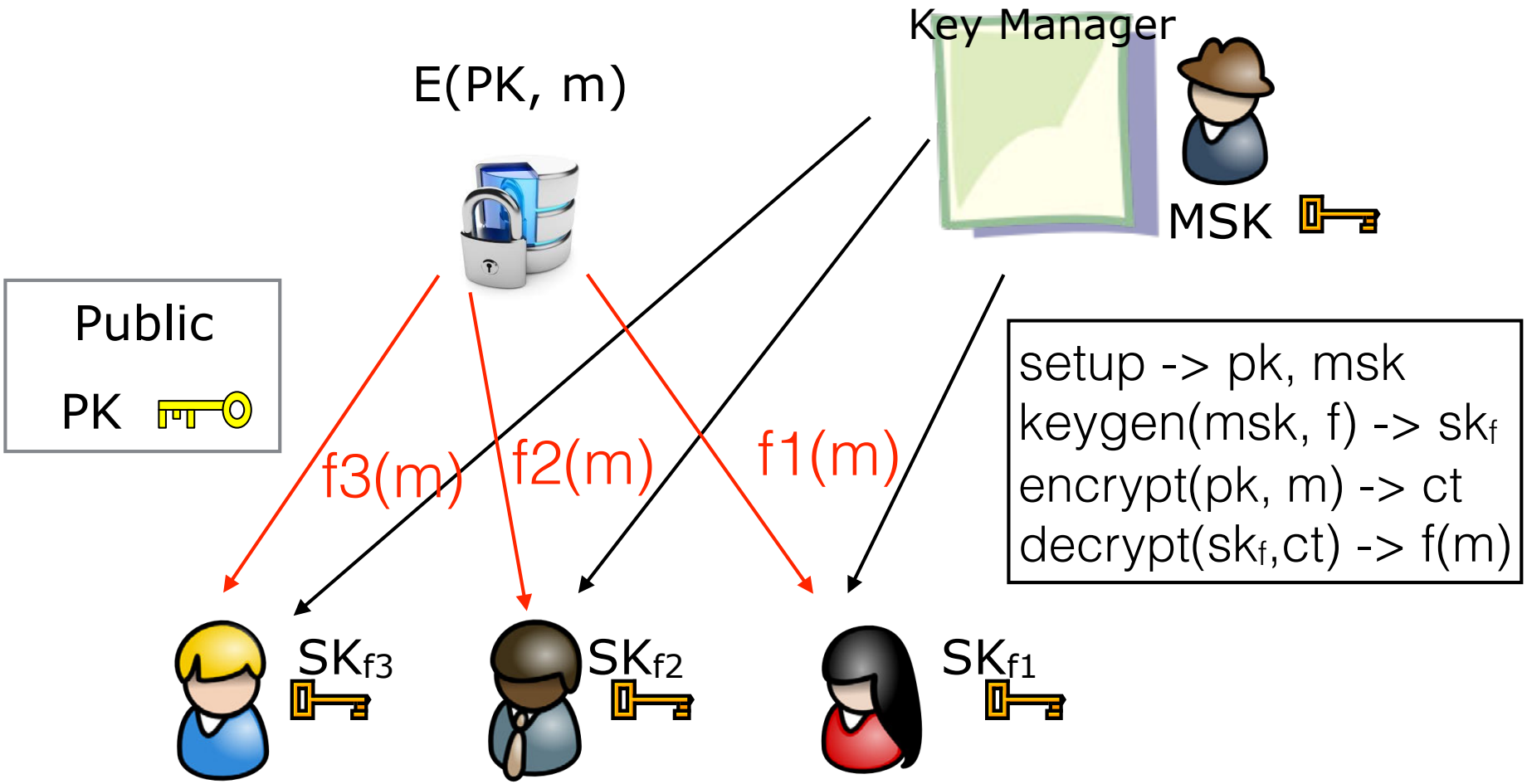
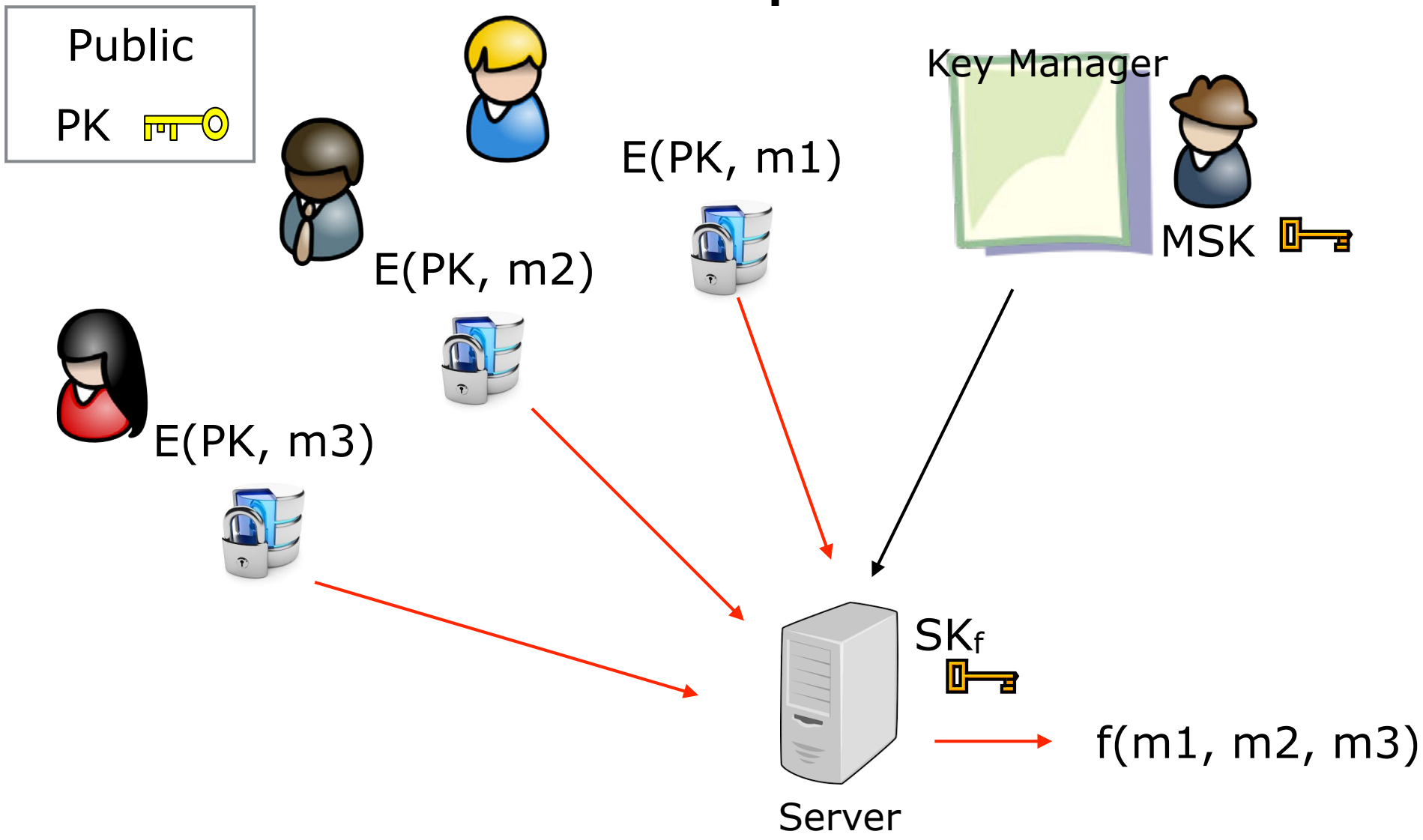
SGX Enabled Functional Encryption and Applications

Ben Fisch
Stanford University

Functional Encryption (FE)

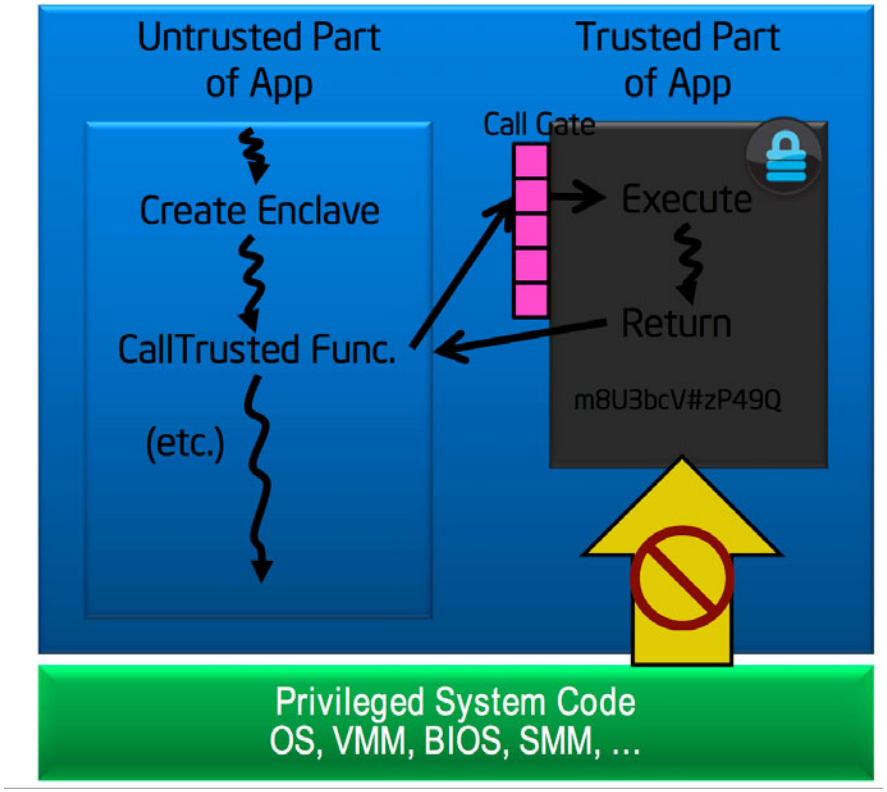


Multi-Input FE

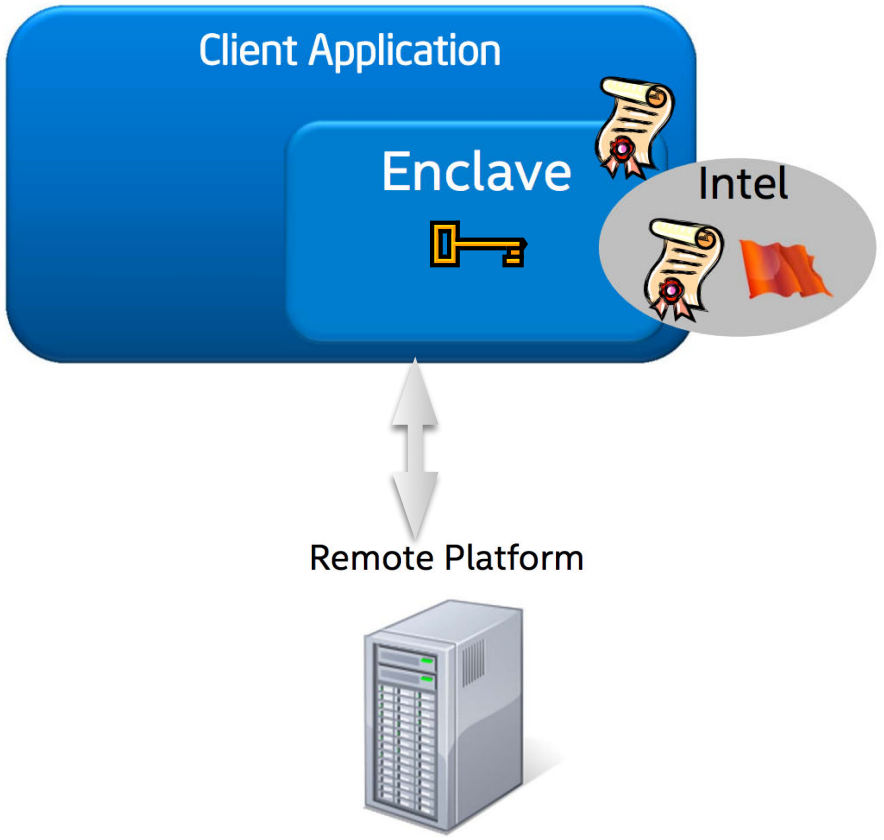


Intel SGX

Enclave Application

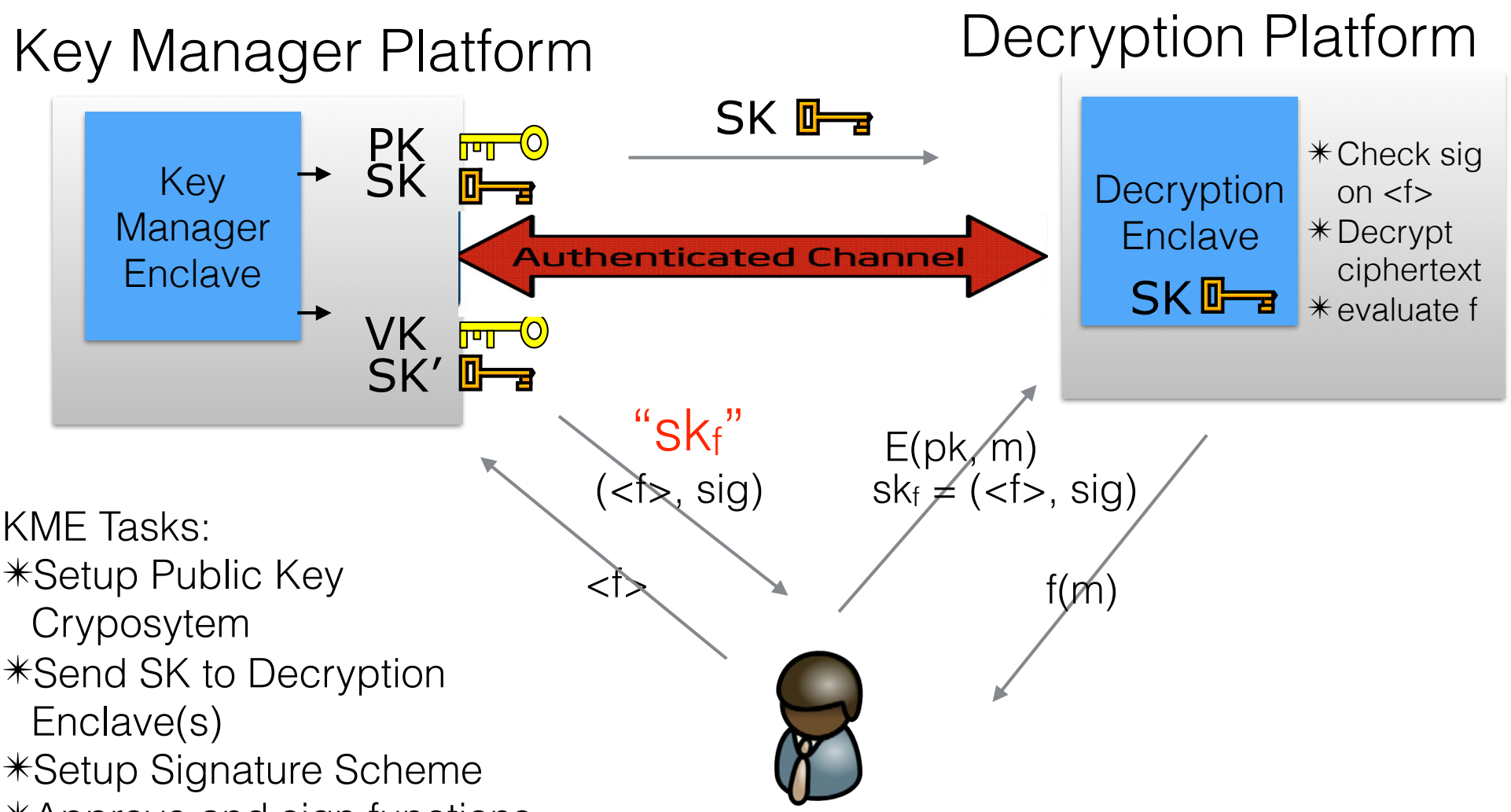


Remote Attestation



Source: ISCA 2015 tutorial slides for Intel SGX

SGX FE Design



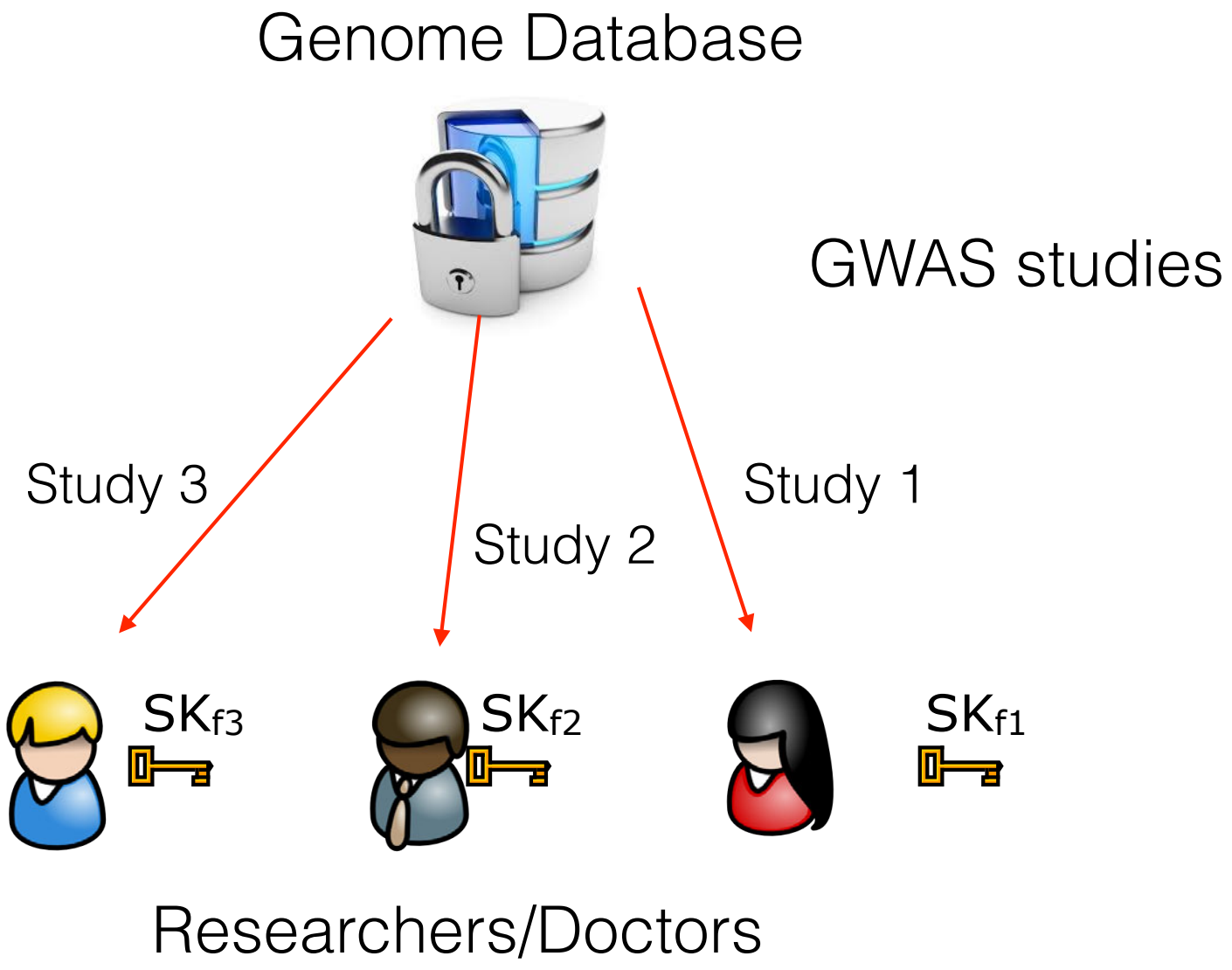
KME Tasks:

- ✳Setup Public Key Cryposytem
- ✳Send SK to Decryption Enclave(s)
- ✳Setup Signature Scheme
- ✳Approve and sign functions

Implementation Challenges

- How to represent the function f:
  - ✳ Cannot move code into enclave after EINIT
  - ✳ Implement interpreter in enclave?
  - ✳ Separate enclave runs f and gives attestation?
- Memory access patterns leaked
- Timing attacks

Genomics Application

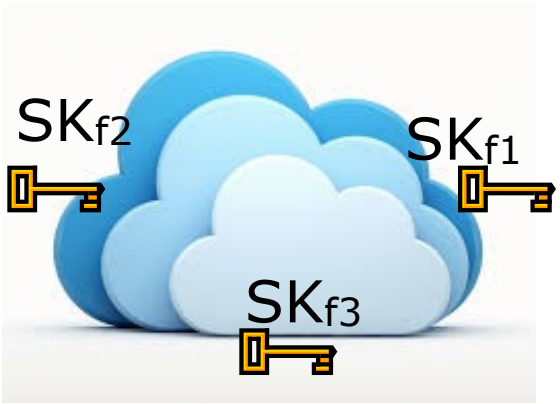


Smart Devices Application

Devices collect data

Encrypt data independently

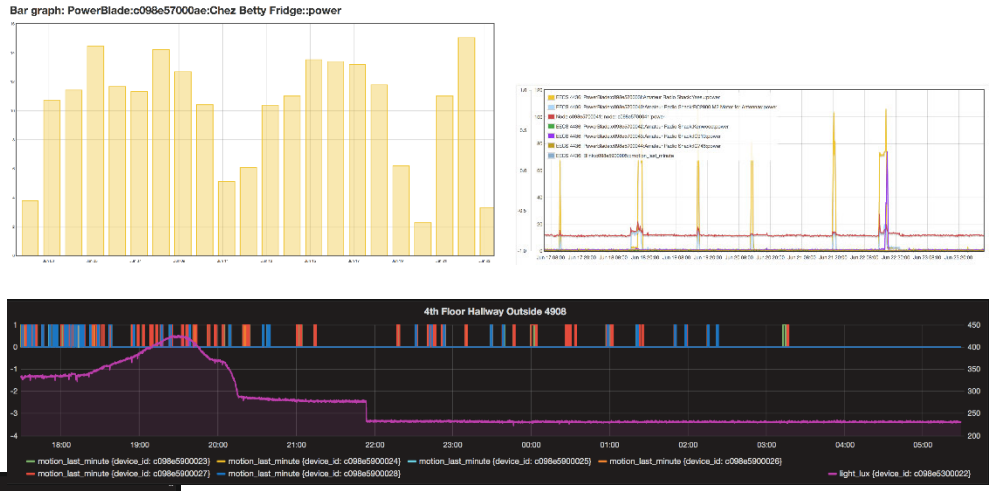
Apps in cloud learn restricted functions



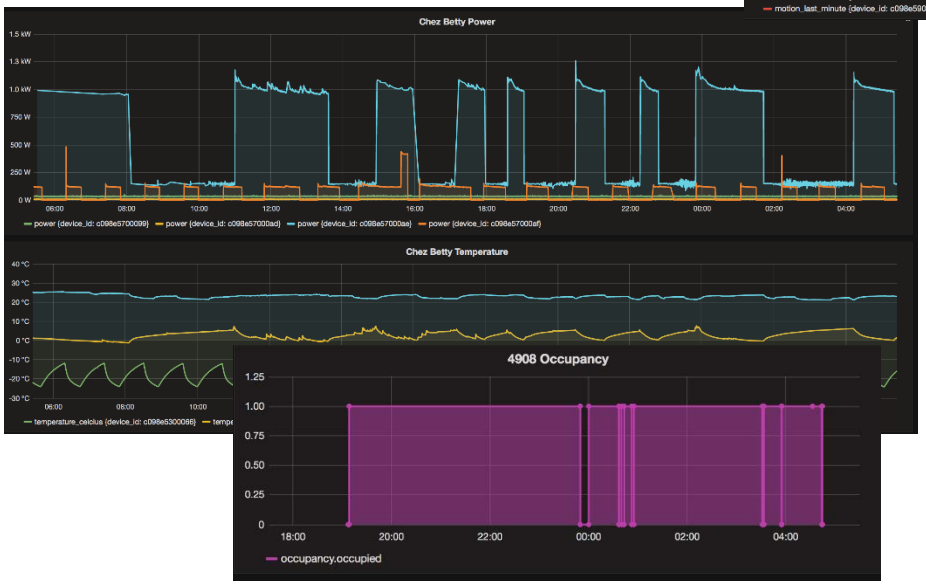
Better Buildings with BLE Sensors and Gateways

Brad Campbell – bradjc@umich.edu

Distributed applications running on gateways



Energy attribution and analysis



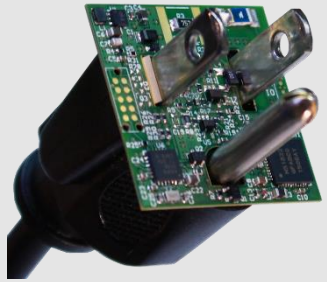
Real-time monitoring

Branden Ghena (brghena@umich.edu)

Michigan Architecture for Sensing Systems

Motion

Temperature
Pressure
Humidity
Light
Acceleration



Power
Voltage
Watt-Hours
Power Factor

Reference to interface description

Device specific data

Data translation script

Translation sandbox

Human readable data stream

```
{
  "device": "BLEES",
  "id": "c098e5300069",
  "pressure_pascals": 98423.5,
  "humidity_percent": 41.33,
  "temperature_celcius": 23.62,
  "light_lux": 49,
}
```

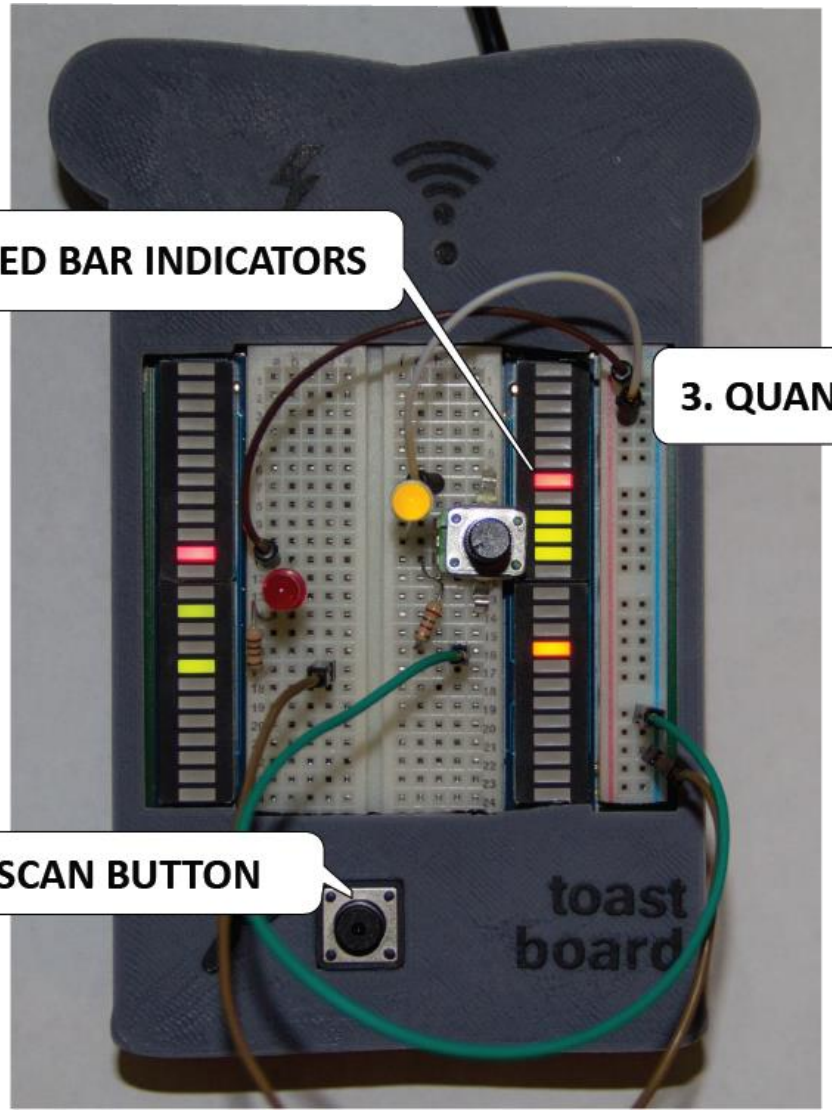
MQTT

The Toastboard

> “Professional software development environments usually provide a debugger, which helps programmers to locate and fix faults … Unfortunately, physical computing does not have analogous support tools and thus it was sometimes difficult for participants in our study to identify what the problem was. “
>
> Booth, Tracey, et al. "Crossed Wires: Investigating the Problems of End-User Developers in a Physical Computing Task."

Daniel Drew

Prof. Bjoern Hartmann



1. LED BAR INDICATORS

2. SCAN BUTTON

3. QUANTITATIVE DATA

4. COMPONENT TESTERS

5. TRANSIENT ANALYSIS

Toastboard

Add a Diode Add

Scan once Scan continuously

Row to Graph: 11 Rigl Start graphing

3.3V
1.9V
1.9V
3.3V
1.3V
1.5V
1.5V
0.0V
POT

This wire may not be inserted correctly!
How I know: The voltage at row 17 is not the same as the voltage at GND

3.0V
2.5V
2.0V
1.5V
1.0V
0.5V
0.0V
2s 4s 6s 8s 10s 12s 14s

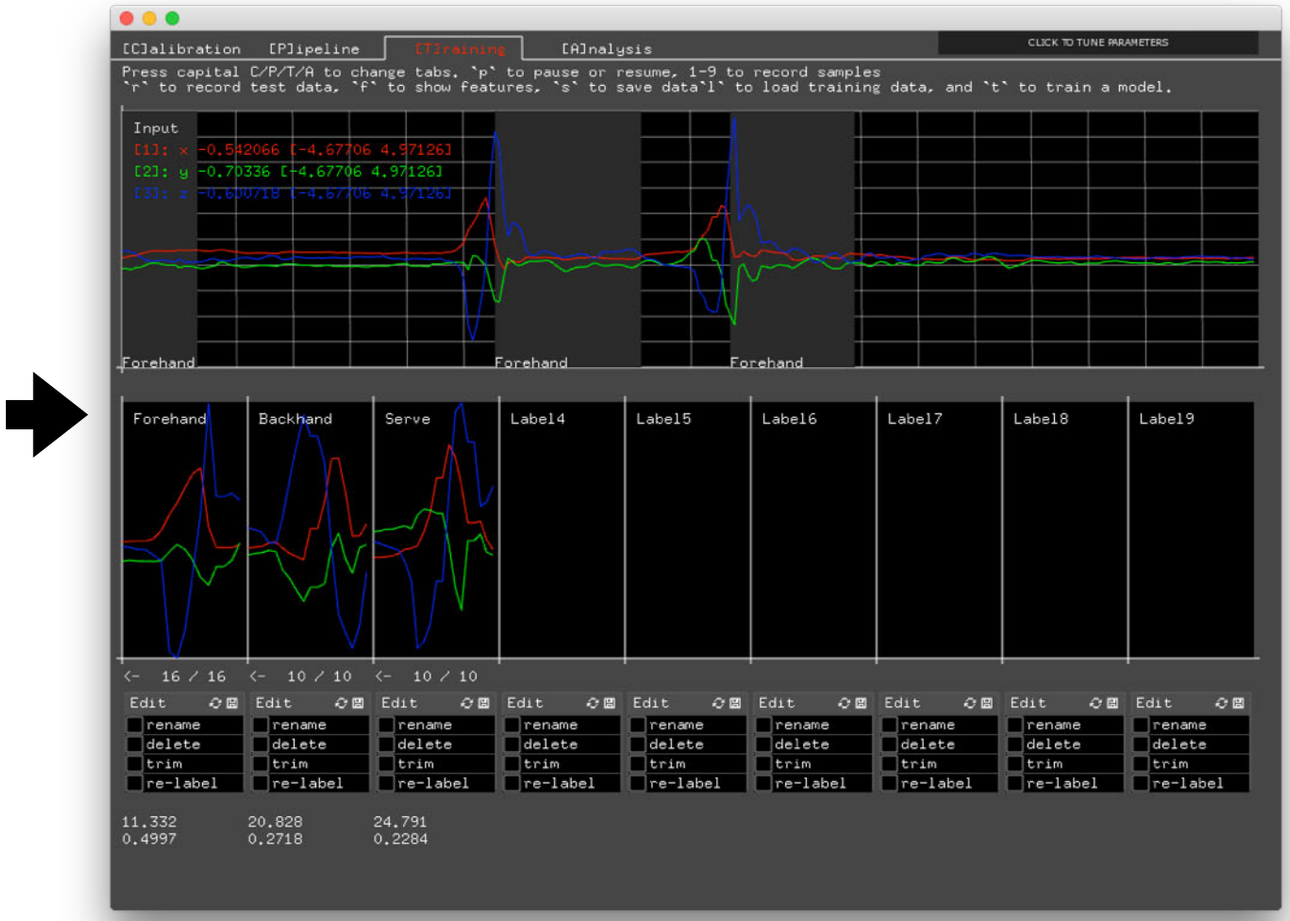
Machine Learning for Makers

David Mellis, Ben Zhang, Audrey Leung, Bjoern Hartmann

Experts Author Rich Code Examples

```
void setup()
{
    stream.setLabelsForAllDimensions({"x", "y", "z"});
    useStream(stream);

    calibrator.setCalibrateFunction(processAccelerometerData);
    calibrator.addCalibrateProcess("Resting",
        "Rest accelerometer on flat surface.", restingDataCollected);
    useCalibrator(calibrator);

    pipeline.setClassifier(DTW(false, true, threshold));
    pipeline.addPostProcessingModule(ClassLabelTimeoutFilter(timeout));
    usePipeline(pipeline);

    registerTuneable(threshold, 0.1, 3.0,
        "Similarity",
        "How similar a live gesture needs to be to a training sample. "
        "The lower the number, the more similar it needs to be.");
    registerTuneable(timeout, 1, 1000,
        "Timeout",
        "How long (in milliseconds) to wait after recognizing a "
        "gesture before recognizing another one.");

    useOStream(oStream);
}
```

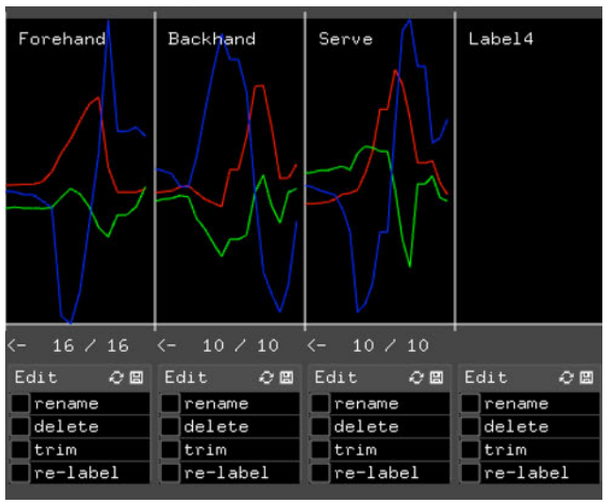
Interface Supports *Makers* in Applying the Examples



Machine Learning for Makers

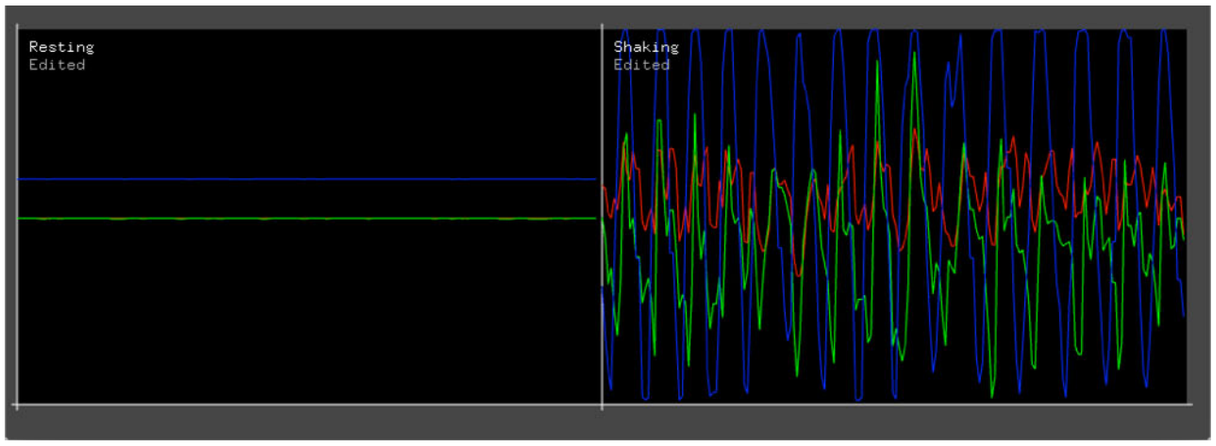
David Mellis, Ben Zhang, Audrey Leung, Bjoern Hartmann

Iterative Refinement of Training Data



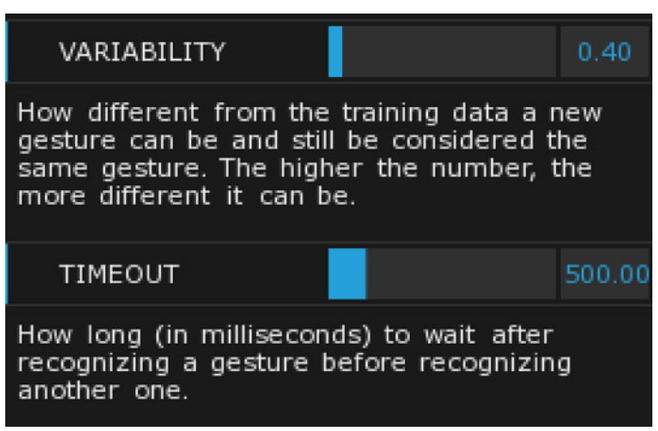
Calibration Across Sensor Models

```
calibrator.setCalibrateFunction(processAccelerometerData);
calibrator.addCalibrateProcess("Resting", "Rest accelerometer on flat surface, w/ z-axis vertical.", restingDataCollected);
calibrator.addCalibrateProcess("Shaking", "Shake accelerometer vigorously in all directions.", shakingDataCollected);
useCalibrator(calibrator);
```



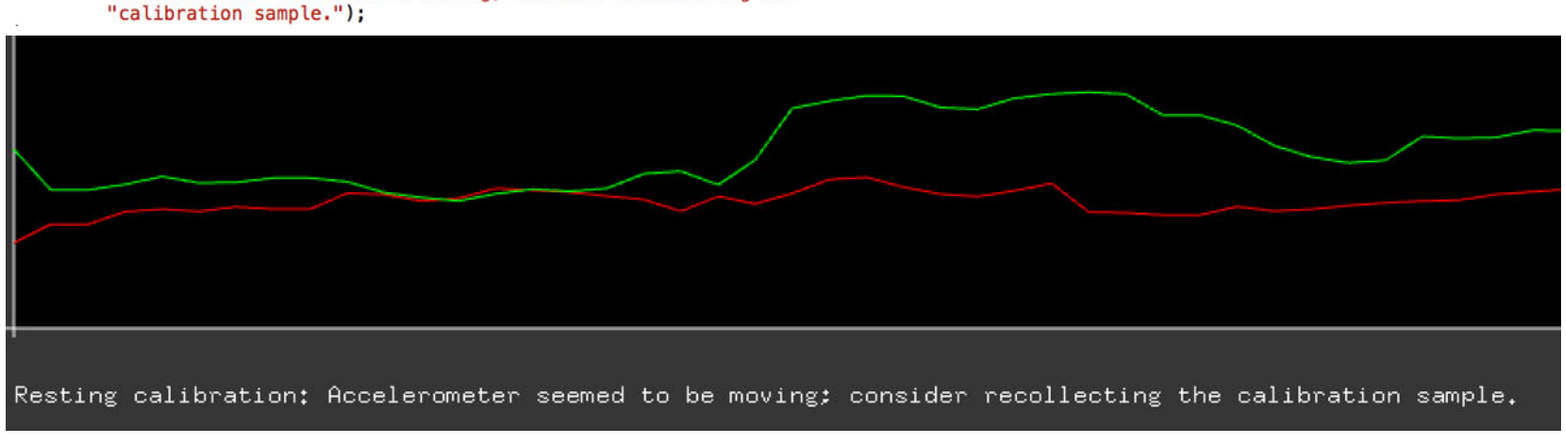
Tuning of Parameters

```
registerTuneable(null_rej, 0.1, 5.0, "Variability",
     "How different from the training data a new gesture can be and "
     "still be considered the same gesture. The higher the number, the "
     "more different it can be.");
registerTuneable(timeout, 1, 3000,
    "Timeout",
    "How long (in milliseconds) to wait after recognizing a "
    "gesture before recognizing another one.");
```



Feedback on Signal Quality

```
if (stddev[0] / range > 0.05 ||
    stddev[1] / range > 0.05 ||
    stddev[2] / range > 0.05)
    return CalibrateResult(CalibrateResult::WARNING,
        "Accelerometer seemed to be moving; consider recollecting the "
        "calibration sample.");
```

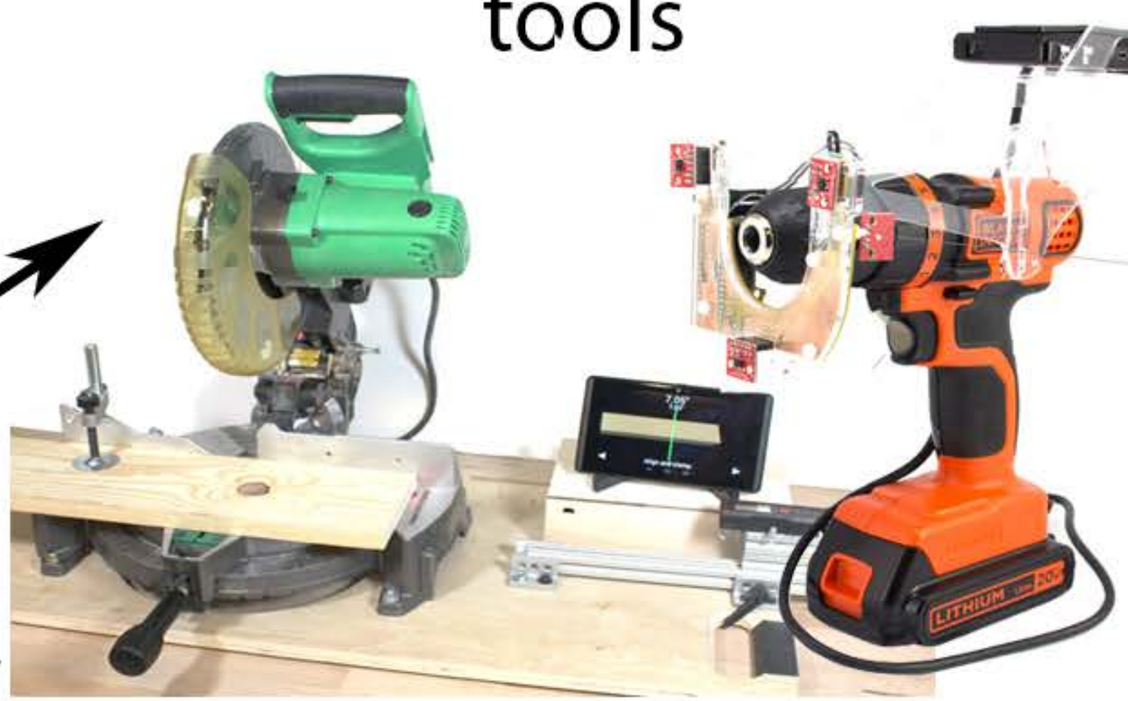


Drill Sergeant:

Supporting Physical Construction Projects through an Ecosystem of Augmented Tools

Eldon Schoop & Michelle Nguyen, Mitchell Karchemsky, Daniel Lim,
Valkyrie Savage, Bjoern Hartmann & Sean Follmer

tools

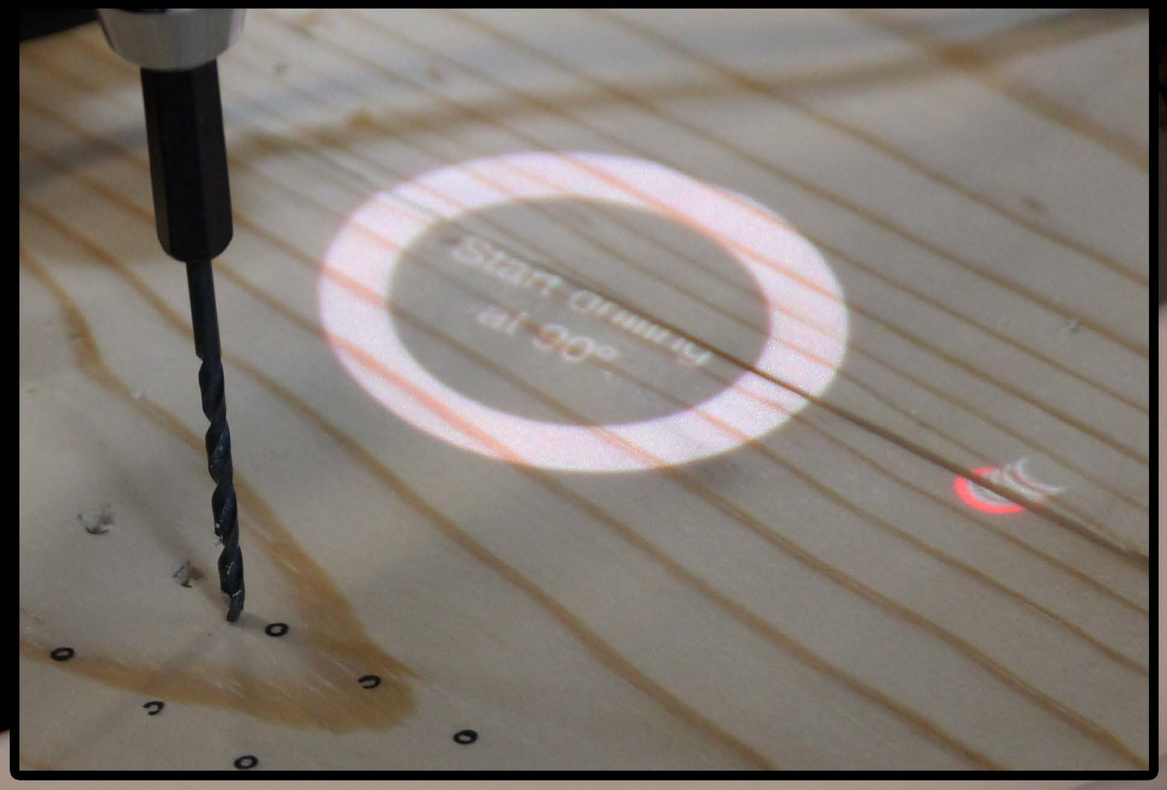


```
... {
      "tool": "drill",
      "task": "drillToDepth",
      "dynamic_params": ["depth"],
      "depth": "screw_depth"
}, ...
```

instructions

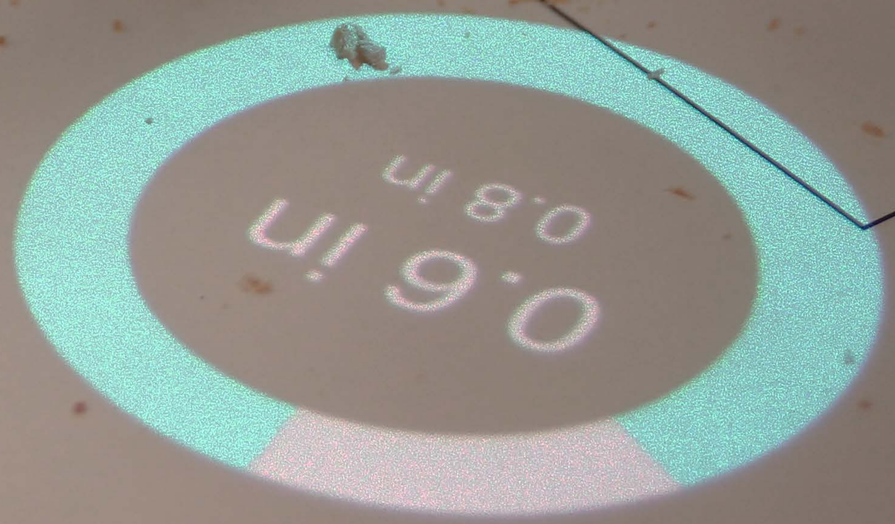
paper templates

object



incorrect orientation feedback

correct orientation feedback



depth feedback


BLACK+DECKER
LITHIUM
LITHIUM 1.5AH 20V MAX

Private Data Collection in an Internet of Things

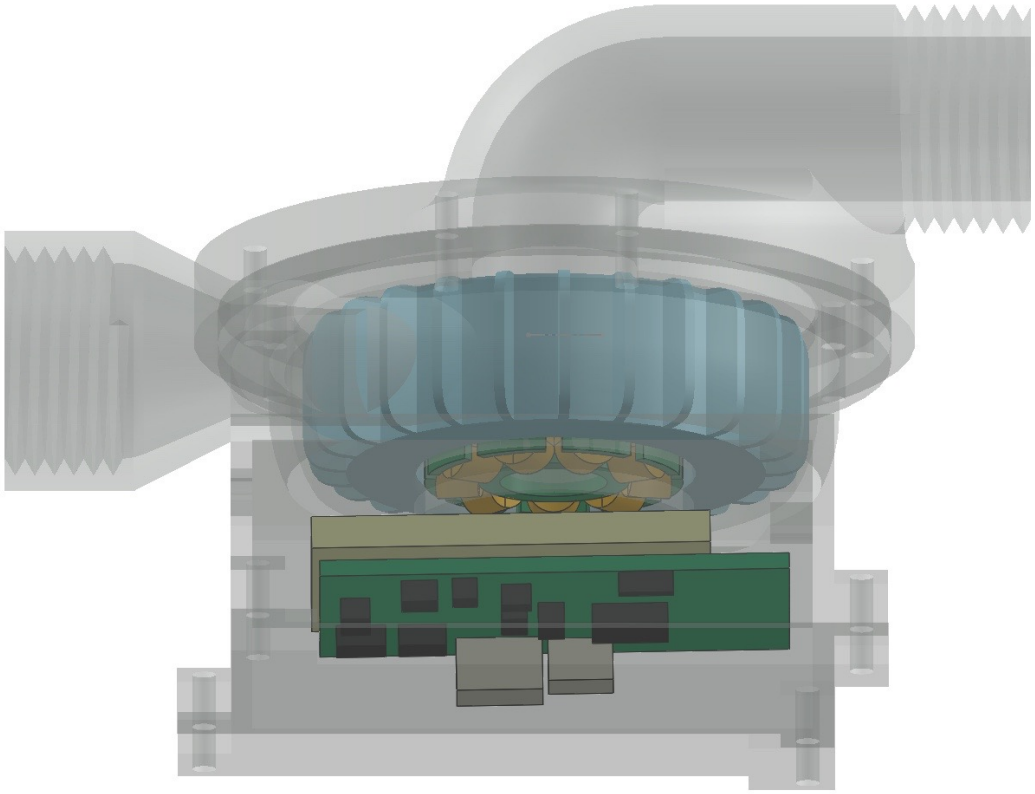
Henry Corrigan-Gibbs and Dan Boneh

- How can we collect privacy-sensitive data in a privacy-preserving way?
- Can we *efficiently* collect interesting aggregate statistics without collecting any individual user's data?
- What are the theoretical limits on the efficiency of these techniques?

Motivation

- California drought
- Collaboration with earth sciences and residential housing departments to reduce water usage across campus
- Gain an understanding of the patterns driving water consumption

Tethys: Energy Harvesting Networked Water Flow Sensor

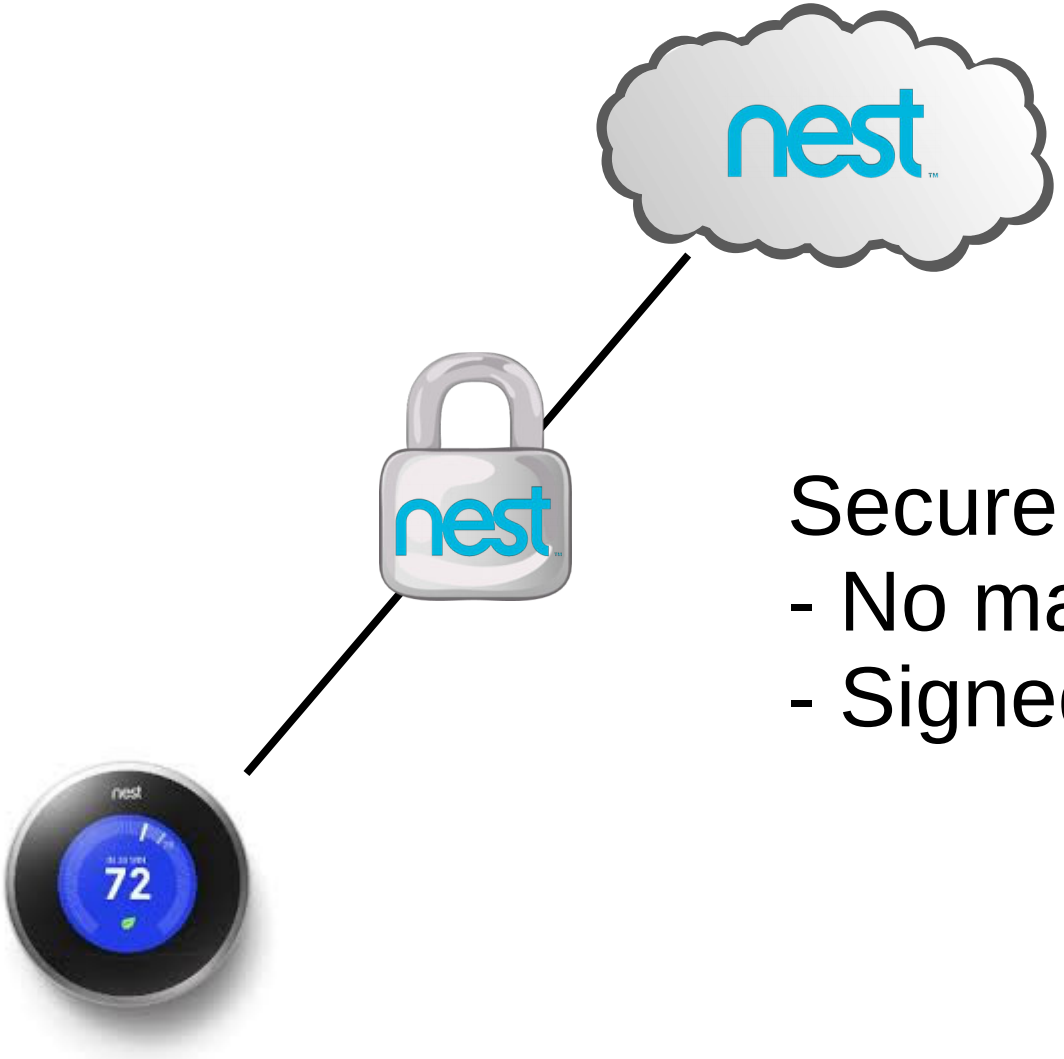


Auditing IoT Communications with TLS-RaR

Judson Wilson, Henry Corrigan-Gibbs, Riad S. Wahby, Keith Winstein, Philip Levis, Dan Boneh

Stanford University

How can we audit TLS communication between our IoT devices and the cloud?



Secure Devices:
- No man-in-the-middle
- Signed firmware

*Nest used for illustrative purposes only.

Goals

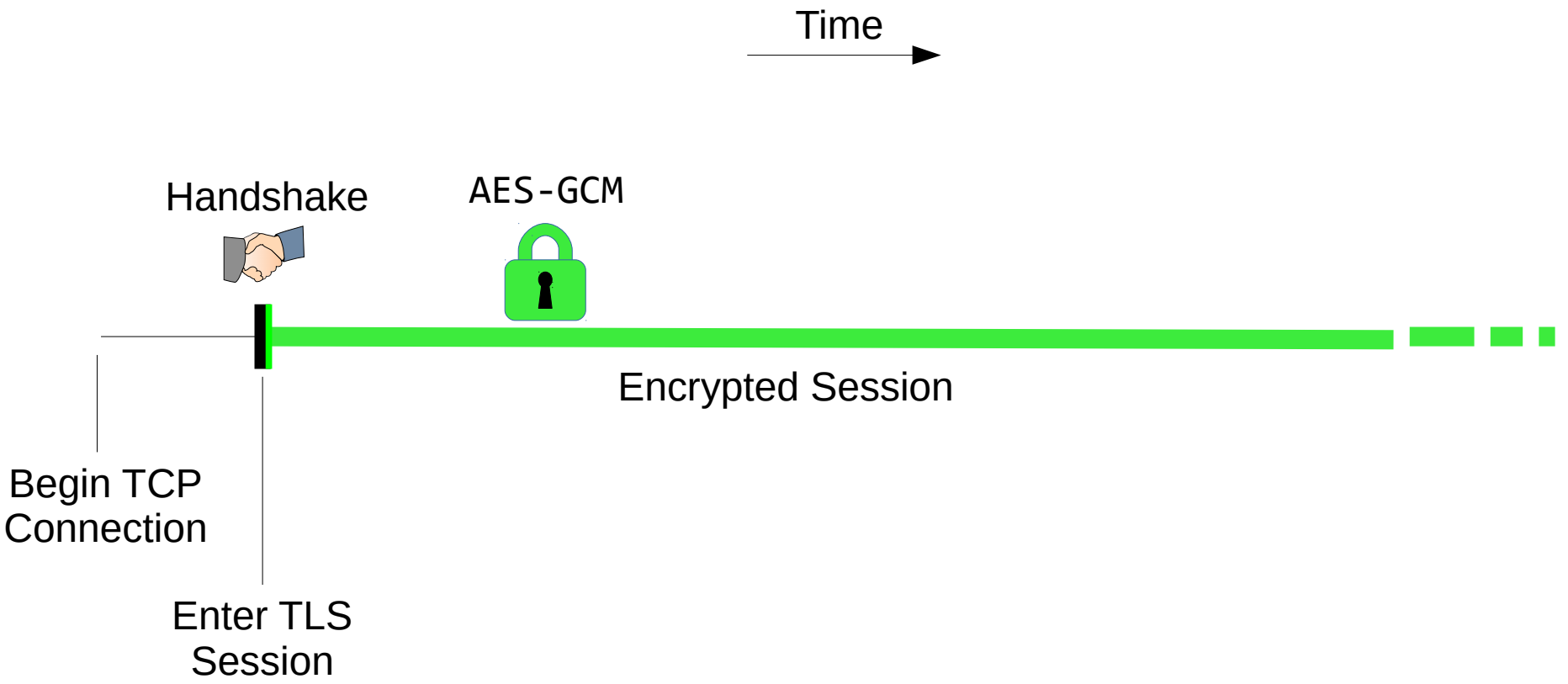
1) Preserve End-to-End Integrity

2) Fewest changes possible to TLS

3) Avoid changes to server side

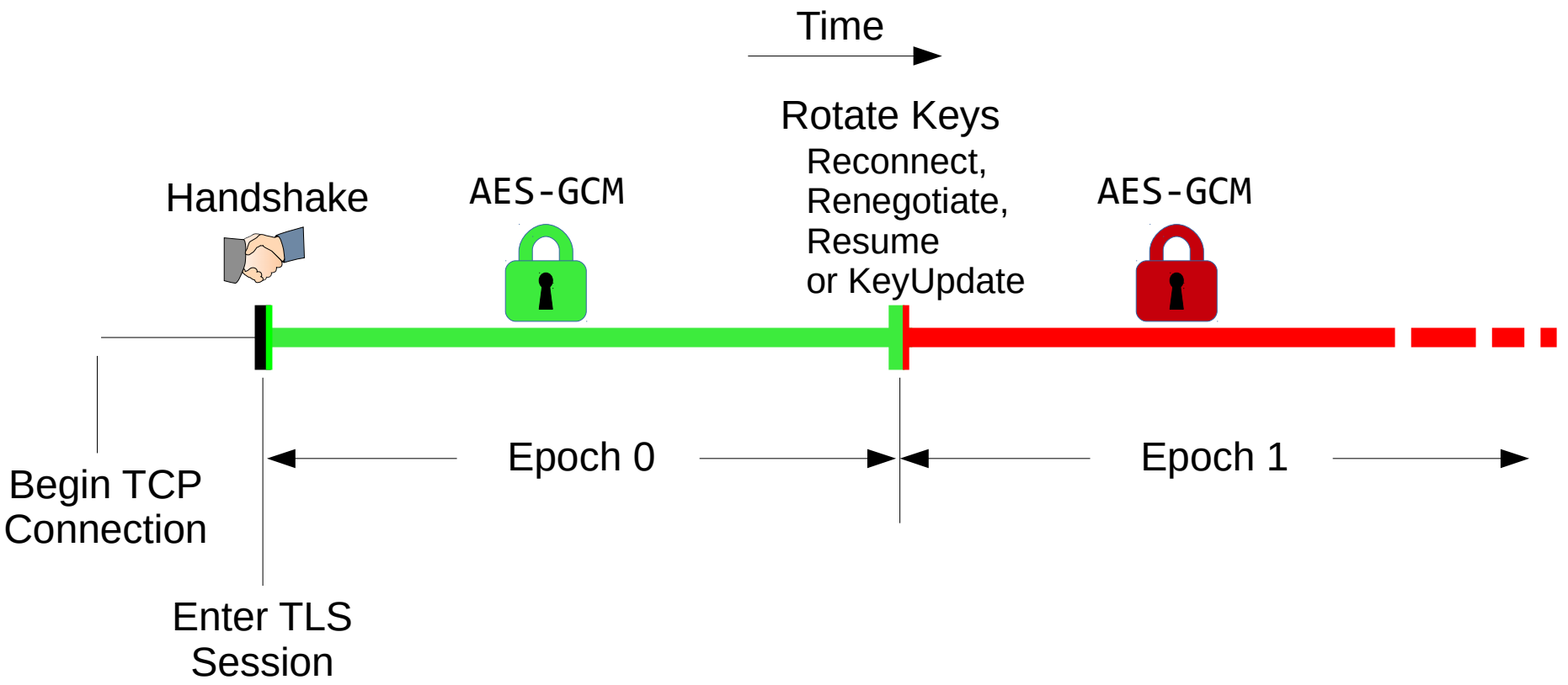
Solution: TLS-RaR

A standard TLS connection...



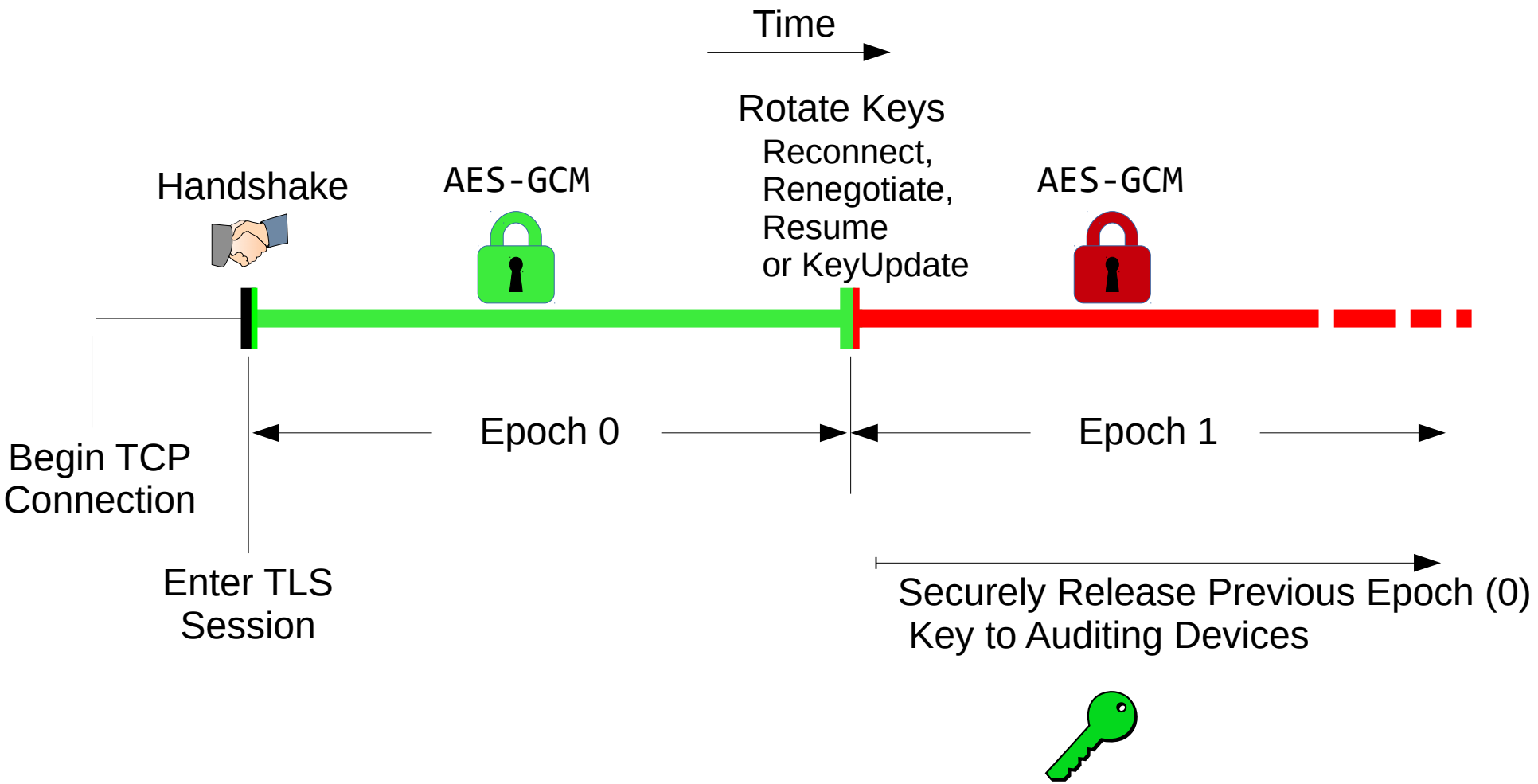
Solution: TLS-RaR

Use standard TLS features to **R**otate keys,



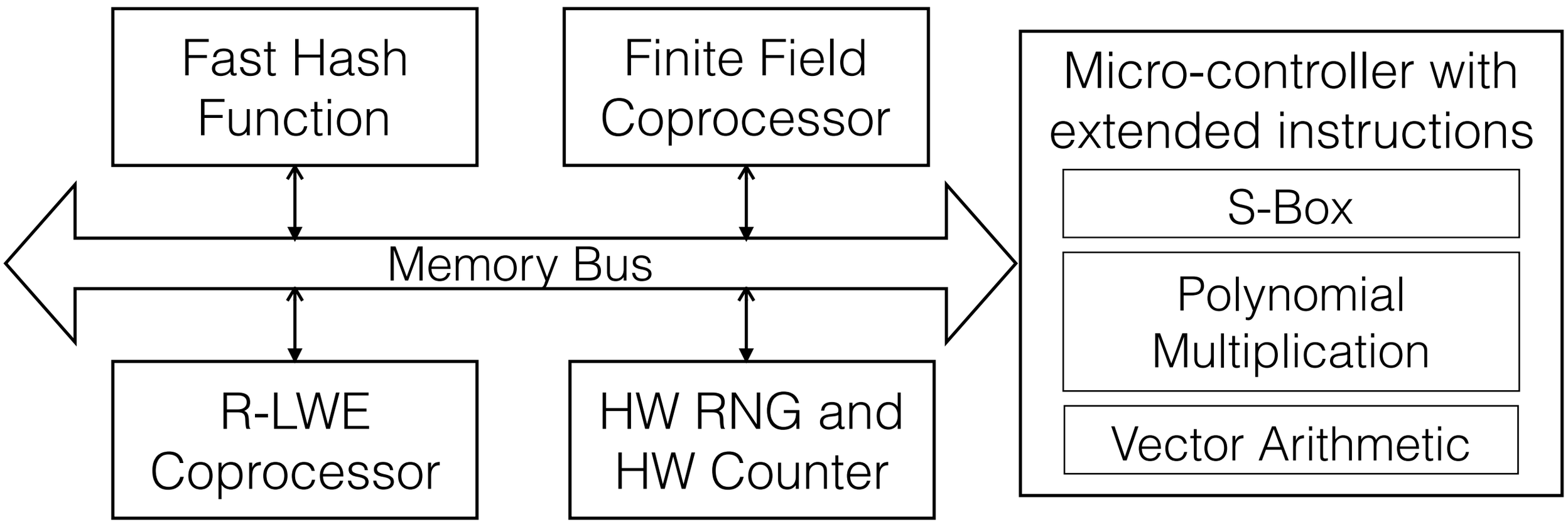
Solution: TLS-RaR

Use standard TLS features to **R**otate keys, **a**nd then securely **R**elease the previous keys to auditing devices.



CESEL

- Flexible hardware architecture for accelerating cryptography



Problem

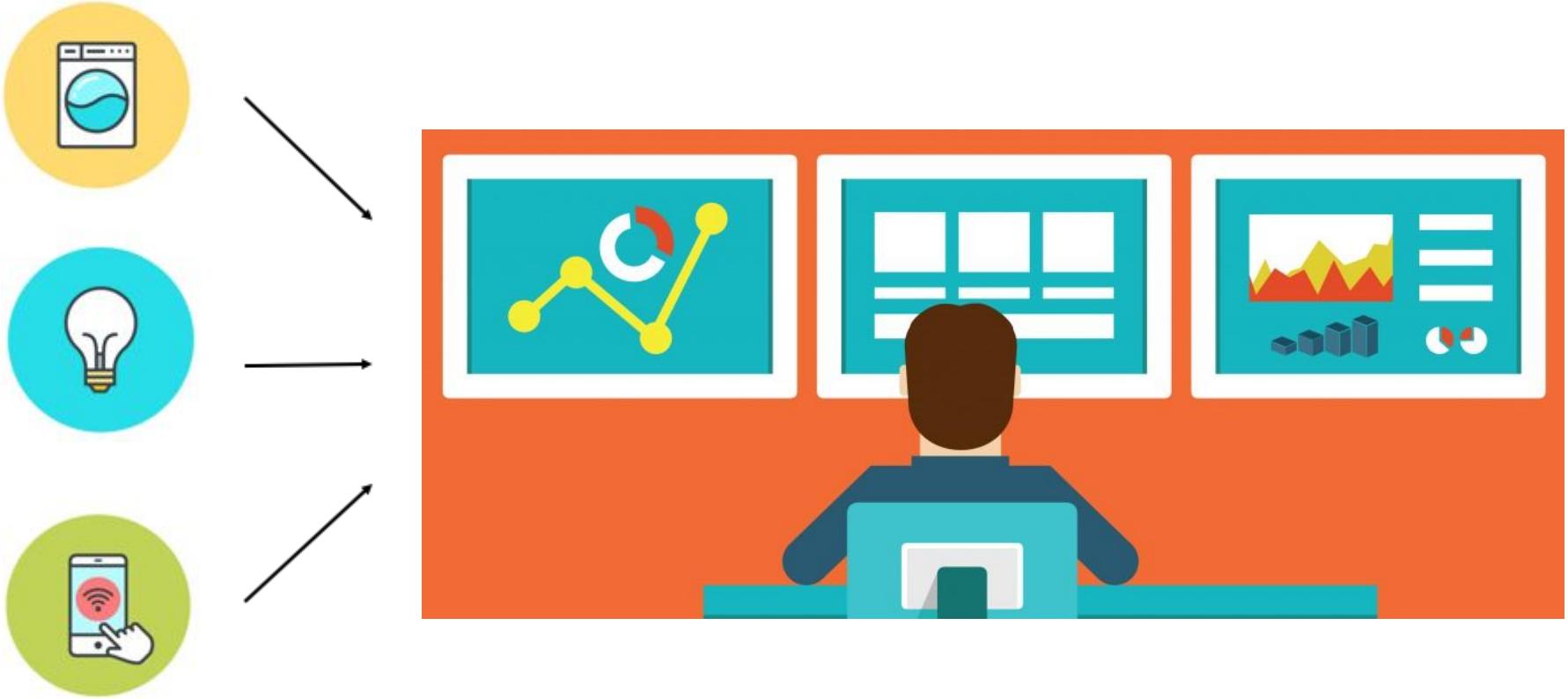
Internet of Things applications are complex distributed systems that include embedded devices, Internet gateways, and backend cloud services.

Their software often uses three or more programming languages, operating systems, and processor architectures for devices with dramatically different resources. This heterogeneity makes applications error-prone, laborious to develop and notoriously insecure.

How can we make the development process of IoT applications simpler and more secure?

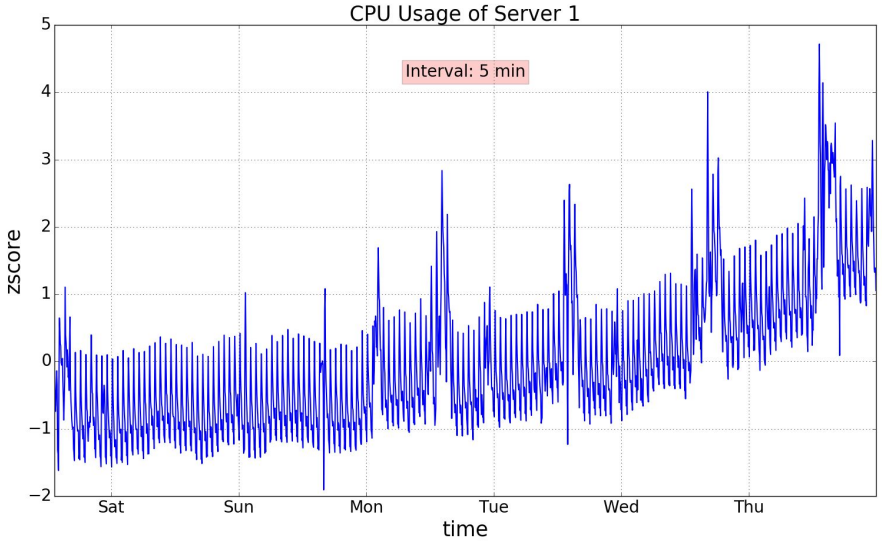
If you know the answer or want to hear an idea come and talk with me!

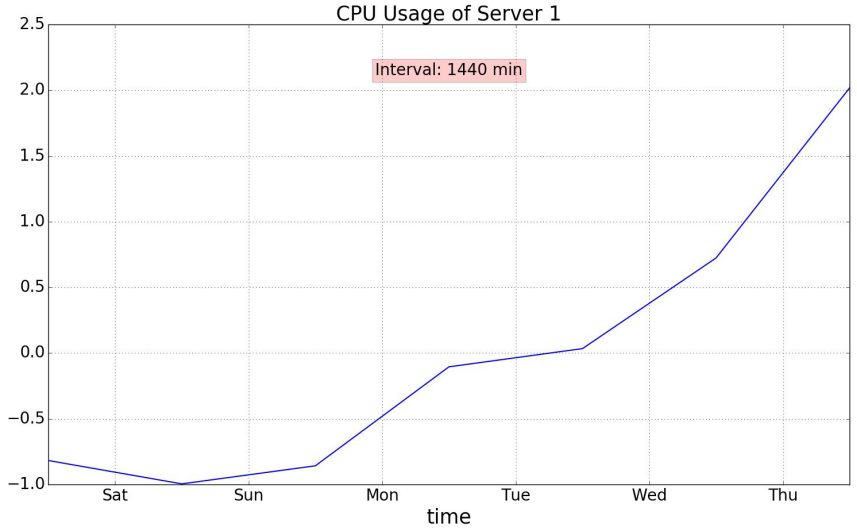
IoT generates a huge amount of times series data



Time series can be hard to interpret - *how to prioritize human attention*?

Challenge: reduce noise while preserving interesting features





ASAP: Automatic Smoothing Parameter Selection for Time Series Visualization

(Kexin Rong, Peter Bailis)

Motivation

- Prioritize user attention

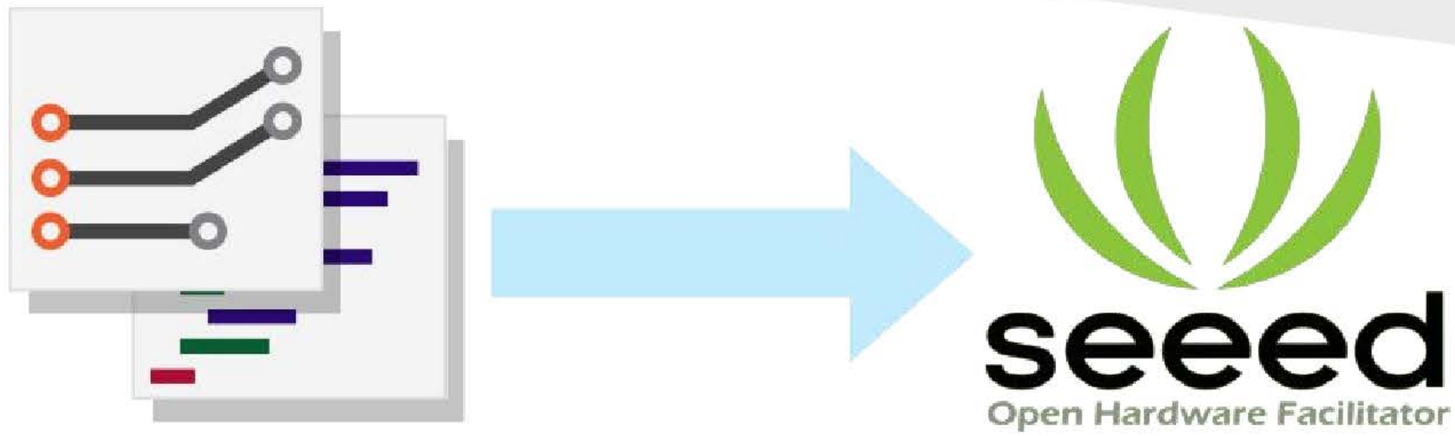
Key Insight

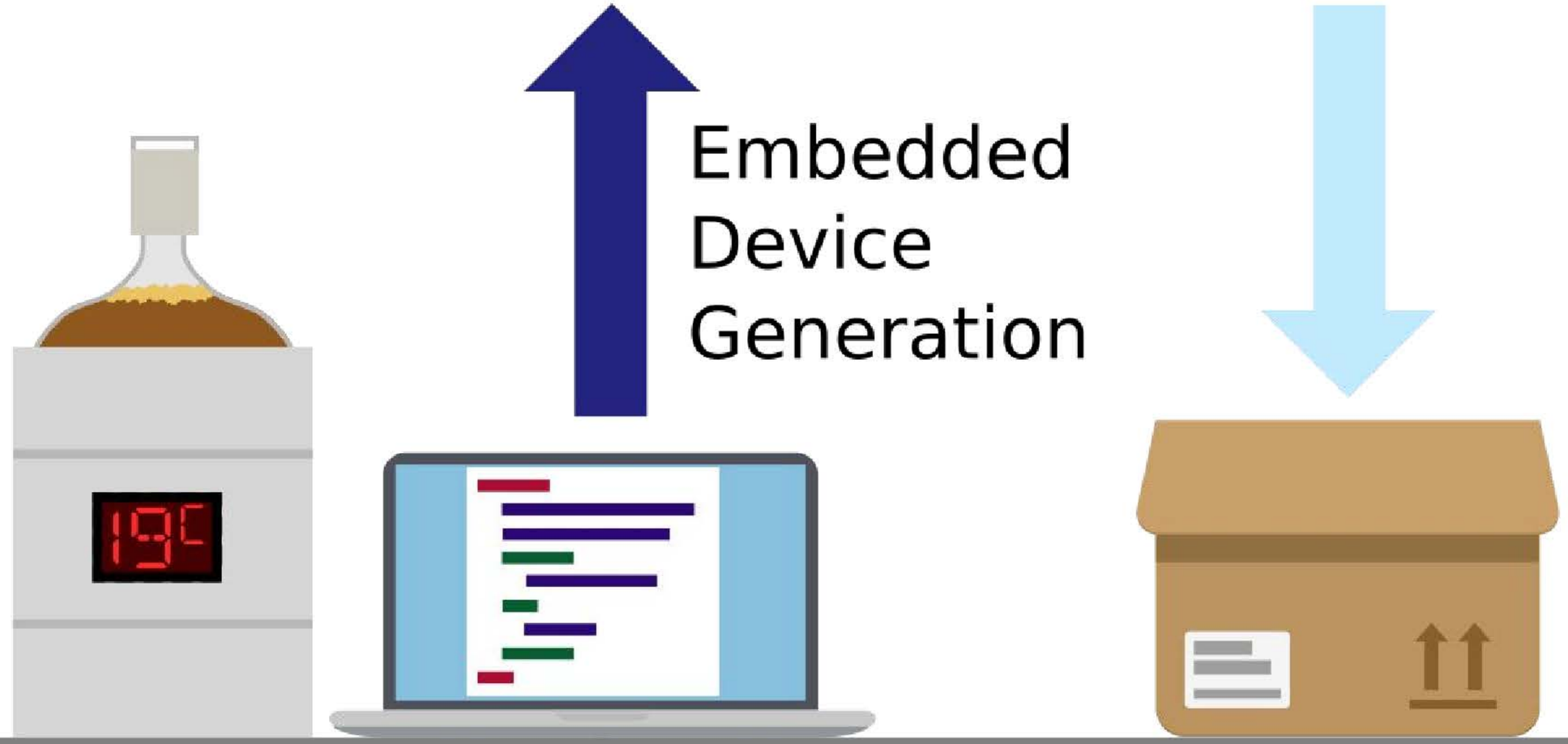
- Smooth as much as possible without losing “interesting” features

Approach

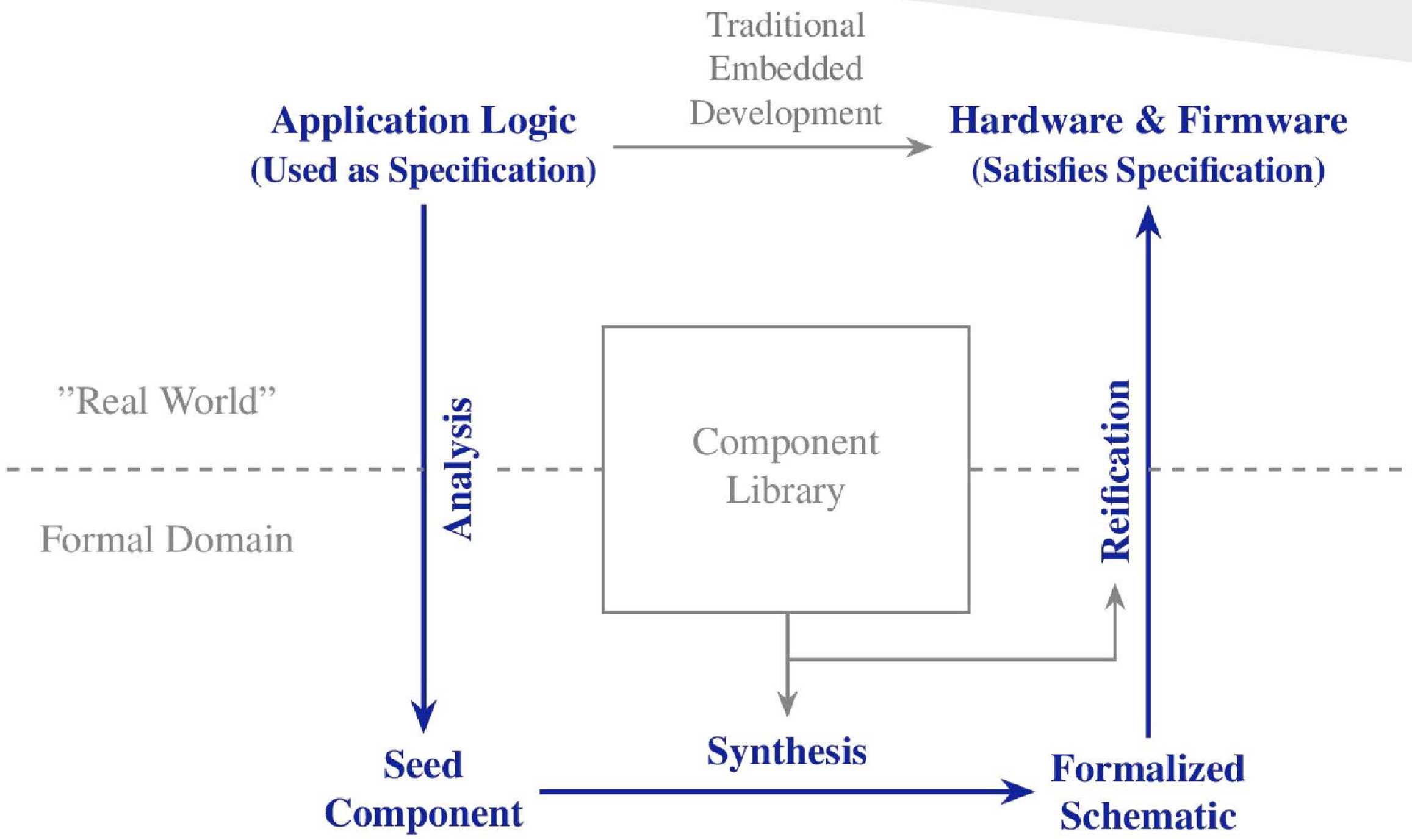
- Optimize signal smoothing parameters on the fly

Embedded Device Generation





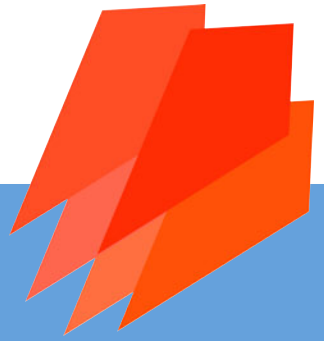
Embedded Device Generation

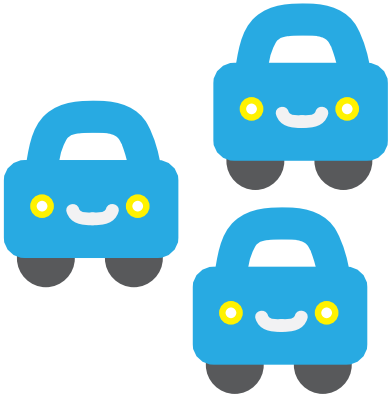


MIC DROP:

Massive IoT Computations via Dimensionality Reduction OPtimization

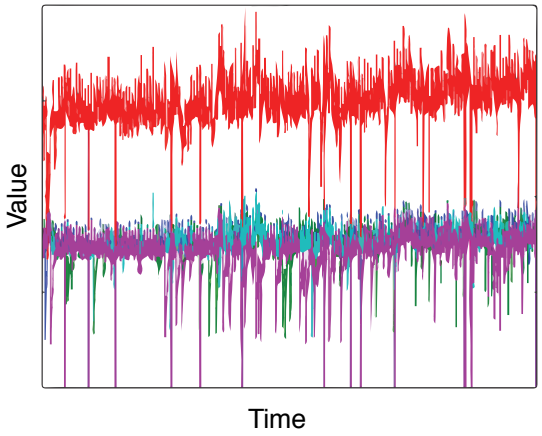
Sahaana Suri

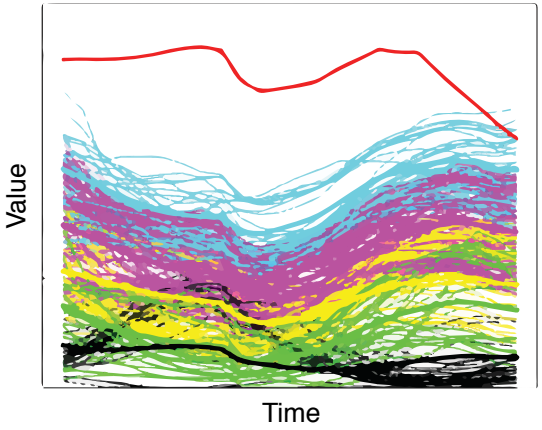


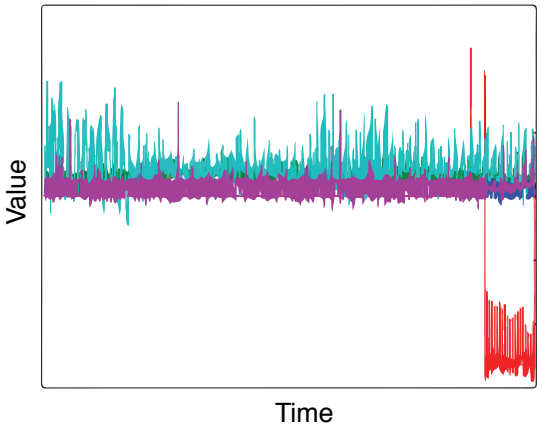


Internet of Things

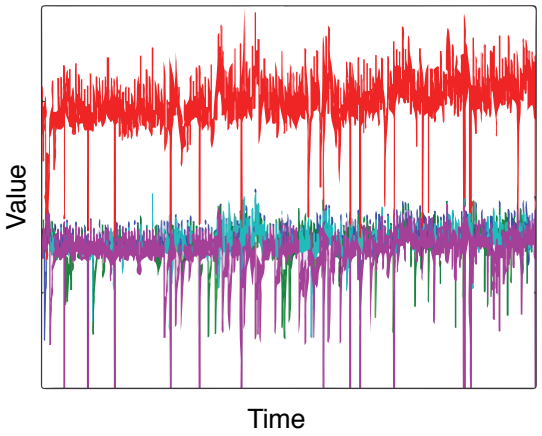
Millions of heterogeneous sensor readings per second

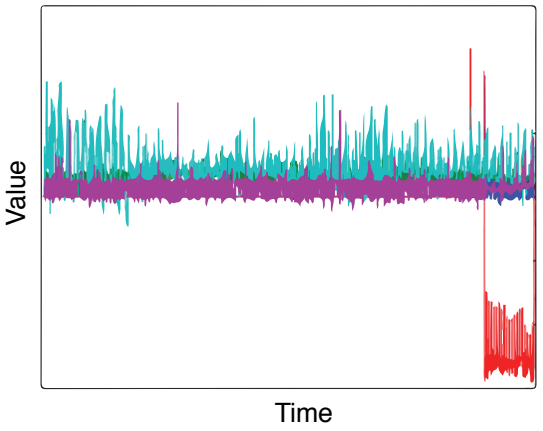






Millions of heterogeneous sensor readings per second





Gain real-time insight into system behavior via automated tools and machine learning systems

Millions of heterogeneous sensor readings per second

Automatically select dimensionality reduction technique that preserves information of interest

Quickly determine how to uncover data's **intrinsic dimensionality** subject to **latency constraints**

Gain real-time insight into system behavior via automated tools and machine learning systems

Millions of heterogeneous sensor readings per second

Learn quickly and bound error by training on bootstrap-based samples

Train multiple models at once via memoization

Seed techniques by using results from previously tested techniques

Gain real-time insight into system behavior via automated tools and machine learning systems

Automated Arbitrarily Complete Full-Loopback Driver Verification

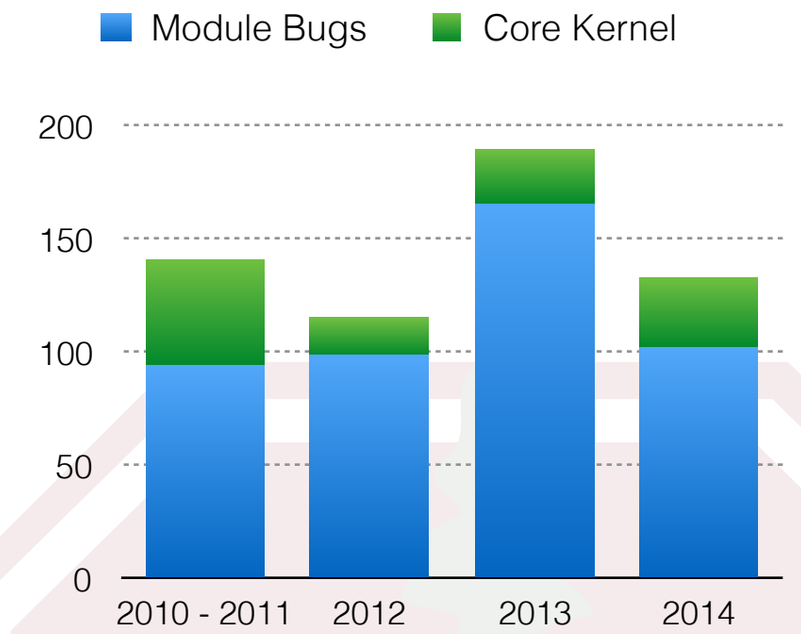
Sergio Benitez, Alejandro Russo, David Mazieres

Problem:

91% of critical CVEs caused by drivers leading to *serious* vulnerabilities

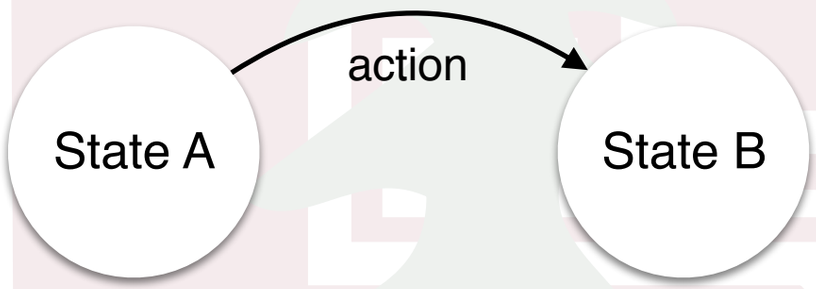
NetUSB flaw leaves 'millions' of routers, IoT devices vulnerable to hacking

The flaw can be exploited to conduct denial-of-service attacks or remote hijacking.



Observation:

Devices modeled by state machines



Our Approach:

Guarantee device is programmed according to state machine at compile time via programming language properties

Automated Arbitrarily Complete Full-Loopback Driver Verification

Sergio Benitez, Alejandro Russo, David Mazieres

Completed:

- ✓ Formalized and proved type system.
- ✓ Formalized and proved correctness properties.
- ✓ Developed (small) OS kernel and drivers.
- ✓ Found bugs in manuals and hardware!

Observation:

Device specification/hardware still susceptible to bugs.

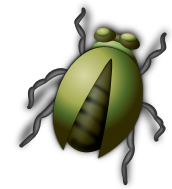
State “actions” cannot be statically checked.

Approach:

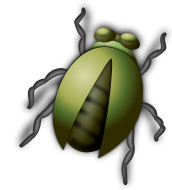
Generate (arbitrarily) complete correctness tests using SM model.

Run tests against real hardware (full-loopback).

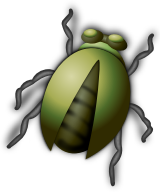
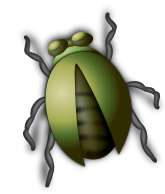
When developers **write code**,

they make **mistakes**. 

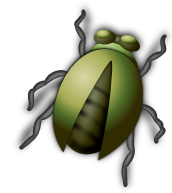
When developers **write code**,

they make **mistakes**. 

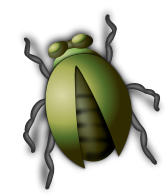
When developers **design hardware**,

 they make **mistakes**. 

When developers **write code**,

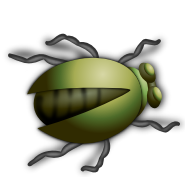
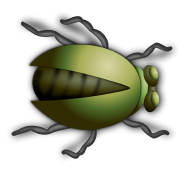
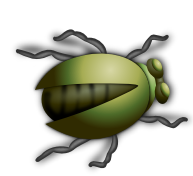
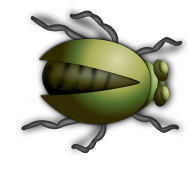
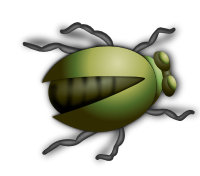
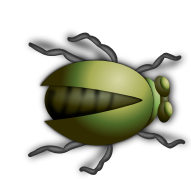
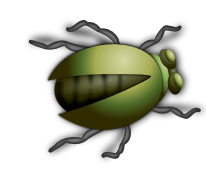
they make **mistakes**. 

When developers **design hardware**,

 they make **mistakes**. 

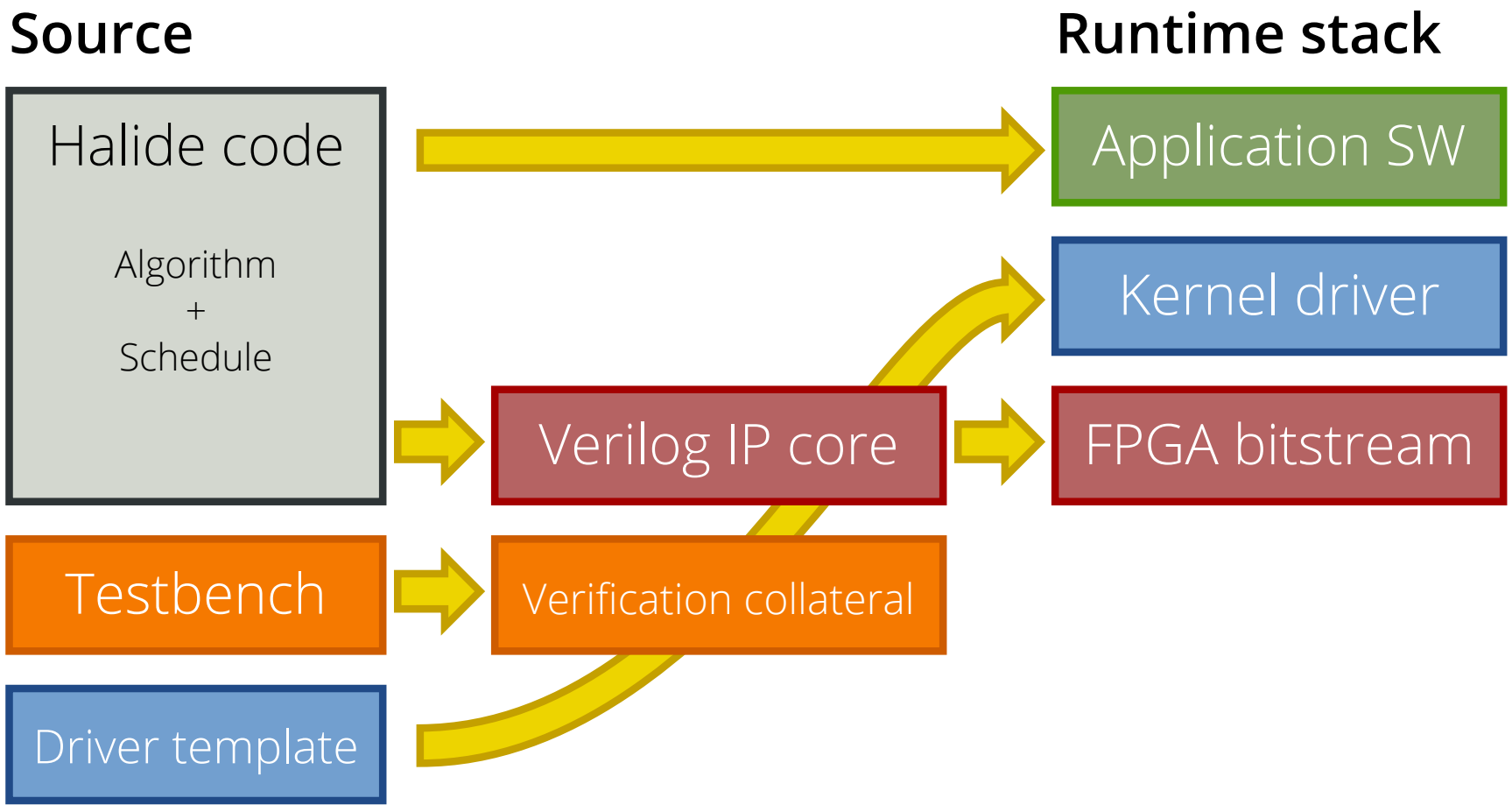
And when they write code for **custom hardware**,

bugs come from the HW and SW, but
also from the interface.

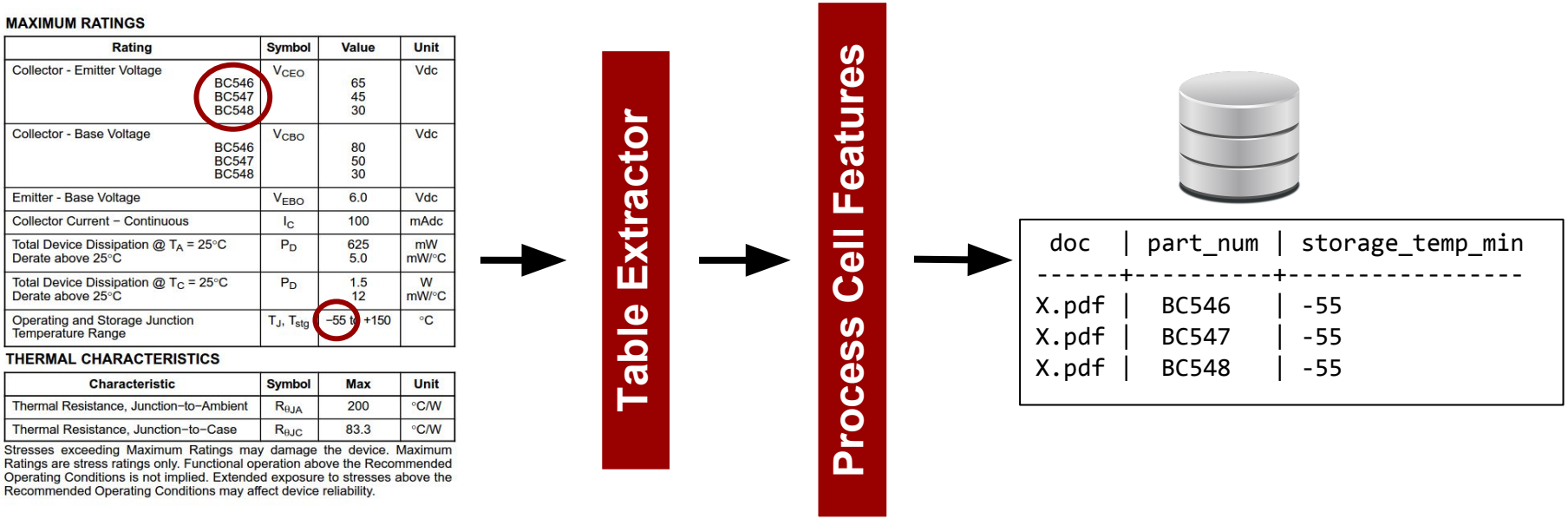
High-level languages and generators can help.

We're automatically building complete HW/SW stacks for image processing algorithms.



Come ask questions!

What we're doing...



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Collector - Emitter Voltage BC546 BC547 BC548	V_{CEO}	 65 45 30	Vdc
Collector - Base Voltage BC546 BC547 BC548	V_{CBO}	 80 50 30	Vdc
Emitter - Base Voltage	V_{EBO}	6.0	Vdc
Collector Current − Continuous	I_C	100	mAdc
Total Device Dissipation @ T_A = 25°C Derate above 25°C	P_D	625 5.0	mW mW/°C
Total Device Dissipation @ T_C = 25°C Derate above 25°C	P_D	1.5 12	W mW/°C
Operating and Storage Junction Temperature Range	T_J, T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction−to−Ambient	$R_{\theta JA}$	200	°C/W
Thermal Resistance, Junction−to−Case	$R_{\theta JC}$	83.3	°C/W

Stresses exceeding Maximum Ratings may damage the device. Maximum Ratings are stress ratings only. Functional operation above the Recommended Operating Conditions is not implied. Extended exposure to stresses above the Recommended Operating Conditions may affect device reliability.

Table Extractor

Process Cell Features

doc	part_num	storage_temp_min
X.pdf	BC546	-55
X.pdf	BC547	-55
X.pdf	BC548	-55

Poster Title: Building a Component Library from Datasheets

Come chat with us if you're interested in...

- Embedded hardware design productivity
- New applications built on a detailed component library
- Extracting data from PDFs and tables
- Giving feedback