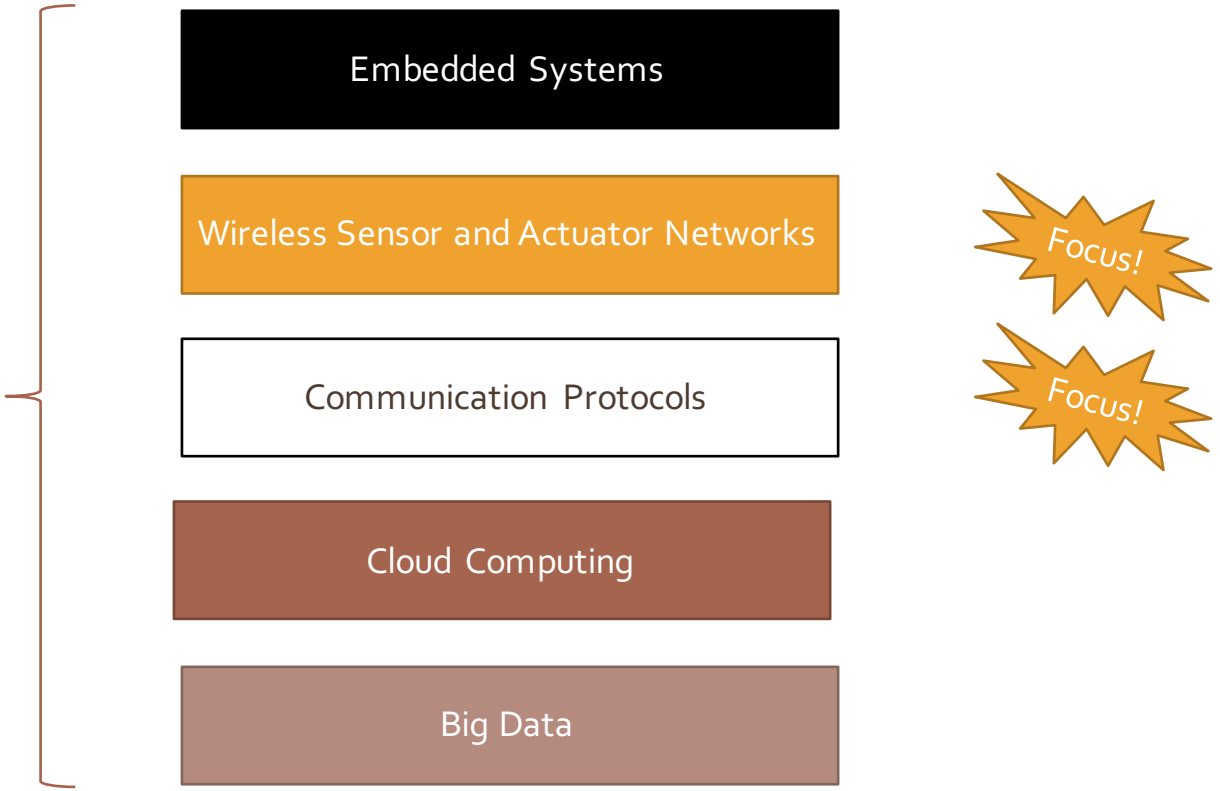
IoT Intro

Fernando Solano – Warsaw University of Technology

fs@tele.pw.edu.pl

Enabling technologies



What's IoT?

„A **dynamic** global network infrastructure with **self-configuring** capabilities based on **standard and interoperable communication protocols** where *physical and virtual* "**things**" have **identities**, physical attributes, and virtual personalities, use intelligent interfaces, and are **seamlessly integrated** into the information network."

Chapter 2 in Internet of Things - Global Technological and Societal Trends, River Publishers, 2011, ISBN 978-87-92329-67-7.

Characteristics

Things Characteristics

„A **dynamic** global network infrastructure with **self-configuring** capabilities based on **standard and interoperable communication protocols** where *physical and virtual* "**things**" have **identities**, physical attributes, and virtual personalities, use intelligent interfaces, and are **seamlessly integrated** into the information network."

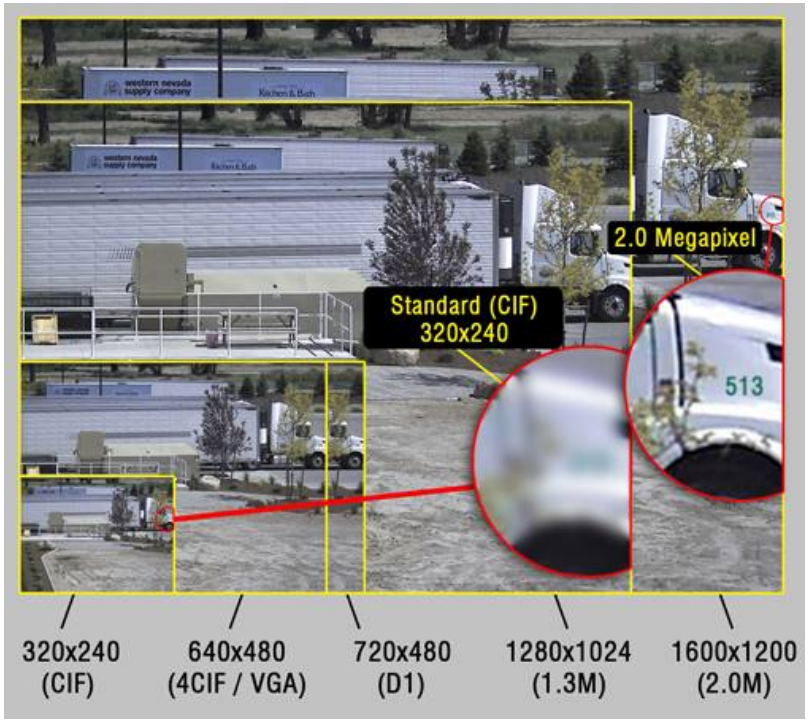
- Dynamic
- Self-configuring
- Interoperable protocols
- Identities
- Integrated into the net

Things Characteristics

- **Dynamic**: *Adapt to changing context and take actions based on operating conditions, user context and sensed environment.*

Example:

automatic adjustement of surveillance image quality based on motion detection



Things Characteristics

- **Self-configuring**: *to configure themselves (with respect to the IoT infrastructure), setup the network, upgrade, etc with minimal user intervention*

Example of such capability on mobile phones

Things Characteristics

- **Interoperable protocols**: *to communication with other IoT devices and with the IoT infrastructure*

Things Characteristics

- **Identities**: each IoT device has a unique identity and a uique identifier (e.g., IP address or URI), allowing:
  - Query the device
  - Monitor the state of the device
  - Control and configure them remotely

Things Characteristics

- **Integrated into the net**: in order to communicate and exchange data with other devices and systems
  - Dynamic discovery of other devices
  - Capability of describing itself and understanding others
  - Capability to interact with others

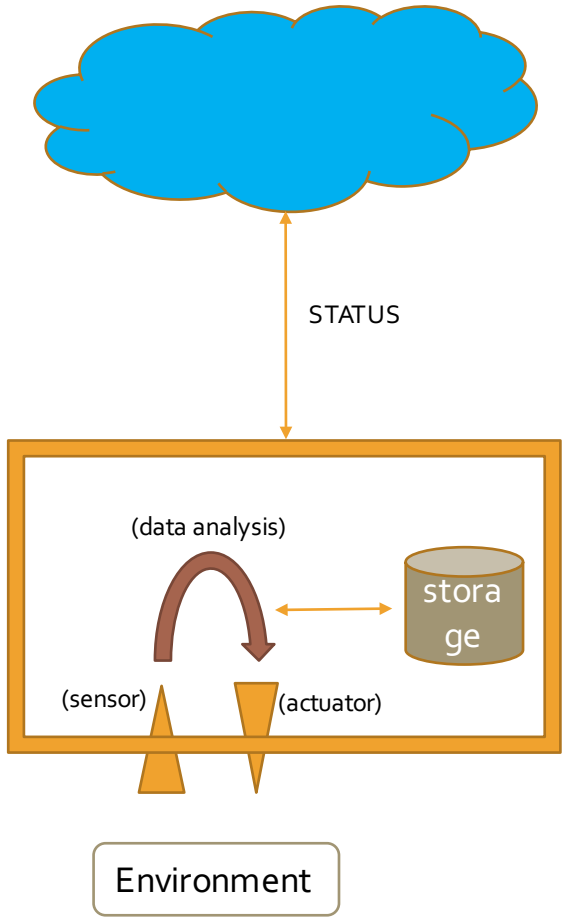
In concrete, a Thing usually has:

- Unique identities
- Sensors and actuators
- Capabilities for:
  - Exchange data and connect to other devices
  - Collect data
  - Process data or send data to be processed centralized
  - Perform tasks locally with some temporal and space constraints
    - Memory, processing capability, communication latencies, speeds and deadlines

IoT Levels and Deployment templates

Smart irrigation

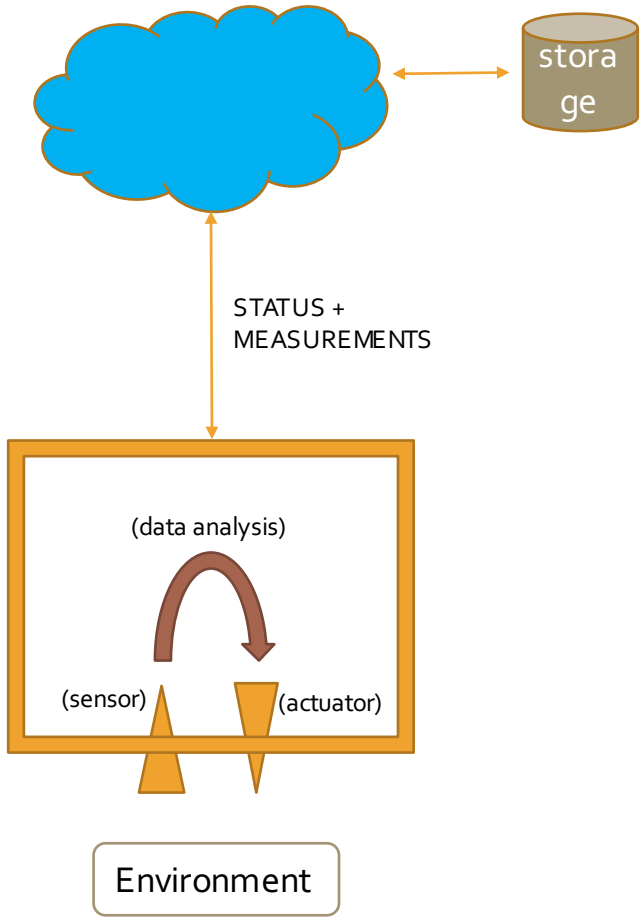
Level 1



- One node
- Simple data analysis: detects low humidity, open irrigation valves
  - Local analysis
- Reports state/status to the cloud
  - Local storage

Smart irrigation

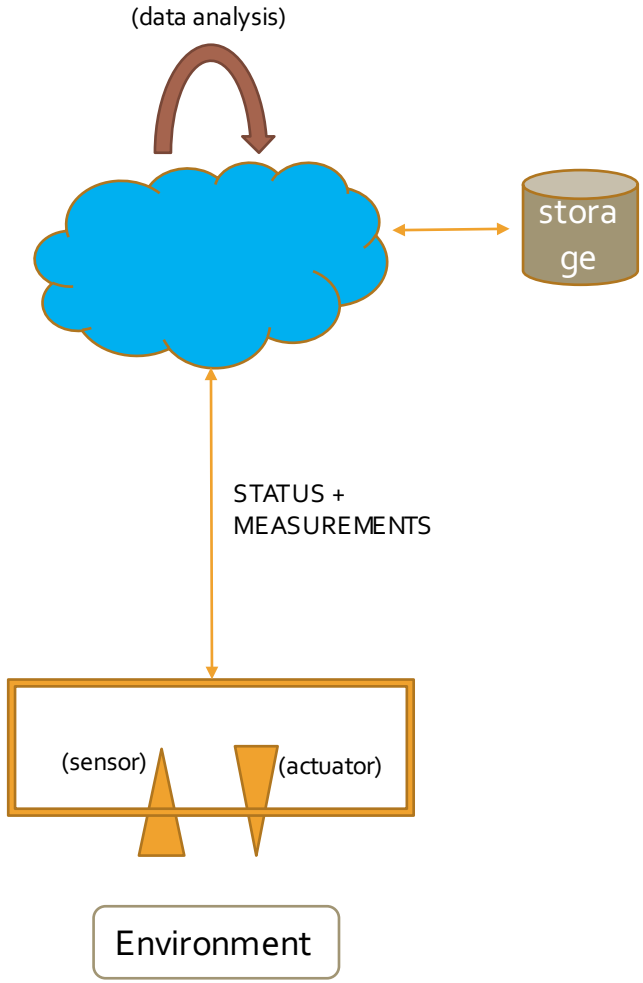
Level 2



- One node
- Simple data analysis: detects humidity, open irrigation valves
  - Local analysis
- Reports collected measurements and state/status to the cloud
  - Cloud storage

Smart irrigation

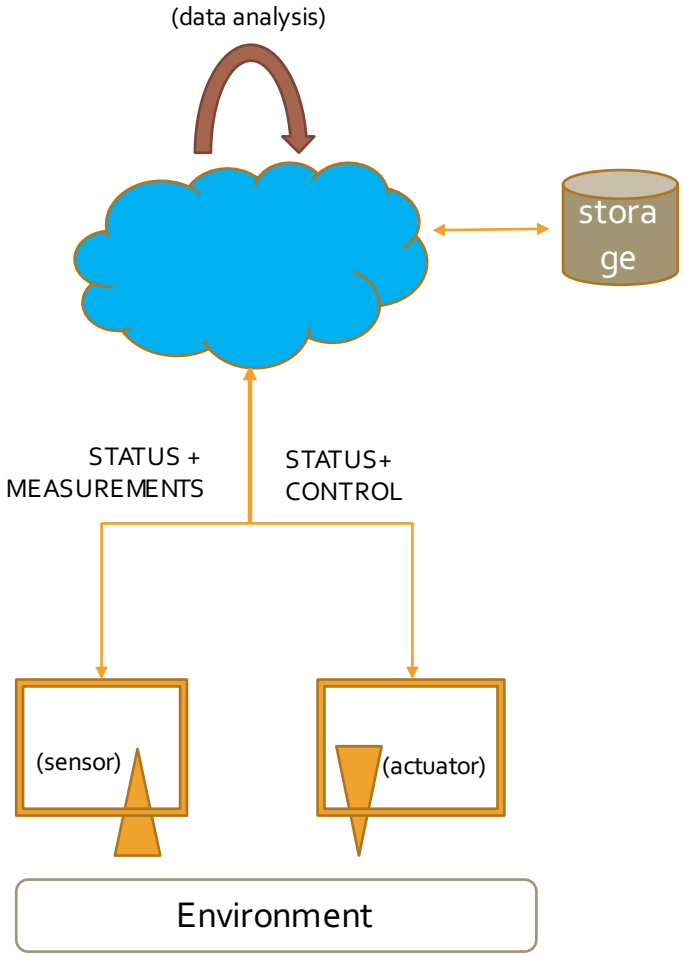
Level 3



- One node
- Complex data analysis: different sensors, different nutrients tanks
  - Cloud analysis
- Reports collected measurements and state/status to the cloud
  - Cloud storage

Smart irrigation

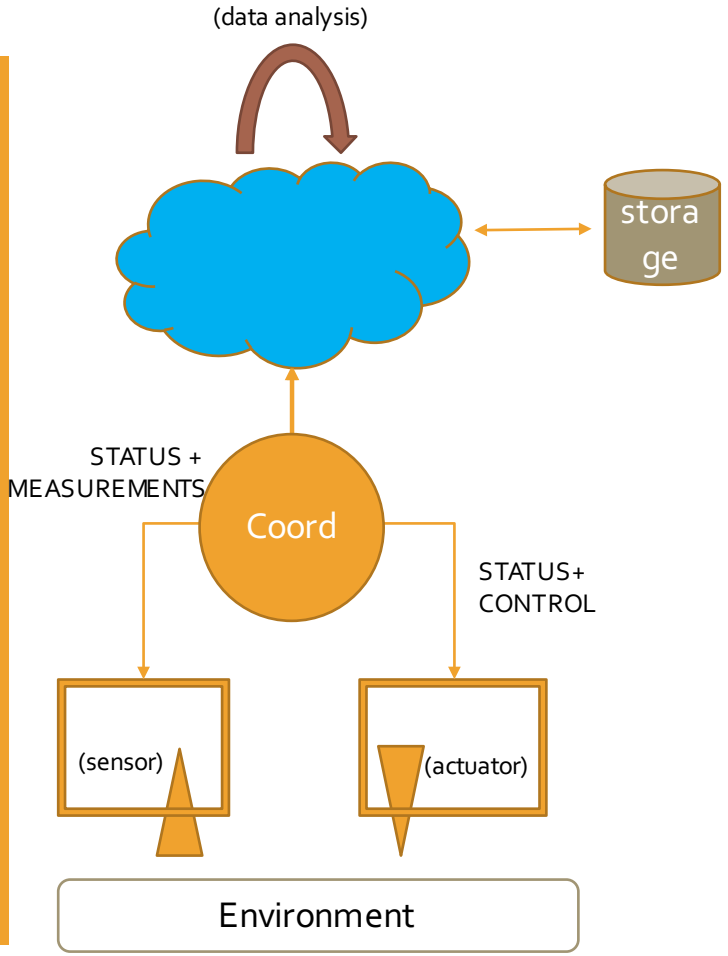
Level 4



- Several nodes:
  - Actuators and sensors separetely (Observers)
- Complex data analysis: different sensors, different nutrients tanks
  - Cloud analysis
- Reports collected measurements and state/status to the cloud
  - Cloud storage

Smart irrigation

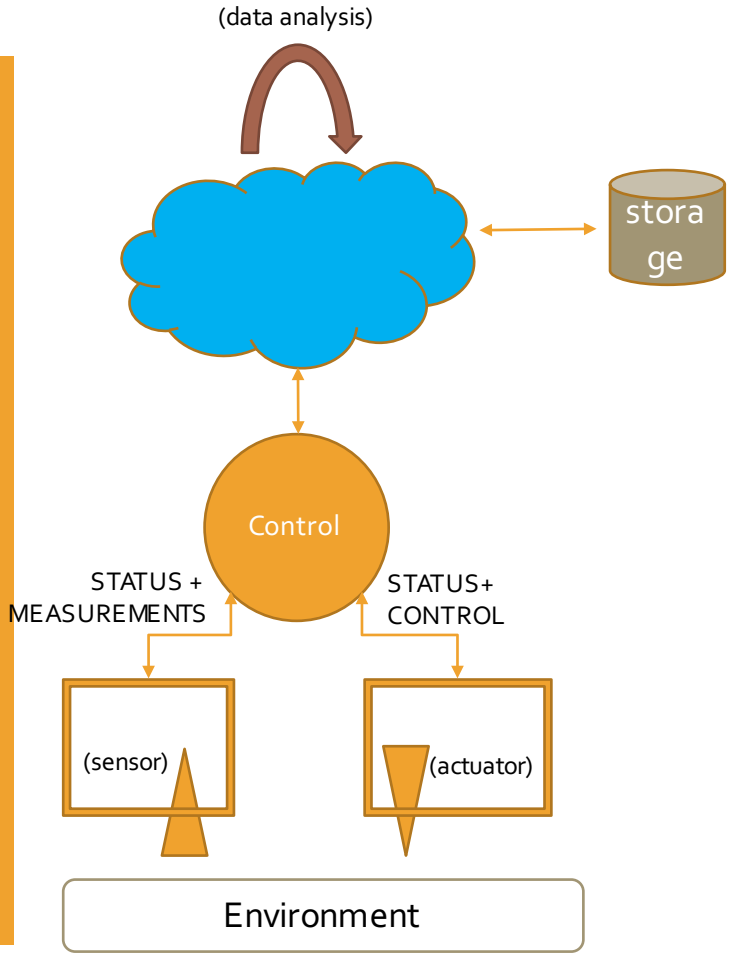
Level 5



- Several nodes:
  - Actuators and sensors separetely (Observers)
  - 1 coordinator (Gateway) – WSN-based
- Complex data analysis: different sensors, different nutrients tanks
  - Cloud analysis
- Reports collected measurements and state/status to the cloud
  - Cloud storage

Smart irrigation

Level 6



- Several nodes:
  - Actuators and sensors separetely (Observers)
  - 1 controller (Gateway) – WSN-based
- Complex data analysis: different sensors, different nutrients tanks
  - Cloud analysis
- Reports collected measurements and state/status to the cloud
  - Cloud storage

IoT levels

Summary

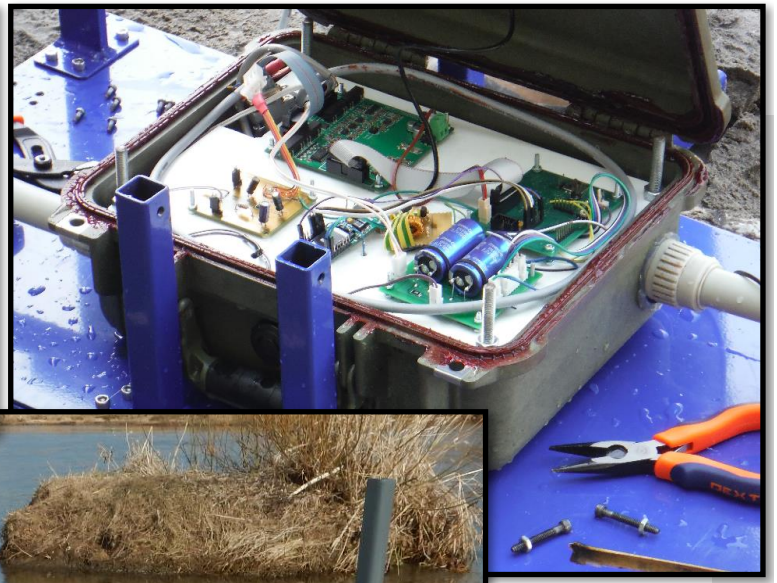
Feature	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
Cloud Storage	-	YES	YES	YES	YES	YES
Cloud Analysis	-	-	YES	YES	YES	YES
Ext Observers	-	-	-	YES	YES	YES
Coordinator	-	-	-	-	YES	YES
Controller	-	-	-	-	-	YES

Examples

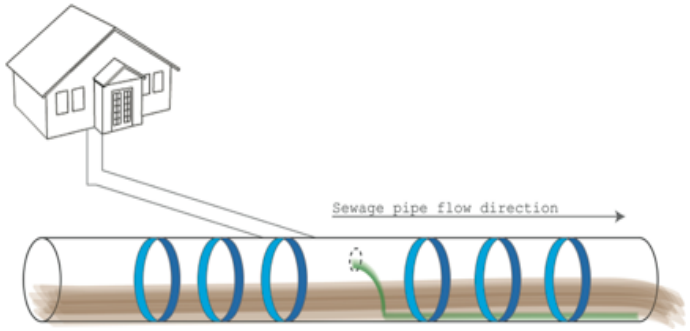
Two from PW + some others

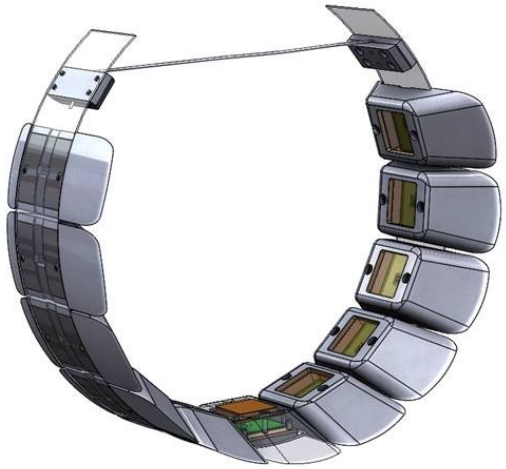
Goldfish


Retractable floating antenna
Biochemical sensors
Gateway
Underwater Controller and Processing Unit



microMole

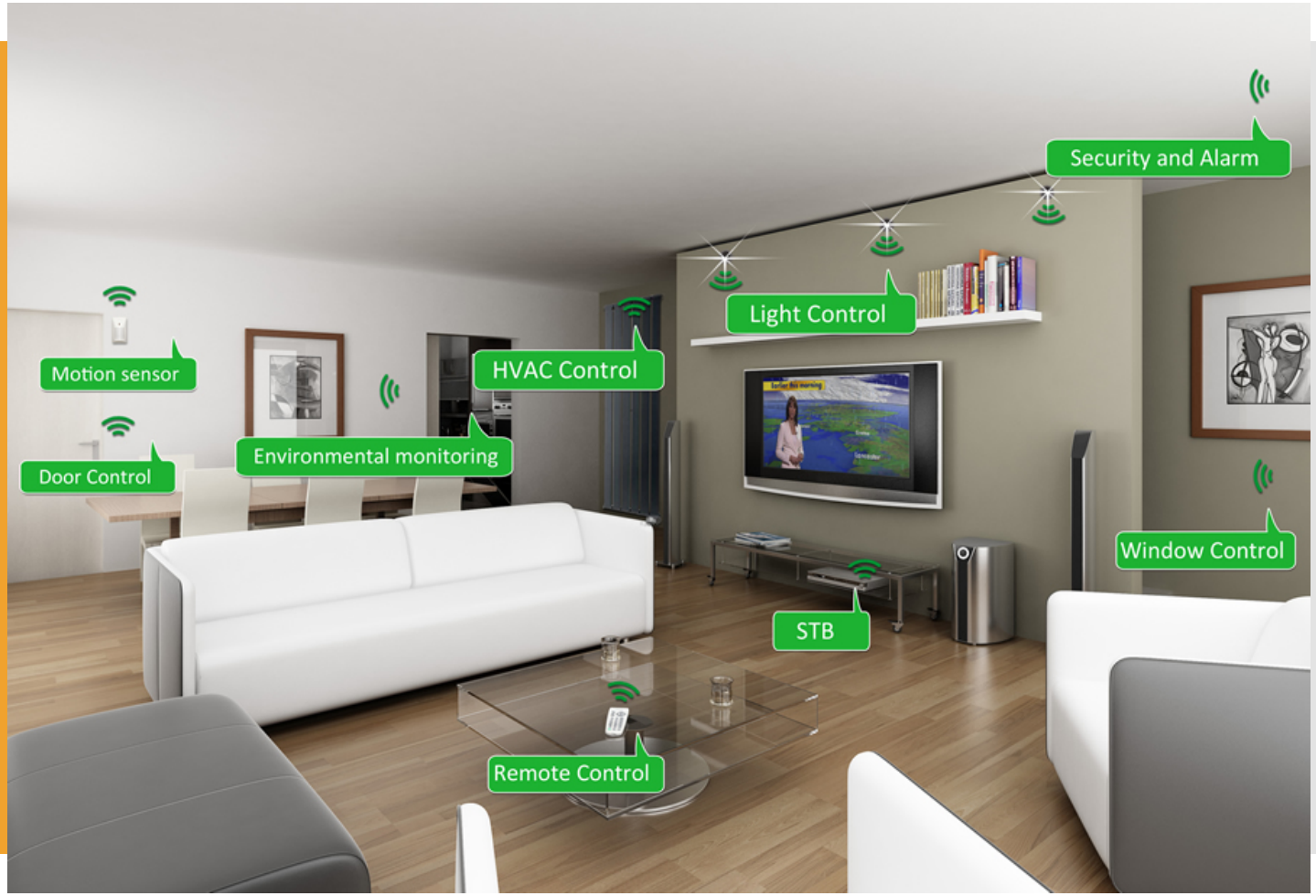




IoT levels?

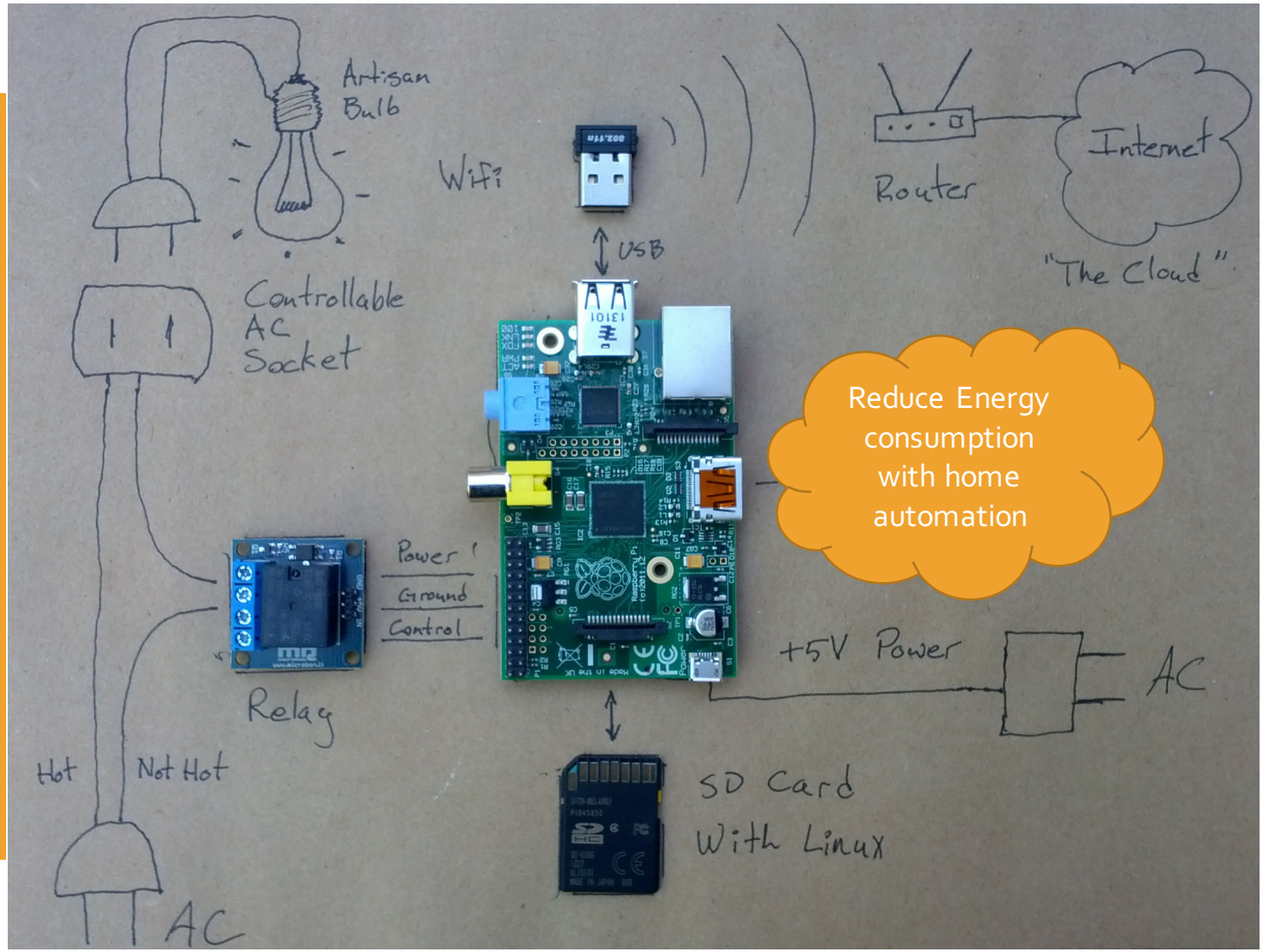
Feature	GF1	GF2
Cloud Storage	YES	YES
Cloud Analysis	-	-
Ext Observers	YES	YES
Coordinator	YES	-
Controller	-	-

Wireless Sensor Networks and Hardware



A bad example

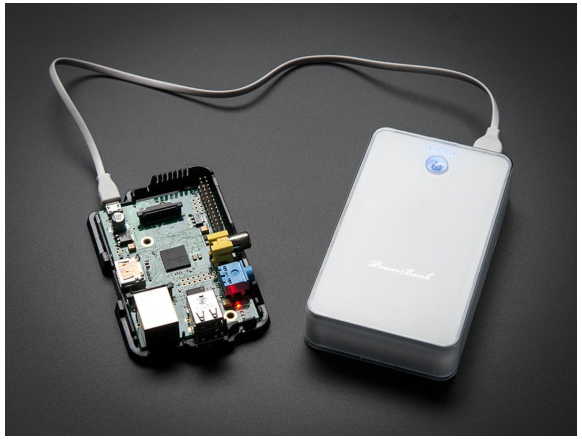
Remote bulb control



A bad example

Remote bulb control

Power Mgte



How often will I need to replace/recharge the battery?

Where can I connect it?
Do I have a plug nearby?

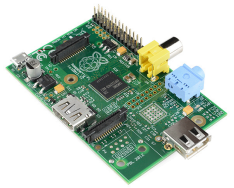
Even if so, what's the point?
Am I saving Energy with this system?

A bad example

Remote bulb control

- Wireless interface:
  - How many bytes do we need to transfer for switching on a bulb?
  - What is the WiFi speed?
  - What would be the overhead for transfering a small packet?
- Processor
  - How complex could be our controlling application at the Rpi? (memory/CPU cycles)
  - Can a delay of milliseconds be tolerable? Or is it microseconds justified?
- OS:
  - Can Linux be put into sleep?
  - If not, what's the boot time?
  - Can Linux put to sleep the WiFi independently?
  - Can we reduce the clock speed arbitrarely?

Hardware

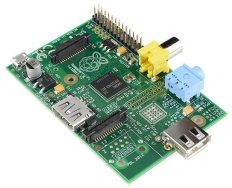


	PC	Smartphone	Raspberry Pi
CPU	2.1-3.2 GHz	1.2-1.8GHz	900 MHz
Memory	16-32GB	8-16GB	512 MB
Wireless data rate		~ 1Mbps (BLE) > 11 Mbps (WiFi)	
Power consumption	300 watts	5-12 watts	1 watt (BLE) 2 watts (WiFi)
Battery	-	2000mAh	
Expected battery lifetime	-	1-2 days	4-6 days (if 2000mAh + WiFi)

Assume we deploy a 100 RPI… we'll have to replace 20 batteries a day!

Hardware

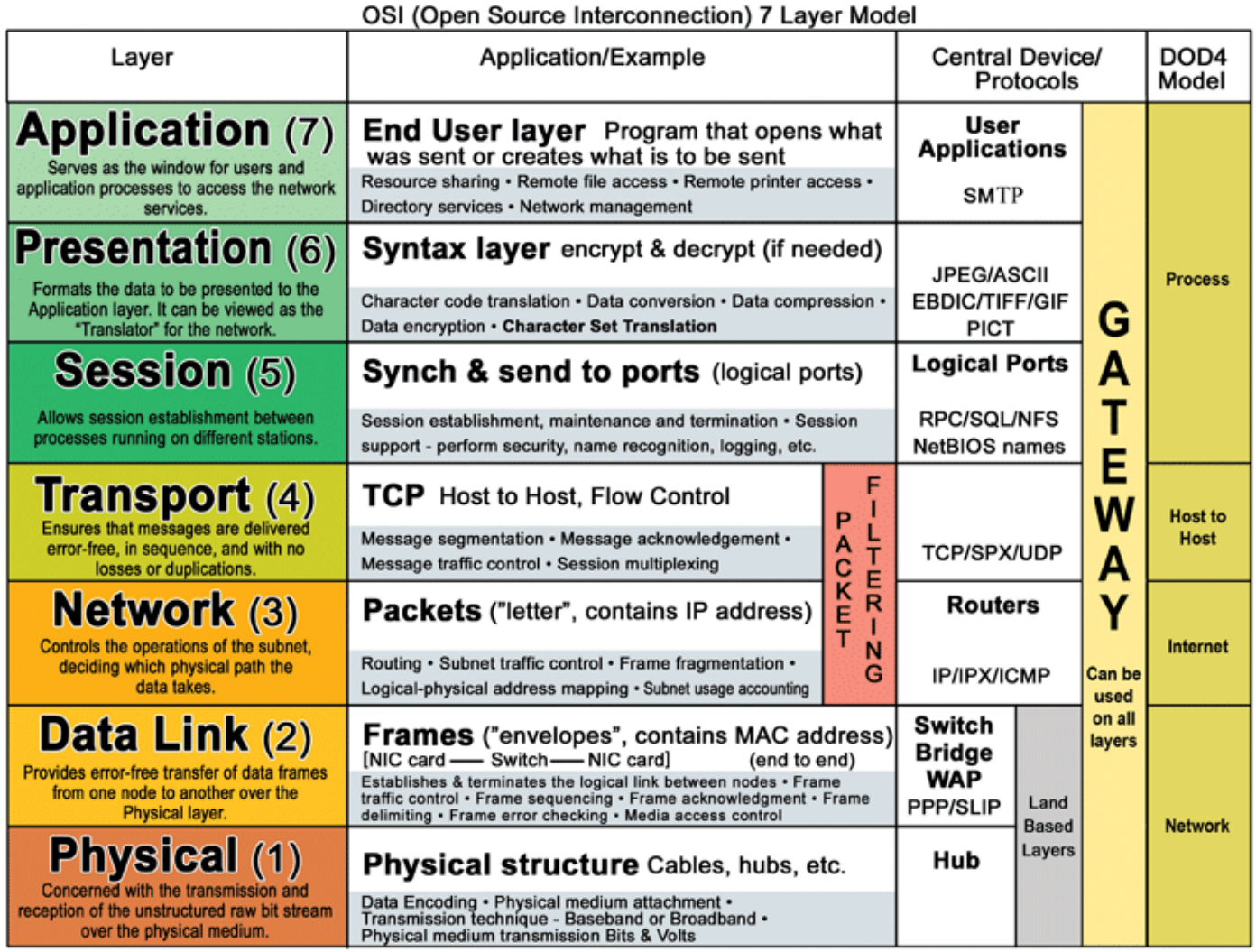





	PC	Smartphone	Raspberry Pi	Wireless sensor
CPU	2.1-3.2 GHz	1.2-1.8GHz	900 MHz	8-48MHz
Memory	16-32GB	8-16GB	512 MB	256 KB
Wireless data rate	> 11 Mbps (802.11)			20-250 Kbps
Power consumption	300 watts	5-12 watts	~ 2 watts	10 mW (on) 10uW (sleep)
Battery	-	2000mAh		750mAh
Expected battery lifetime	-	1-2 days	4-6 days (if 2000mAh)	4 years @ duty cycle 0.1%

Layers and Protocols

OSI model


OSI (Open Source Interconnection) 7 Layer Model
Layer
Application/Example
Central Device/ Protocols
DOD4 Model
Application (7)
Serves as the window for users and application processes to access the network services.
End User layer Program that opens what was sent or creates what is to be sent
Resource sharing • Remote file access • Remote printer access • Directory services • Network management
User Applications
SMTP
Presentation (6)
Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.
Syntax layer encrypt & decrypt (if needed)
Character code translation • Data conversion • Data compression • Data encryption • Character Set Translation
JPEG/ASCII
EBDIC/TIFF/GIF
PICT
Process
Session (5)
Allows session establishment between processes running on different stations.
Synch & send to ports (logical ports)
Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.
Logical Ports
RPC/SQL/NFS
NetBIOS names
Transport (4)
Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.
TCP Host to Host, Flow Control
Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing
PACKET FILTERING
TCP/SPX/UDP
Host to Host
Network (3)
Controls the operations of the subnet, deciding which physical path the data takes.
Packets ("letter", contains IP address)
Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting
Routers
IP/IPX/ICMP
Internet
Data Link (2)
Provides error-free transfer of data frames from one node to another over the Physical layer.
Frames ("envelopes", contains MAC address)
[NIC card — Switch — NIC card] (end to end)
Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control
Switch
Bridge
WAP
PPP/SLIP
Land Based Layers
Network
Physical (1)
Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.
Physical structure Cables, hubs, etc.
Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts
Hub
GATEWAY
Can be used on all layers

IoT Link Layer Protocols

- How data is sent over the physical medium between two (physically) interconnected devices.
- Some examples:
  - 802.3 – Ethernet (10Mbps – 40Gbps, coaxial, twisted pair)
  - 802.11 – WiFi (1Mbps – 6,75 Gbps, wireless)
  - 2G/3G/4G – Mobile Communications
  - 802.16 – WiMax (1.5 Mbps – 1 Gbps, wireless)
  - 802.15.1 – Bluetooth (? , wireless)
  - 802.15.4 – LR-WPAN (40 Kbps – 250 Kbps, wireless)
- Which one of them is mostly used in constrained devices?

IoT Network Protocols

- Responsible for sending IP packets from the source node in a network of devices to the end node
- Provides (global hierarchical) addressing of nodes in the network and routing of packets across the network
- Some standards:
  - IPv4 – 32-bits addresses – until 2011
  - IPv6 – 128-bit addresses
  - 6loWPAN – IPv6 compressed
- Some protocols:
  - ICMPv6
  - RPL
- Which one of them is mostly used in constrained devices?

IoT Transport Layer

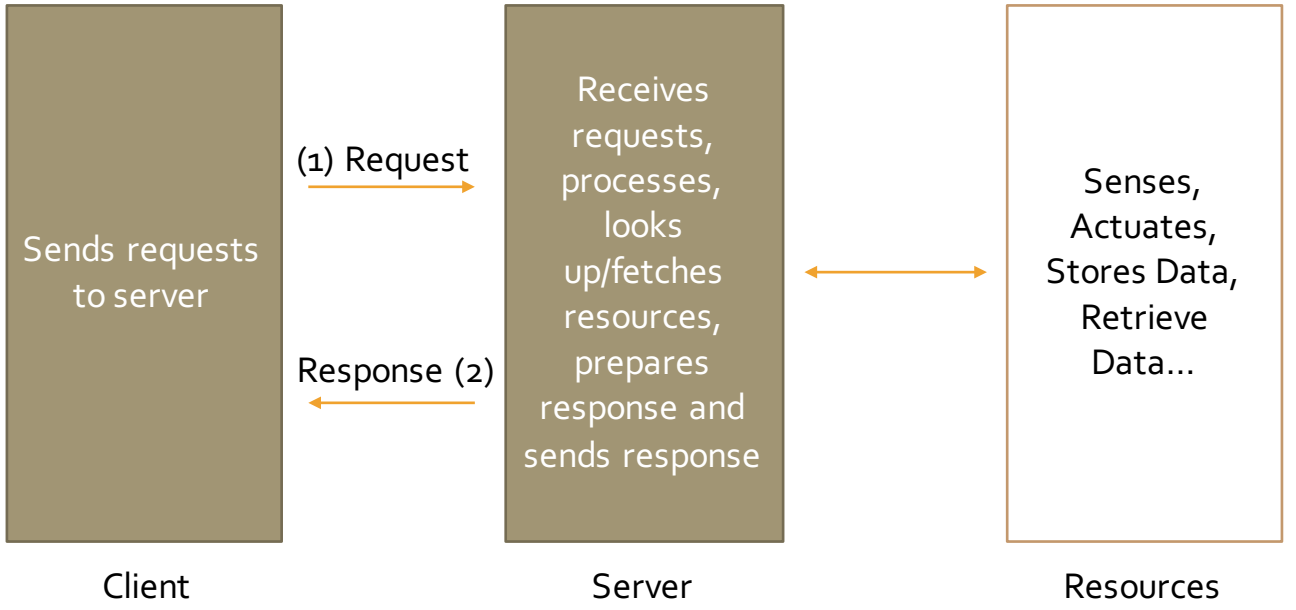
- End-to-end datagram transfer independent of underlying network and identification of en-points (ports)
- Some protocols
  - UDP: connectionless (no handsakes)
    - Small data-units to exchange
    - No overhead from connection setup
  - TCP: connection-oriented and statefull
    - Ensures reliable transmission in order
    - Flow control
- Which one of them is mostly used in constrained devices?

IoT Application Layer

- Enable process-to-process connections using ports
- Some examples:
  - HTTP - Hyper-Text Transfer Protocol
  - CoAP – Constrained Application Protocol
  - WebSocket
  - MQTT – Message Queue Telemetry Transport
  - XMPP – Extensible Messaging and Presence Protocol
  - DDS – Data Distribution Service
  - AMQP – Advanced Message Queueing Protocol
  - …

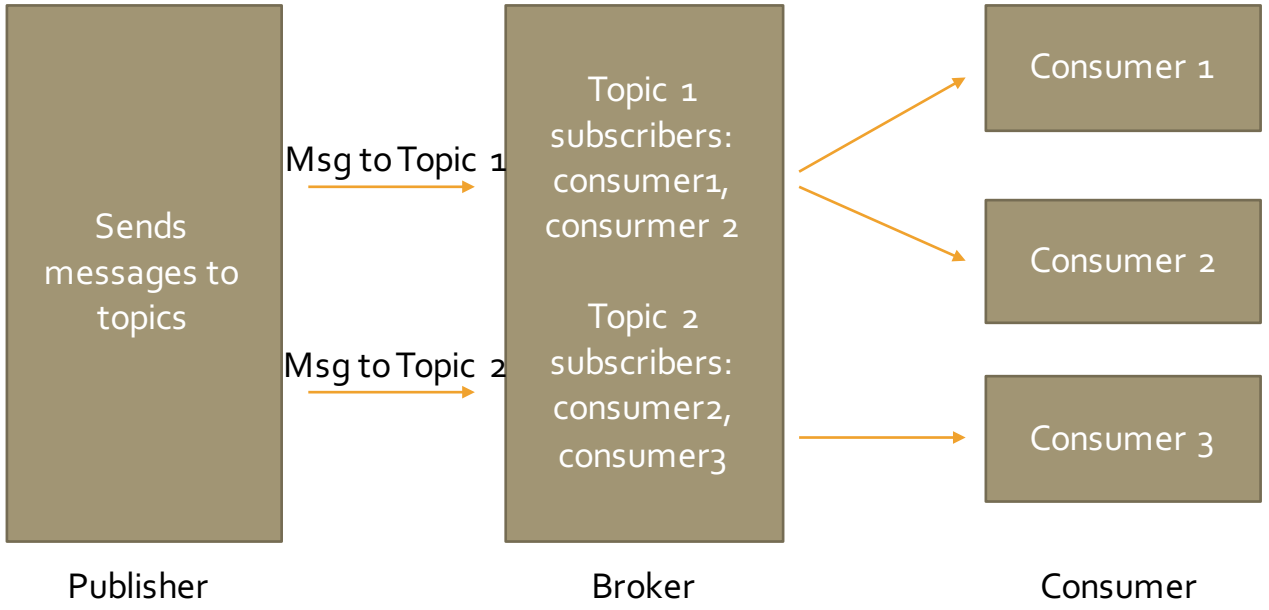
Communication Models

Request-Response

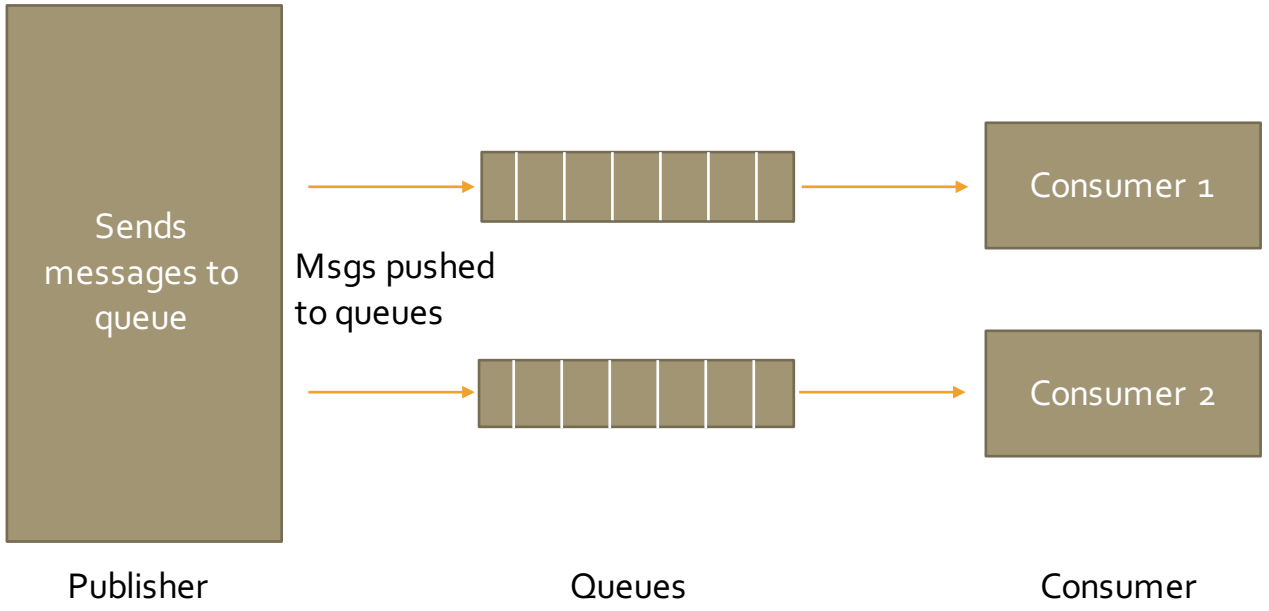


Request-Response is stateles: each request-response is independent from others

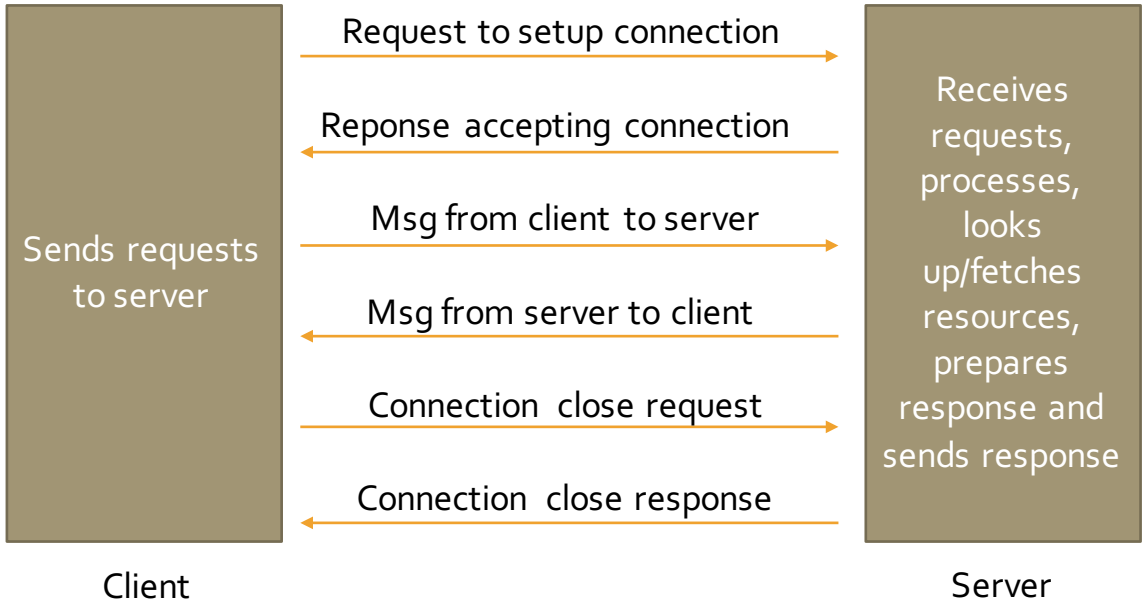
Publish-Subscribe



Push-Pull


Sends messages to queue
Msgs pushed to queues
Consumer 1
Consumer 2
Publisher
Queues
Consumer

Exclusive Pair



Exclusive-Pair is statefull, bi-directional, and fully duplex

REST-based Communication APIs

- Principles (constrains):
  - Client-Server
  - Stateless: request contains all information needed to understand it
  - Cache-able: Response can be marked as cache-able. If so, client can reuse response later on.
  - Layered system: a component cannot see beyond its scope
  - Uniform system: communication method between client and server is uniform
  - Code on demand: servers can provide executable code or scripts for clients (optional)

Student topics

- Home automation
  - Smart Lighting
  - Home Intrusion Detection
- Cities
  - Smart Parking
- Environment
  - Weather Monitoring System
  - Weather Reporting Bot
  - Air Pollution Monitoring
  - Forest Fire Detection
  - River Pollution Monitoring
- ~~Agriculture~~
  - ~~Smart Irrigation~~
  - ~~Crops Monitoring~~
- Health and Wellness
  - For the elderly
  - For the physically impaired
- Sports
- Performing Arts
- …

"All things appear and disappear because of the concurrence of causes and conditions.
Nothing ever exists entirely alone; everything is in relation to everything else."

Hindu Prince Gautama Siddharta