21ES614 – Internet of Things

Sivraj P, Asst. Professor,
Dept. of EEE, Amrita School of Engineering
Amrita Vishwa Vidyapeetham

Syllabus

Unit 1

Introduction to IoT - Definitions, frameworks and key technologies. Functional blocks of IoT systems: hardware and software elements- devices, communications, services, management, security, and application. Challenges to solve in IoT

Unit 2

Basics of Networking & Sensor Networks - Applications, challenges - ISO/OSI Model, TCP/IP Model, Sensor network architecture and design principles, IoT technology stack, Communication models. Communication Protocols - Overview of protocols in each layer, Application protocols for the transfer of sensor data, Infrastructure for IoT: LoRa-Wan, 6LoWPAN, 5G and Sigfox.

Unit 3

Introduction to Cloud, Fog and Edge Computing. Modern trends in IoT – Industrial IoT, Wearable. Applications of IoT - Smart Homes/Buildings, Smart Cities, Smart Industry, and Smart Medical care, Smart Automation etc.

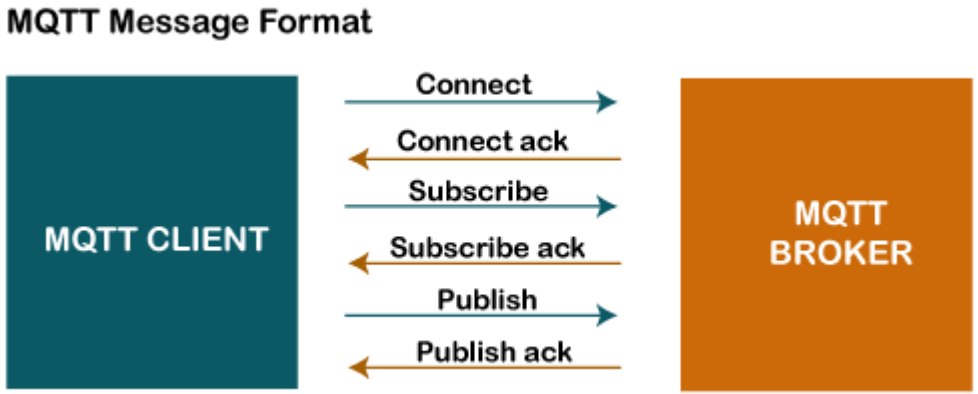
MQTT

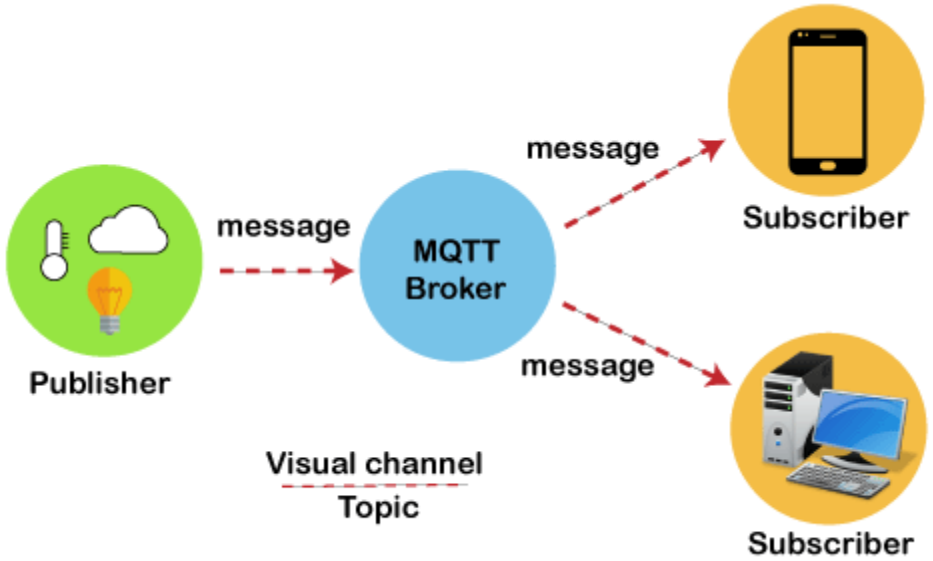
- MQTT is a standards-based messaging protocol, or set of rules, used for machine-to-machine communication.
- Lightweight and efficient
- Scalable
- Reliable
- Secure
- Well-supported

Ref: https://aws.amazon.com/what-is/mqtt/
https://www.javatpoint.com/mqtt-protocol

MQTT Architecture

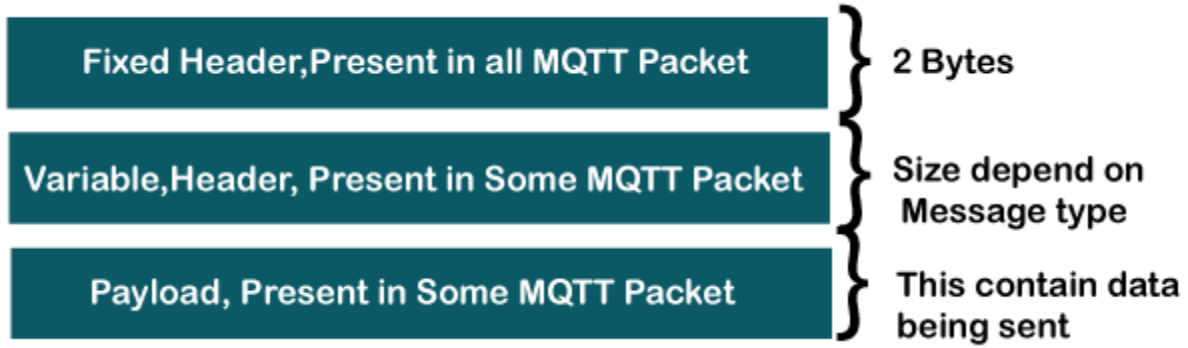
- Message
  - Payload data, Quality of Service (QoS), Collection of Properties, Topic Name
- Client
  - Publish, Subscribe
- Server or Broker
- Topic





Ref: https://aws.amazon.com/what-is/mqtt/; https://www.javatpoint.com/mqtt-protocol

MQTT Packet Structure



Fixed Header

BIT	7	6	5	4	3	2	1	0
Byte1	MQTT Control Packet Type				Flag specific to each MQTT Packet type			
Byte2...	Remaining Length							

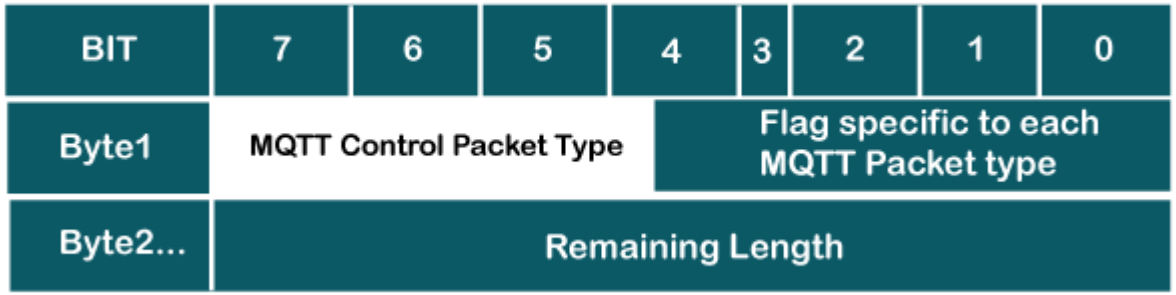
MQTT Control Packet Types

Name	Value	Direction of flow	Description
Reserved	0	Forbidden	Reserved
CONNECT	1	Client to Server	Connection request
CONNACK	2	Server to Client	Connect acknowledgment
PUBLISH	3	Client to Server or Server to Client	Publish message
PUBACK	4	Client to Server or Server to Client	Publish acknowlegment(QoS1)
PUBREC	5	Client to Server or Server to Client	Publish received(QoS2 delivery part 1)
PUBREL	6	Client to Server or Server to Client	Publish release(QoS 2 delivery part 2)
PUBCOMP	7	Client to Server or Server to Client	Publish complete (QoS 2 delivery part 3)
SUBSCRIBE	8	Client to Server	Subscribe request
SUBACK	9	Server to Client	Subscribe acknowledgment
UNSUBSCRIBE	10	Client to Server	Unsubscribe request
UNSUBACK	11	Server to Client	Unsubscribe acknowledgment
PINGREQ	12	Client to Server	PING request
PINGRESP	13	Server to Client	PING response
DISCONNECT	14	Client to Server or Server to Client	Disconnect notification
AUTH	15	Client to Server or Server to Client	Authentication exchange

Ref: https://www.javatpoint.com/mqtt-protocol

MQTT Packet Structure

Fixed Header



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		

Ref: https://www.javatpoint.com/mqtt-protocol
https://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html

MQTT Packet Structure

- Variable Header
  - Protocol name
  - Protocol version
  - Connect flags
    - Clean session flag
    - Will flag
    - Will QoS
    - Will Retain flag
    - Username & password flags
  - Keep Alive timer

Enumeration	HEX	Meaning
0	0x00	Connection Accepted
1	0x01	Connection Refused: unacceptable protocol version
2	0x02	Connection Refused: identifier rejected
3	0x03	Connection Refused: server unavailable
4	0x04	Connection Refused: bad user name or password
5	0x05	Connection Refused: not authorized
6-255		Reserved for future use

- Connect Return Code
- Topic Name

Ref: https://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html

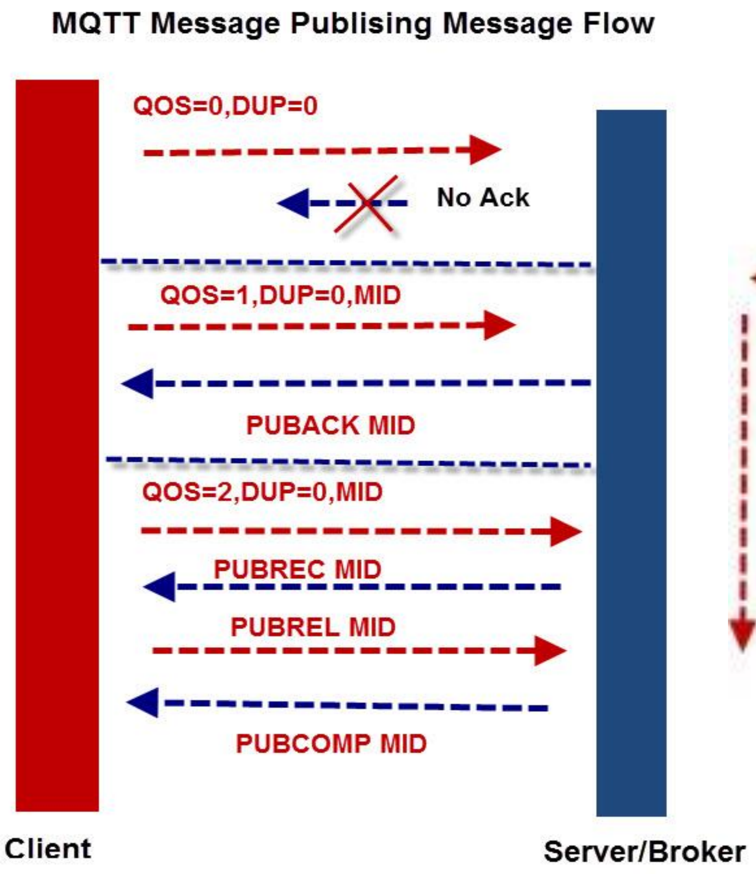
MQTT Packet Structure

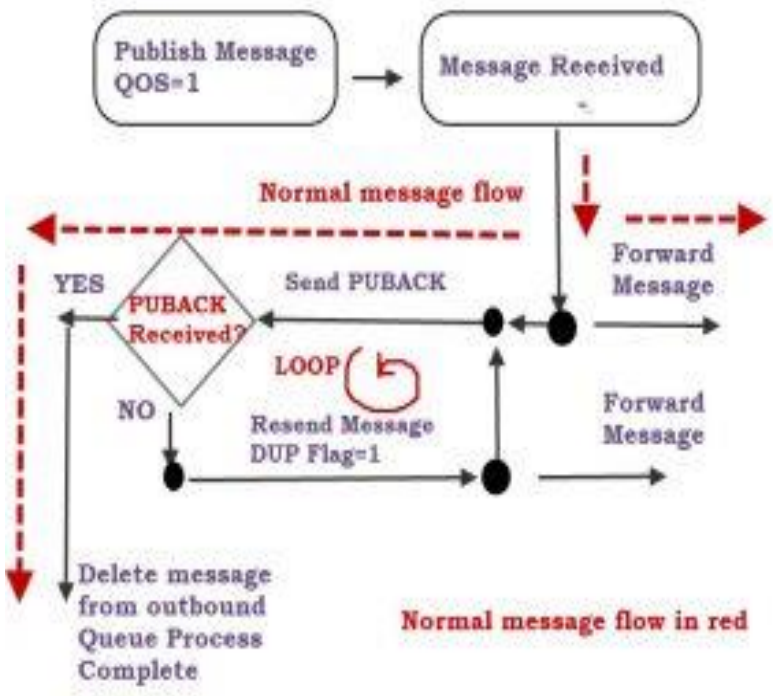
MQTT control Packet that contain Payload

MQTT Control Packet	Payload
CONNECT	Required
CONNACK	None
PUBLISH	Optional
PUBACK	None
PUBREC	None
PUBREL	None
PUBCOMP	None
SUBSCRIBE	Required
SUBACK	Required
UNSUBSCRIBE	Required
UNSUBACK	Required
PINGREQ	None
PINGRESP	None
DISCONNECT	None
AUTH	None

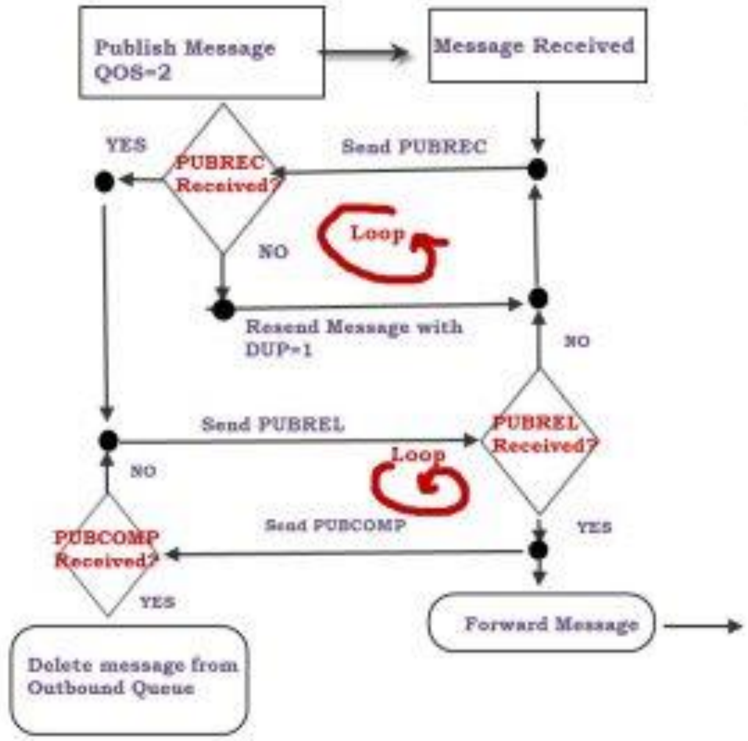
Ref: https://www.javatpoint.com/mqtt-protocol

MQTT QoS





MQTT QOS 1 Message Flow Diagram



MQTT QOS 2 Message Flow Diagram

Ref: http://www.steves-internet-guide.com/mqtt-publish-subscribe/; http://www.steves-internet-guide.com/understanding-mqtt-qos-levels-part-1/ http://www.steves-internet-guide.com/understanding-mqtt-qos-2/

MQTT Topics

- Structured using the forward slash (/)as a delimiter
- Topic names are - Case sensitive; use UTF-8 strings, must consist of at least **one character** to be valid.
- **$SYS topic -** reserved topic and is used by most MQTT brokers to publish information about the broker.
- **Wildcards**
  - **# (hash character)** – multi level wildcard
  - **+ (plus character)** -single level wildcard

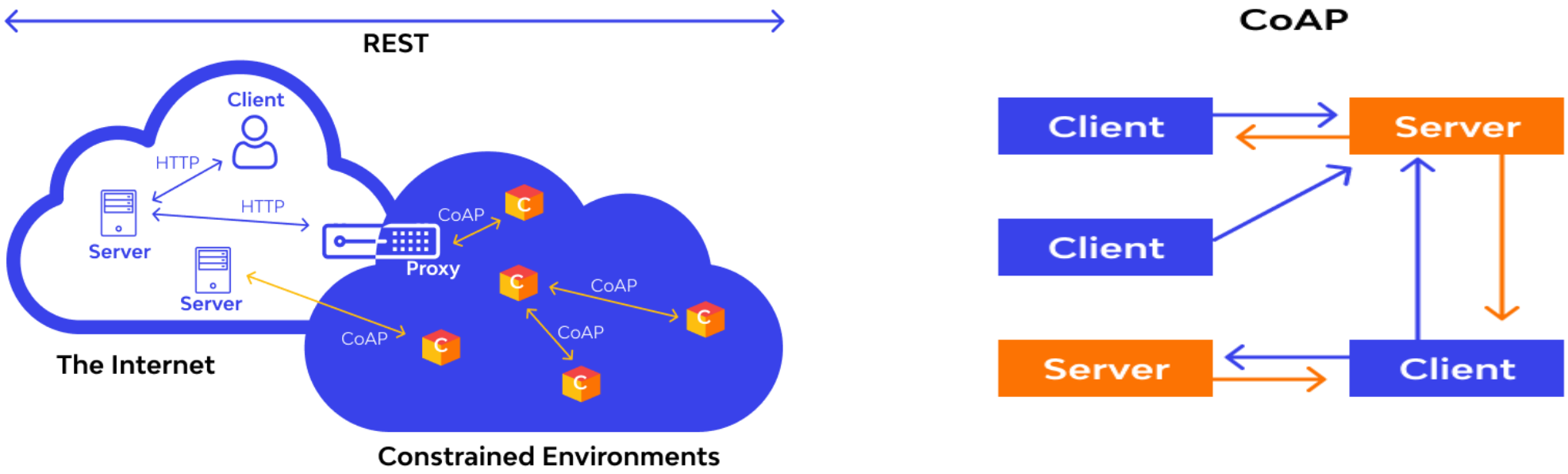
Ref: http://www.steves-internet-guide.com/understanding-mqtt-topics/

CoAP

- Constrained Application Protocol
- Web protocol used in M2M with constrained requirements
- Asynchronous Message Exchange
- Low Overhead
- Very Simple To Perform Syntactic Analysis
- Uniform Resource Identifier
- Proxy and Caching Capabilities
- Functions as sort of HTTP for restricted nodes

Ref: https://radiocrafts.com/technologies/coap-constrained-application-protocol/

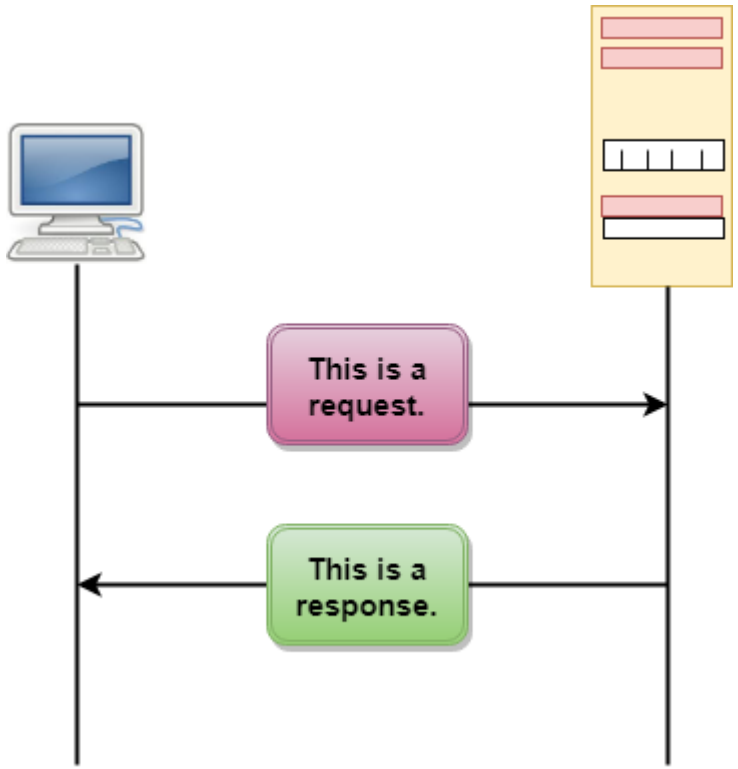
CoAP

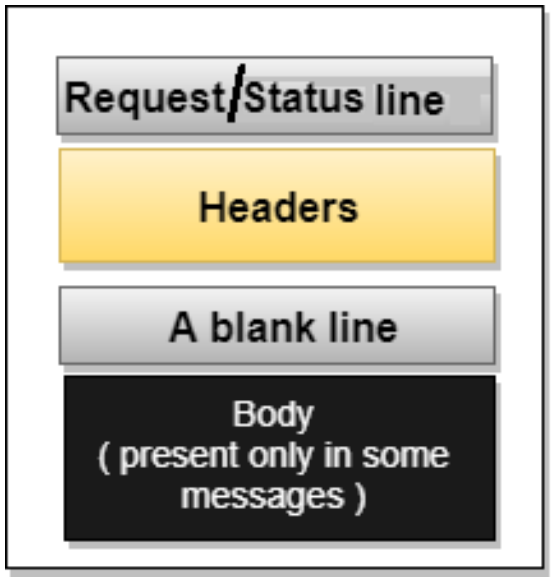


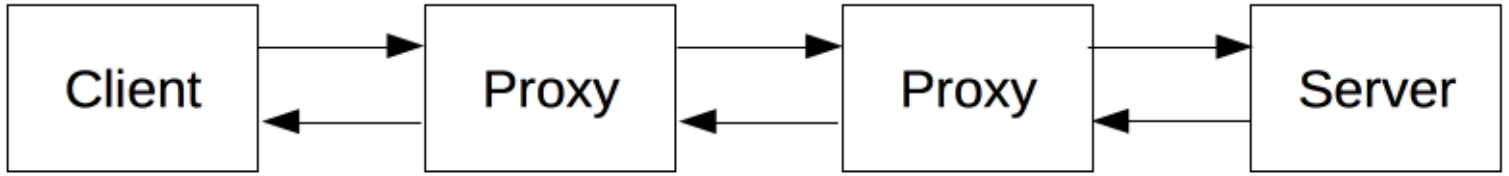
- Makes UDP transactions possible at endpoints in the confirmable (CON) or non-confirmable (NON) format.
- Every CoAP message features a distinct ID to avoid message duplications.

Ref: https://www.wallarm.com/what/coap-protocol-definition
https://www.geeksforgeeks.org/rest-api-introduction/

HTTP



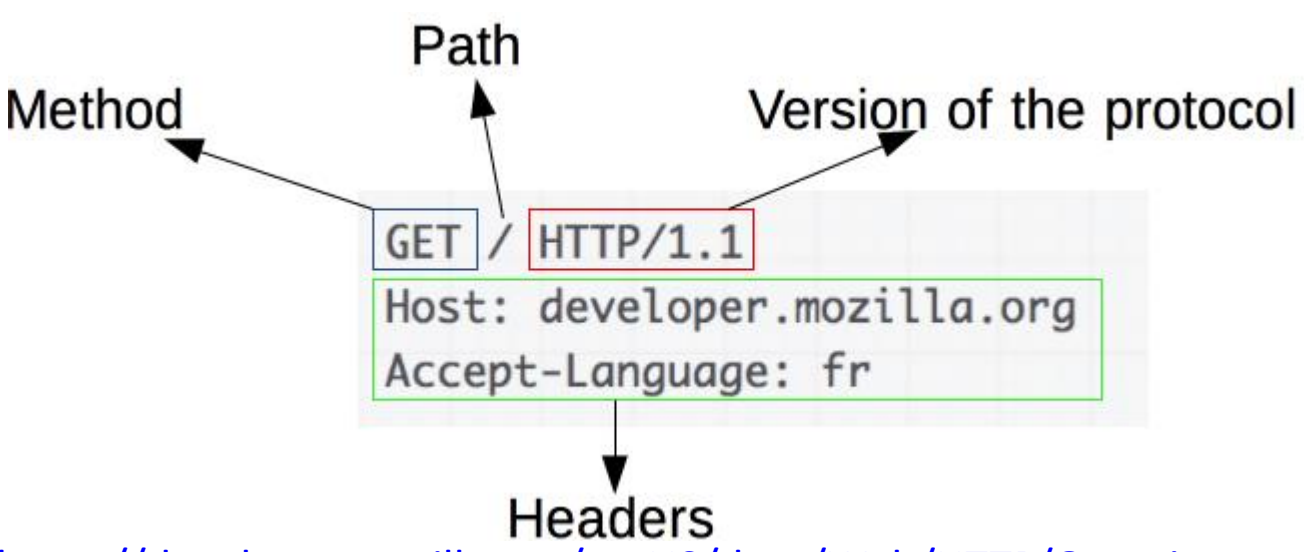


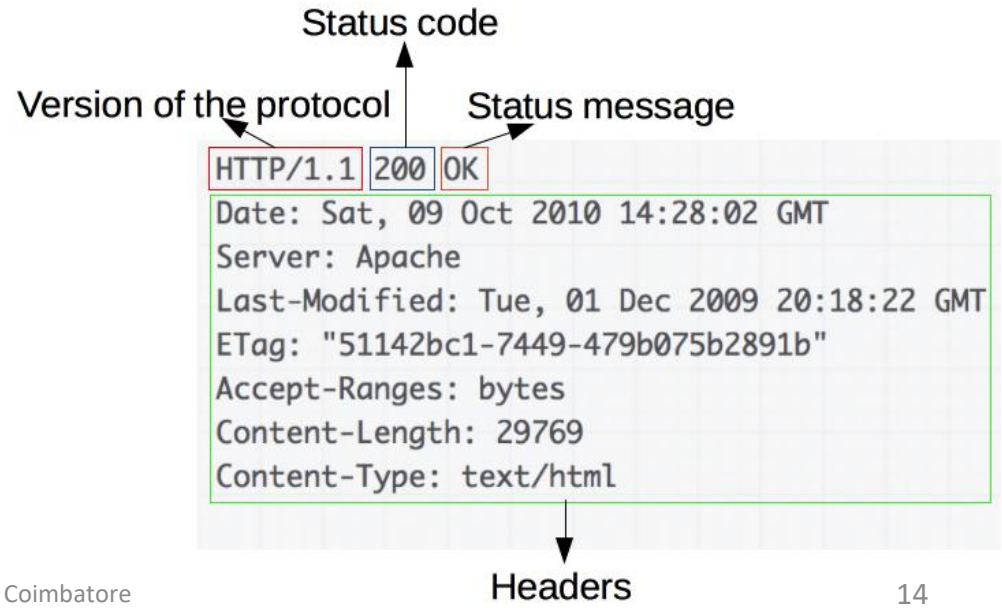


Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview
https://www.javatpoint.com/computer-network-http

HTTP

- HTTP Client flow
  - Open a TCP connection, Send an HTTP message, Read the response
- HTTP Server flow
  - Open a TCP connection, Read the HTTP message, Send the response

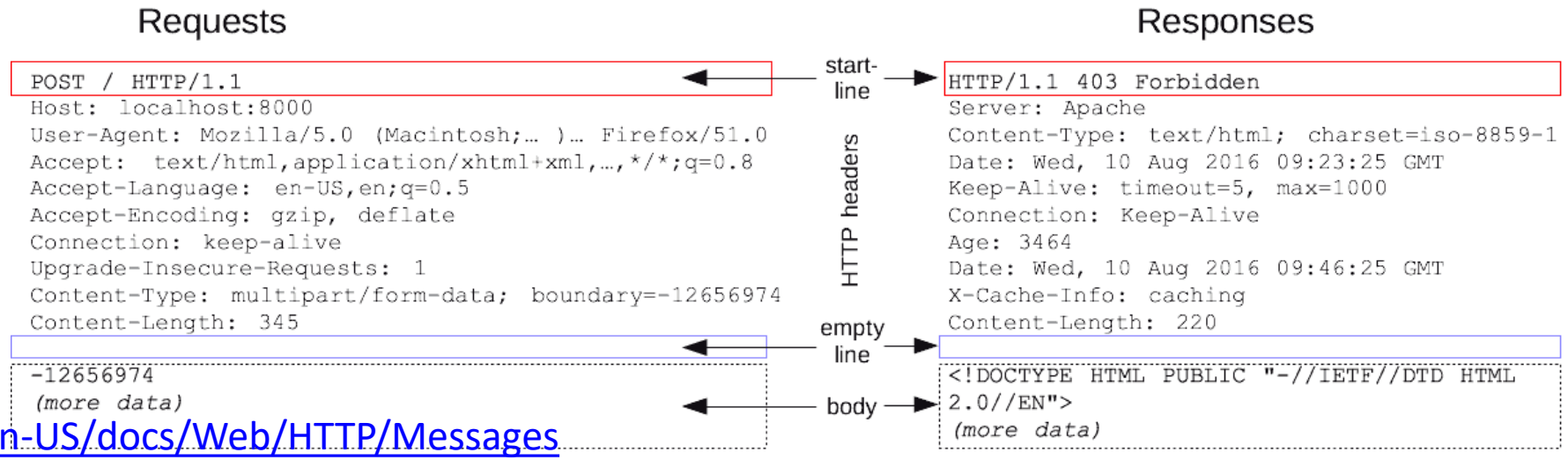




Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview
https://www.javatpoint.com/computer-network-http

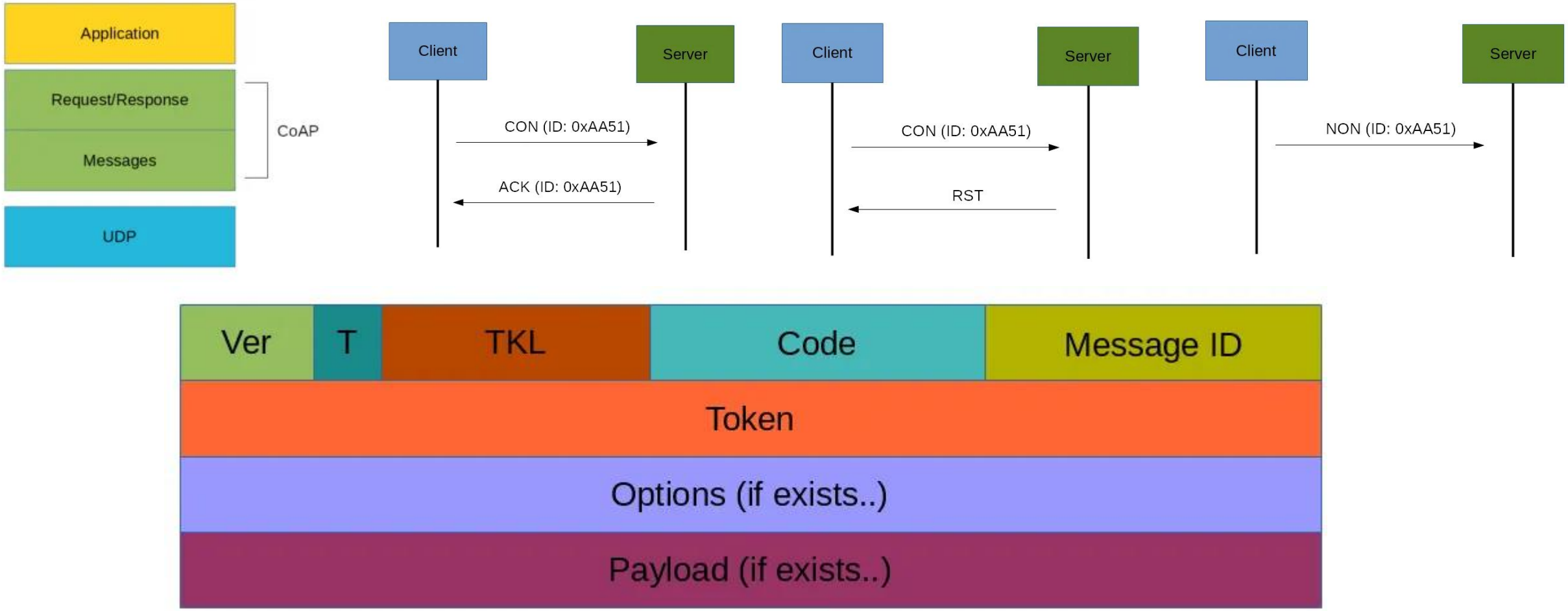
HTTP Messages

- Client – Request
  - Get, Head, Post, Put, Delete, Connect, Options, Trace, Patch
- Server – Response
  - Informational, Successful, Redirection, Client & Server Error Responses



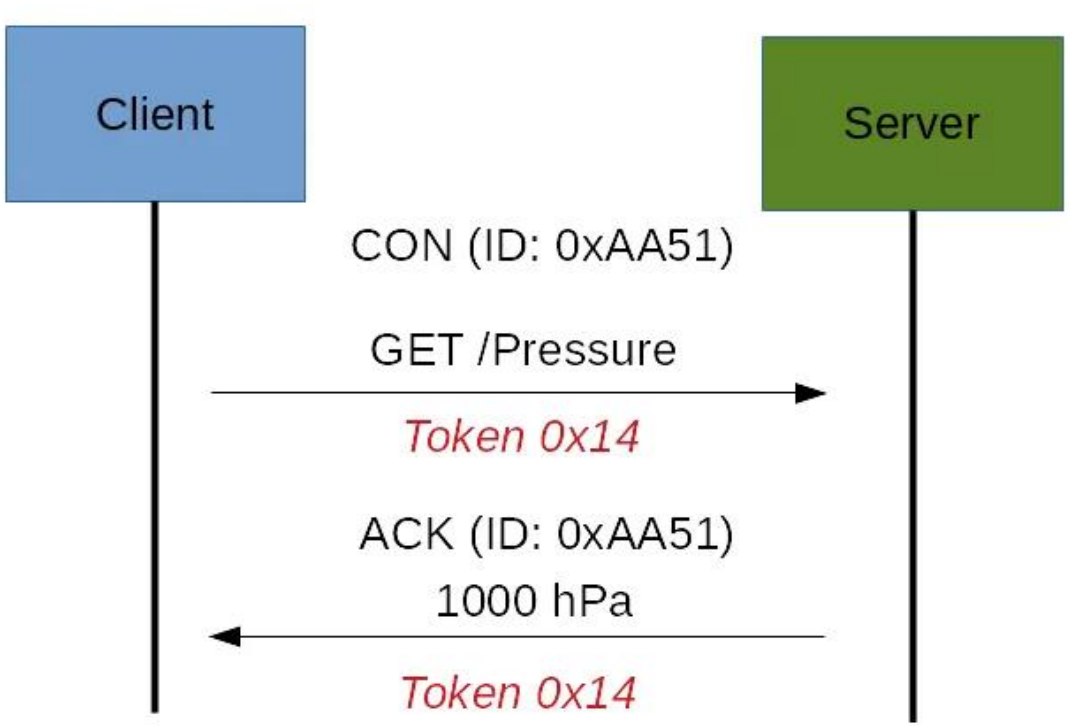
Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages
https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods
https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

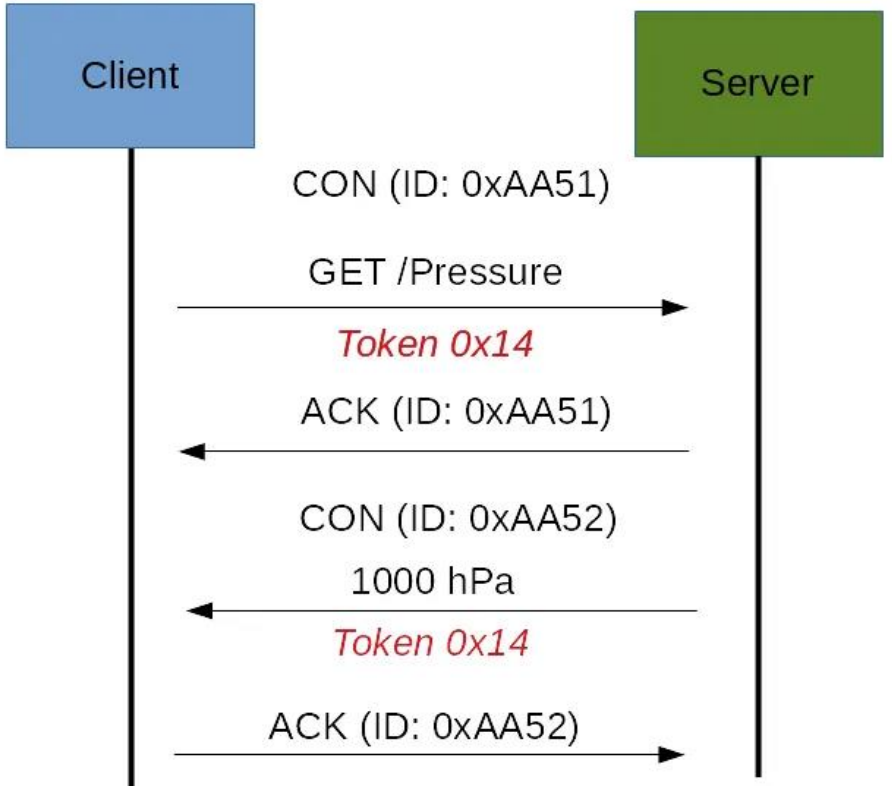
CoAP Messages



Ref: https://dzone.com/articles/coap-protocol-step-by-step-guide

CoAP Request-Response

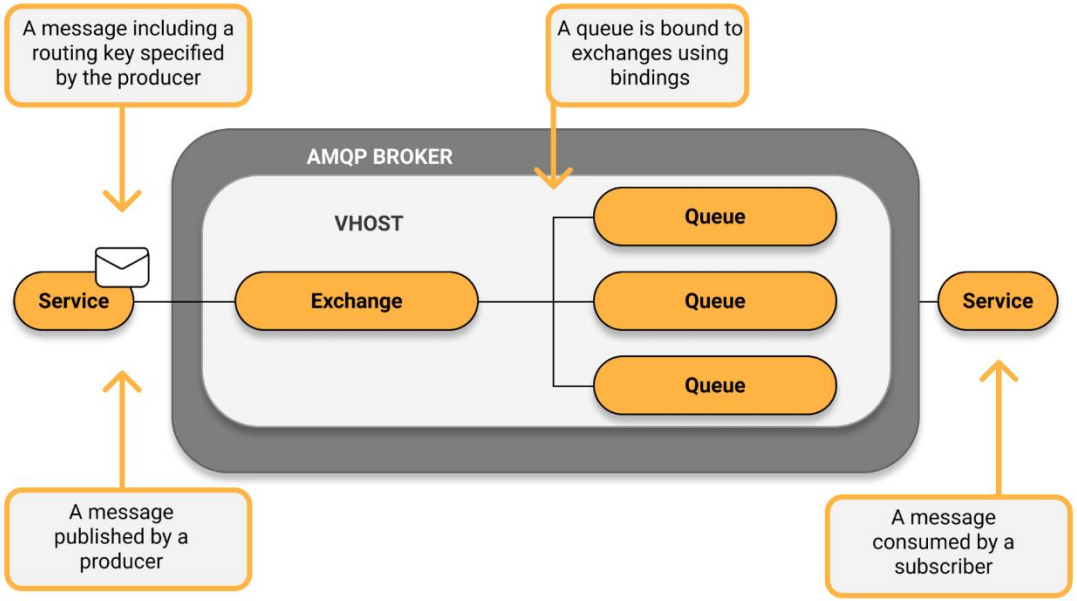


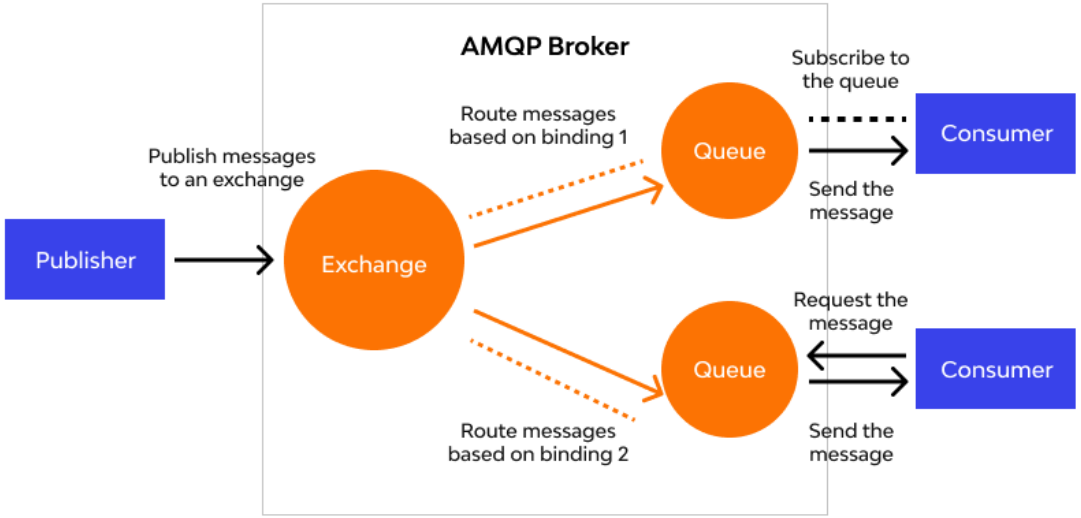


Ref: https://dzone.com/articles/coap-protocol-step-by-step-guide

AMQP

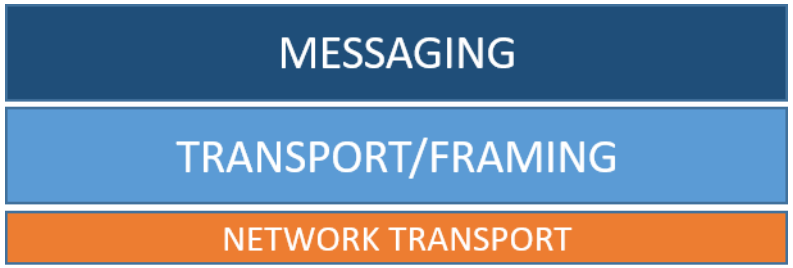
- Advanced Message Queuing Protocol

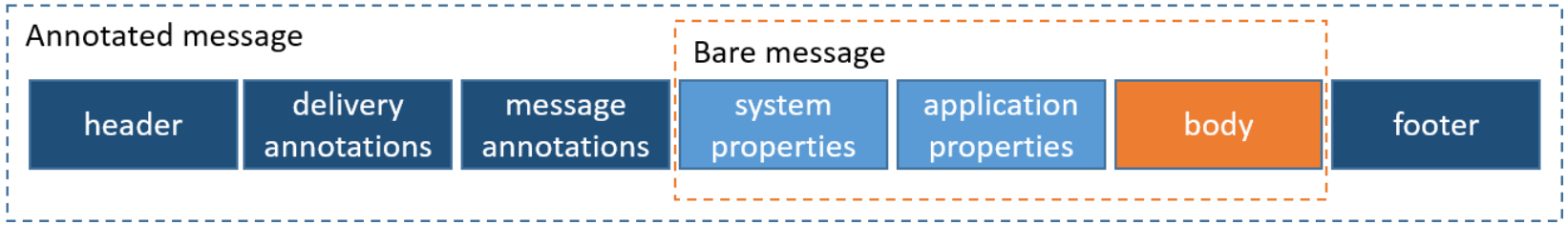




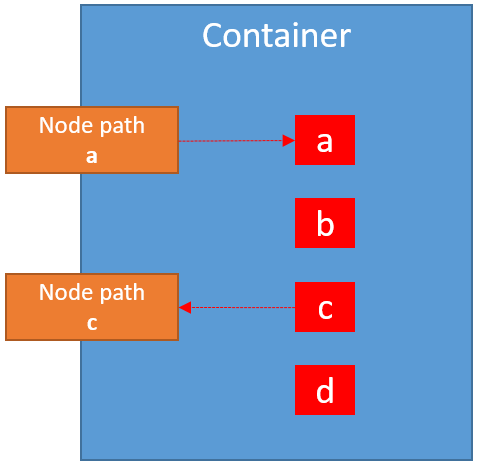
Ref: https://www.wallarm.com/what/what-is-amqp
https://www.cloudamqp.com/blog/what-is-amqp-and-why-is-it-used-in-rabbitmq.html

AMQP

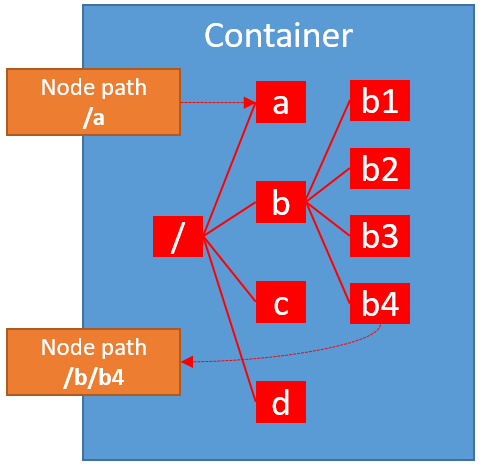


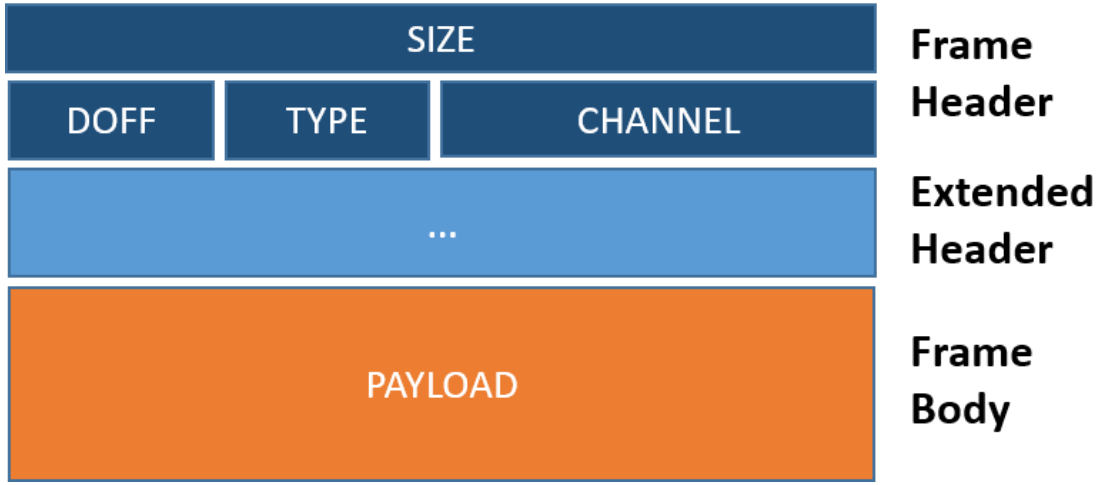


Flat Node Topology



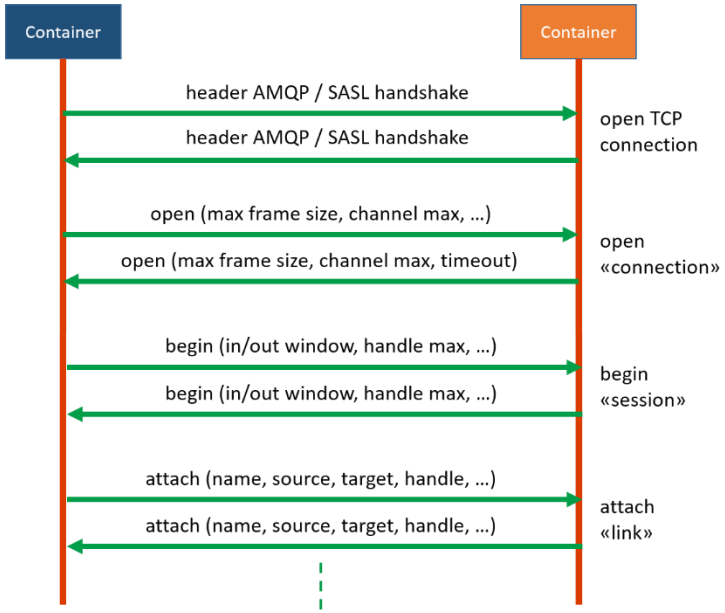
Hierarchical Node Topology

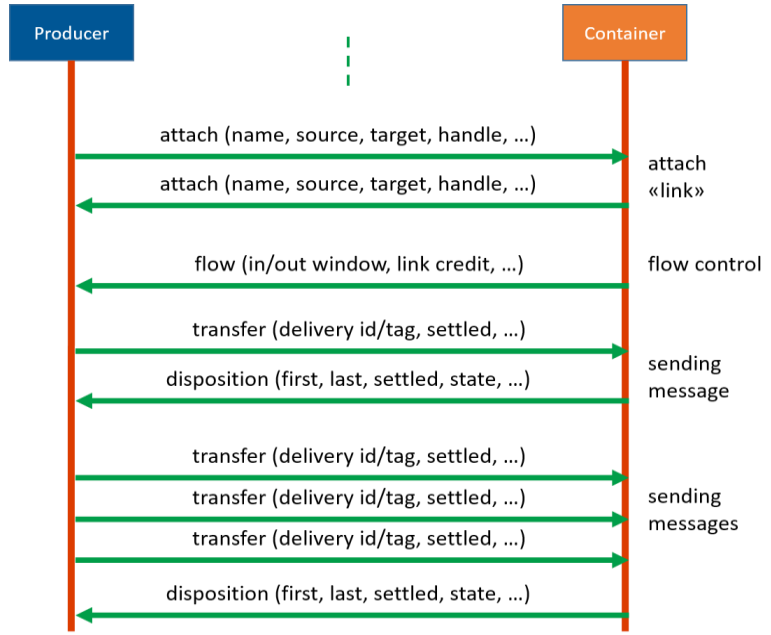


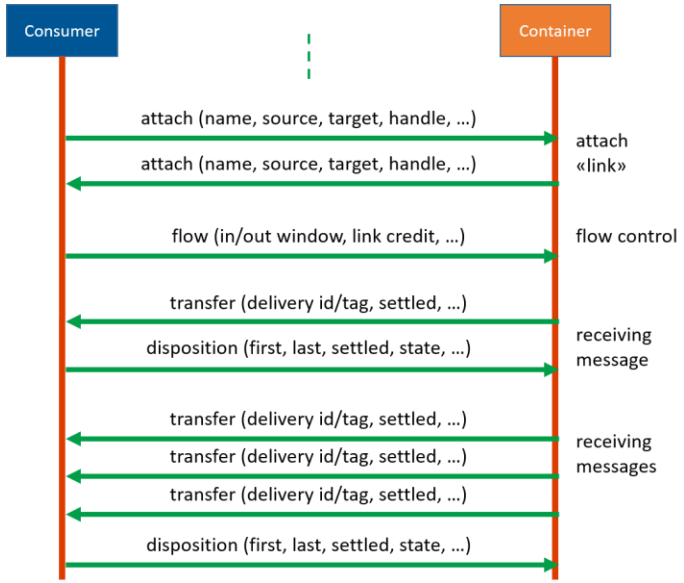


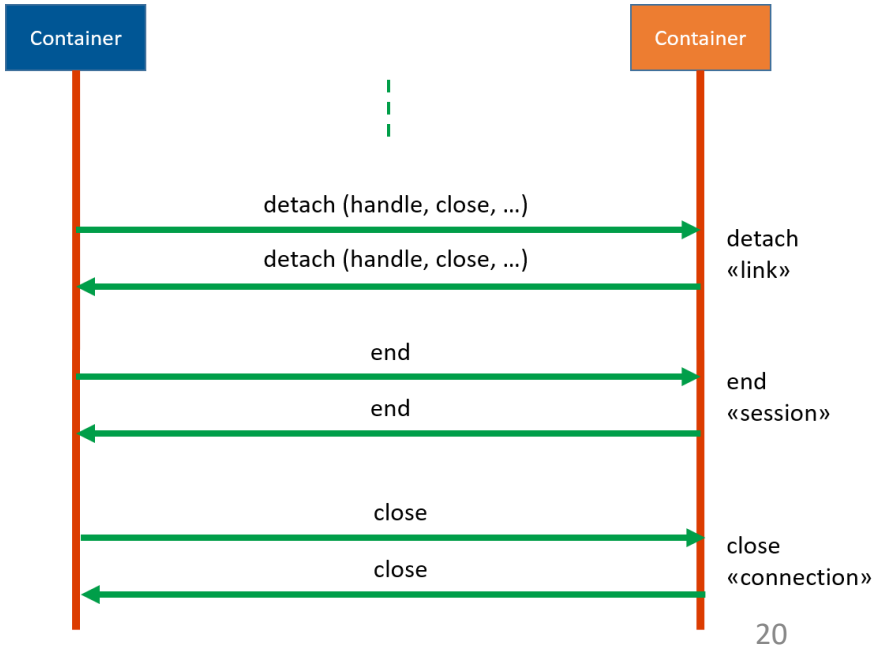
Ref: https://dzone.com/refcardz/amqp-essentials

AMQP









Ref: https://dzone.com/refcardz/amqp-essentials

Thank You…