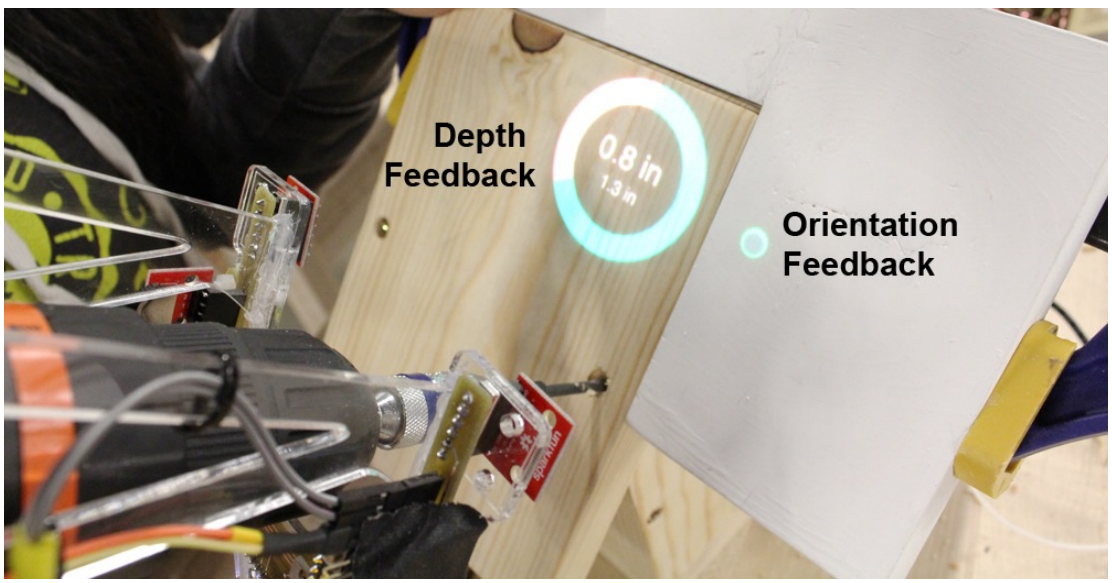


Streaming Model API



Drill Sergeant





Data Security Results

- Arx: A Strongly Encrypted Database System
- Embark: Securely Outsourcing Middleboxes to the Cloud
- BlindBox: Deep Packet Inspection for Encrypted Traffic
- Machine Learning Classification Over Encrypted Data
- Verena: End-to-End Integrity Protection for Web Applications
- Privacy, Discovery, and Authentication for the Internet of Things
- Hosting Services on an Untrusted Cloud
- Semantically Secure Order-Revealing Encryption: Multi-input Functional Encryption Without Obfuscation

System Security Results

- CESEL: Securing a Mote for 20 Years
- Robust, low-cost, auditable random number generation for embedded system security
- Auditing IoT Communications with TLS-RaR
- How to Build Static Checking Systems Using Orders of Magnitude Less Code.
- Ownership is Theft: Experiences Building an Embedded OS in Rust
- Beetle: Flexible Communication for Bluetooth Low Energy.
- Toastboard: Ubiquitous Instrumentation and Automated Checking of Breadboarded Circuits
- Ravel: Programming IoT Applications as Distributed Models, Views, and Controllers
- MBus: An Ultra-Low Power Interconnect Bus for Next Generation Nanopower Systems
- PowerBlade: A Low-Profile, True-Power, Plug-Through Energy Meter
- Cinamin: A Perpetual and Nearly Invisible BLE Beacon

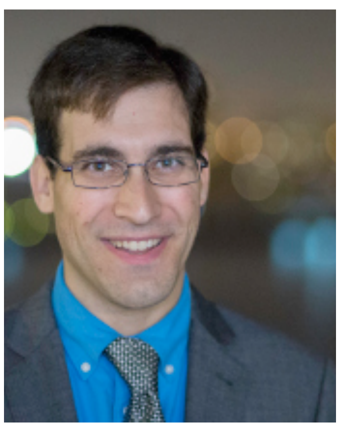
Thank you!

Philip Levis
Stanford
Embedded Systems

Mark Horowitz
Stanford
Hardware

Dan Boneh
Stanford
Cryptography

Dawson Engler
Stanford
Software



Keith Winstein
Stanford
Networks

Björn Hartmann
Berkeley
Prototyping

Raluca Ada Popa
Berkeley
Security

Prabal Dutta
Berkeley/Michigan
Embedded Hardware