IoT: Cloud Services

Why Microservices?

So why Microservices?

- **Fast deployment:** Code changes for each service can be made independently (as long as the API contract with other services isn't violated) and therefore, build+test+deploy cycles speed up dramatically.
  - Netflix, for example, deploys code a hundred times in a single day thanks to the early adoption of the microservices architecture!
- **Efficient scaling:** Each microservice can be scaled independently,
  - a much more efficient way of scaling an application, because not every part of an application experiences the same amount of load and needs to be scaled equally.
- **Design autonomy:** Developers get the freedom to employ different technologies, frameworks, and design patterns to design and implement each microservice,
  - pursuing a *horses for the courses* strategy as necessary.

How can IoT exploit Microservices

- The monolithic IT architectures do not align with an environment where every device has a computer and wireless connection.

- Example:

- In my own house, I may have 7 light bulbs and 10 meters of light strips that all have their own processors, plus an Apple TV and an XBox.

- And how I want to interact with them is probably different than what you would want to do.

- This requires some level of decoupling: I want devices announcing themselves and reacting to the actions of other devices

- A fuzzy problem, a domain ideally suited for Microservices.

IoT/Microservices

- In the previous example:
    - I envision a MicroService that simply indicates whether I am home or away (probably via my iPhone and it geo-location services).
        - Another Microservice reacts to that and, based on the time of day, turns lights on or off (via Apple HomeKit and my Philips Hue controller).
- With Microservices, I can keep adding a bit more sophistication through additional services without waiting for one of the big vendors to build an application with that feature.

Challenges of IoT and Microservices

- **Interoperability**. How do I get devices from various vendors working with each other?

- **Security**. How do I protect access to systems in my house from malicious strangers (or hacker acquaintances, in my case)?

Dealing with the challenges

- **Interoperability:** there are a couple of standards emerging, and key vendors recognize that if their hub supports multiple interoperability standards, it is more likely to be used than a competitor's.

- **Security**: best addressed by using a locked, wireless network, and following best practices to secure it. Then you only need to protect the externally-facing facade from assault.

    - Apple TV, Microsoft XBox, and Amazon Echo all seem to be competing to be that facade.

Open APIs for IoT

- A successful vendor in this space will recognize the need for interoperability, and rather than solving it by successive features in proprietary products, will open up API's to allow rich communication to their hubs and devices over standard protocols (like RESTful interfaces with http protocols).
- A rich open source model is emerging
  - Much like some the toy robot and drone markets.
  - Market share is the reward for getting there first with open protocols.

Vendor Direction

- Recently, Amazon is pushing hard to be that integration vendor (see Amazon to flex internet of things).
- Philips just reversed a decision to exclude external devices from their Hub; the negative community reaction to exclusion was so vehement that Philips understood they would lose market share by such a move (see Philips Hue is getting back its third party smart bulb support). The industry seems to understand what they need to do.