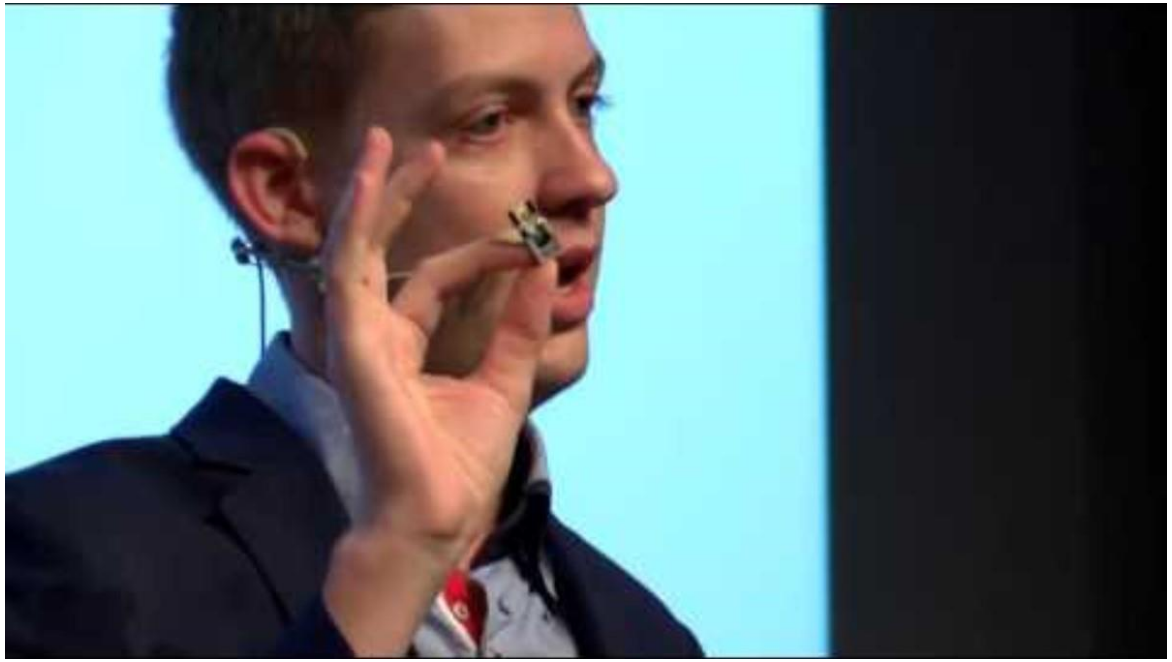
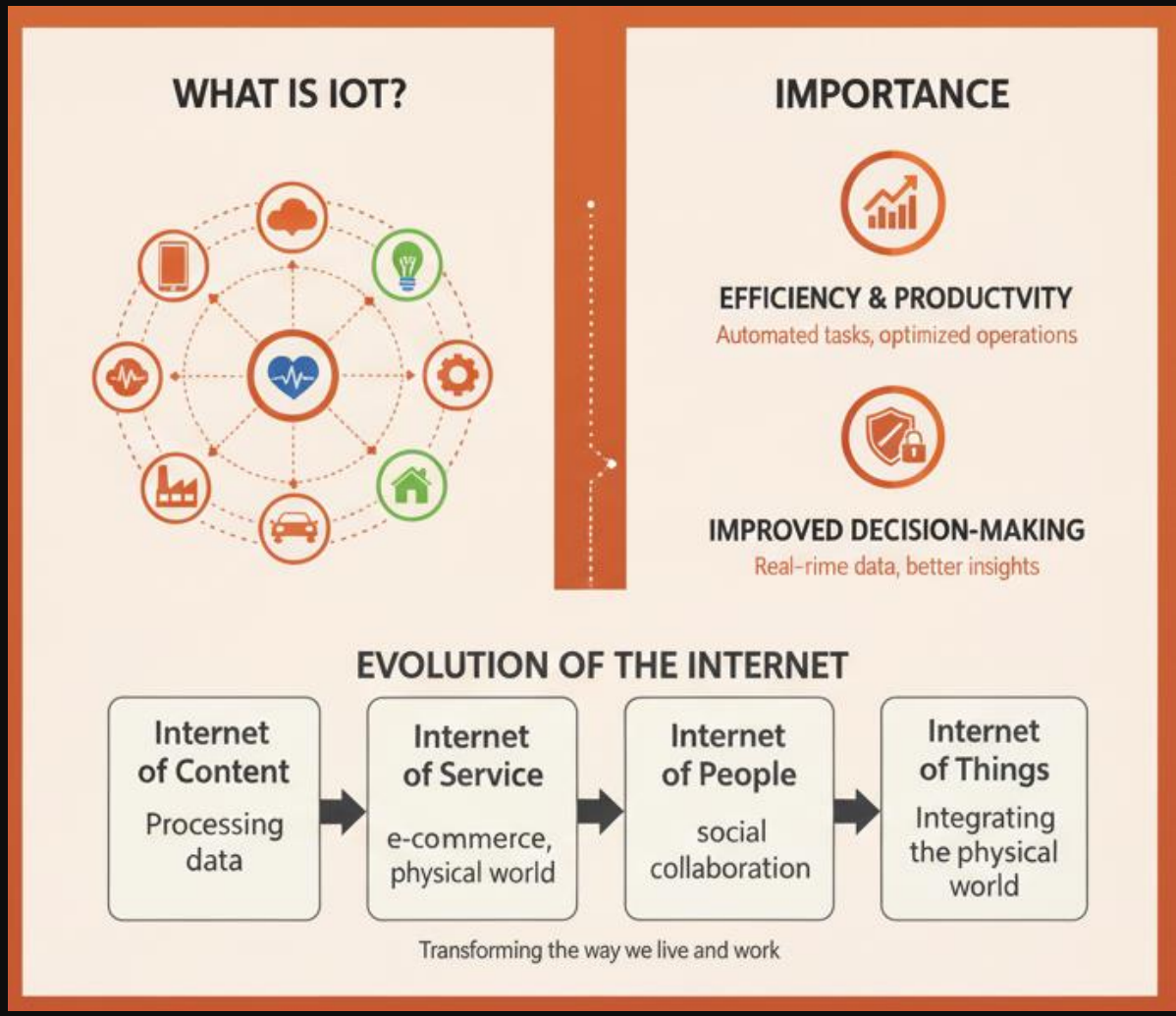
Internet of Things (IoT)-1

Recap: What is Internet of Things (IoT)?



Definition of IoT

The Internet of Things (IoT) describes physical objects that are embedded with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the internet or other communication networks.



Deep Dive: IoT Solution Development Methodology

Define & Sense
- Define the problem domain and requirements (what you want the IoT system to achieve, who the users are, what physical things are involved).
- Determine how to *sense* the physical world (which sensors/devices will detect relevant phenomena).

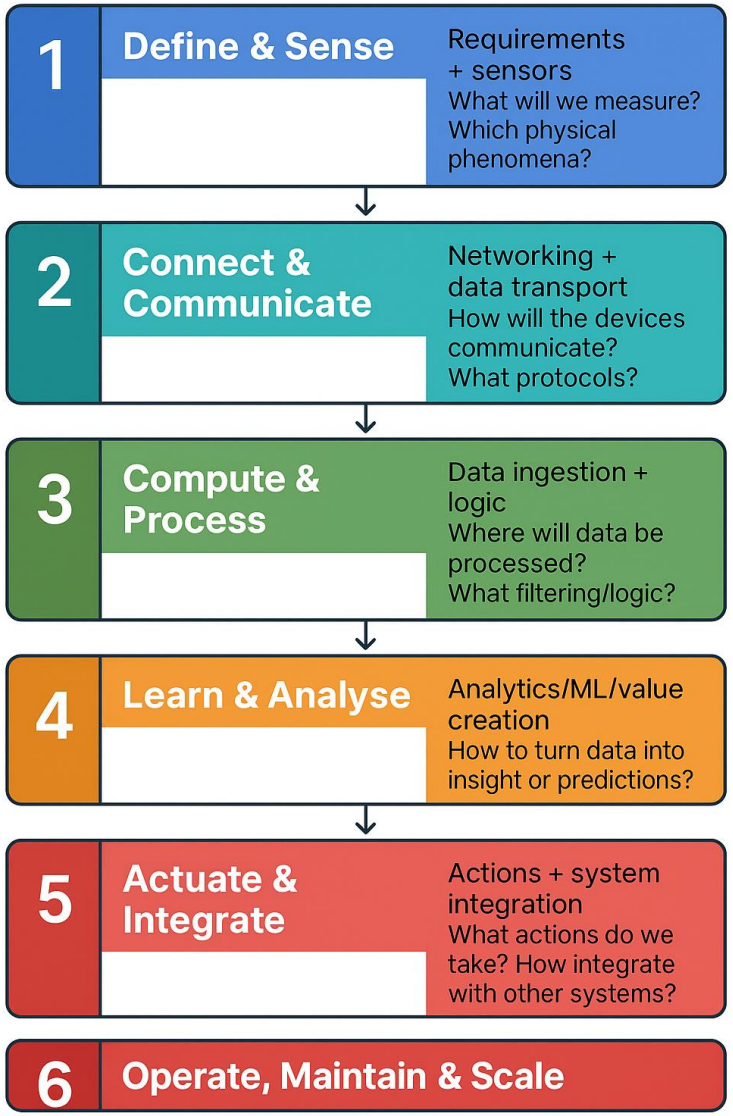
Connect & Communicate
- Choose how data from the “things” get communicated (wired/wireless, protocols, connectivity, network infrastructure).

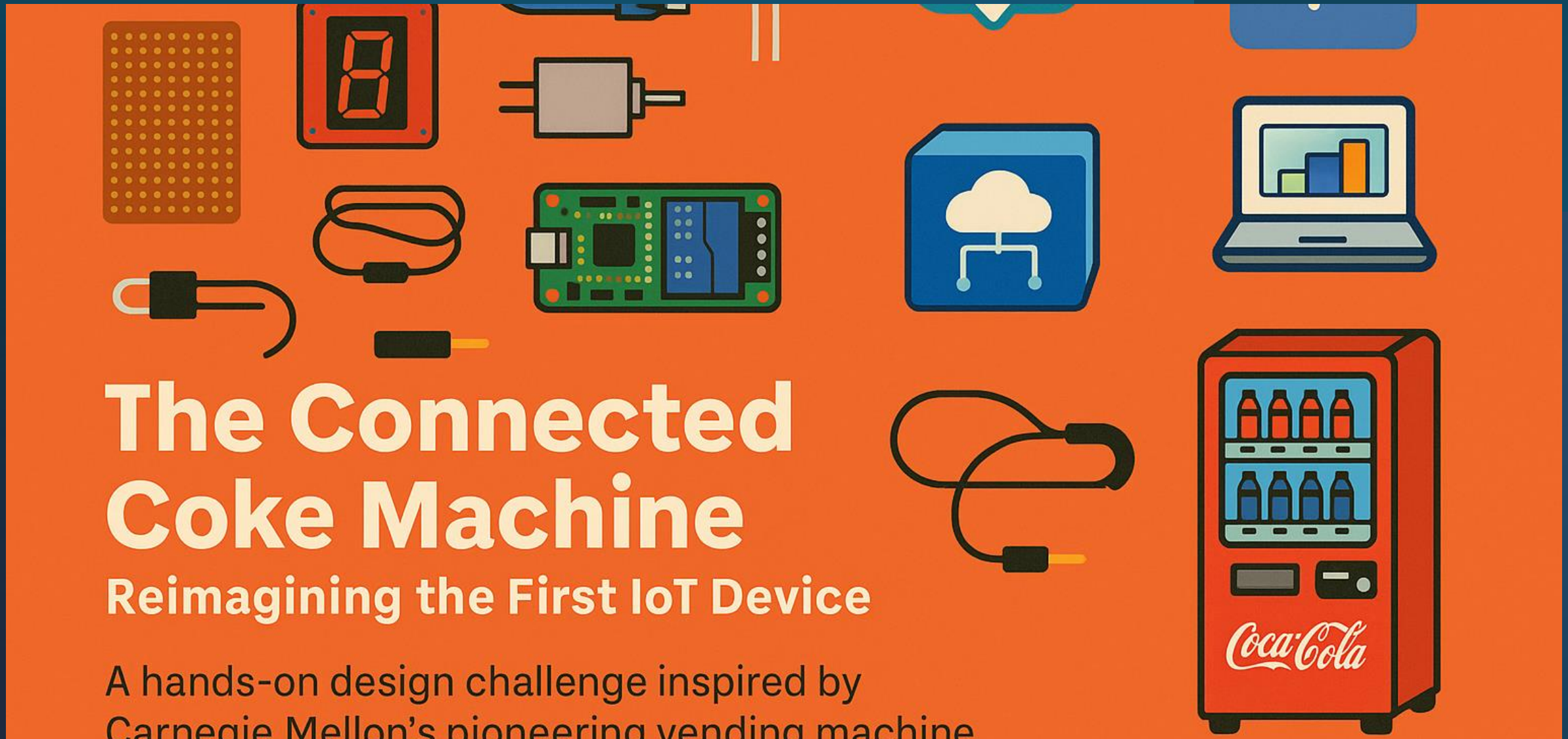
Compute & Process
- Decide where data is processed, aggregated, cleaned, analysed.
- Define the logic: what happens when data arrives? e.g., filtering, threshold detection.
- Also think about storage, state-management, and real-time vs batch processing.

Learn & Analyse
- Use analytics / machine learning / feedback loops to derive value from the collected data: e.g., predictive maintenance, usage patterns, optimisation.
- In an IoT context this may include: anomaly detection, forecasting, adaptive control.

Actuate & Integrate
- Trigger actions or responses based on results of the processing/learning: drive actuators, send alerts, update dashboards, change system behaviour.
- Integrate with external systems, build user-interfaces, dashboards, APIs.

Operate, Maintain & Scale
- Deploy the IoT system, monitor its operation, maintain (firmware updates, security patches, sensor replacement).
- Plan for system evolution, scalability, lifecycle (decommissioning, reuse).

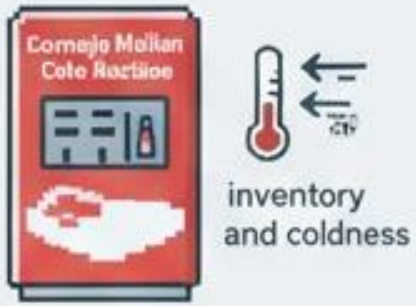



The Connected
Coke Machine
Reimagining the First IoT Device
A hands-on design challenge inspired by
Carnegie Mellon's pioneering vending machine
Coca-Cola

Challenge Worksheet

Steps	Key Focus	Your Design ideas?
Step 1: Define & Sense	Requirements + sensors, what to measure.	
Step 2: Connect & Communicate	Networking + data transport, protocols.	
Step 3: Compute & Process	Data ingestion + logic, edge/cloud filtering.	
Step 4: Learn & Analyse	Analytics/ML/value creation, turn data into insight.	
Step 5: Actuate & Integrate	Actions + system integration, external systems interface.	
Step 6: Operate, Maintain & Scale	Lifecycle, updates, scalability, and security.	

A BRIEF HISTORY OF THE INTERNET OF THINGS



inventory and coldness

1980s

THE FIRST CONNECTINE:

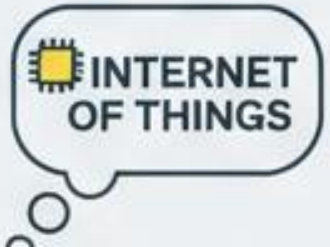
Applicarce: Remotely checked inventory

1993

THE INTERNET TOASTER

First device controled over

2000



2009

PRESENT DAY

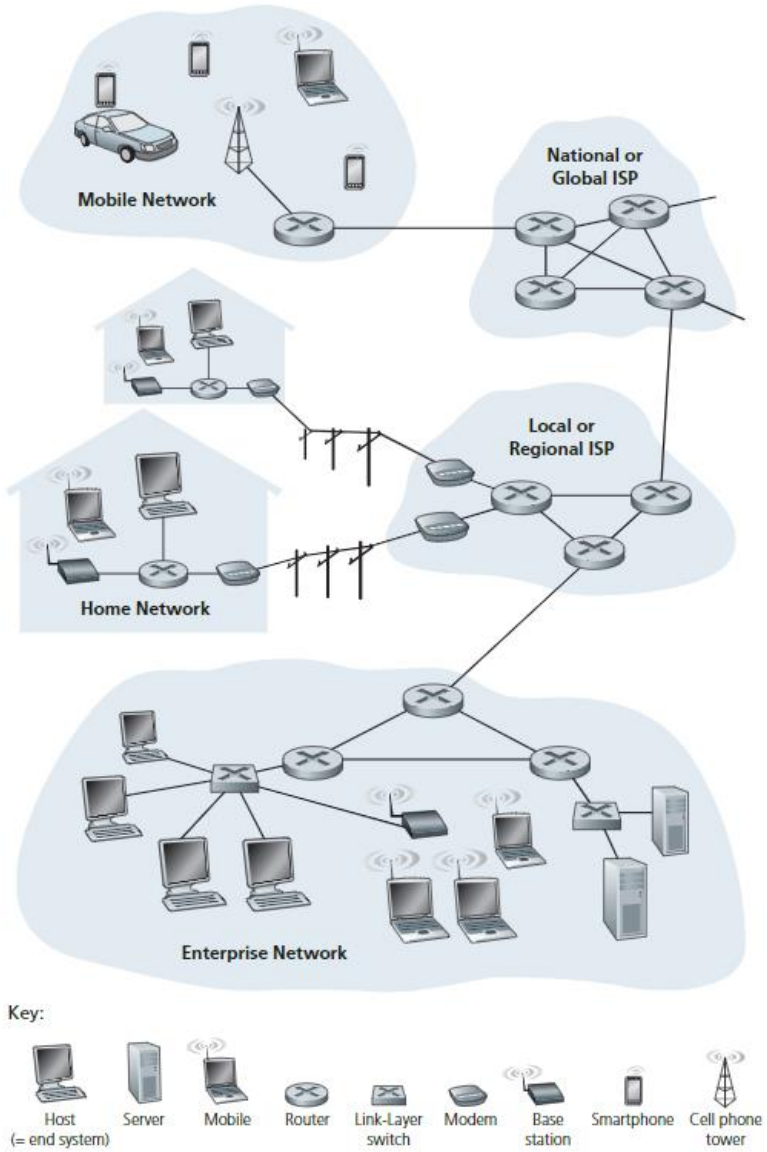
IOT BECOMES PERVAIVE:

Billions of devices, AI/ML, 5G, and

So how does the Internet work?

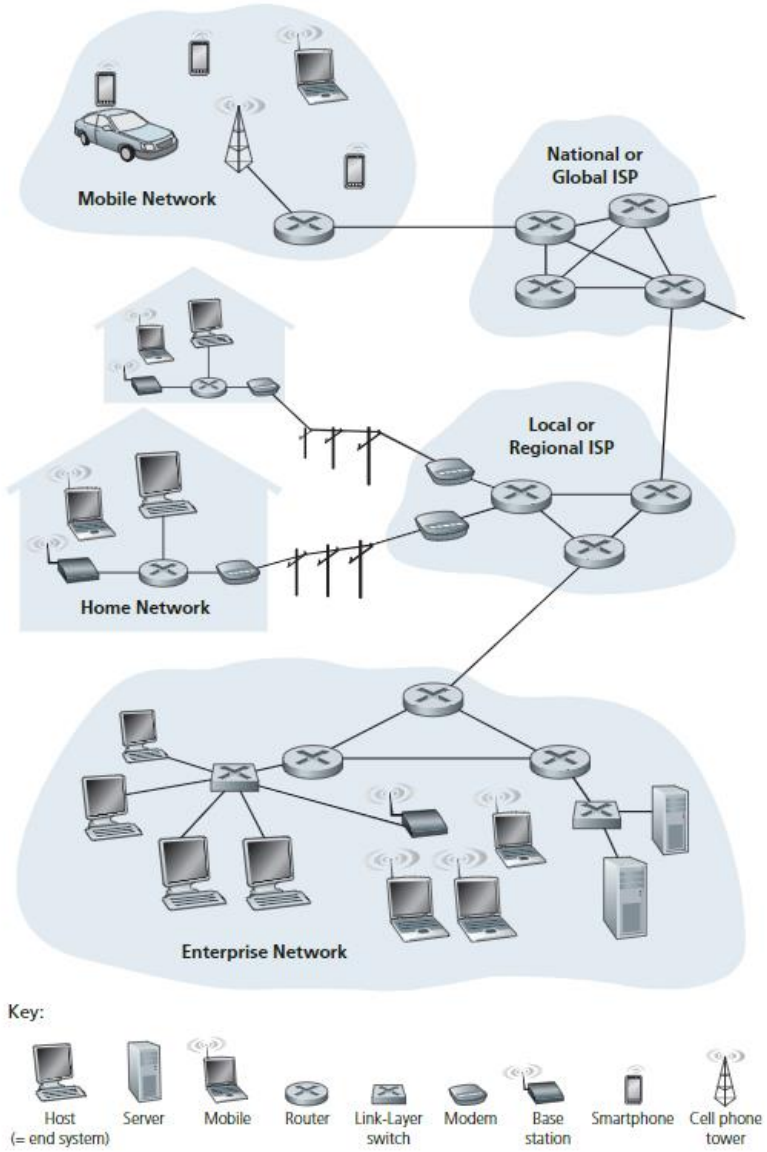
Architectural Elements of the Internet

- Internet is a wide area network that interconnects billions of devices ranging from sensors to computer throughout the world.
- The path from source to destination can be as simple as single cable or as complex as a network spanning the entire globe.
- The link from source-to-destination is generally called uplink/upstream and similarly the reverse path is known as downlink/downstream.



Architectural Elements of the Internet

- Network infrastructure contains three categories of equipment:
  1. End Systems: Also known as hosts, i.e., client/server devices.
  2. Intermediate Devices: Devices capable of directing packets or performing other functions on the packets.
  3. Network Media: Communication medium which provides a channel for communication. Think of this as a pipe for transferring water between tanks.



Source: [1] Kurose, James F. Computer networking: A top-down approach featuring the internet, 3/E. Pearson Education India, 2005 – Chapter 1

Architectural Elements of the Internet

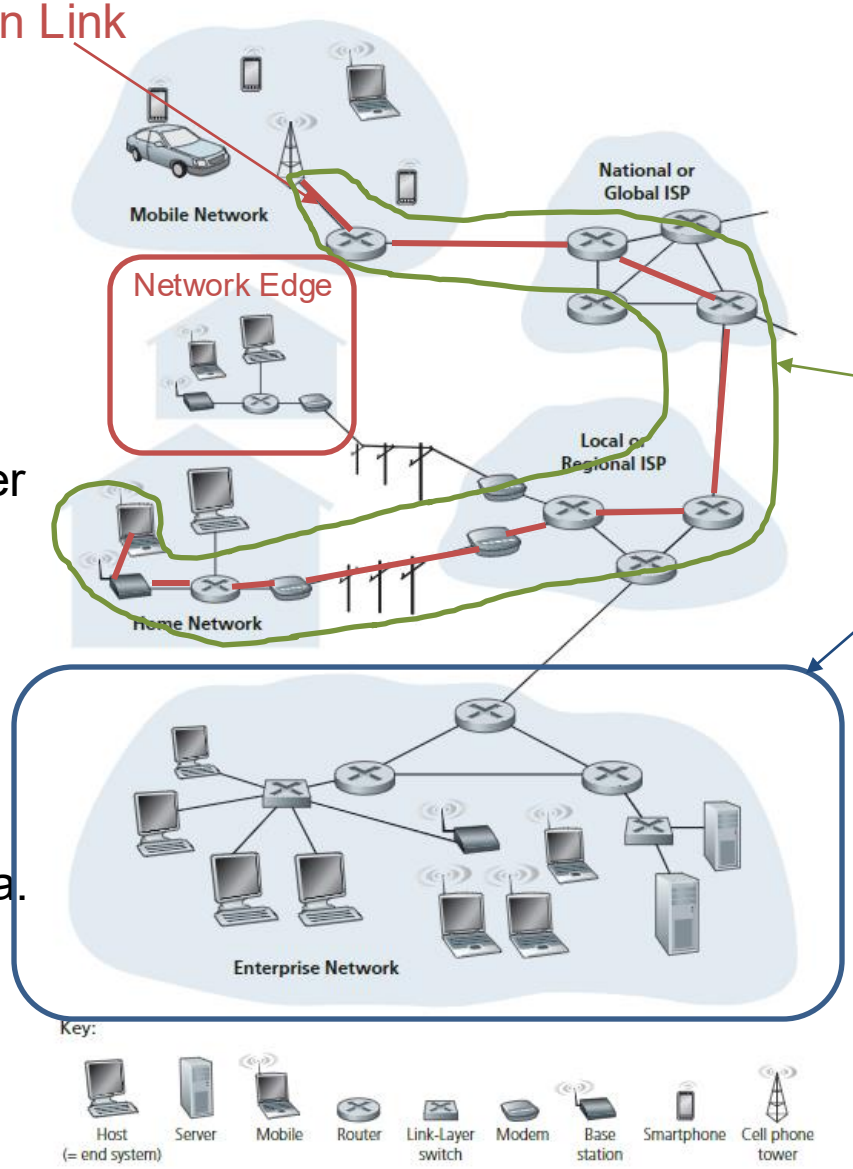
Communication Link

Transmission Latency: Delay between transmission and reception on a communication link

Link Throughput: Number of Bits per second.

Media Considerations:

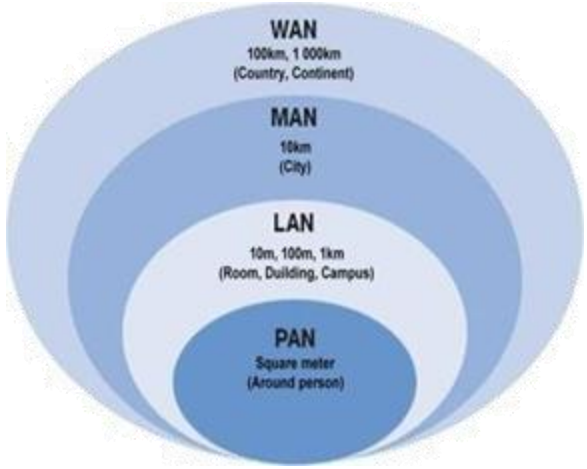
- Distance of link is dictated by media.
- Impairments associated with media.
- Maximum sustained throughput.
- Cost (CAPEX and OPEX)



Hop Count: Number of Communication Links traversed in a path.

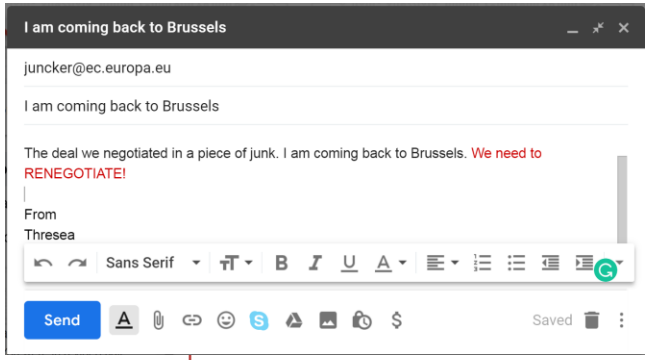
Path

Topology Diagram

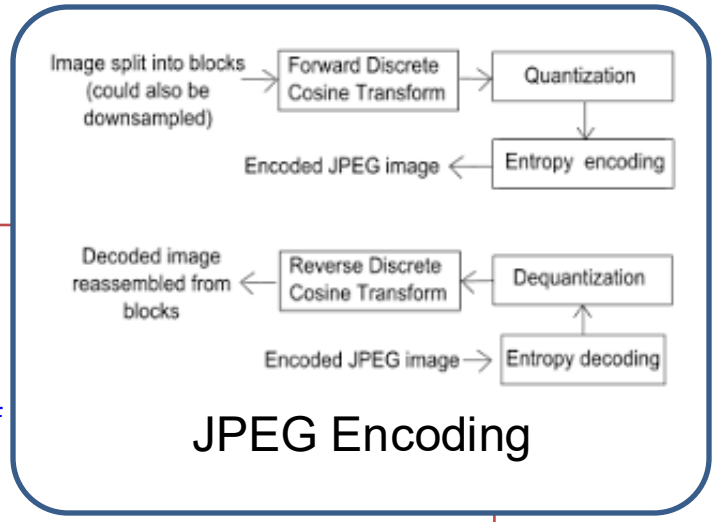


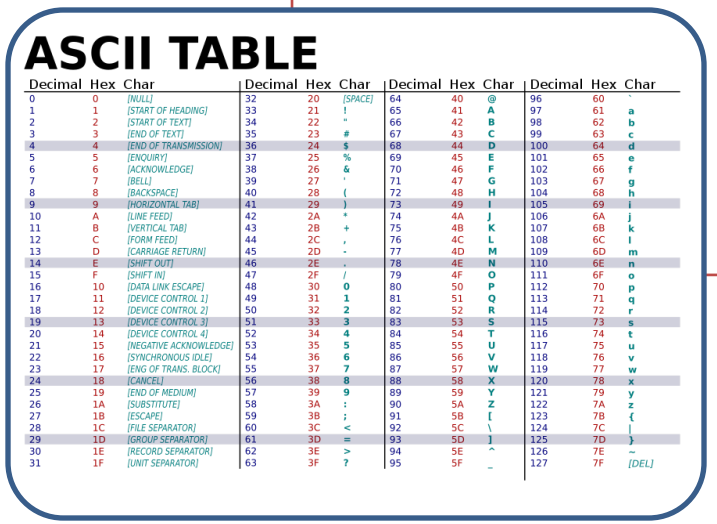
Inter-Networking Challenges

- Difference in form and nature of content which needs to be transmitted over the network. For example, text in different languages, video etc.



https://dutchreview.com/featured/brexit-on-the-sidelines/





ASCII TABLE

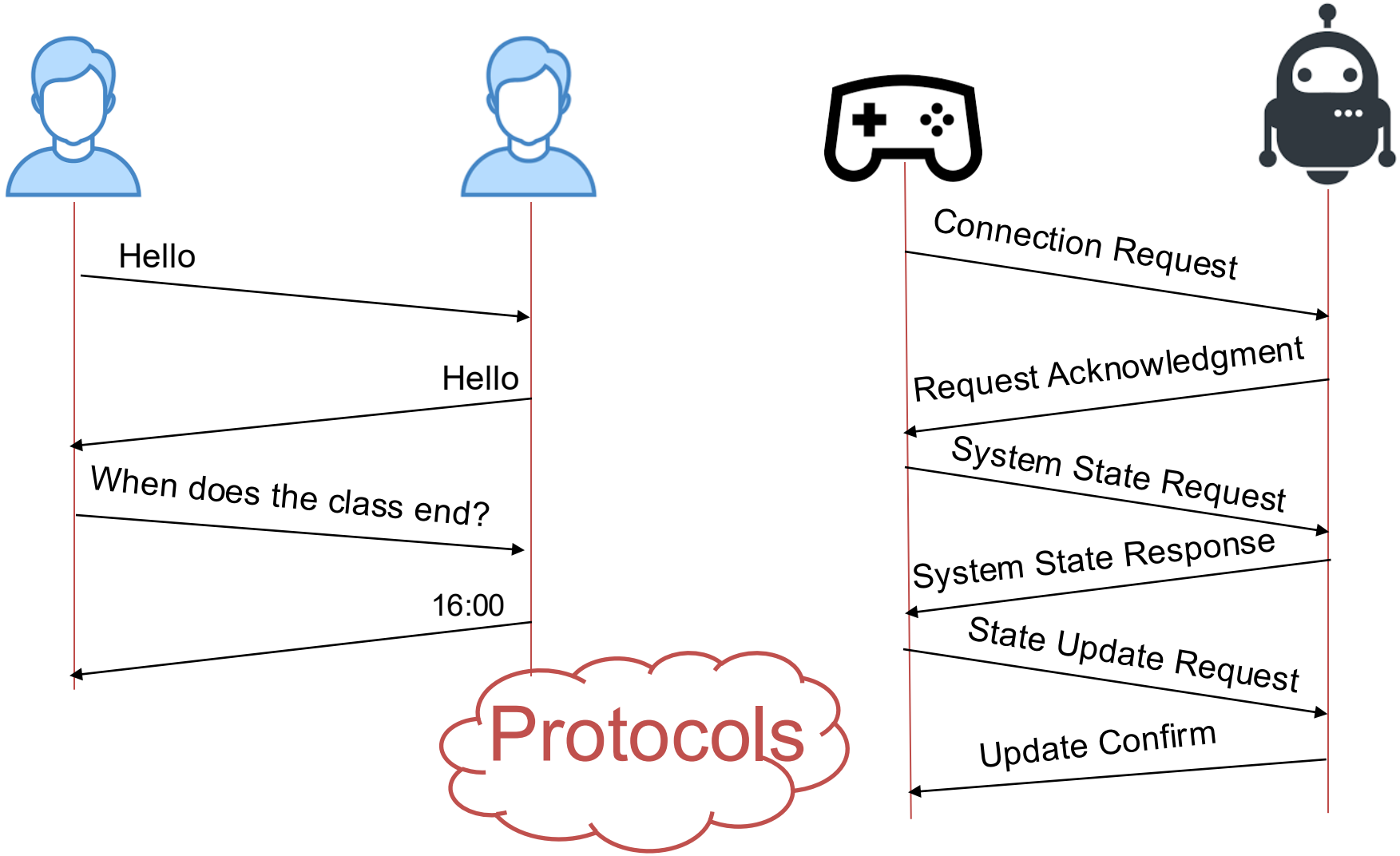
Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Encoding/Source Coding

1010101010000011001

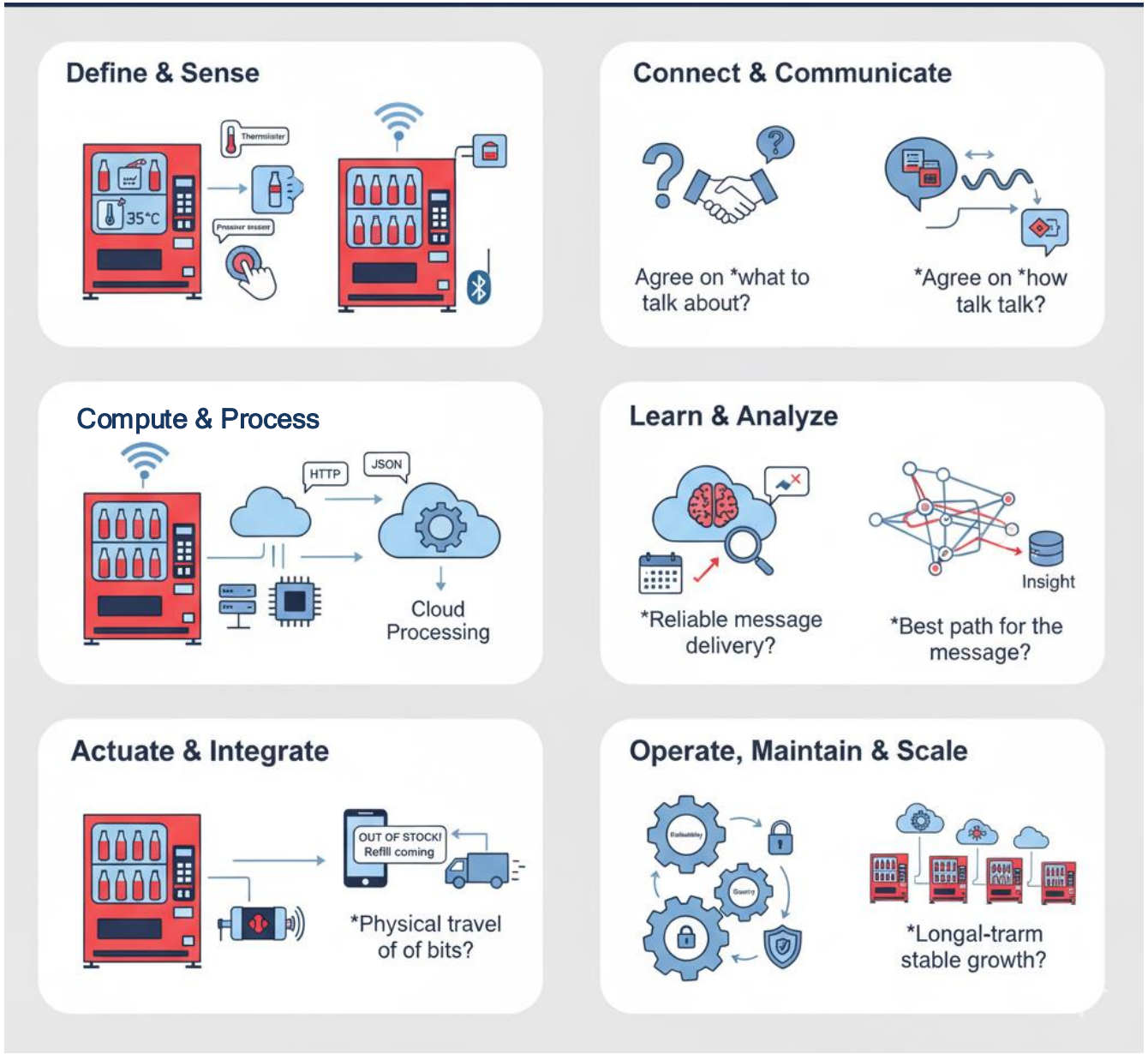
Inter-Networking Challenges

- How do we structure our conversation?

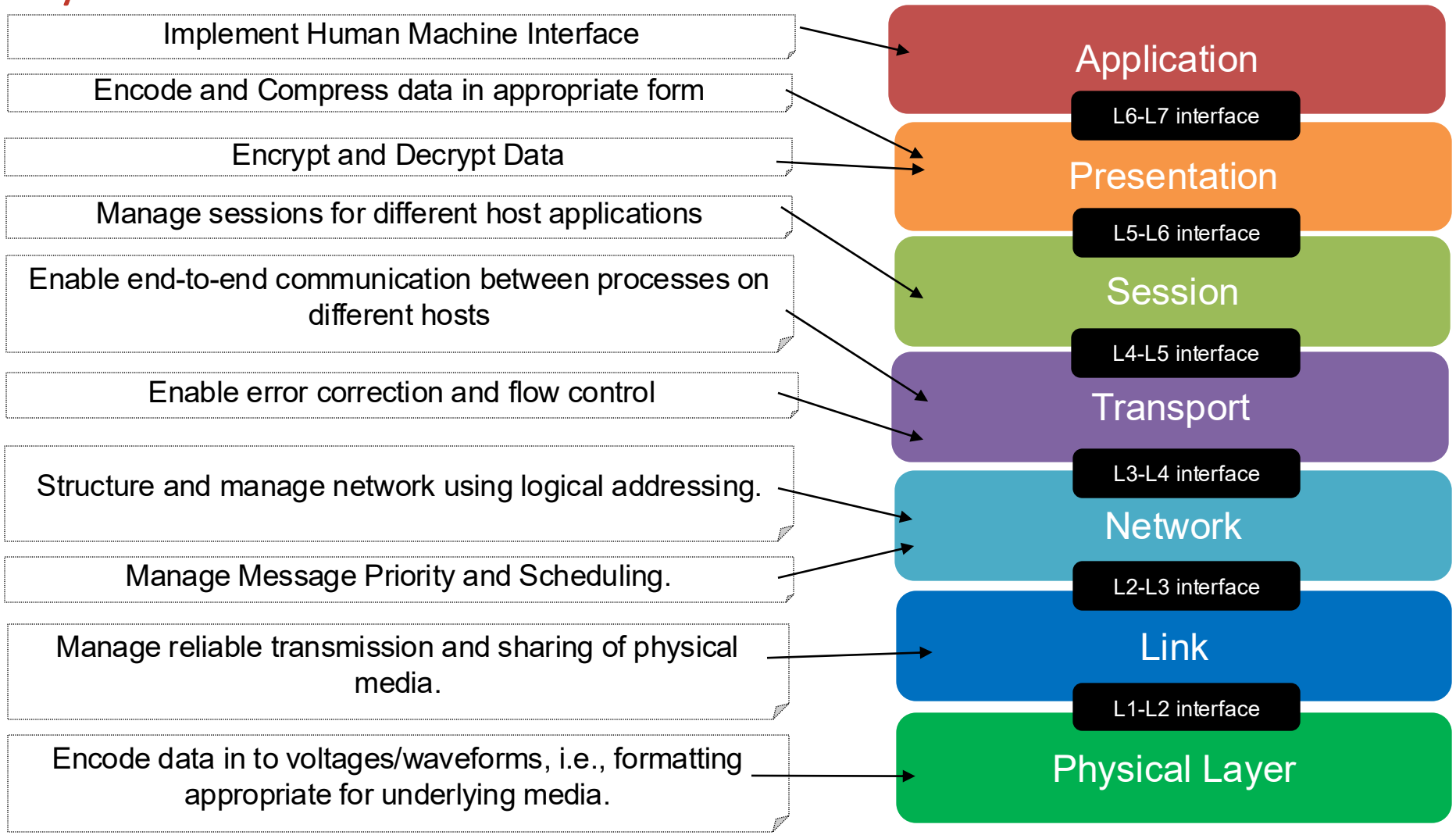


Point-to-Point Networking and Beyond

- “How do two devices determine and agree on how to start and end a conversation?”
- "How do they know what language or data format to use?”
- “What happens if a message is lost or arrives out of order?”
- “How does the system decide which path the message should take through the network?”
- “How do the bits and bytes of that message physically travel from one device to another?”



Layered Architecture



Open Systems Interconnection model

OSI and Pizza Deliver

Application Layer

Eat Pizza

Cook Pizza

Presentation Layer

Remove Pizza from Box

Pack in Pizza Box

Remove bock packaging.

Session Layer

Pack the box with address label

Whitehouse is on shutdown and I have guests, Send me two 14 inch, one 11 inch and one 10-inch Mexican Pizza?

Sure Sir, will be delivered in 20 mins. Where would you like it to be delivered?

Whitehouse Of course.

Collect from courier and acknowledge receipt via app.

Transport Layer

Dispatch it to right courier and track delivery via app.

Arrive at location provided by GPS.

Network Layer

Use GPS to Locate Delivery Address

Link Layer

Physical Layer

Share the road with other traffic according to rules.

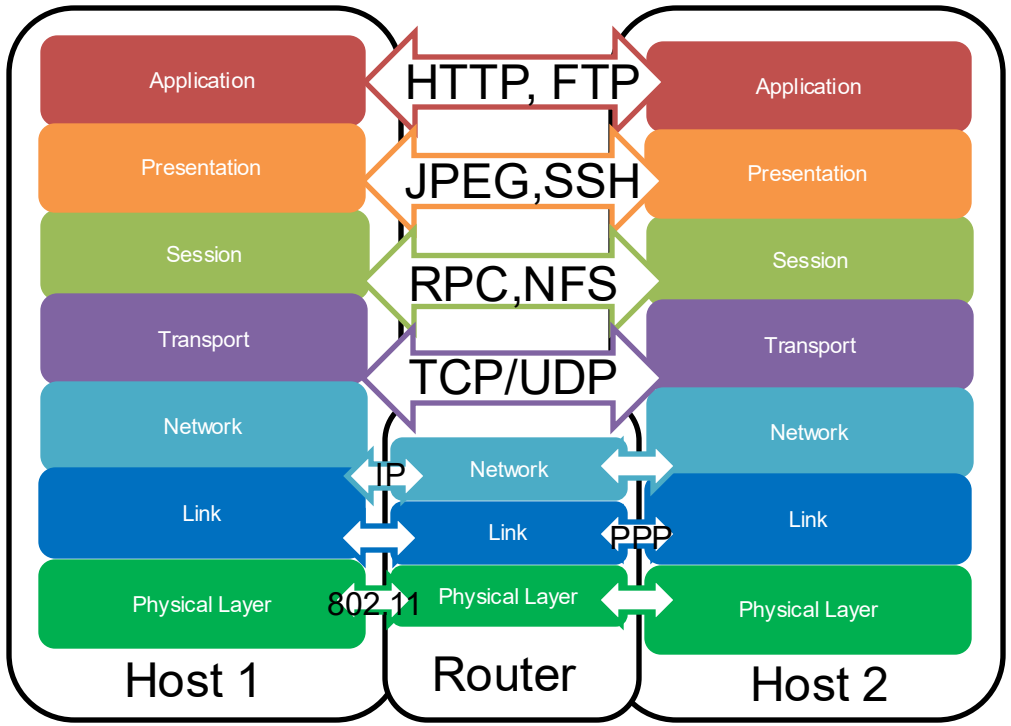
Select appropriate vehicle and commute on the road..

Select appropriate vehicle and commute on the road..

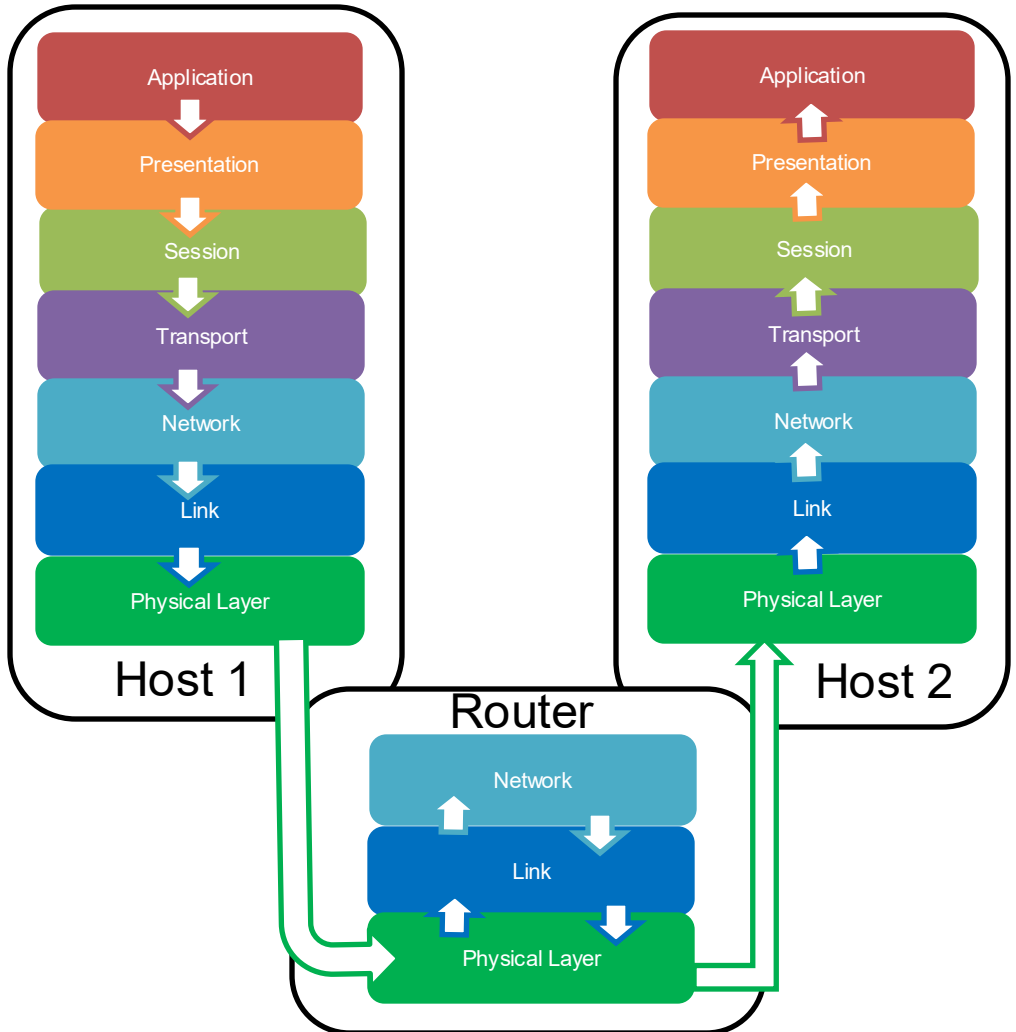
Share the road with other traffic according to rules.

Logical vs. Physical Interconnects

Logical Interconnect: Each layer interacts with corresponding layer via protocol interface.

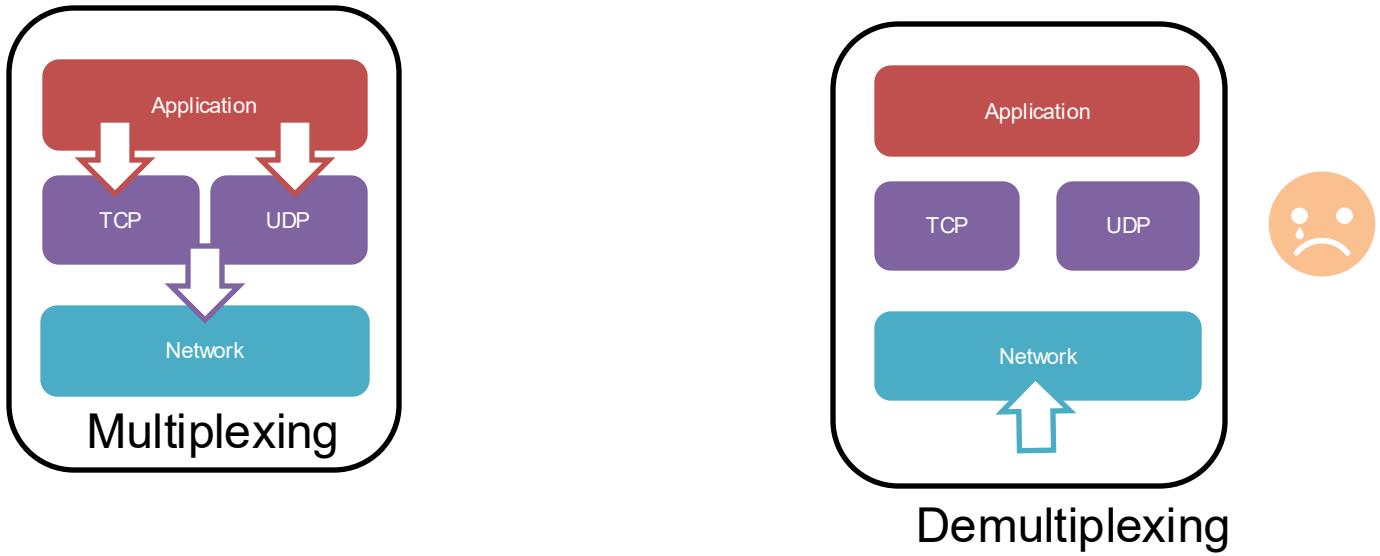


Physical Interconnect: Each layer interacts with the bottom layer through service interface enabling data flow.



Protocol Multiplexing

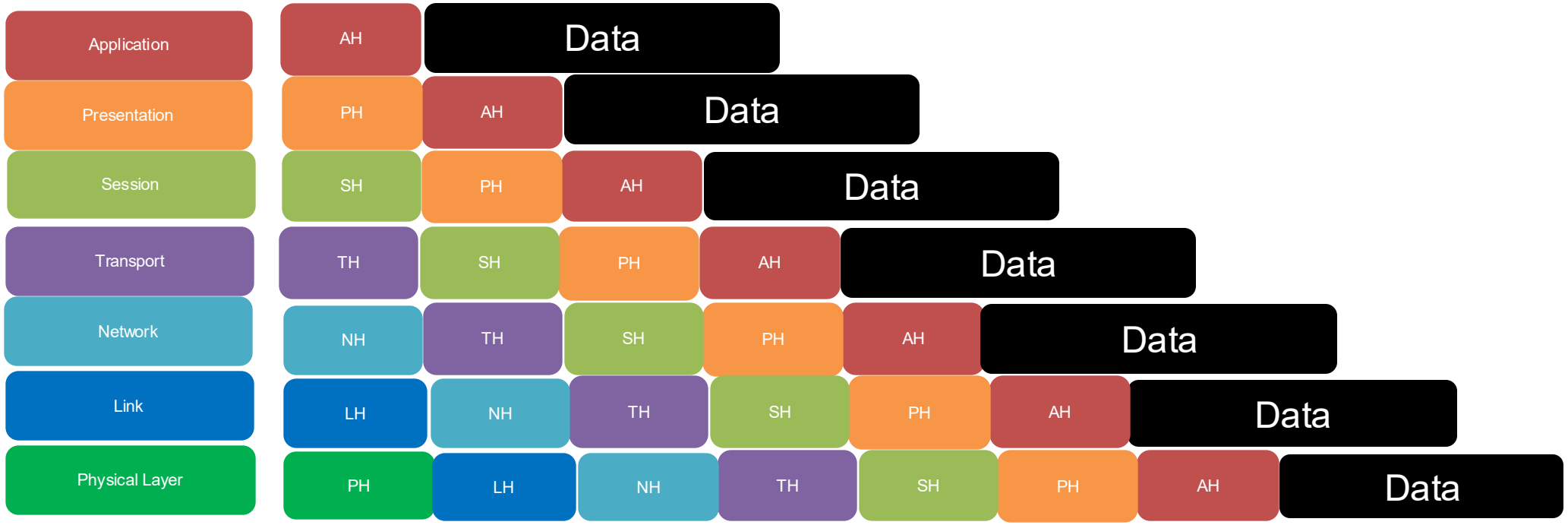
Multiplexing: Multiple overlying protocols can use same underlying protocol. On the receiving end how do we de-multiplex the messages so that the underlying layer can decide which overlying layer receives messages.



https://www.cs.princeton.edu/courses/archive/spring18/cos463/lectures/L02-transport1.pdf

Solution

Protocol Headers: Each layer attaches a tag or a label with a fixed format. Higher layers do not see tags from lower layers which they remove before forwarding the payload upwards.



https://www.cs.princeton.edu/courses/archive/spring18/cos463/lectures/L02-transport1.pdf

T3: Fundamentals of Networking

T3.1. OSI vs. TCP/IP

T3.2. Layered Architecture for LAN, MAN, and WAN

T3.3. Layer Functionality, Interfaces, and Services

T3.4. Logical vs Physical Interconnects

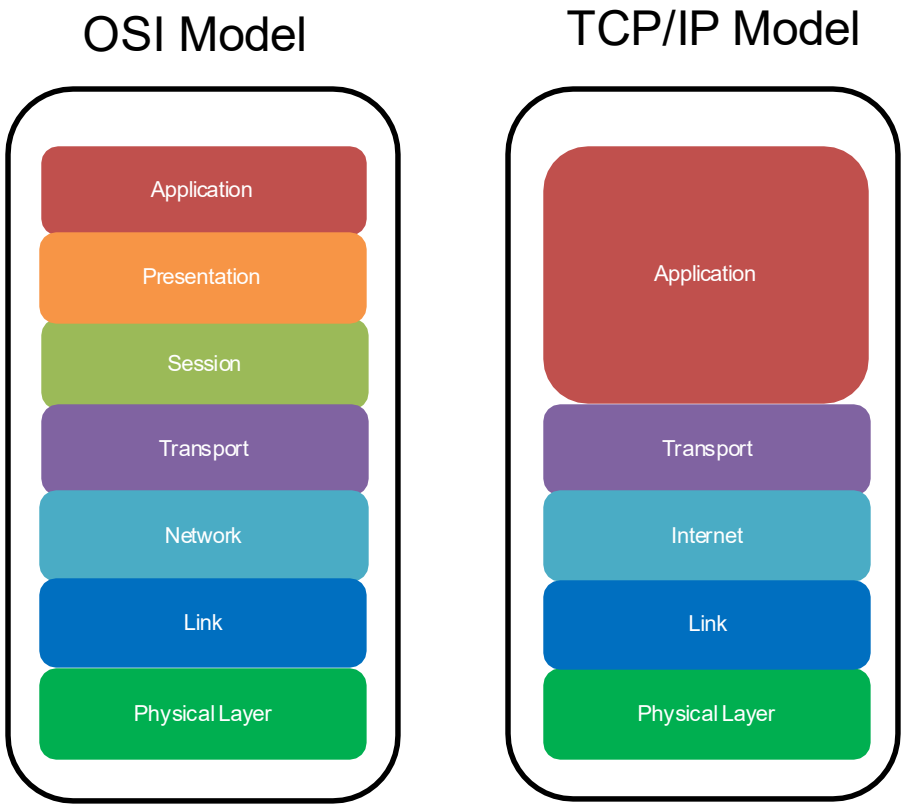
T3.5. Networking Devices: Router, Switches, and Hubs

T3.6. End-to-End Delays

T3.7. Multiplexing: Circuit and Packet Switching

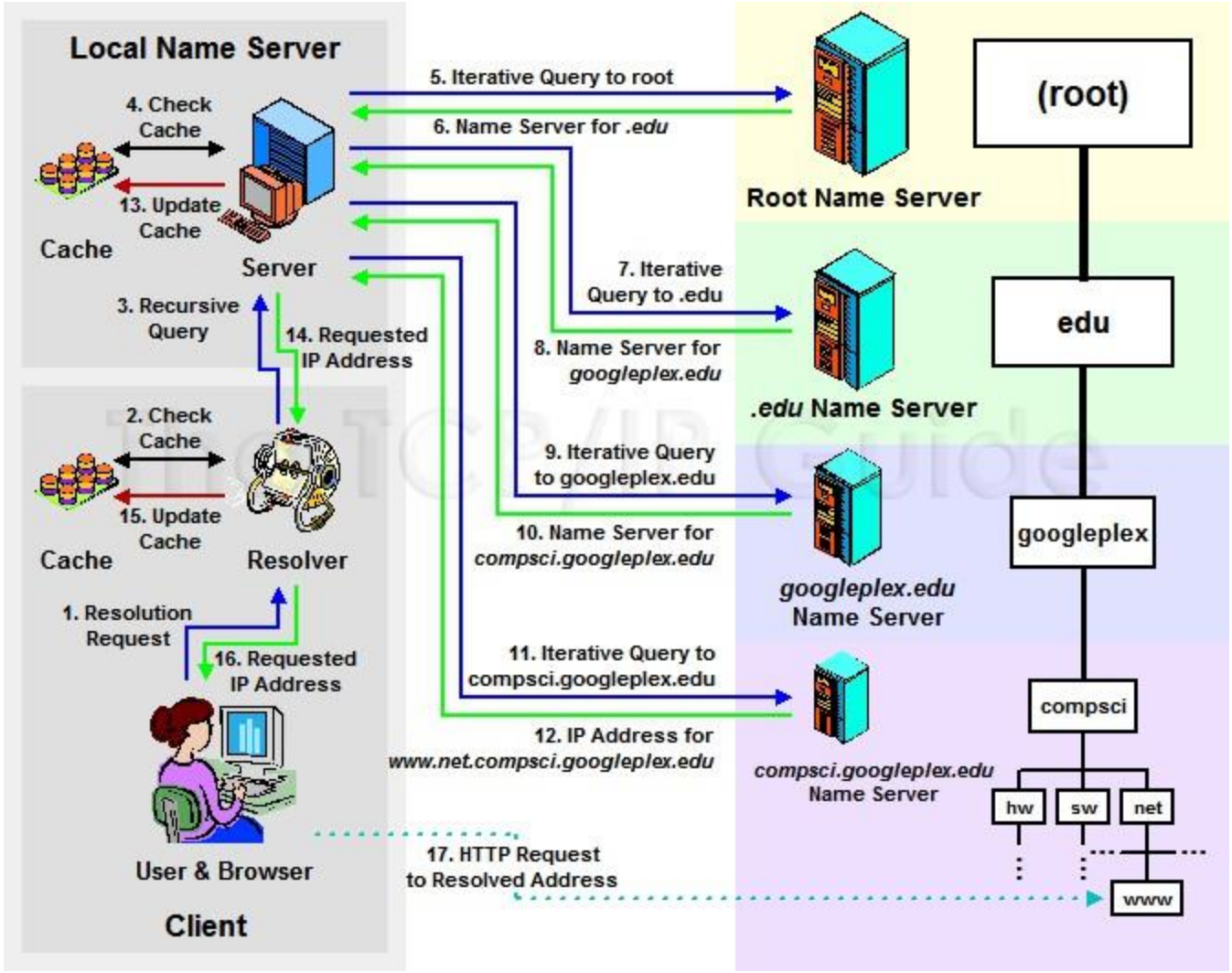
OSI vs. TCP/IP

OSI	TCP/IP
OSI is a protocol agnostic reference architecture.	TCP/IP is based around the specific standardized transport and internet protocols.
Designed mainly from vertical interconnect perspective.	Design philosophy was more host-centric, i.e. logical and horizontal.
Separate presentation and session layers.	Merges all three layers into application domain.
Session layer maintains host sessions.	Some of the session layer responsibilities are moved across to transport layer.



Application Layer Processes

Client-Server & Domain Name System

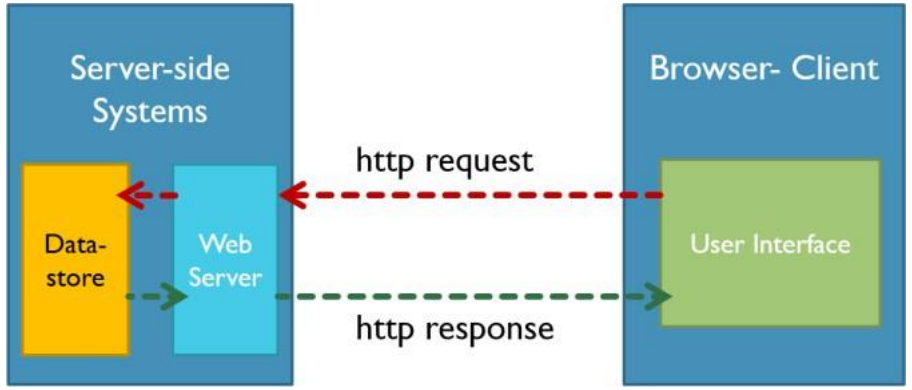


Traditional web-architecture

Courtesy: http://www.tcpipguide.com/free/t_DNSNameResolutionProcess-2.htm

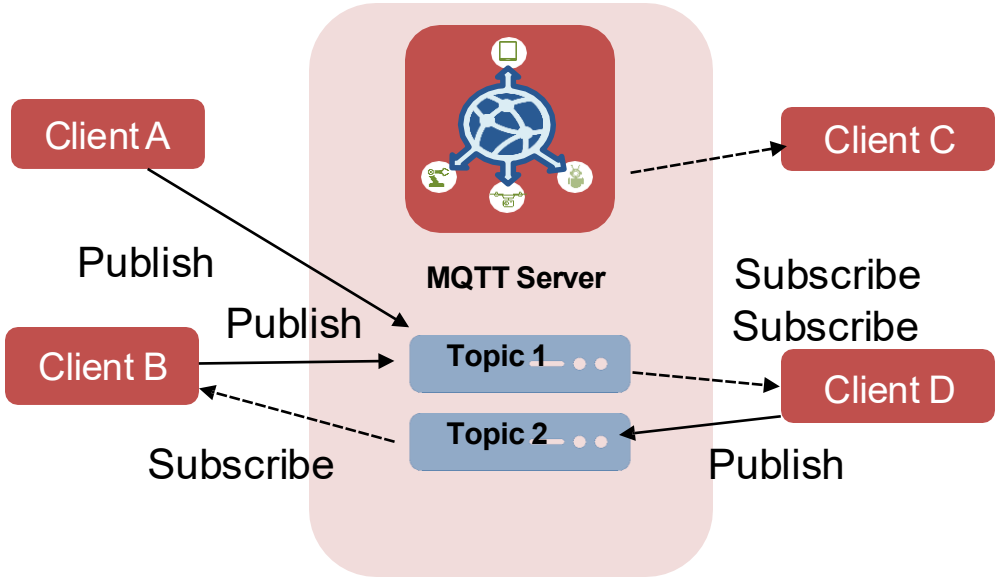
MQTT

Message Queuing Telemetry Transport (MQTT) is a publish-subscribe architecture based lightweight asynchronous messaging protocol for the machine-to-machine (M2M) communication.



HTTP

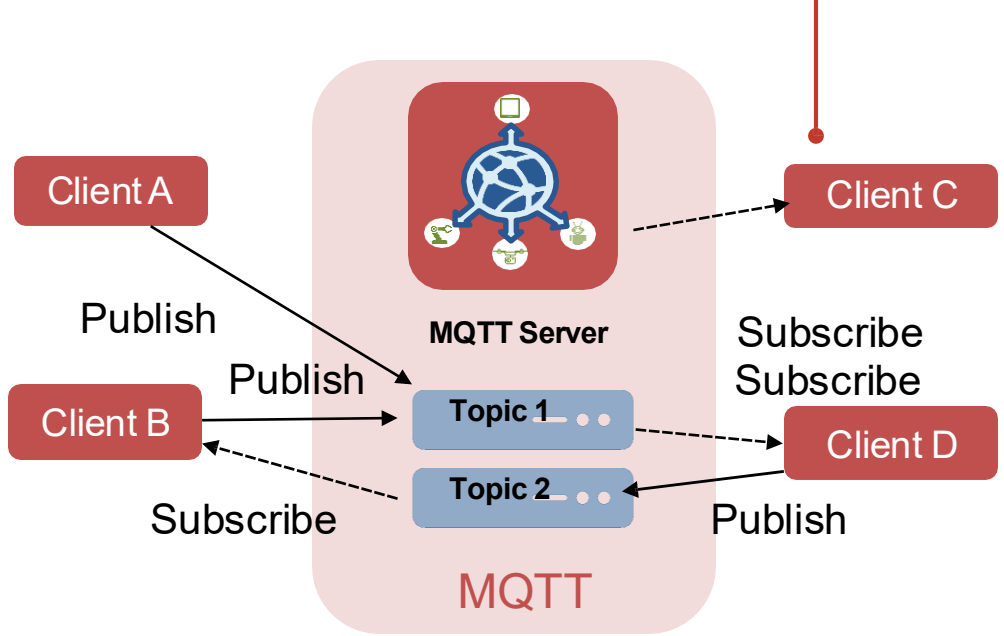
From, Data Transmission Opportunities for Collaborative Cloud-Based Building Information Modeling



MQTT

MQTT

MQTT is a part of OASIS suite, i.e., it is an open standard application layer protocol which runs over TCP and IP protocols. There is also a variant of MQTT known as MQTT-SN which is capable of operating over non TCP/IP wireless sensor networks.



Key Features of MQTT:

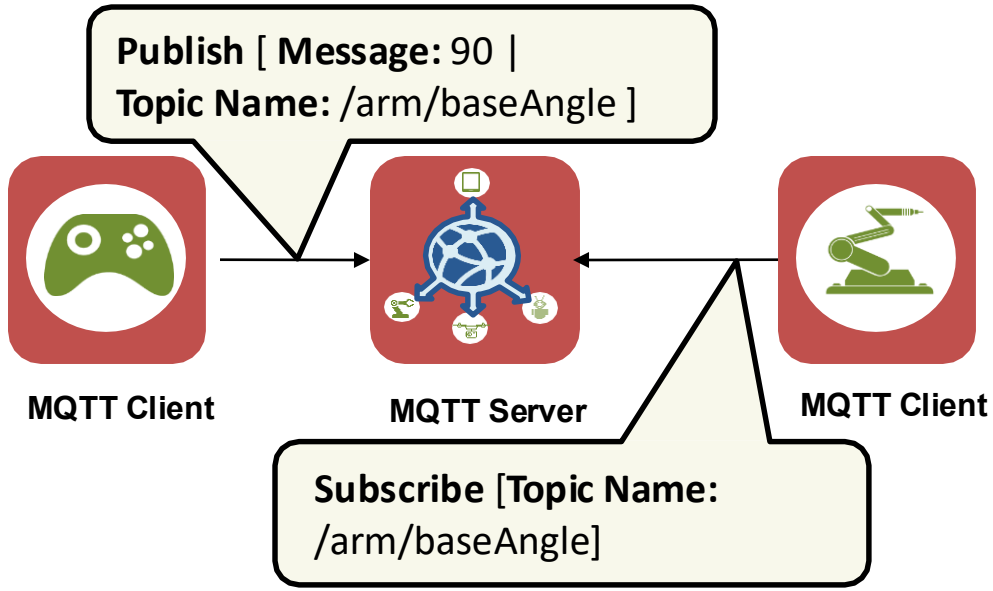
1. Provisions one-to-many communication;
2. Supports three different qualities of services for the messaging;
3. Content agnostic payload transmission, i.e. can send any form of data over MQTT;
4. It is scalable, i.e. capable of scaling from small number of devices to larger networks;
5. Optimized design for network overhead minimization;
6. Support for reporting abnormal connection termination

MQTT Architecture

MQTT

MQTT Client: An MQTT client is capable of publishing messages to or receiving messages from the MQTT Server. MQTT messages published by a client are grouped using a label called ‘Topic Name’. Clients can subscribe to one or more topics and can specify this in ‘Topic Filter’ which they provide with subscription request.

MQTT Server: An MQTT Server acts as an arbitrator between various MQTT clients. An MQTT Server receives the messages published by clients under various topics. The server subsequently forwards these messages to those clients who have subscribed to the corresponding topic. MQTT server is often also known as ‘MQTT Broker’.



Topic Naming Conventions

MQTT servers are responsible for routing the published messages from clients to subscribing clients based on their subscription topic list.

- MQTT protocol employs topic names for routing messages.
- MQTT topics follow hierarchical structure for strings separated by a forward slash (‘/’) which represents levels within hierarchy.
- Levels within the hierarchy organize topic space into a tree like structure.
- The structure of MQTT topic is similar to a file path which is expressed as a string sequence with forward slash as separator between various parent folders for the file.

Listing 1.1: Example MQTT Topic Names

```
1 'myNao/arm/gripper/angle'
2 'diningroom/sensor/temperature'
3 'diningroom/sensor/ambientLight'
4 'myHusky/leftEncoder/ticks'
```

Topic Naming Conventions

Listing 1.1: Example MQTT Topic Names

```
1 'myNao/arm/gripper/angle'
2 'diningroom/sensor/temperature'
3 'diningroom/sensor/ambientLight'
4 'myHusky/leftEncoder/ticks'
```

1. Topic Names are case sensitive, i.e., temp ≠Temp1.
2. All topic names and topic filters should be at least be of a one character in length.
3. Use of space character is permissible in the topic name.
4. Leading and trailing forward slashes would create a distinct topic name.
5. Topic name must not include NULL character.
6. Topic names should not exceed 65535 bytes

Special Topic Name

- There is a special reserved topic name '$SYS' which may be created by default on MQTT Broker.
- The $SYS prefix is generally employed by the MQTT Server to publish meta-topics which correspond to server specific information or information pertaining to control APIs.
- Consequently, a topic name with leading $' character is not allowed for publishing clients and are read-only topics.

```
C:\Python34\steve\mos>mosquitto_sub -h ws4  -t $SYS/# -v
$SYS/broker/version mosquitto version 1.4.9
$SYS/broker/timestamp 2016-06-08 11:40:24+0100
$SYS/broker/uptime 165 seconds
$SYS/broker/clients/total 0
$SYS/broker/clients/inactive 0
$SYS/broker/clients/disconnected 0
$SYS/broker/clients/active 0
$SYS/broker/clients/connected 0
$SYS/broker/clients/expired 0
$SYS/broker/clients/maximum 1
$SYS/broker/messages/stored 45
$SYS/broker/messages/received 6
$SYS/broker/messages/sent 271
$SYS/broker/subscriptions/count 0
$SYS/broker/retained messages/count 45
$SYS/broker/heap/current 8728
$SYS/broker/heap/maximum 12692
$SYS/broker/publish/messages/dropped 0
$SYS/broker/publish/messages/received 0
```

Topic

Steve's Internet Guide $SYS Log

Subscribing to the Topics

MQTT Wildcards:

MQTT allows two wildcards, i.e., '#' and '+' to be used in topic filter for the subscribing MQTT clients. These wildcards cannot be used in topic names from the publishing clients. The '#' is known as a multi-level wildcard while the '+' is referred to as the single level wild-card.

Sample Topics:	Single Level Wildcard (+)	Multi Level Wildcard (#)
Home/LivingRoom/DHT22/Humidity Home/BedRoom/DHT22/Temperature Home/BedRoom/DHT22/Humidity Home/Balcony/LDR/DayLight Home/Kitchen/SmokeSensor/Smoke	Topic with Wild Card: Home/+ /DHT22/Humidity Matches from Sample Topics: Home/LivingRoom/DHT22/Humidity Home/BedRoom/DHT22/Humidity	Topic with Wild Card: Home/# Matches from Sample Topics: Home/LivingRoom/DHT22/Humidity Home/BedRoom/DHT22/Temperature Home/Balcony/LDR/DayLight

Courtesy:https://iotbytes.files.wordpress.com/2017/01/singlelevelmqtt.png?w=375&h=301

Subscribing to the Topics

- Multi-level Wildcards: Multi-level wildcards A multi-level wildcard can match any number of levels within specified topic. Essentially, a multi-level wildcard represent all child nodes in the topic-hierarchy. A multi-level wildcard must be specified either on its own or at a certain level within the topic name following the separator, i.e. it must be the last character specified by the topic filter. For instance, 'robot/motors/#' would subscribe to all of the following topics. Notice that subscription '#' is valid subscription and will receive all published messages, also subscription to 'robot/#' is valid and will receive all messages published under the topic name 'robot' on child levels.

Listing 1.2: Example for Multi-level wildcards

```
1 'robot/motors/dcmotorLeft'
2 'robot/motors/dcmotorRight'
3 'robot/motors/servoMotor/angle'
```

Subscribing to the Topics

Listing 1.2: Example for Multi-level wildcards

```
1 'robot/motors/dcmotorLeft'
2 'robot/motors/dcmotorRight'
3 'robot/motors/servoMotor/angle'
```

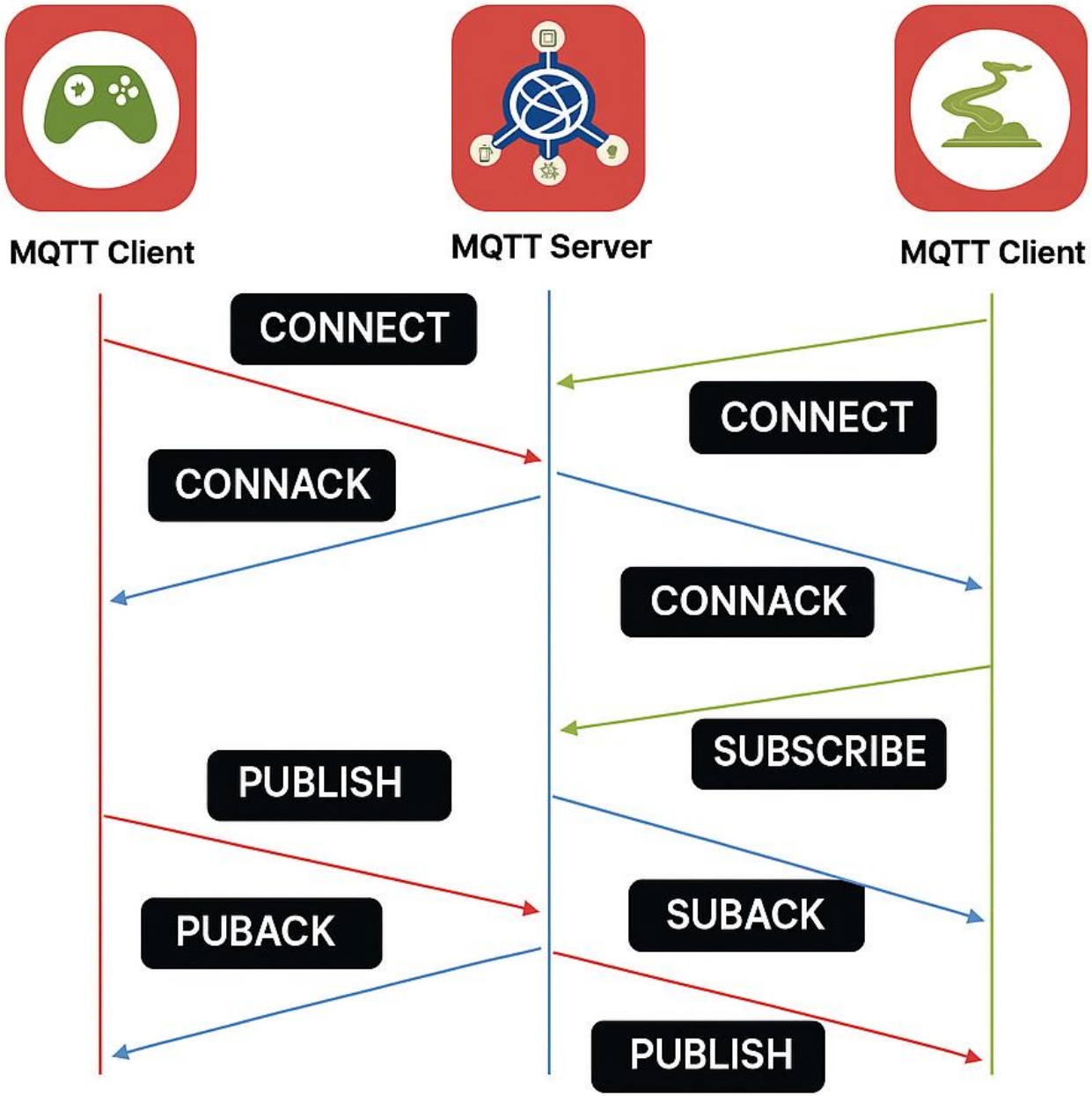
Single-level Wildcards: A single-level wildcard can only match at a particular level. Consequently, it can be used at any level with in topic filter, including the first and the last levels. It can also be repeated within the topic filter. For instance, 'robot/motors/+' is a valid topic name and would result in subscription to

Listing 1.3: Example for Single-level wildcards

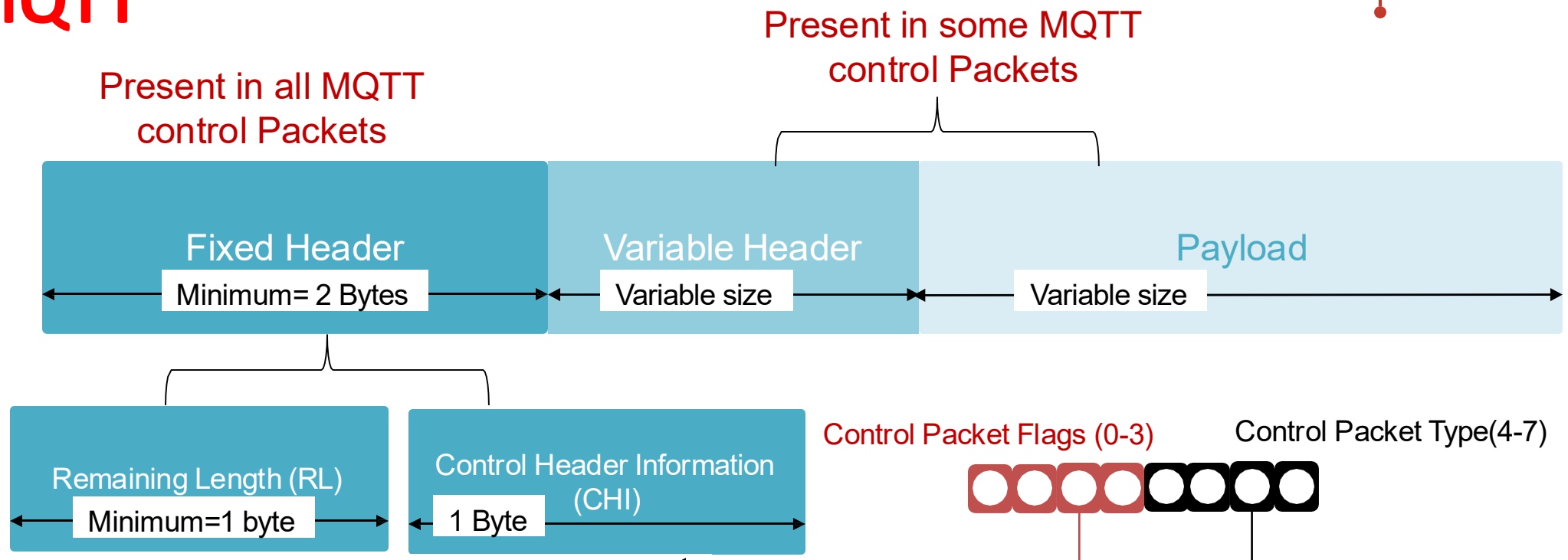
```
1 'robot/motors/dcmotorLeft'
2 'robot/motors/dcmotorRight'
3 'robot/motors/servoMotor'
```

However, it will not subscribe to the 'robot/motors/servoMotor/angle'. Notice that the server does not match the topic filters which start with wildcard character if the topic name begins with the '$' sign

MQTT Protocol



MQTT



Name	Value	Description	Direction of flow
Reserved	0	Reserved	Forbidden
CONNECT	1	Client requests to connect to Server	Client → Server
CONNACK	2	Server acknowledges the client's request for connection	Server → Client
PUBLISH	3	Message publication request	Client ⇆Server
PUBACK	4	Acknowledgment of publish request	Client ⇆Server
PUBREC	5	Publish received	Client ⇆Server
PUBREL	6	Publish released	Client ⇆Server
PUBCOMP	7	Publish complete	Client ⇆Server
SUBSCRIBE	8	Client's subscribe request	Client → Server
SUBACK	9	Server's acknowledgment of client's subscription request	Server → Client
UNSUBSCRIBE	10	Client's request for unsubscription	Client → Server
UNSUBACK	11	Server's acknowledgment of client's unsubscription request	Server → Client
PINGREQ	12	Ping request	Client → Server
PINGRESP	13	Ping response	Server → Client
DISCONNECT	14	Client disconnecting	Client → Server
Reserved	15	Reserved	Forbidden

Table 1.1: MQTT Control Packet Types and Corresponding 4 bit values.

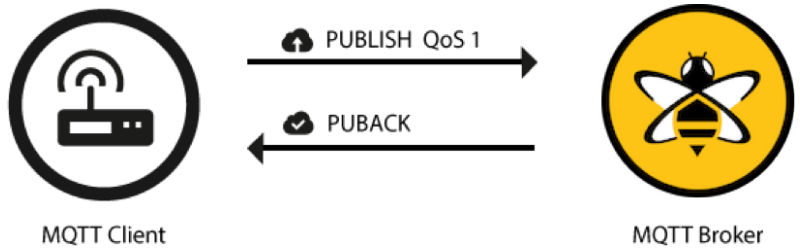
Control Packet	Status	Bit Mask
CONNECT	Reserved	0 0 0 0
CONNACK	Reserved	0 0 0 0
PUBLISH	Used in version 3.1.1	Dup QoS QoS Retain
PUBACK	Reserved	0 0 0 0
PUBREC	Reserved	0 0 0 0
PUBREL	Reserved	0 0 1 0
PUBCOMP	Reserved	0 0 0 0
SUBSCRIBE	Reserved	0 0 1 0
SUBACK	Reserved	0 0 0 0
UNSUBSCRIBE	Reserved	0 0 1 0
UNSUBACK	Reserved	0 0 0 0
PINGREQ	Reserved	0 0 0 0
PINGRESP	Reserved	0 0 0 0
DISCONNECT	Reserved	0 0 0 0

Table 1.2: MQTT Control Packet Types and Corresponding 4 bit Flag values.

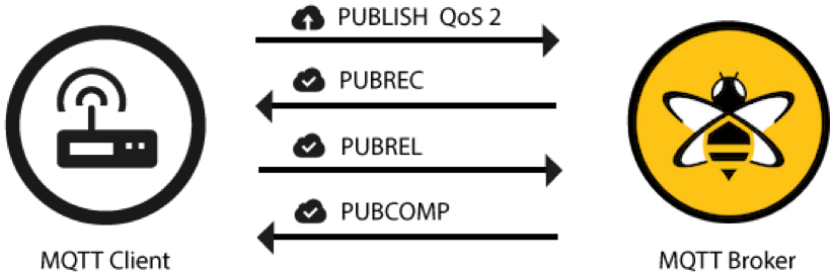
MQTT QoS

QoS 0: At most once (0) Default
QoS 1: At least once (1)
QoS 2: Exactly once (2)

QoS 0: Fire and Forget

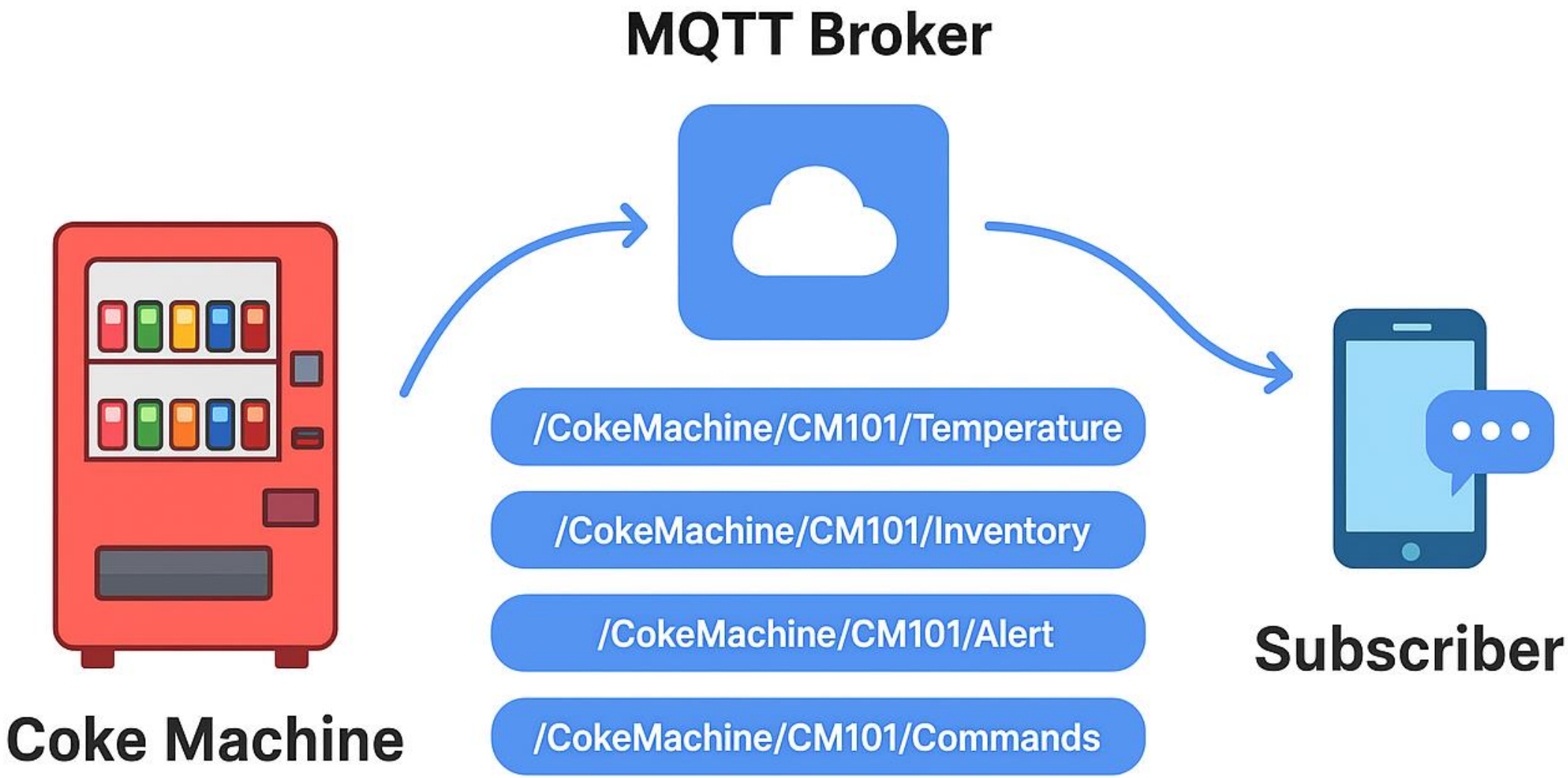


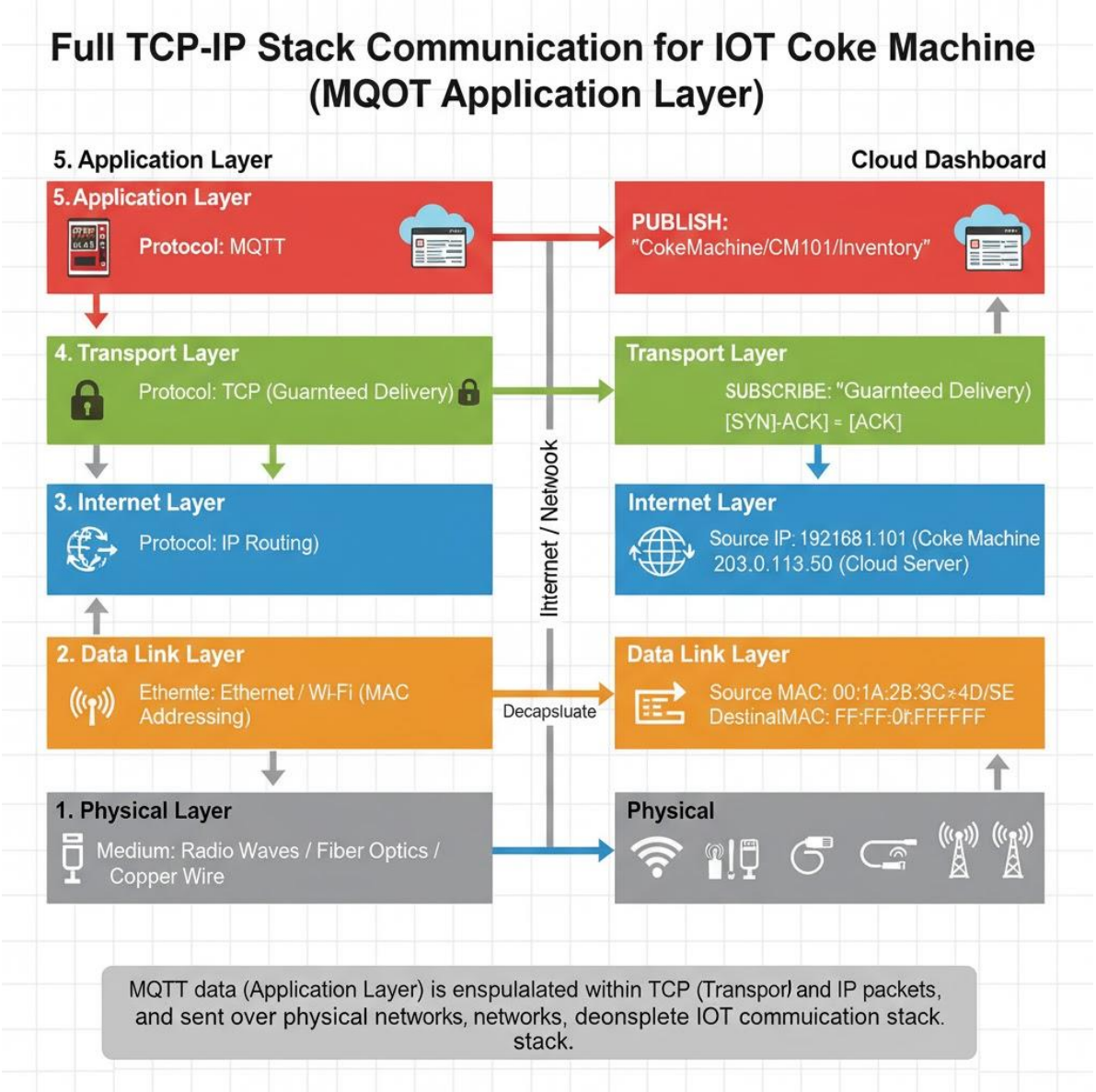
QoS 1: Repeat Until Acknowledged



QoS 2: At most once

IOT Coke Machine Using MQTT

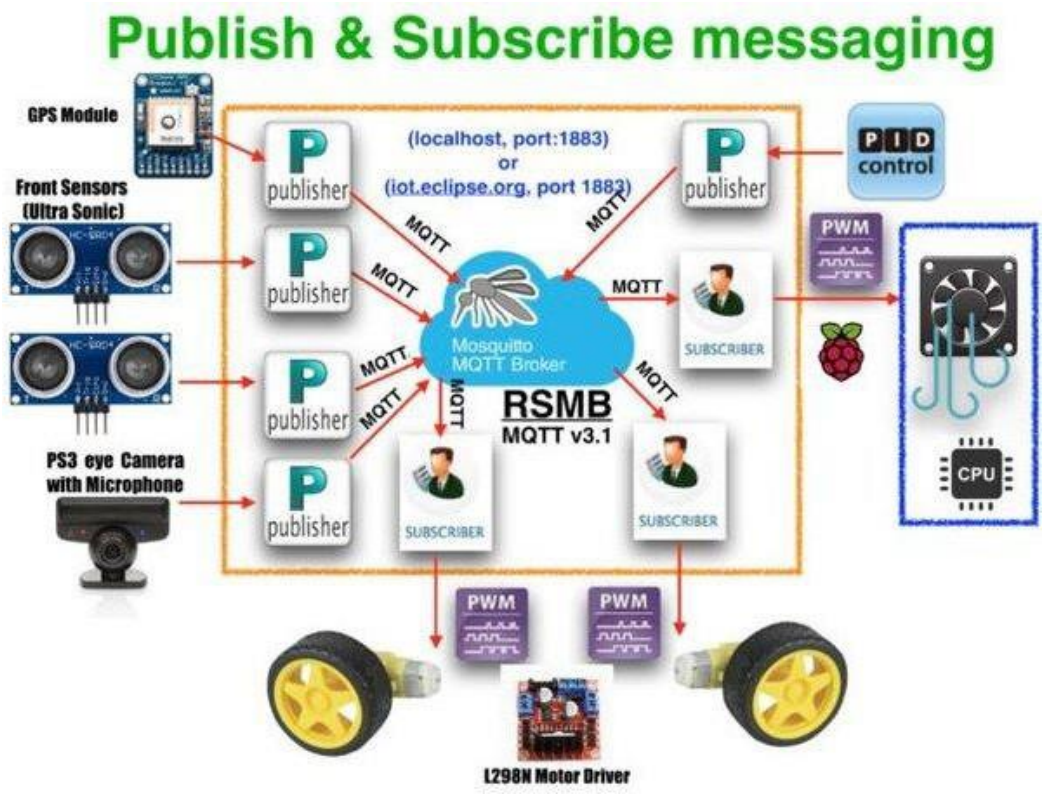



Full TCP-IP Stack Communication for IOT Coke Machine (MQOT Application Layer)
5. Application Layer
Cloud Dashboard
5. Application Layer
Protocol: MQTT
PUBLISH:
"CokeMachine/CM101/Inventory"
4. Transport Layer
Protocol: TCP (Guarnteed Delivery)
Transport Layer
SUBSCRIBE: "Guarnteed Delivery)
[SYN]-ACK] = [ACK]
3. Internet Layer
Protocol: IP Routing)
Internet / Netwook
Internet Layer
Source IP: 192168 1.101 (Coke Machine
203.0.113.50 (Cloud Server)
2. Data Link Layer
Ethernte: Ethernet / Wi-Fi (MAC Addressing)
Decapsluate
Data Link Layer
Source MAC: 00:1A:2B.'3C÷4D/SE
DestinalMAC: FF:FF:0f:FFFFFF
1. Physical Layer
Medium: Radio Waves / Fiber Optics / Copper Wire
Physical
MQTT data (Application Layer) is enspulalated within TCP (Transpor and IP packets, and sent over physical networks, networks, deonsplete IOT commuication stack. stack.

IoT Coke Machine Design

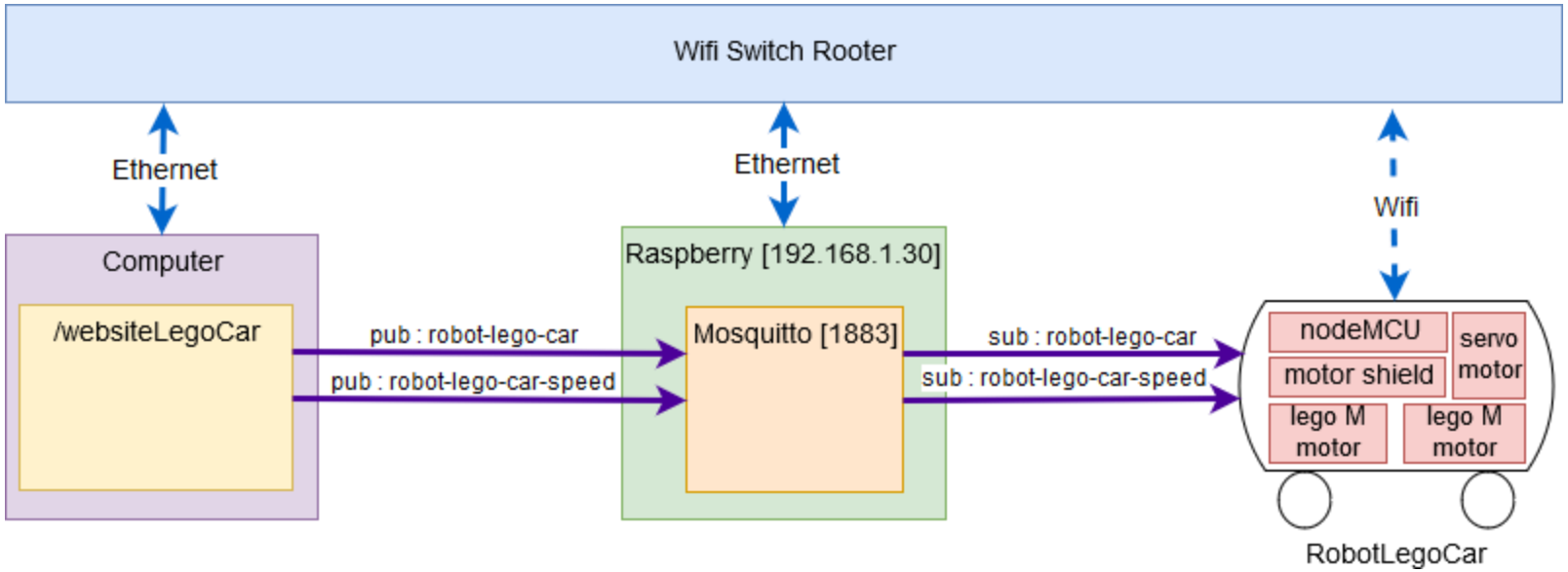
Steps	Key Focus	Your Design ideas?
Define & Sense	Requirements + sensors, what to measure.	**Requirements:** Track inventory, ensure optimal temperature (4-6°C), monitor power status, enable remote commands. **Sensors: Temperature probe** (thermistor), **Optical/Weight sensors** (for inventory count), **Power sensor** (to detect outages).
Connect & Communicate	Networking + data transport, protocols.	**Network:** Wi-Fi (preferred) or Cellular (3G/4G) for remote locations. **Protocols: MQTT** (Application Layer) over **TCP/IP** (Transport/Internet Layers) for lightweight, persistent data transport to the Cloud Broker. **DNS** for resolving the Broker's IP address.
Compute & Process	Data ingestion + logic, edge/cloud filtering.	**Edge Processing (On Machine):** Simple local logic to **Filter out redundant readings** (e.g., only send temp if it changes by $> 0.5^\circ$C). **Trigger immediate local alerts** (e.g., siren if temperature is critical). **Cloud Ingestion:** Use a managed MQTT Broker service to ingest all data.
Learn & Analyse	Analytics/ML/value creation, turn data into insight.	**Business Insight: Predictive Refill Scheduling** (ML model analyzes sales/inventory rates to predict the exact time a refill crew should be dispatched). **Anomaly Detection** (Analyze temp and power data to predict component failure *before* it happens).
Actuate & Integrate	Actions + system integration, external systems interface.	**Actuation:** Remote command from the Cloud (via MQTT) to **reboot the system** or **adjust the cooling settings**. **Integration:** Interface with an **Enterprise Resource Planning (ERP) system** for inventory management and **logistics/scheduling software** for refill crew assignments.
Operate, Maintain & Scale	Lifecycle, updates, scalability, and security.	**Security:** Use **TLS/SSL** for MQTT (MQTTS) to encrypt traffic. Implement **Client Authentication** (Client IDs/Passwords). **Scaling:** Use an auto-scaling managed MQTT Broker and a cloud backend to handle thousands of machines. **Maintenance:** Over-the-Air (OTA) firmware updates for the machine's embedded software.

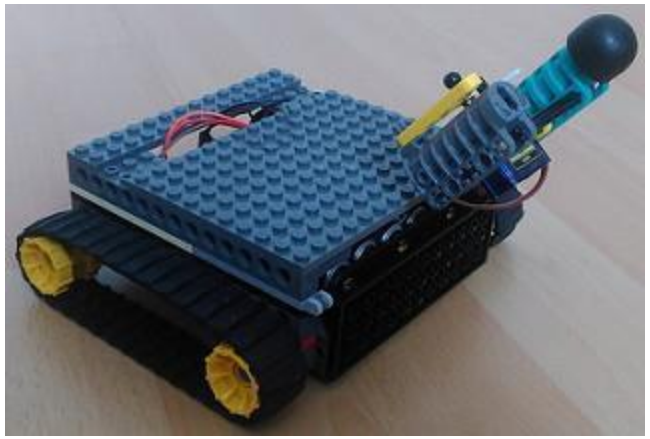
Implementation Examples

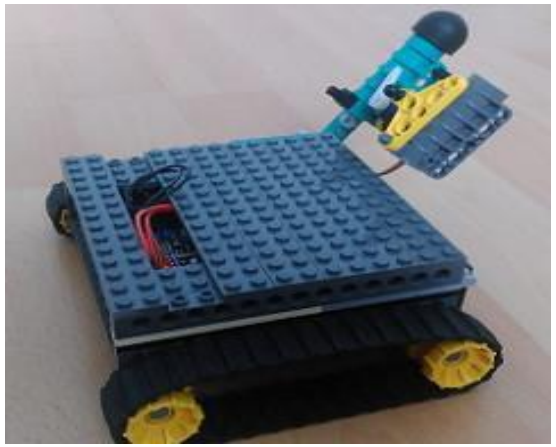


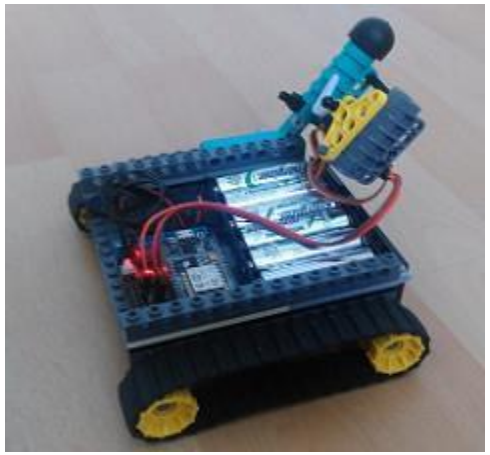
https://iotmakerblog.wordpress.com/2017/01/19/ibm-watson-cloud-robot/

Implementation Examples



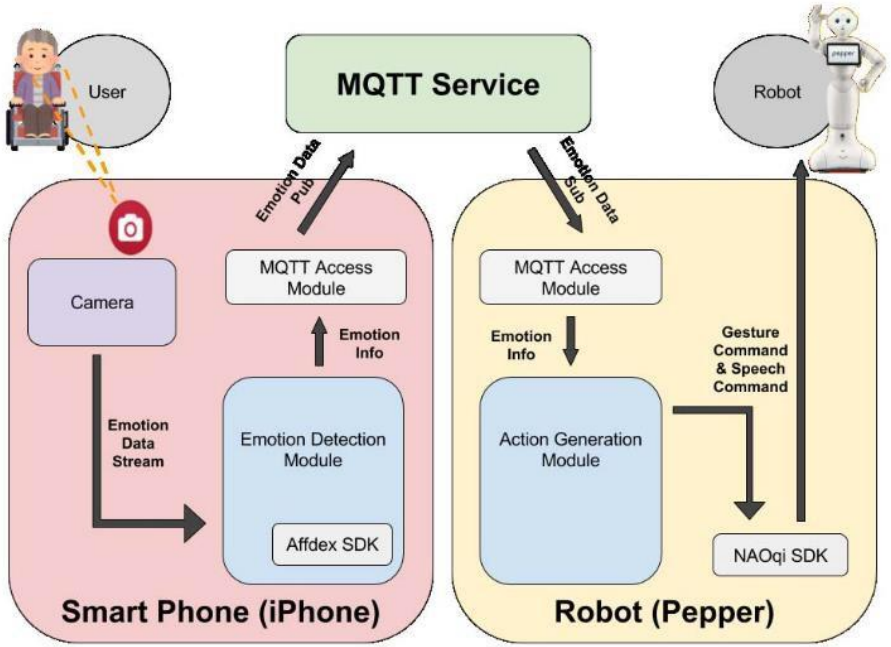


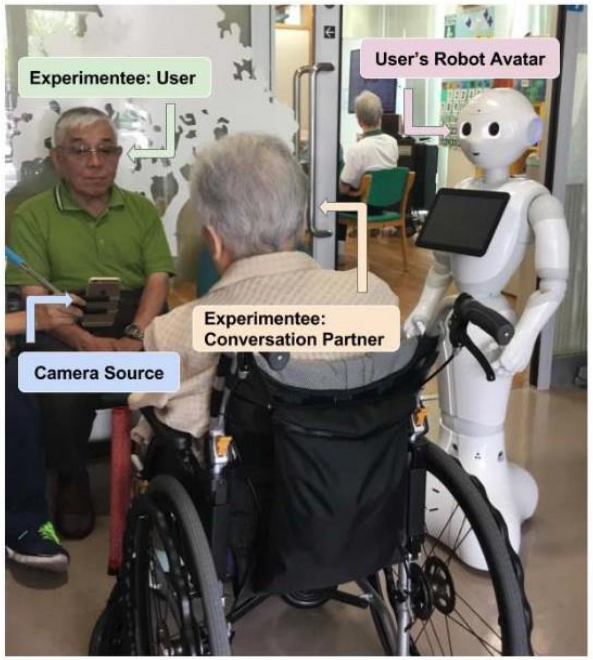




https://github.com/Clemaul/NodeMCU-MQTT-wireless-lego-robot-car

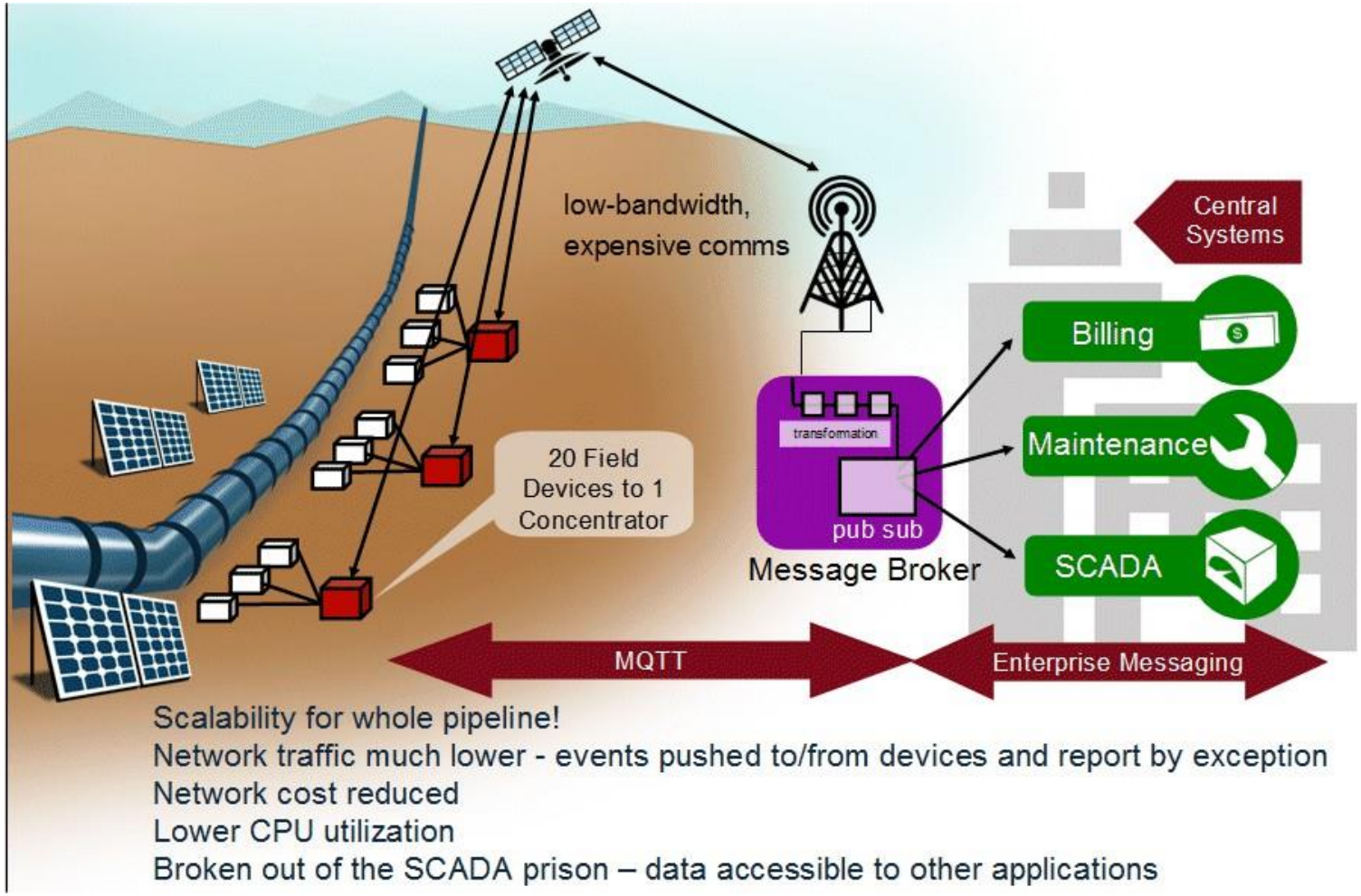
Implementation





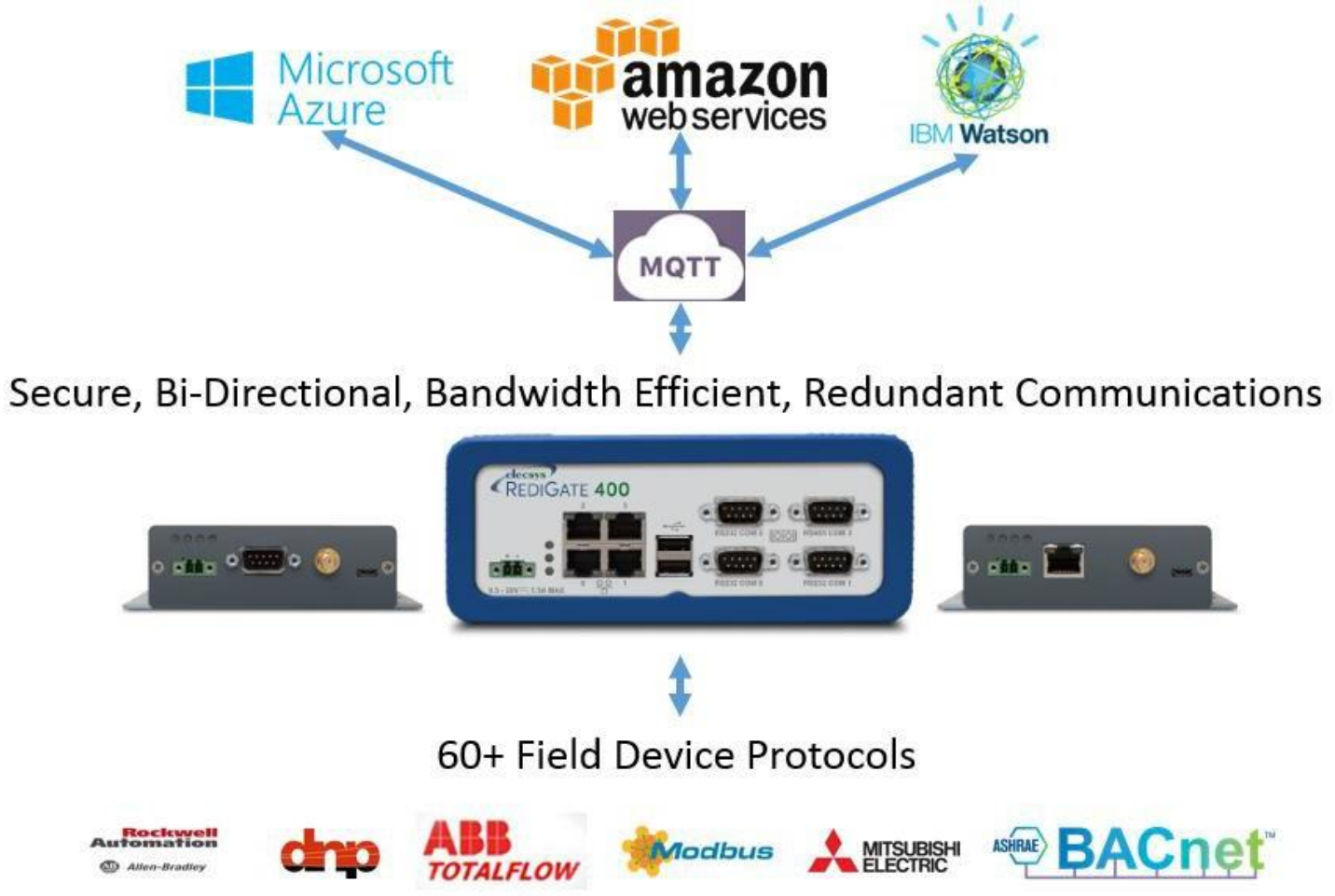
Ex-Amp Robot: Expressive Robotic Avatar with Multimodal EmotionDetection to Enhance Communication of Users with Motor Disabilities, IEEE ROMAN,2017

Implementation



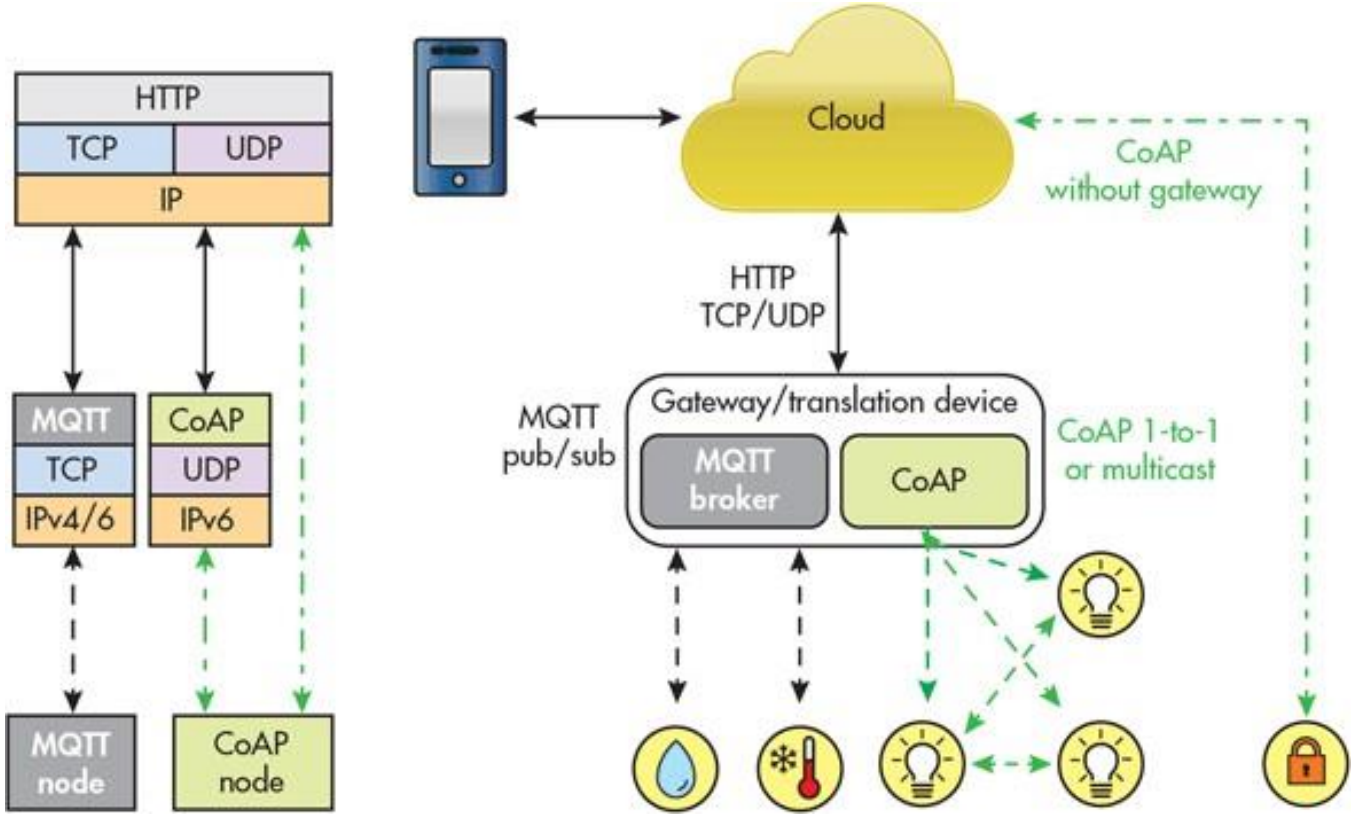
https://developer.ibm.com/messaging/2013/04/26/mqtt-enabling-internet-things/

Implementation



https://elecsyscorp.com/2017/08/09/redigate-supports-azure-aws-ibm/

Other Protocols



CoAP and MQTT

https://www.electronicdesign.com/iot/mqtt-and-coap-underlying-protocols-iot