On Enabling Technologies for the Internet of Important Things

Marten Lohstroh, Hokeun Kim, John Eidson, Chadlia Jerad, Beth Osyk, and Ed Lee

Presented by: Gabe Parmer

Mission Statement

- The Internet has a number of useful things:
  - global namespaces
  - reliable delivery
  - security through asymmetric encrypton
  - certificate-based authentication
  - aggregation and mass processing of data

Mission Statement II

- But lets consider
  - Timeliness
  - Quality of service (QoS)
  - Physical safety
  - Security & privacy

Challenges

- Software longevity (cloud service/app/device)
- Security
  - Non-safety critical devices?
    YES (Mirai) → IoDDoSes
  - Safety critical?
    YES → hack = ouch
- Networking potential vs. challenges
  - Autonomy & error handling (no human ITL)

Focus: IIoT

- Internet of **Important** Things
  - Safety critical cyber-physical systems
  - Sense *and* interact with the world

    ...and talk to the Internet
- Driving Question:

> Can CPSes and IIoT achieve a **balance** where the *benefits* of the network out-weight the *risks*? Can the risks be *understood*?

Focus: IIoT

- Internet of **Important** Things

Brainstorm: Come up with at least three examples of systems that would be considered the IoIT.
Do you think this is a large segment of IoT?
Important segment?

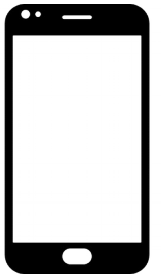
- Driving Question:

Can CPSes and IIoT achieve a **balance** where the *benefits* of the network out-weight the *risks*?
Can the risks be *understood*?

Edge Computing

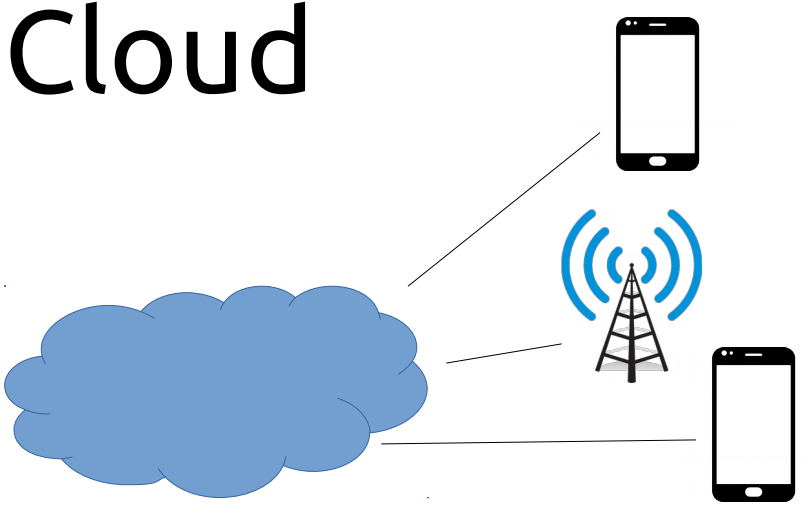
"a computing device that can act as an internet gateway or a router"

- Mobile vs. immobile edge computer
- Physical proximity to devices it serves
  - *Leverage locality* – low latency, variables = {wifi}
  - *Leverage locality* – keep data local (priv. & sec)
  - *Leverage locality* – offload computation
  - *Leverage locality* – offload storage/memory
  - *Leverage locality* – discovery based on proximity



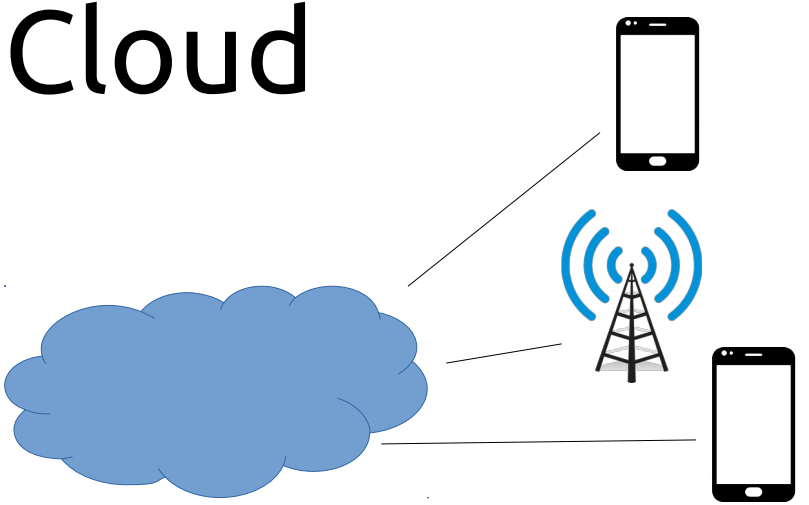
Versus Cloud

- Cloud excels at
  - Aggregation
  - Batch processing
  - Massive resources



Versus Cloud

- Cloud excels at
  - Aggregation
  - Batch processing
  - Massive resources



How can we leverage the **physical locality** of the edge *and* the **scale** of the cloud?

[Sec] Security: traditional view

- **C**onfidentiality
  - Only appropriate principles can access information
  - *Mitigation*: Controlling the *flow of information*
- **I**ntegrity
  - Data and computation cannot be interfered with
  - *Mitigation*: Isolation and “many walls”
- **A**vailability
  - Requests can be processed within a reasonable span
  - *Mitigation*: Distribution, scale, and rate-limiting

[Sec] Attacks

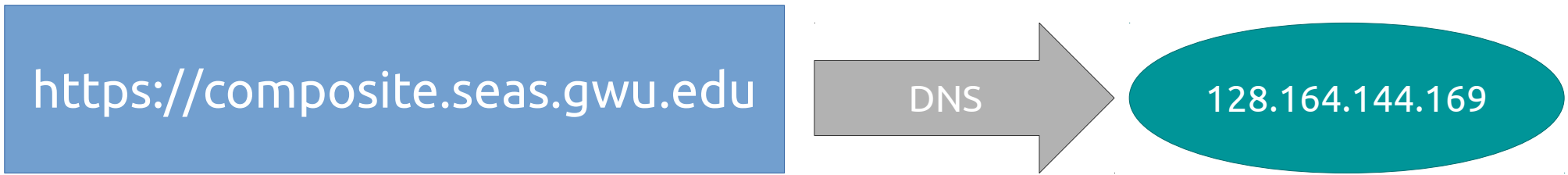
- Disruption of timing (DoT) attacks
- Limited battery power
- Physical disruption
  - Hammer, sensor, *microscope*
- Actuator manipulation
- Wireless – jamming, snooping

[Sec] Attacks

- Disruption of timing (DoT) attacks
- Limited battery power
-
-
- Wireless – jamming, snooping

How do these fit into classical CIA security models?

[Sec] IIoT: Naming & Identification



✓ Business

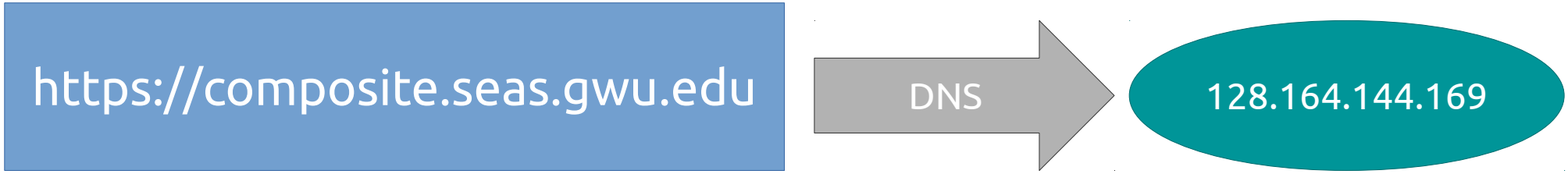
✓ Server

✗ IoT device

✗ IoT dev

- Billions of devs, humans → device & dev → dev

[Sec] IIoT: Naming & Identification



✓ Business ✓ Server

Solutions? Options?

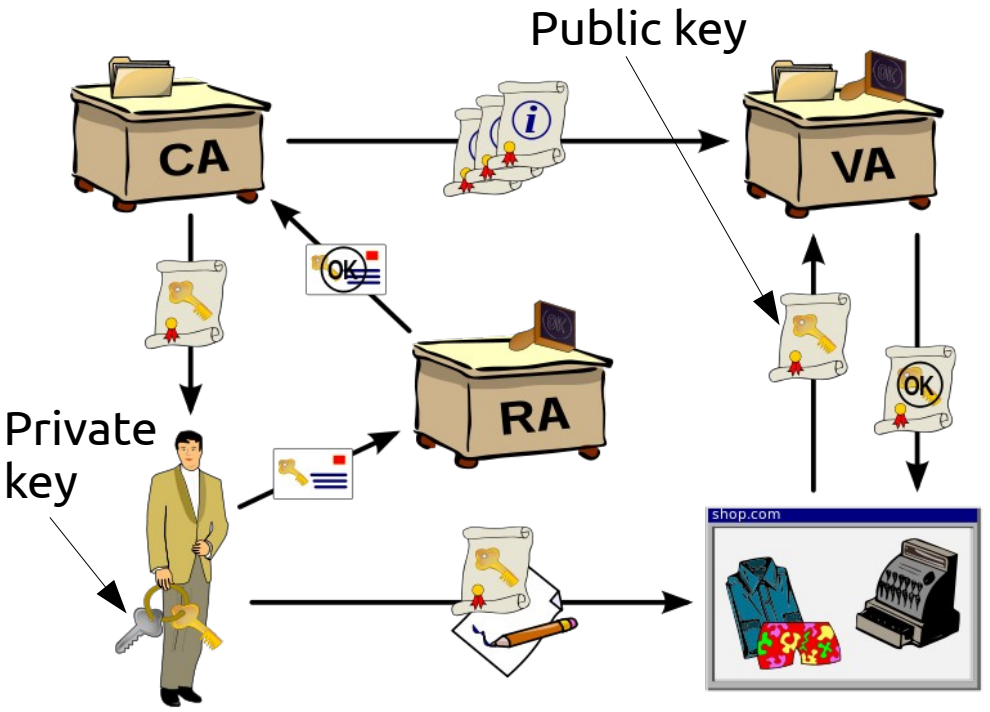
✗ IoT device ✗ IoT dev

- Billions of devs, humans → device & dev → dev

[Sec] Authentication

Solving:
Am I talking to X?

- X = business/org
- SSL/TLS (used in http**s**://...)



CA/RA/VA =
Certificate/Registration/Validation Authority

Image: Thanks Wikipedia!

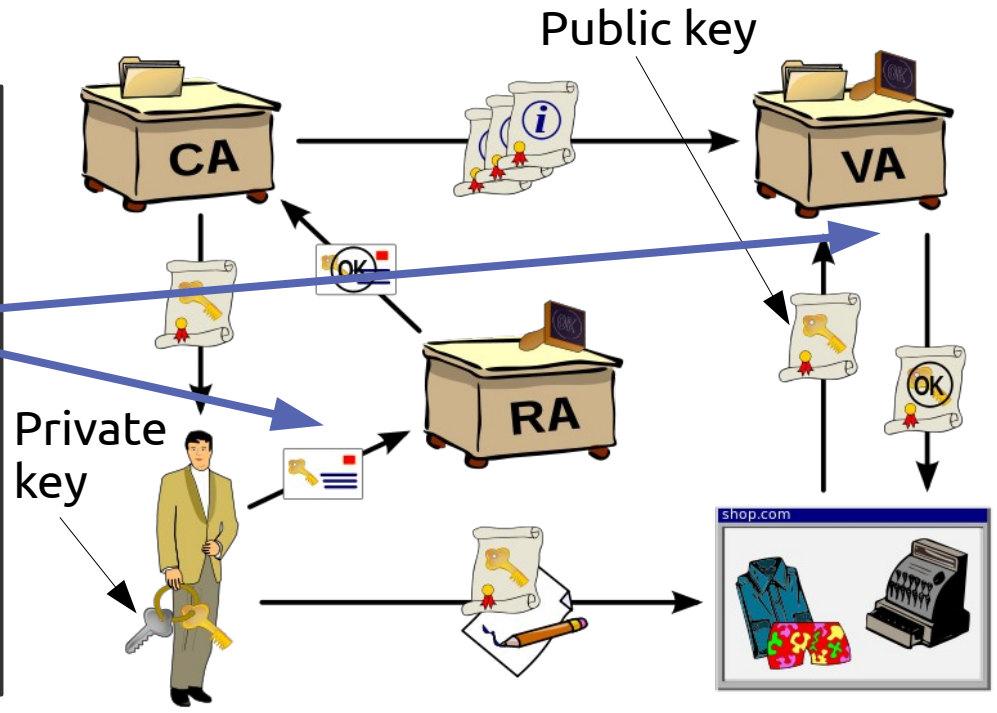
[Sec] Authentication

Solving:

IoIT: **Billions** of devices.

Will this scale to billions of devices?

Brainstorm:
Any solutions?



CA/RA/VA =
Certificate/Registration/Validation Authority

Image: Thanks Wikipedia!

[Sec] Availability

- DDoS attacks on necessary internet connections? (Hospitals, heating controllers)
- Local VAs distribute load to physical locations
  - How do you know the VA you're talking to is actually the VA? *Needs authentication* (circular)

Timing and Coordination

- Wide area network is best-effort
  (due to router contention, routing, datacenter proc)
- Accurately computing the *current time* is hard
  - Uses network and NTP – *local times* skew
- Important for *coordination*
- Synchronize local clocks using services

  accuracy = *f* (network jitter)

Timing and Coordination

- Wide area network is best-effort
  (due to router contention, routing, datacenter
  pr
- A
  –

Brainstorm: What would you need accurate, coordinated time for in the IoT?

- Important for *coordination*
- Synchronize local clocks using services

  accuracy = *f* (network jitter)

[Time] Robust Time Coordination

- GNSS (GPS sat broadcasts) – easily jammed
- eLoran – WAN wireless standard, stronger signals, lower frequency → jamming challenge
- Redundancy: NTP + PTP + …
- But network is always a challenge
  - Attack to increase jitter of time sync messages?

[Time] Network Management

Predictability – controlled latency

- TDMA – Time Division, Multiple Access
  - Each frequency divided into windows
  - Devices allocated *periodic windows*
- Examples: time-triggered ethernet, GSM
- *What are the downsides?*

[Time] Network Management

Predictability – controlled latency

Brainstorm:
1) If you assumed no malicious intents, do you think you can have a predictable wireless, edge system?
2) If you assumed a malicious environment?
3) What is a reasonable assumption here?

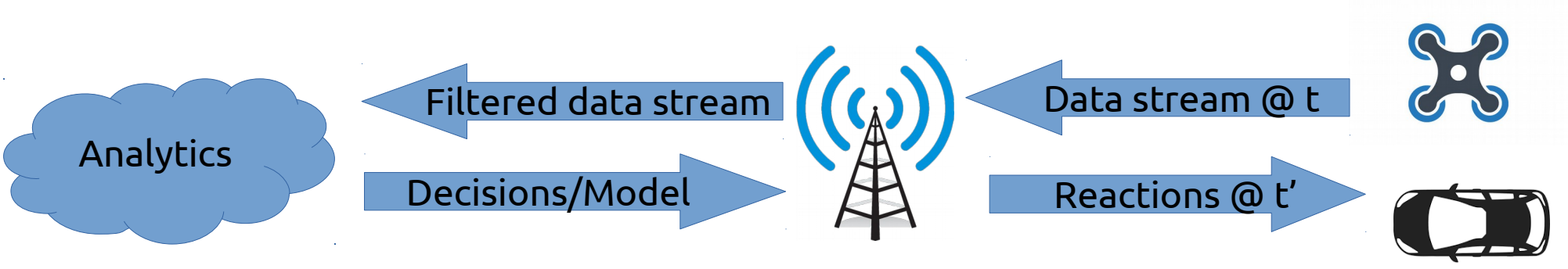
- *What are the downsides?*

Programming the IIoT

- Goal: Data → Knowledge → Decisions

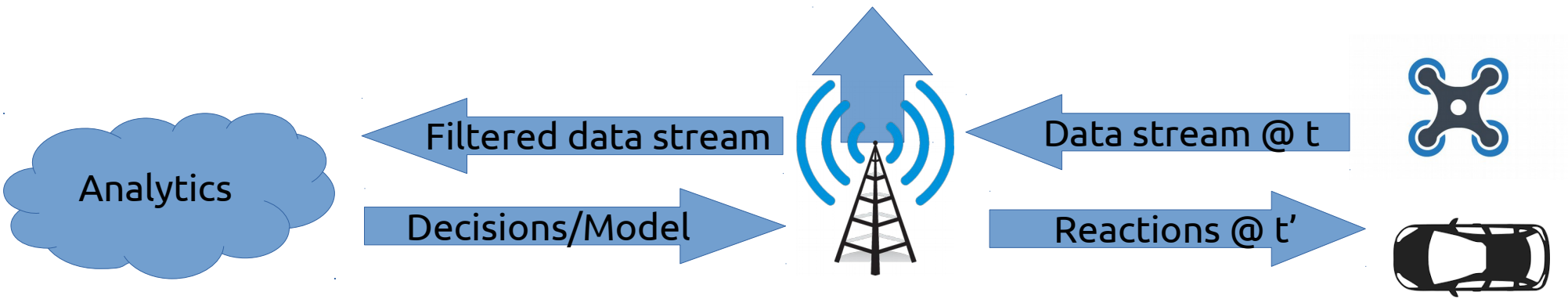
sense → streaming(time)

Analytics



Programming the IIoT

- Transient devices
- Admission control
- Temporal isolation between devices
- Scale to many devices/many tenants
- Real-time computation



Event vs. Thread-based Programming

- Event (callback-based) programming
  do_IO(io_done(io) { /* process I/O */ });
- Thread-based programming
  io = do_IO(); // block, switch to another thread
  /* process I/O */
- *Brainstorm:*
  - *Trade-offs between both of these?*
  - *Applicability for IoIT?*

[Prog] Evts vs. Threads

- Events
  - Serial, non-preemptive execution
  - Computation state in events, not stacks
  - “stack ripping” – logic not linear
- Threads
  - Preemptive = low latency for prioritized comp.
  - Preemptive = race conditions ;-(
  - Stacks might waste more memory

[Prog] Time and Simultaneity

- Can simultaneous operations (separate devices, threads, or “observers”) see actions in the same order?
- Order events uniformly, process them in order
  - E.g. for a periodic task model?

  - *What will it take to achieve deterministic concurrency on a single devices?*
  - *Across distributed devices?*

Conclusions

- Areas that need love:
  - Authentication/authorization services
  - Time synchronization and coordination
  - Programming models that increase determinism
- Edge computation to recover locality, but increase complexity