Internet of Things: An Introduction

- IoT Overview and Architecture
- IoT Communication Protocols
- Acknowledgements

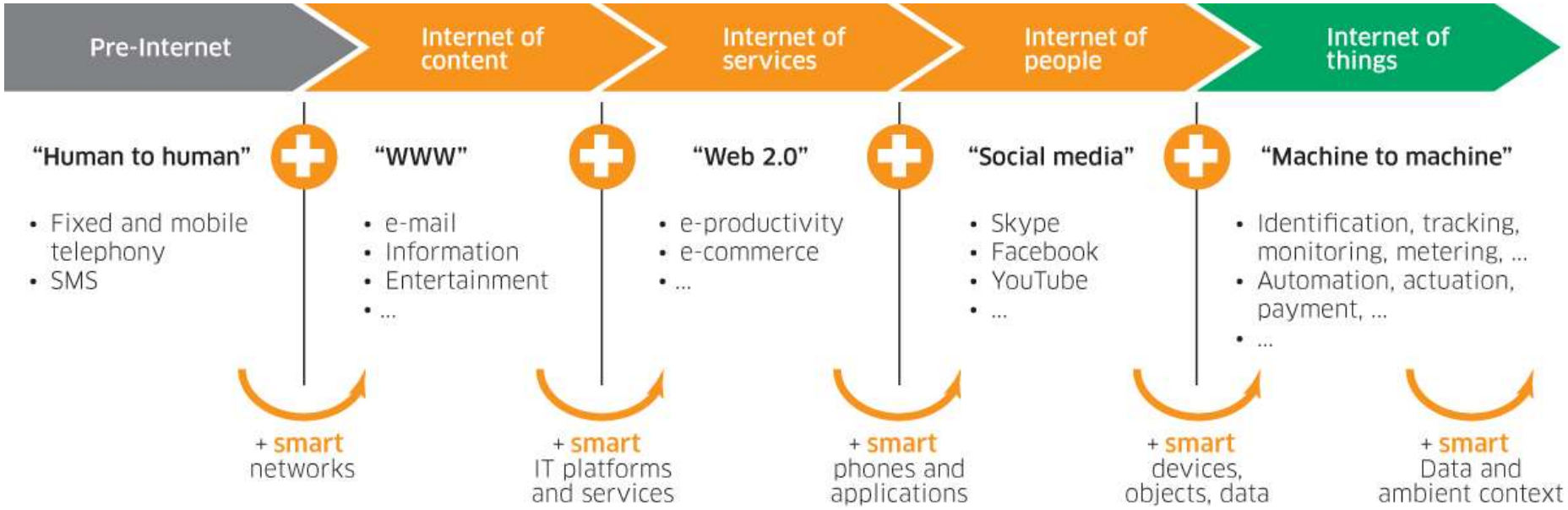
What is IoT?

- **Internet of Things (IoT) comprises things that have unique identities and are connected to the Internet**

- **The focus on IoT is in the configuration, control and networking via the Internet of devices or "Things" that are traditionally not associated with the internet**
  - **Eg: pump, utility meter, car engine**

- **IoT is a new revolution in the capabilities of the endpoints that are connected to the internet**

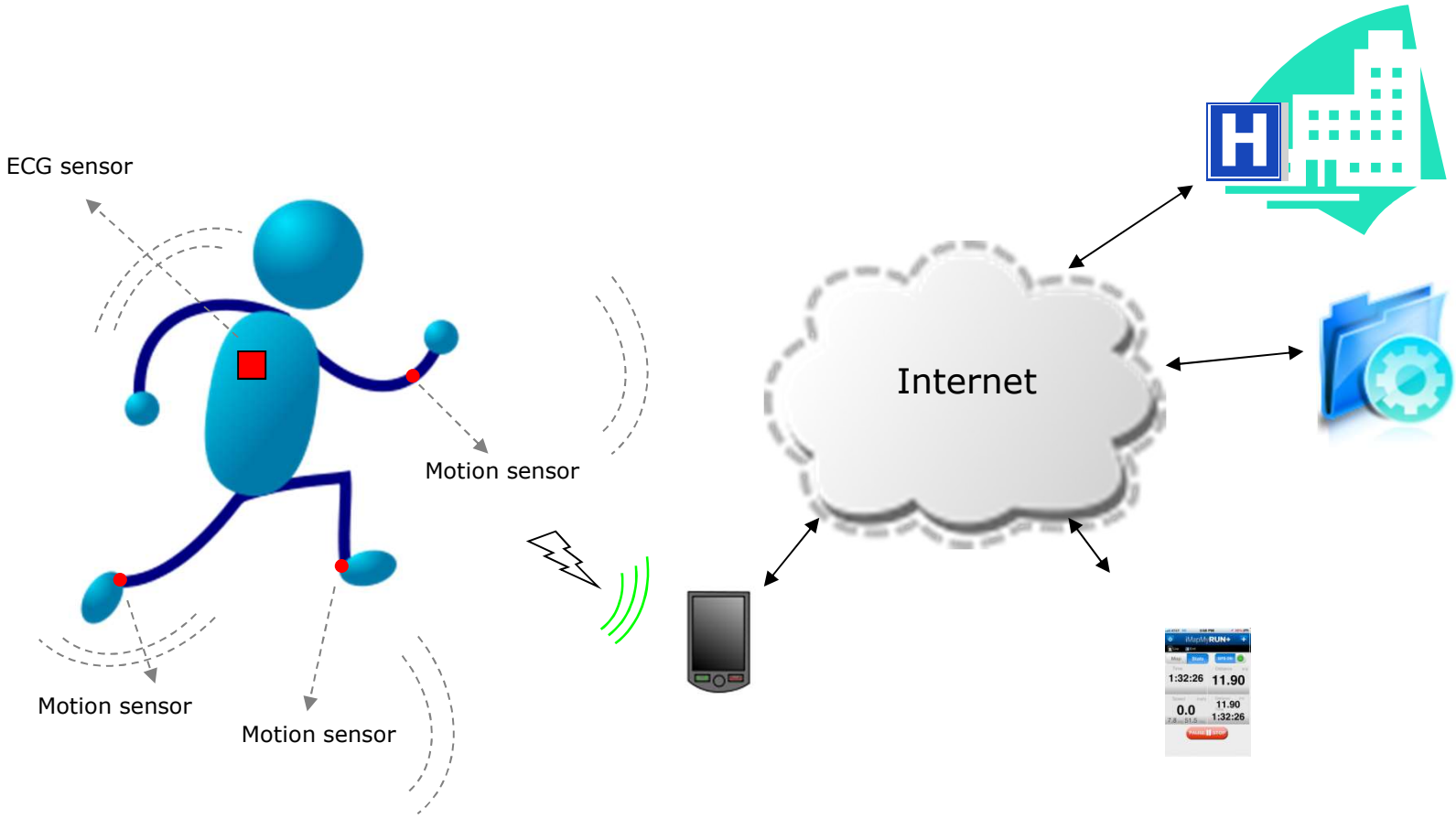
IoT Evolution



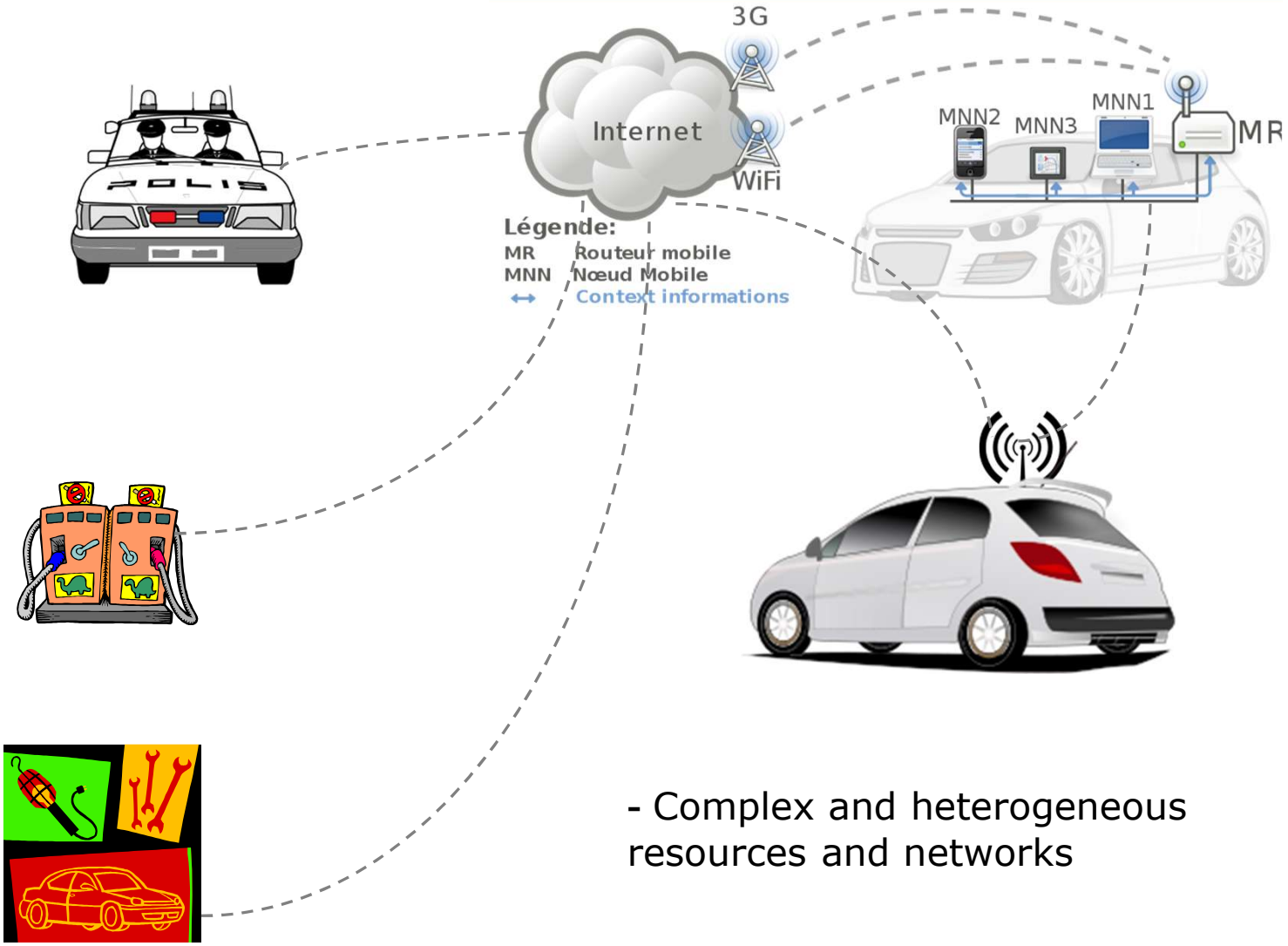
M2M vs IoT

M2M	IoT
Point-to-point communication usually embedded within hardware at the customer site	Devices communicate using IP Networks, incorporating with varying communication protocols
Many devices use cellular or wired networks	Data delivery is relayed through a middle layer hosted in the cloud
Devices do not necessarily rely on an Internet connection	In the majority of cases, devices require an active Internet connection
Limited integration options, as devices must have corresponding communication standards	Unlimited integration options, but requires a solution that can manage all of the communications

IoT: People Connecting with Things



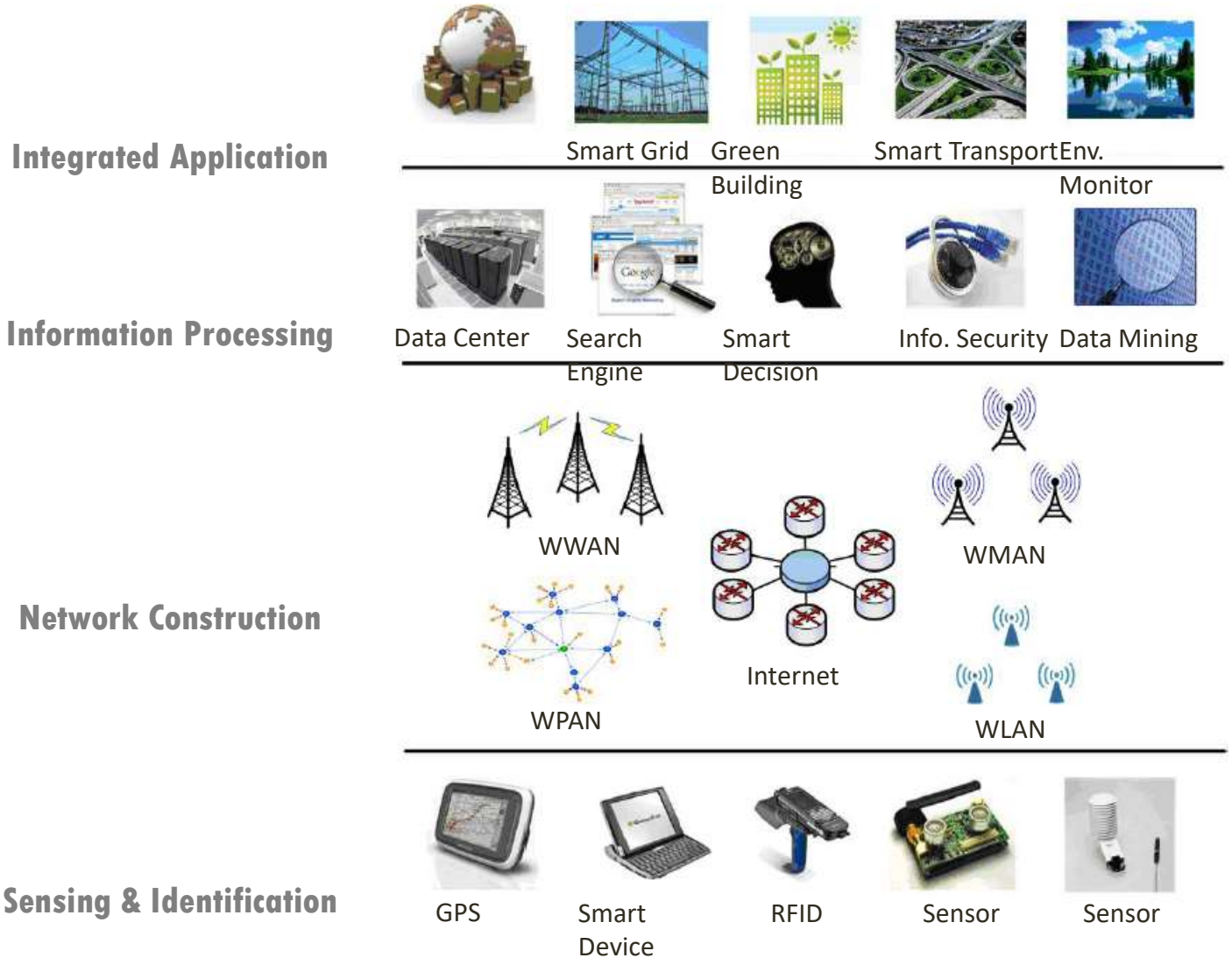
IoT: Things Connecting with Things



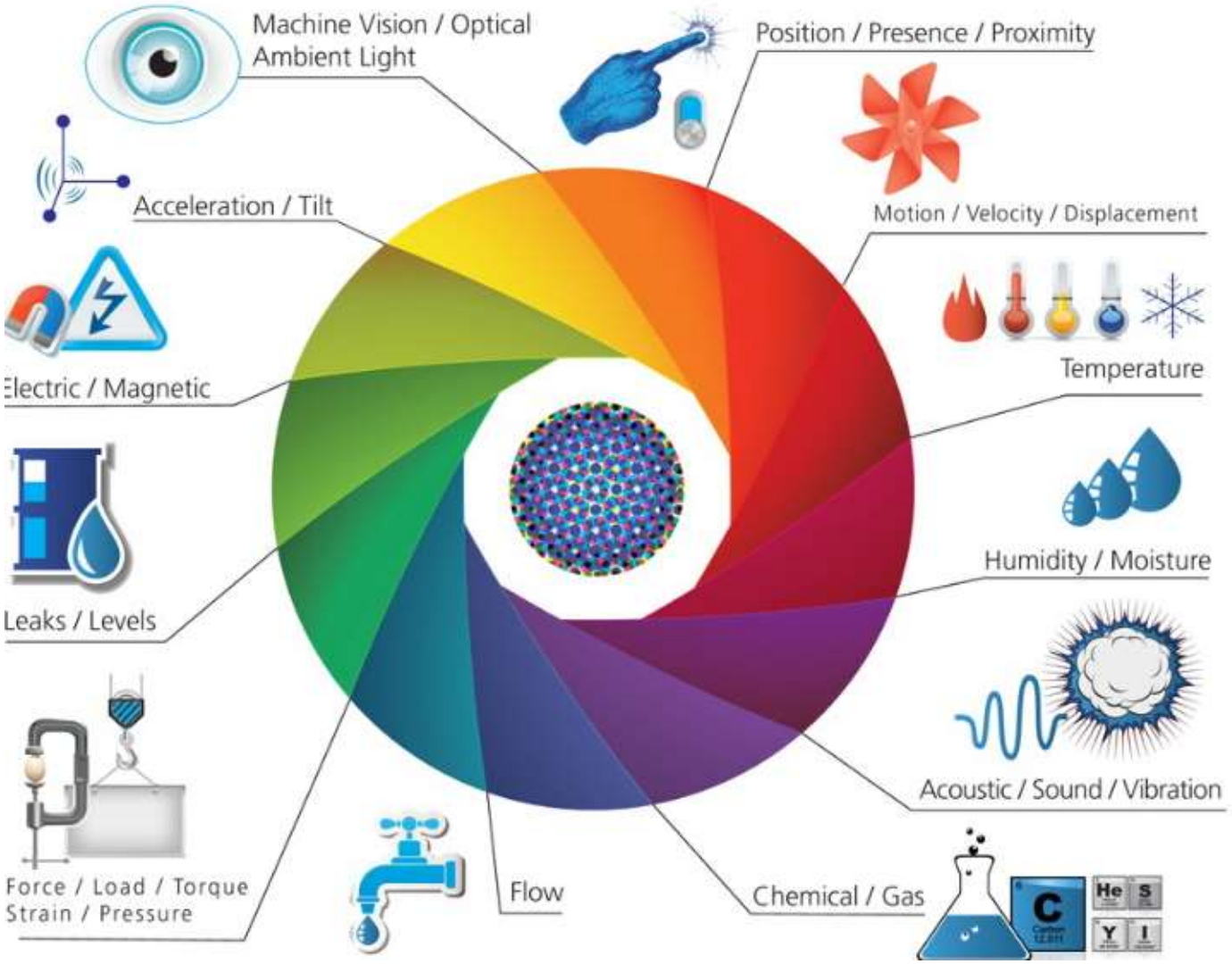
- Complex and heterogeneous resources and networks

IoT: Application Areas

IoT Architecture

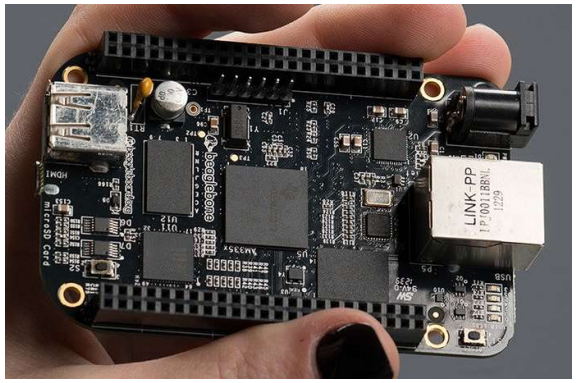


IoT: Sensors and Actuators

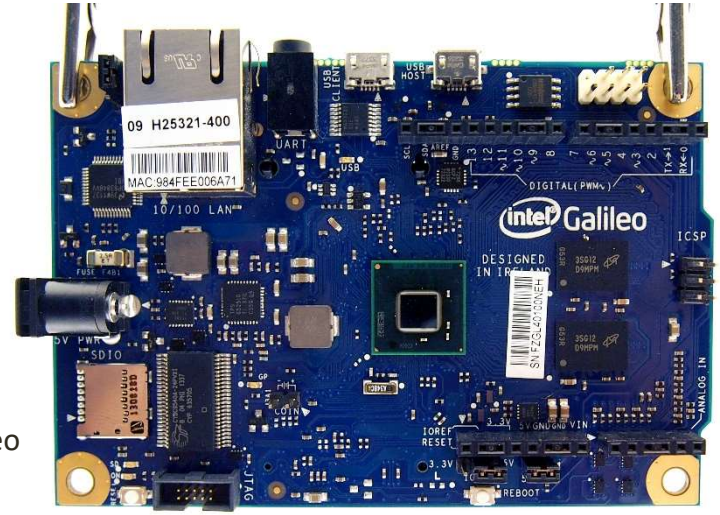


IoT: Sensors Available in the Market (examples)

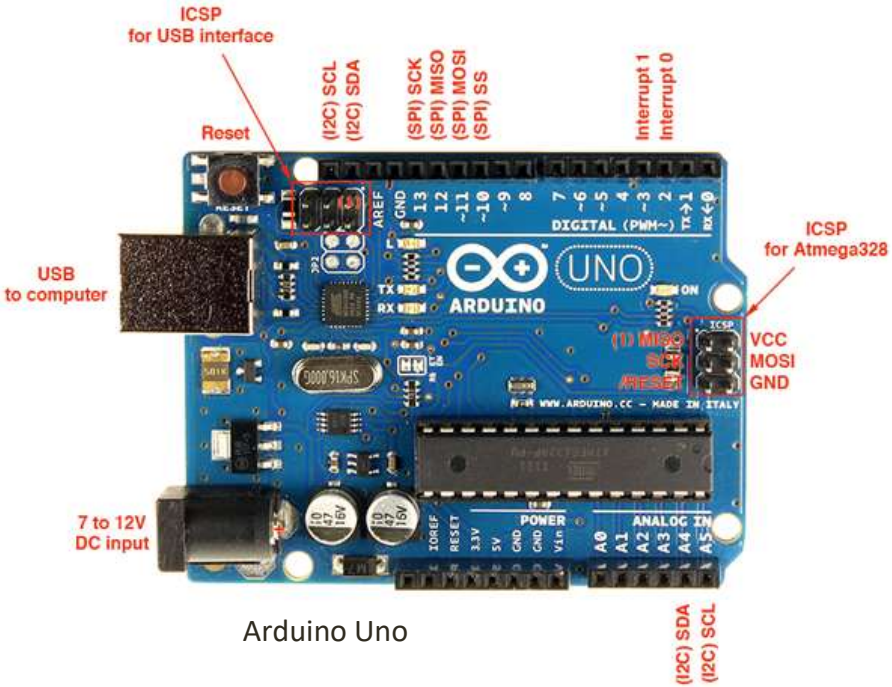
IoT: Smart Objects (examples)



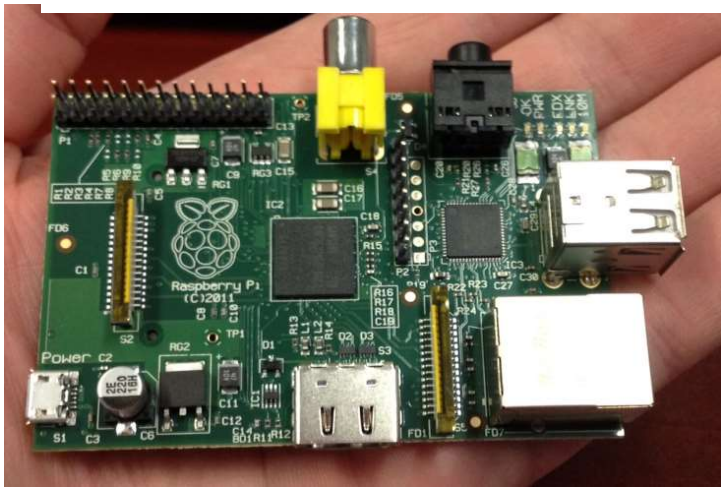
Beaglebone black



Intel Galileo



Arduino Uno



Raspberry Pi

IoT Communication Technologies

IoT Network Topology

Unified & Horizontal IoT Platform

Device Management/Cloud

2G/3G/LTE/WiFi/Fixed

Unlicensed LPWA Networks in ISM bands (+ other bands)

3GPP Licensed LPWA Network(s) NB-IoT + EC-GSM

2G/3G/LTE

2G/3G/LTE/Fixed

2G/3G

+20dB Link budget gain

Concentrator

RF Mesh

Smart Meters

Chillers

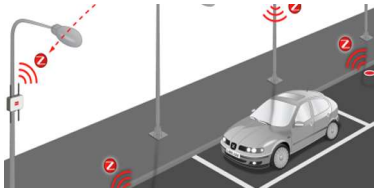
UPS

Fire

Security

CCTV

Power



Smart Meter

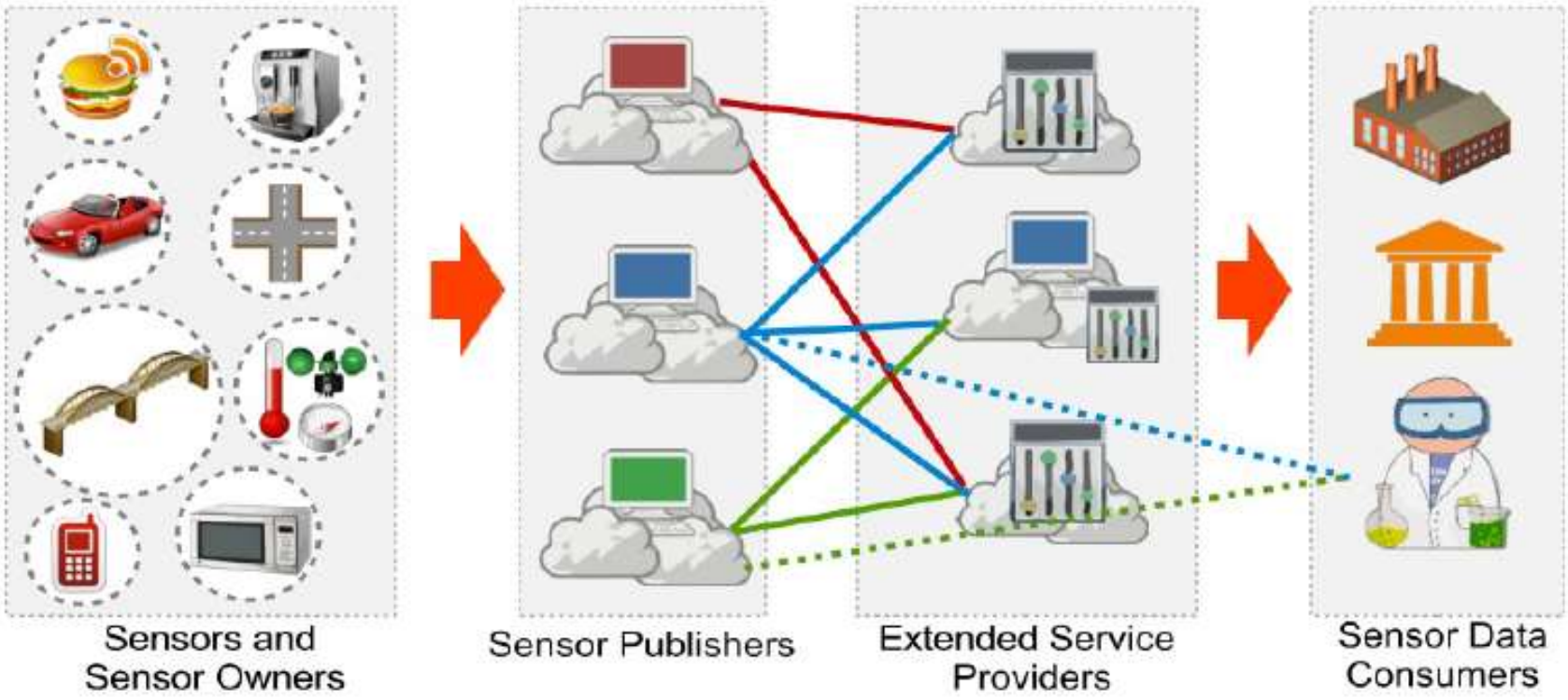
Smart Building Management

Fleet Management

Smart Waste Management with battery-powered

Smart Parking with battery-powered sensors

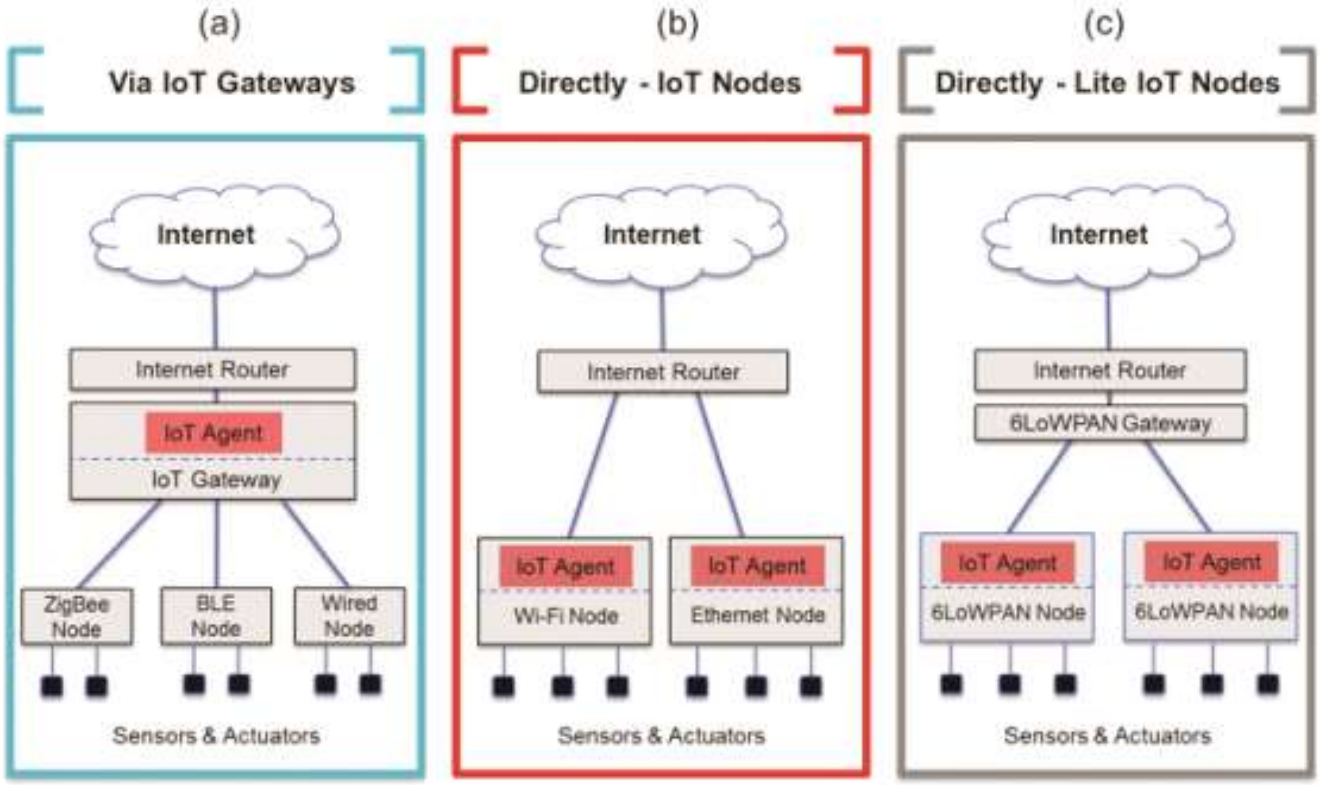
IoT Cloud: Sensing-As-A-Service Model



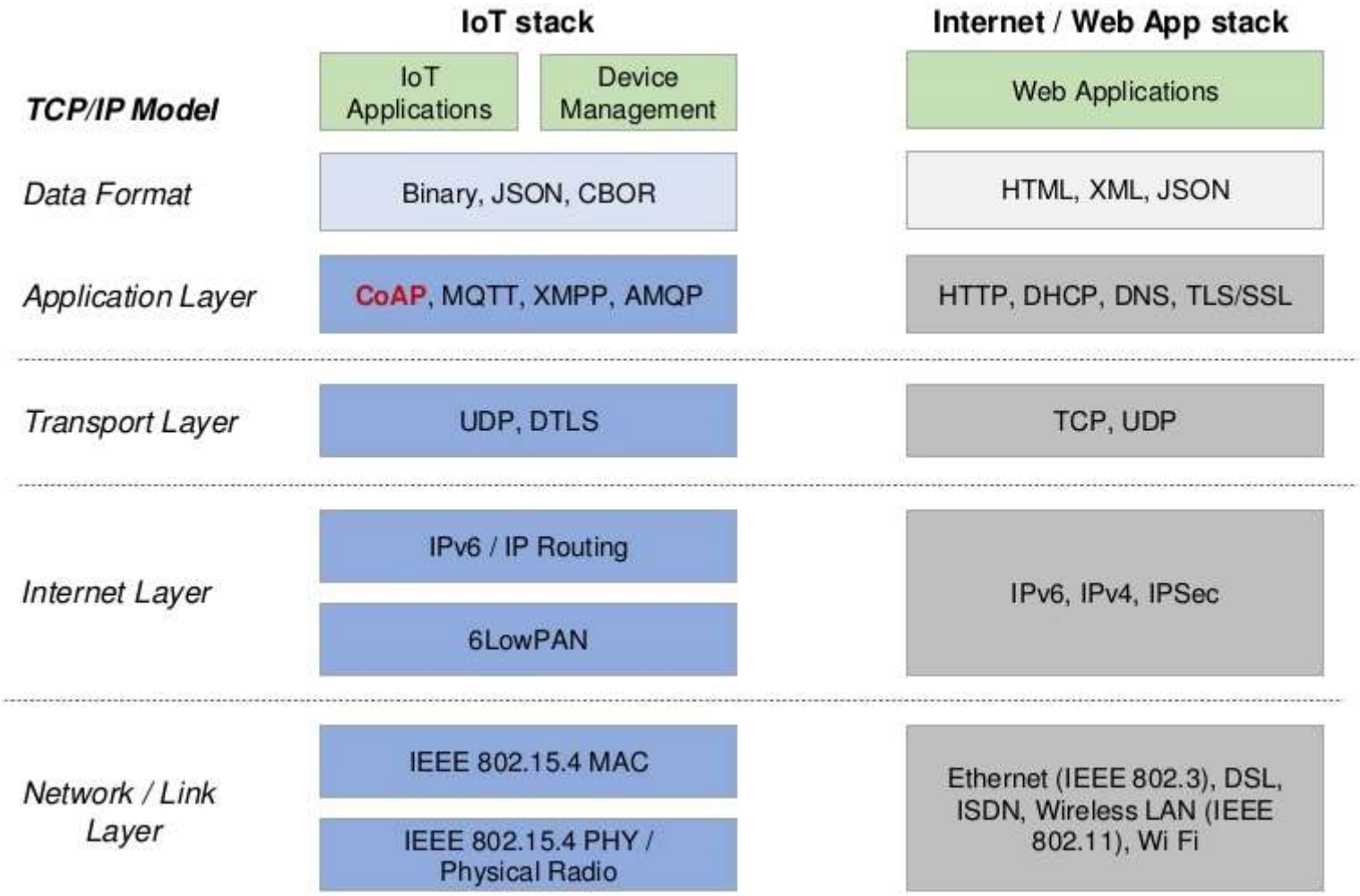
IoT Protocols

- **CoAP (Constrained Application Protocol)**
- **MQTT (Message Queue Telemetry Transport)**
- **XMPP (Extensible Messaging and Presence Protocol)**
- **6LoWPAN (Low power Wireless Personal Area Networks)**

IoT Protocol Architectures

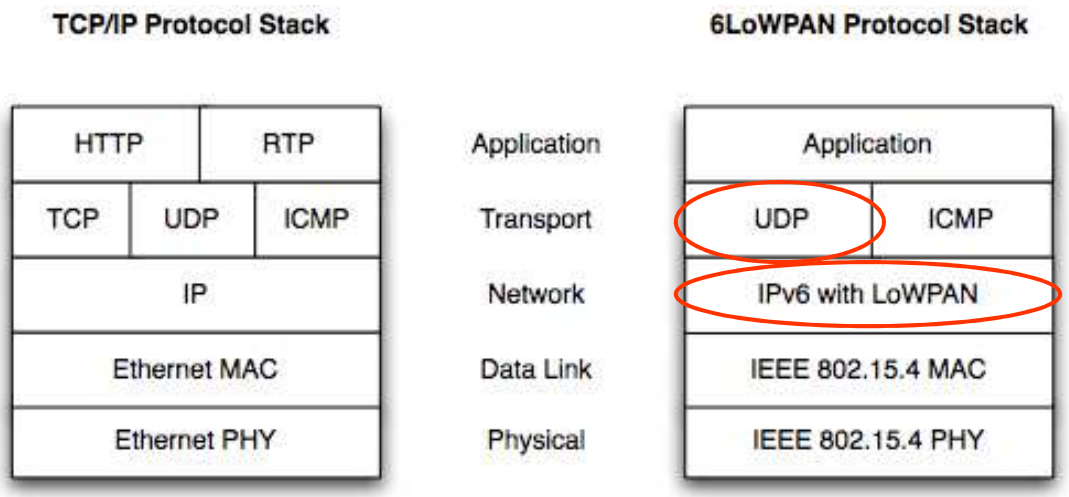


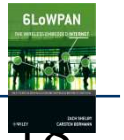
IoT Protocol Stack



The 6LoWPAN Format

- 6LoWPAN is an adaptation header format
  - Enables the use of IPv6 over low-power wireless links
  - IPv6 header compression
  - UDP header compression
- Format initially defined in RFC 4944
- Updated by RFC 6282





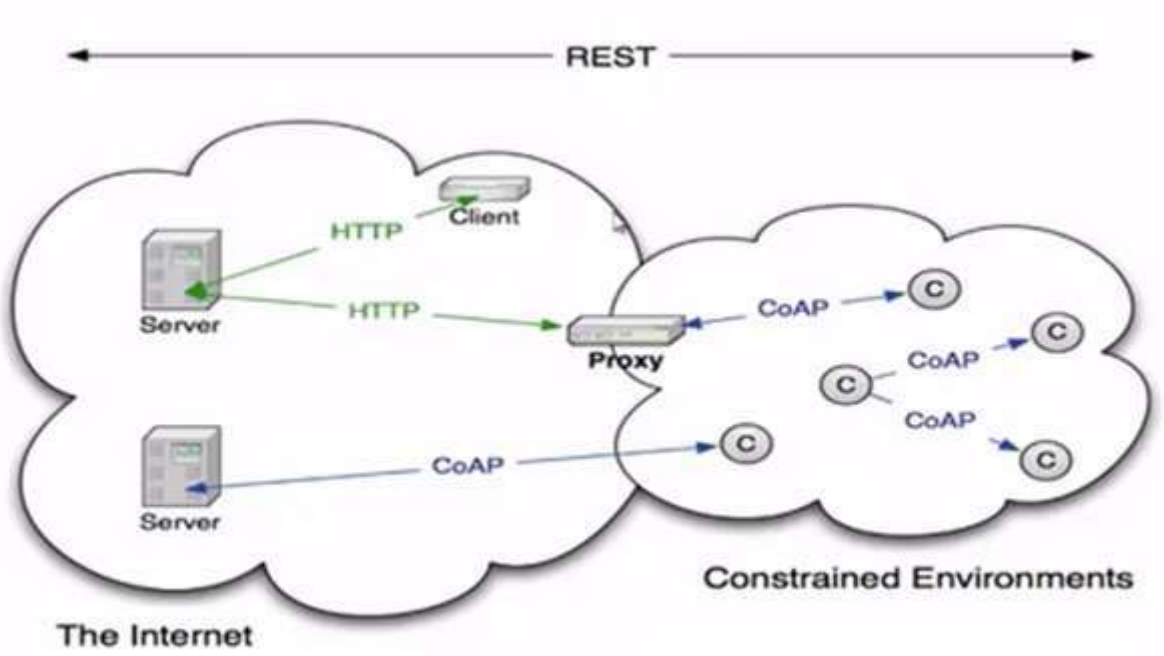
IoT Transport Layer for Smart Objects

- TCP for Smart Objects
  - Advantages
    - Built-in reliability
      - Mechanism to recover lost packets
    - Control of the maximum size of its packets
      - Use of the TCP MSS (Maximum Segment Size) option
  - Drawbacks
    - Many TCP mechanisms e.g., sliding-window, congestion avoidance are not needed in smart object networks
    - Large header size introduces a significant overhead.
- UDP for Smart Objects
  - Advantages
    - Low overhead for header size and protocol logic
      - Less energy for packet transmission and reception
      - More space for application data
      - Small code footprint
    - Well suited for traffic with low reliability demand.
  - Drawbacks
    - No provision of recovery mechanism for lost packets (application has to recover them)
    - No mechanism for splitting application data into appropriate packet sizes.
      - Usually, smart object networks deal w/ small packet sizes.

Constrained Application Protocol (CoAP)

- IoT oriented and less complex alternative to HTTP
- Open IETF standard (RFC 7252)
- Datagram Transport Layer Security (DTLS)
- Easy proxy to/from HTTP
- URIs supported (e.g., coap://hostname:port/leds/red?q=state&on)
- RESTfull client-server model
- Implements reliable unicast over UDP
- Supports best effort multicast
- Client-Server & Publish-Subscribe models.

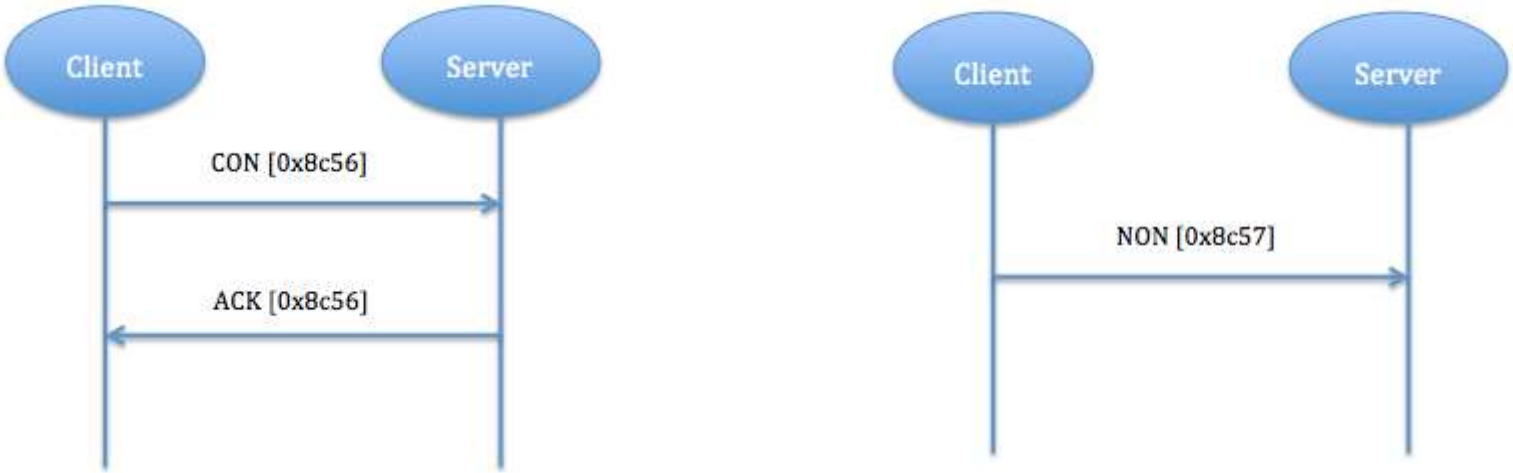
CoAP and HTTP Interworking



CoAP Architecture

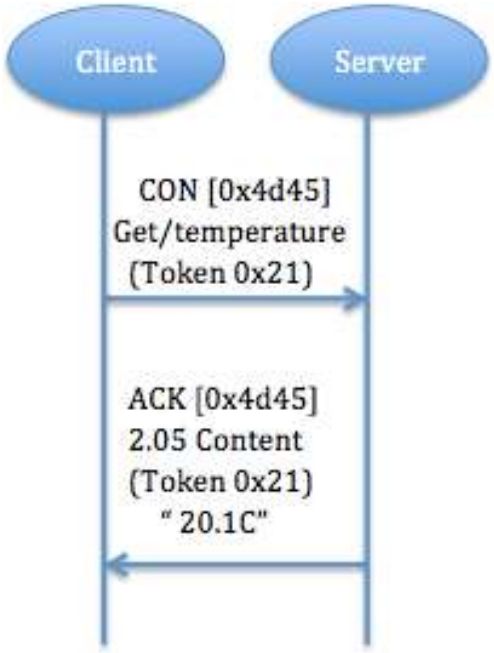
CoAP Message Layer Model

- Confirmed and non-confirmed message exchange models

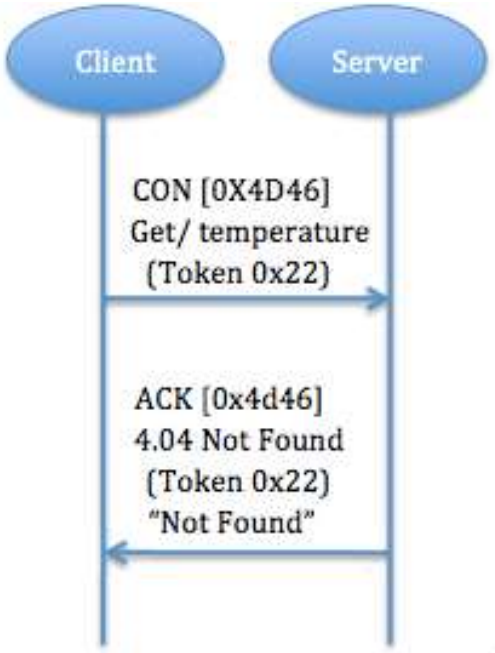


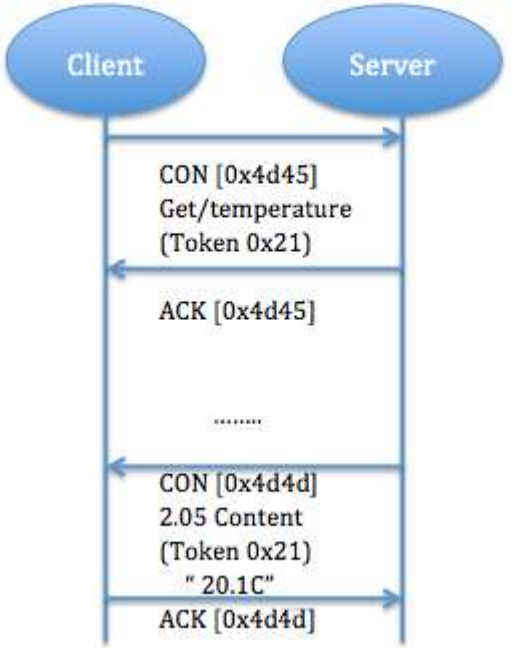
CoAP Request/Response Layer Model

- Piggy-backed Confirmed Response



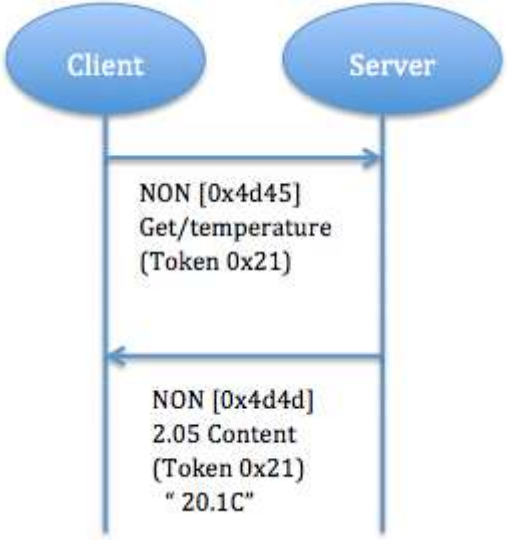
- Separate Confirmed Response



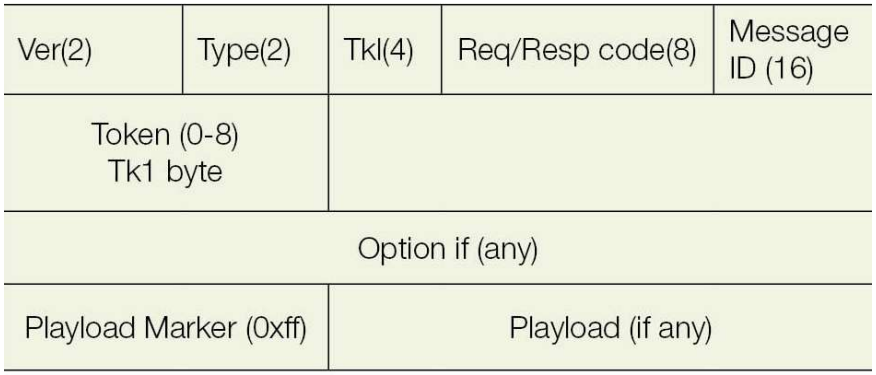


CoAP Request/Response Layer Model

- Non-confirmed Response



CoAP Message Format



CoAP message header	Description
Ver	It is 2 bit unsigned integer. It mentions CoAP version number. Set to one.
T	It is 2 bit unsigned integer. Indicates message type viz. confirmable (0), non-confirmable (1), ACK (2) or RESET(3).
TKL	It is 4 bit unsigned integer, Indicates length of token (0 to 8 bytes).
Code	It is 8 bit unsigned integer, It is split into two parts viz. 3 bit class (MSBs) and 5 bit detail (LSBs).
Message ID	16 bit unsigned integer. Used for matching responses. Used to detect message duplication.
Options	Zero or more option fields may follow a token. A few options are Content Format, Accept, Max-Age, Etag, Uri-Path, Uri-Query, etc.

Message Queuing Telemetry Transport (MQTT)

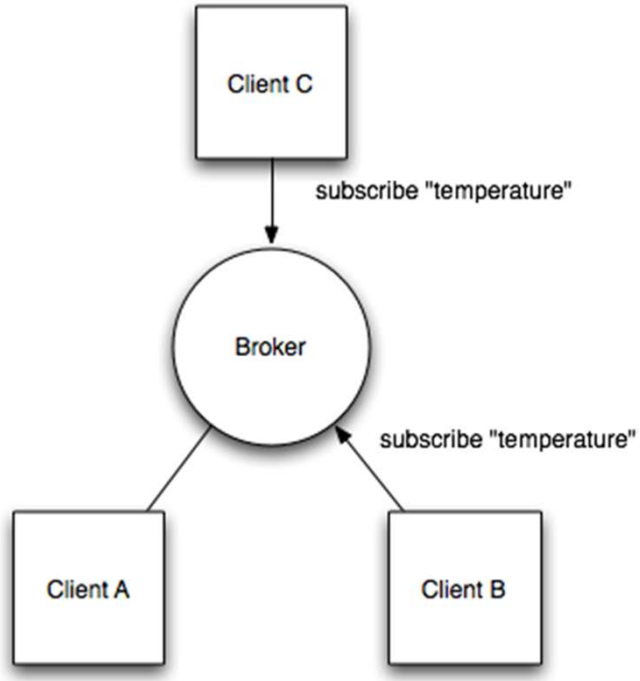
- Lightweight messaging protocol designed for sensors and devices with
  - Flaky network connectivity
  - Low computing power
  - Connections where bandwidth is at a premium
- Works on top of TCP
- Transport Layer Security (TLS)
- Protocol specification is open source
- Applications:
  - A way to obtain real world data
    - Information is gathered by an increasing number of sensors and devices deployed all over
- A way to provide real time information
  - E.g. Locate an item in a supply chain
  - Accurate current load of a any system (e.g. electricity meters)
  - Current status of a system (level of liquid in a container, temperature, pressure etc.)
  - A way to connect all the devices and sensors directly to your messaging infrastructure

MQTT Features

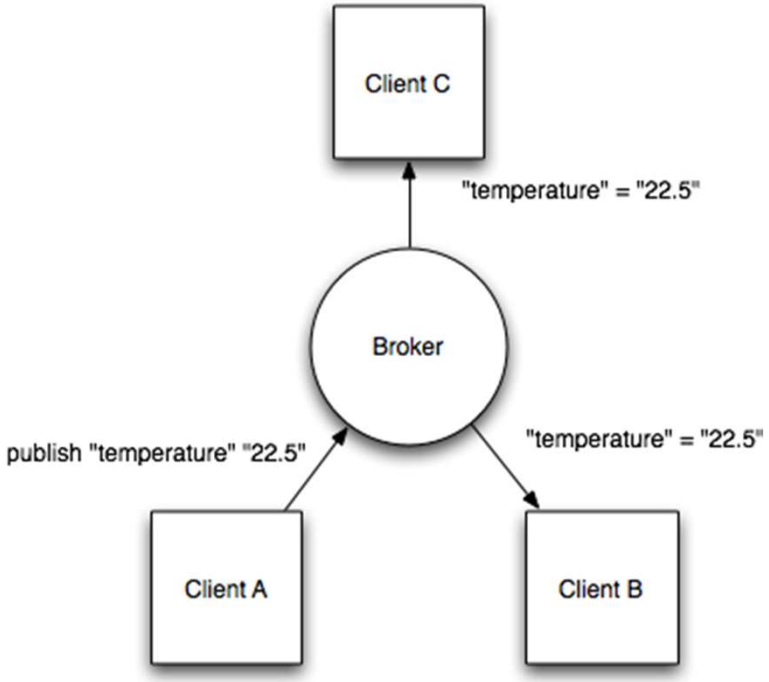
- Client/Server model with Clients and Brokers
- Publish and subscribe to topics
  - Managed by the broker
- 3 qualities of service
  - 0 Best effort to deliver a message
  - 1 Deliver at least once
  - 2 Deliver exactly once
- Supports persistent messages (only most recent per topic)
- Minimal transport overhead to reduce network traffic
  - As little as 2 bytes
- Last Will and Testament
  - MQTT clients can register a custom “last will and testament” message to be sent by the broker if they disconnect.
  - These messages can be used to signal to subscribers when a device disconnects.

MQTT Architecture

- All three clients open TCP connections with the broker. Clients B and C subscribe to the topic temperature .



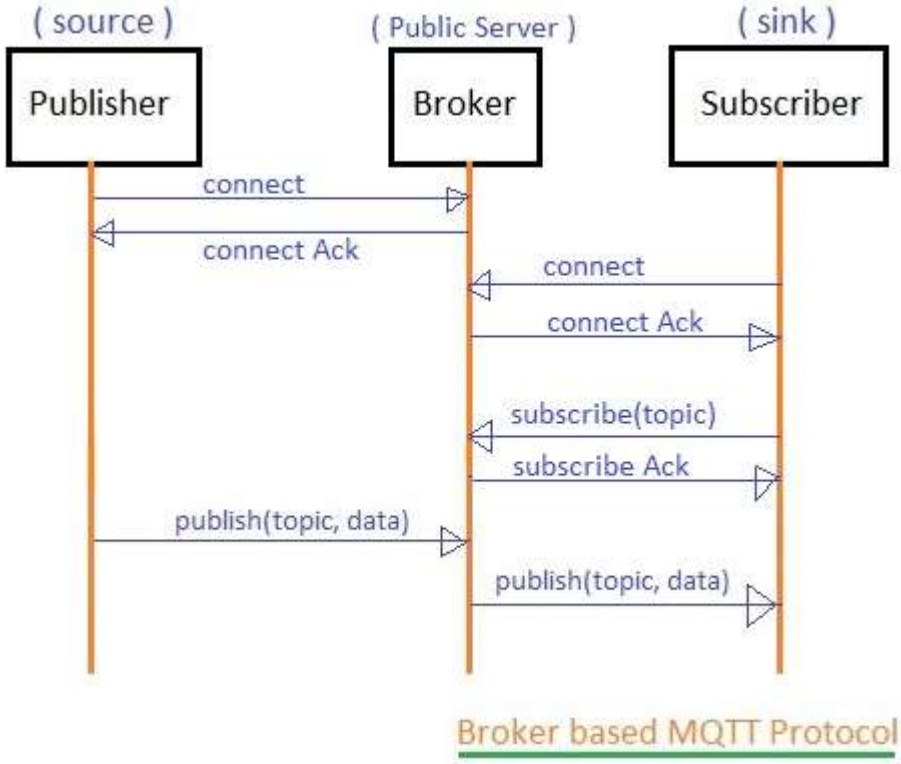
- At a later time, Client A publishes a value of 22.5 for topic temperature. The broker forwards the message to all subscribed clients.



MQTT Topics and Topic Matching

- In MQTT, topics are hierarchical, like a filing system (e.g., kitchen/oven/temperature).
- Wildcards are allowed when registering a subscription (but not when publishing) allowing whole hierarchies to be observed by clients.
- The wildcard + matches any single directory name, # matches any number of directories of any name.
- Examples:
  - kitchen/+/temperature matches kitchen/foo/temperature but not kitchen/foo/bar/temperature
  - kitchen/# matches kitchen/fridge/compressor/valve1/temperature

MQTT Protocol



Broker based MQTT Protocol

MQTT Message Format

- Fixed header

bit	7	6	5	4	3	2	1	0
byte 1	Message Type				DUP flag	QoS level		RETAIN
byte 2	Remaining Length							

Mnemonic	Enumeration	Description
Reserved	0	Reserved
CONNECT	1	Client request to connect to Server
CONNACK	2	Connect Acknowledgment
PUBLISH	3	Publish message
PUBACK	4	Publish Acknowledgment
PUBREC	5	Publish Received (assured delivery part 1)
PUBREL	6	Publish Release (assured delivery part 2)
PUBCOMP	7	Publish Complete (assured delivery part 3)
SUBSCRIBE	8	Client Subscribe request
SUBACK	9	Subscribe Acknowledgment
UNSUBSCRIBE	10	Client Unsubscribe request
UNSUBACK	11	Unsubscribe Acknowledgment
PINGREQ	12	PING Request
PINGRESP	13	PING Response
DISCONNECT	14	Client is Disconnecting
Reserved	15	Reserved

Bit position	Name	Description
3	DUP	Duplicate delivery
2-1	QoS	Quality of Service
0	RETAIN	RETAIN flag

QoS value	bit 2	bit 1	Description		
0	0	0	At most once	Fire and Forget	<=1
1	0	1	At least once	Acknowledged delivery	>=1
2	1	0	Exactly once	Assured delivery	=1
3	1	1	Reserved		

Digits	From	To
1	0 (0x00)	127 (0x7F)
2	128 (0x80, 0x01)	16 383 (0xFF, 0x7F)
3	16 384 (0x80, 0x80, 0x01)	2 097 151 (0xFF, 0xFF, 0x7F)
4	2 097 152 (0x80, 0x80, 0x80, 0x01)	268 435 455 (0xFF, 0xFF, 0xFF, 0x7F)

MQTT for Sensor Networks (MQTT-SN)

- Even though MQTT is designed to be lightweight, it has two drawbacks for very constrained devices:
  - Every MQTT client must support TCP and will typically hold a connection open to the broker at all times. For some environments where packet loss is high or computing resources are scarce, this is a problem.
  - MQTT topic names are often long strings which make them impractical for 802.15.4 and other low bitrate small packet protocols.
- Both of these shortcomings are addressed by the MQTT-SN protocol:
  - MQTT does not require TCP (can use UDP or serial link)
  - Broker support for indexing topic names (short topic IDs).
- Requires MQTT-SN to MQTT gateway.

MQTT vs CoAP

Features	MQTT	CoAP
Full Form	Message Queue Telemetry Transport	Constrained Application Protocol
Messages used	Connect, connect ack, publish, publish ack, subscribe, subscribe ack, disconnect etc.	GET, PUT, POST and DELETE
Architecture	Publish/Subscribe	Request/Response
Need of centralized broker	required, end devices communicate via broker	not required, end devices direct communicate
Transport protocol	TCP/IP	UDP/IP
Security protocol	TLS	DTLS
fault tolerance	broker is SPoF	server is SPoF
Interoperability	foundational	semantic
scope	device to cloud cloud to cloud	device to cloud cloud to cloud

Acknowledgements

1. Dr.Kayarvizhy, “Internet of Things”, http://studyslide.com/doc/20337/iot---dr.-kayarvizhy
2. Augusto Casaca, “Internet of Things”, IFIP TC6 LATIN AMERICA TUTORIALS IN NETWORKING
3. Xi Chen, “Constrained Application Protocol for Internet of Things”, https://www.cse.wustl.edu/~jain/cse574-14/ftp/coap/index.html
4. Toby Jaffey, “MQTT and CoAP, IoT Protocols”, https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php
5. “MQTT vs REST”, http://www.rfwireless-world.com/Terminology/MQTT-vs-REST.html