Devices as Services

Fatih Bakir[1], Rich Wolski[1], Chandra Krintz[1], Gowri Sankar Ramachandran[2]

1. University of California Santa Barbara
2. University of Southern California

UCSB

USC Viterbi
School of Engineering
Center for Cyber-Physical Systems
and the Internet of Things

Devices as Services

- More and more focus on Edge
  - Low latency
  - Better scaling
- End devices are increasingly powerful
  - More memory, faster processors, more hardware
- Can perform tasks other than telemetry
  - Actuation
  - Computation, inference
- Our approach:
  - Treat end devices as *nanoservices*
  - Composition of nanoservices = applications

Nanoservices

- Services on microcontrollers
  - Difficult to program
  - Need a common paradigm across the board
  - Heterogeneity
  - Low power devices on internet: security
- Overview:
  - End to end FaaS (ie. Serverless)
  - Composition of services
  - Security

Programming

- A single FaaS platform
    - Run the same program on devices, edge and cloud

- Cspot
    - Low level, event driven programming
    - Coupled with append only storage abstraction
    - Custom OS on embedded
    - Open source: https://github.com/Mayhem-lab/cspot

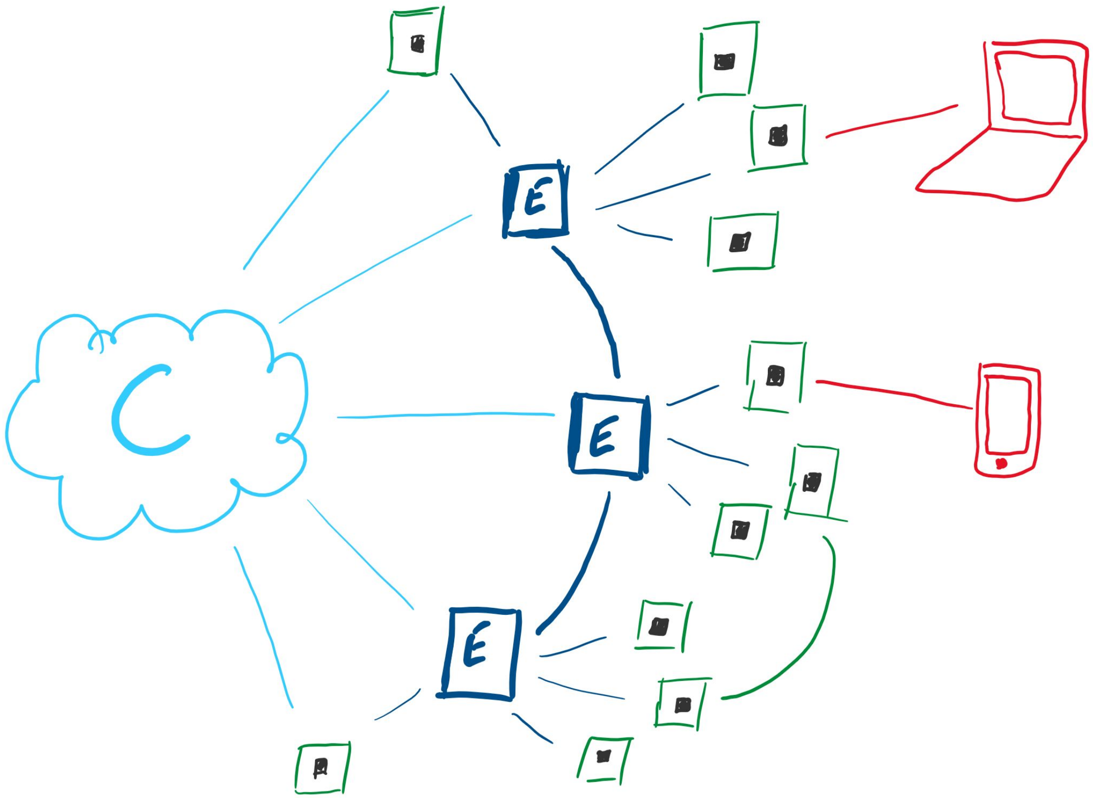
- How to compose these services?

Edgeistry

- Orchestration for nanoservices
- Decentralized & highly distributed

- Service discovery
- Identity management
- Reverse proxying for devices
  - Anonymity
  - Speed matching
  - Caching

Overall Architecture



Security

- Conventional solutions don’t work well on devices
  - Asymmetric crypto computationally intensive
  - TLS needs large buffers (>16K) for packets
- A lightweight security primitive is needed
- Our approach: Capabilities using HMACs
  - Very efficient, runs on 8 bit microcontrollers with 2K RAM
  - Allows for distributed sharing, without the device
  - Full policy delegation to clients
  - Composable
- Privacy and authentication are orthogonal!
- How would end to end numbers look like?

Results & Evaluation

- End to end latency comparison vs Amazon Web Services (AWS)

- *AWS:* Device -> Cloud (Handle with lambda, store in dynamodb, RSA2048)
- *Our:* Device -> Edge -> Cloud (Handle with cspot, store with cspot, capabilities)

- *AWS: 5,578* milliseconds
- *Our:* 608 milliseconds
  - An order of magnitude faster

- Micro benchmarks (on ESP8266 WiFi SoC)

Microbenchmarks for capabilities

Algorithm	Sign Time	Verify Time
PKCS1 (2048 Bits)	3280 ms	187 ms
ECDSA (256 Bits)	214 ms	4340 ms
HMAC-SHA256 (128 bit key)	0.37 ms	0.37 ms
Our Work	0.58 ms	0.9 ms

Operation	Time
RSA Handshake (2048 Bit)	3.95 Seconds

Conclusion

- Can build FaaS end to end
- Portable to many architectures
- Ability for efficient authentication on the device is key
- Devices as services is a viable approach for IoT

Discussion

1. Open questions:
   a. How to incorporate privacy efficiently?
   b. How does discovery get done?
   c. How to commission new devices?
2. Feedback
   a. Are there counterexamples?
3. Controversial points
   a. Internet is built backwards for IoT
   b. Services with duty cycles?
4. How this fails
   a. Assumes devices will increase in power & functionality