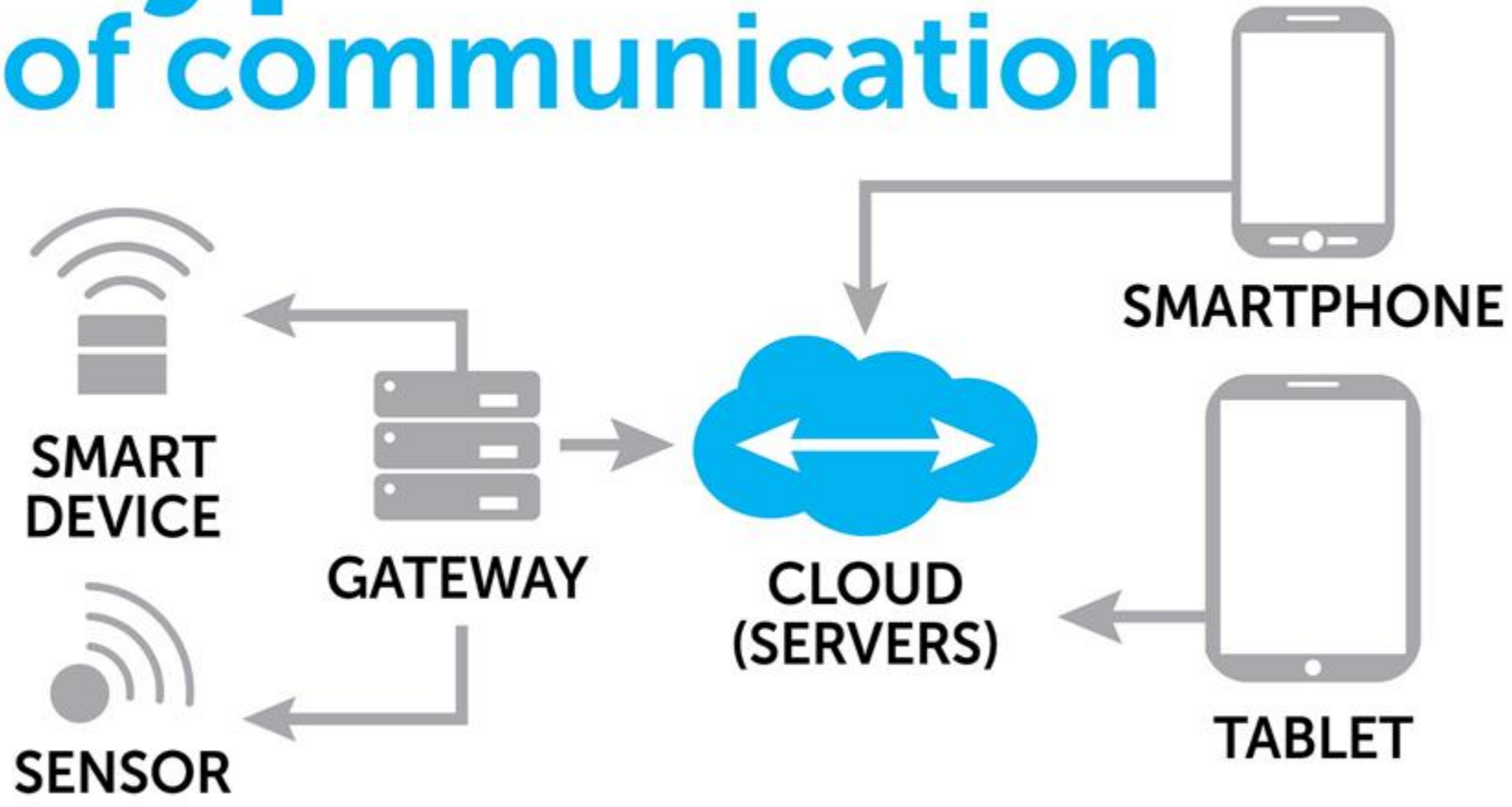
Types of communication

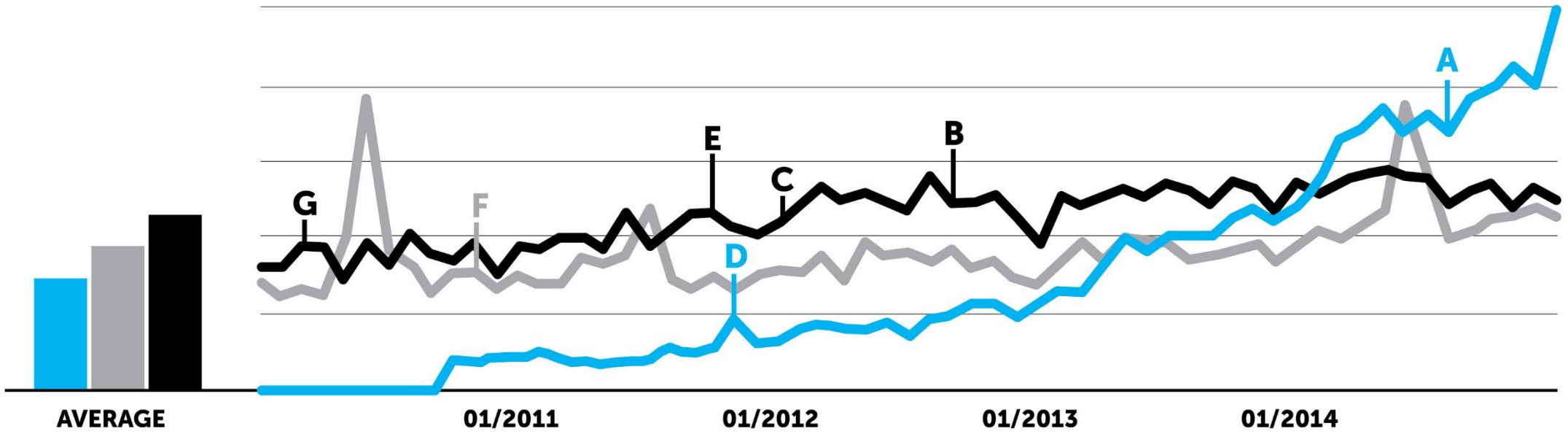
DIGI



- D2D
- D2S
- S2S
- S|D2H

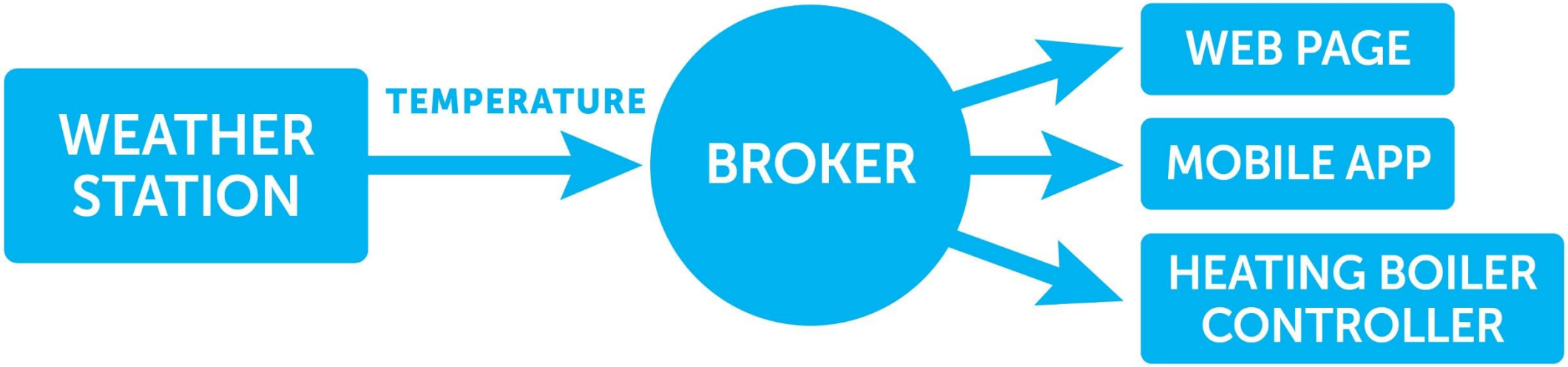
Protocols

HTTP MQTT AMQT CoAP

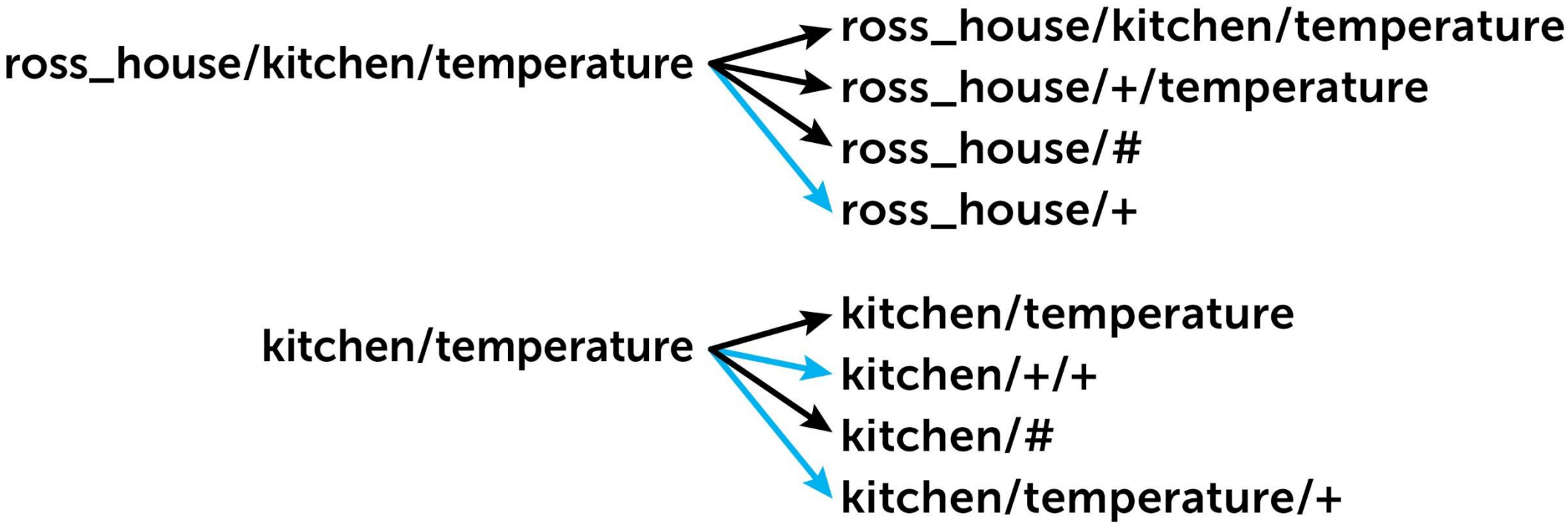


MQTT Architecture

- Client/Server model
- MQTT is message oriented
- Topic system
- Connection over TCP



Topic matching



Main Features

- Assured delivery(3 levels of QoS)
- Retained messages
- Last will & testament(depends on keep_alive value)
- Multiple subscriptions ‘multiplexed’ over one connection
- WebSocket

MQTT vs HTTP

- Request/Response
- One to One
- Heavyweight
- REST - CRUD
- No QoS

MQTT vs CoAP, Constrained Application Protocol

- HTTP-like
- NAT Issue
- UDP, noSSl
- Lightweight(cause of binary)

MQTT vs AMQP,
Advanced Message Queuing Protocol

- Publish/Subscribe
- Metadata
- More complex client
- Queue

Libraries and Brokers

- Hivemq
- Mosquitto + Python lib
- Paho => C/C+, Java, Javascript, Python, Go
- MQTT.js for node.js and the browser.

Code example (Subscribe)

```
import paho.mqtt.client as mqtt

def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("ross_house/kitchen/temperature")
    client.subscribe("ross_house/basement/temperature")
    client.subscribe("ross_house/kitchen/alarm")

def on_message(client, userdata, msg):
    print(msg.topic + " " + str(msg.payload))

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.will_set('Alarm/Iam/Offline', 'My ID Is BLAH BLAH BLAH', qos=2)
client.connect("iot.eclipse.org", 1883, 60)
client.loop_forever()
```

Code example (Publish)

```
import paho.mqtt.client as mqtt

client = mqtt.Client()
client.connect("iot.eclipse.org", 1883, 60)
client.publish('ross_house/kitchen/alarm/', 'Motion was detected')


import paho.mqtt.publish as publish
publish.single('ross_house/kitchen/alarm/', 'Motion was detected',
hostname="iot.eclipse.org")
```