Internet of Things workshop
ST - 2015/2016

IoT exercise

John Carpenter, 1982

Johan Lukkien

Goal

- Get acquainted with IoT protocols
- Exercise SOA / RESTful applications
- Exercise different architectures

- Prepare for an IoT-related lighting project at NXP

Description (1/2)

- Each node in set of nodes *P* represents a parking spot
- Parking spots have
  - identity
  - location: (*x*,*y*) coordinates
  - state: *Free, Empty, Reserved*
  - occupant: *id* of vehicle in case state != *Free*
- Parking spots expose a service that admits
  - to change their states
  - to inspect their states
    - directly or through eventing (after subscription)
- Parking spots advertise their services through mDNS and DNS-SD
- The access to the services is done through CoAP

Description (2/2)

- Each node in a set of nodes *V* represents a vehicle
- Vehicles have an *id*
- Vehicles repeatedly:
  - move through 2D space (takes time proportional to distance travelled)
  - decide that they want to park
    - discover parking spots
    - optionally: reserve a nearby one
    - move to and take a parking spot for some time
      - this releases reservations
      - a parking spot must be reserved or free to take it

Exercise in stages (1/2)

- Develop a fully distributed simulation of this system using CoAP, mDNS, DNS-SD
  - have at least two teams develop a vehicle (simulator) and two teams develop a parking spot (simulator)
  - define the service description and the interfaces carefully such as to admit interoperability
  - each parking spot is mapped onto a single physical node
  - one computer may simulate many vehicles, in total more than there are parking spots

Exercise in stages (2/2)

- Create a client application that visualizes the current state of the parking system
  - overview of (*x,y*), *id, state* and *occupancy*

- Create a broker application
  - used as contact point by the vehicles
  - maintaining the current state of the parking system
  - try to design the system in the earlier stage such that nothing has to change to the vehicles
  - implement a parking spot assignment policy

- Create a data store that records what happens over time
  - parking spots log data into this store, as do vehicles
  - applications work on top of the store
    - make one that visualizes statistics

Equipment

- We use Raspberry Pi's to represent parking spots
  - we use wired ethernet
- Vehicles clients and other components run on arbitrary computers
- We assume Linux as the OS

- CoAP packages to be used:
  - TxThing
  - Californium
    - use command line or browser interface (Copper)
  - any package that handles GET/PUT/OBSERVE correctly
- mDNS/DNS-SD is natively installed
  - avahi