WEBINAR

Machine to Machine Communication with Microsoft Azure IoT Edge & HiveMQ

HIVEMQ

WELCOME

Christoph Schäbel

- Practical MQTT expert with multiple years of experience in the field
- HiveMQ Core Developer
- Background in scalable and reliable distributed systems and robotics

@schaebo

linkedin.com/in/cschaebel

www.hivemq.com

Kresimir Galic

- Independent contractor
- Strong software engineering experience
- Certified Azure Solutions Architect
- Focused on IoT, from concept to the implementation
- Technical blogger
- Speaker

@kgalic1

linkedin.com/in/kresimir-galic-8664a66a/

www.variant-logic.com

IoT Edge

WHAT WE WILL TALK ABOUT

- **Shifting to the Edge**
  → What is edge?
- **Azure IoT Edge Concept**
  → Cloud managed deployments
  → Containerized environment
- **Cloud Managed Deployment of HiveMQ Broker for M2M Communication**

What We Call ‘Edge’?

- Edge of network
- Closer to the devices/machines
- Outside of the cloud, but managed by the cloud

Shifting To The Edge

- **Cloud**
  → Globally available, unlimited compute resources
- **IoT**
  → Signals from sensors and machines, managed centrally by cloud
- **Edge**
  → Intelligence offloaded from the cloud to IoT devices
- **AI**
  → Intelligence capabilities, in the cloud and on the edge

Azure IoT Edge

- **Cloud Managed**

  → Enables rich management of Azure IoT Edge from Azure provide a complete solution instead of just an SDK

- **Cross Platform**

  →Containerized environment, enables targeting most popular edge operating systems such as Windows and Linux

- **Portable**

  → Enables Dev/Test of edge workloads in the cloud with later deployment to the edge as part of a continuous integration / continuous deployment pipeline

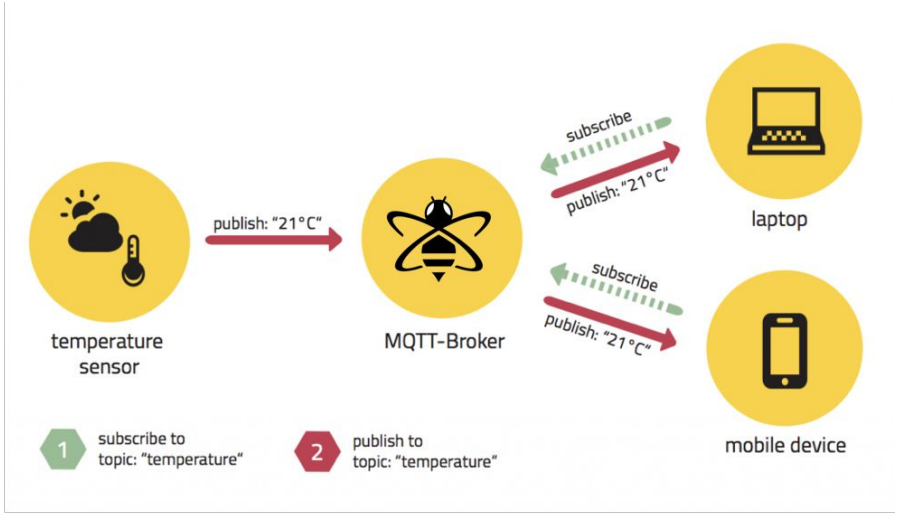
- **Extensible**

  → Deploying custom modules, third party solutions and/or AI

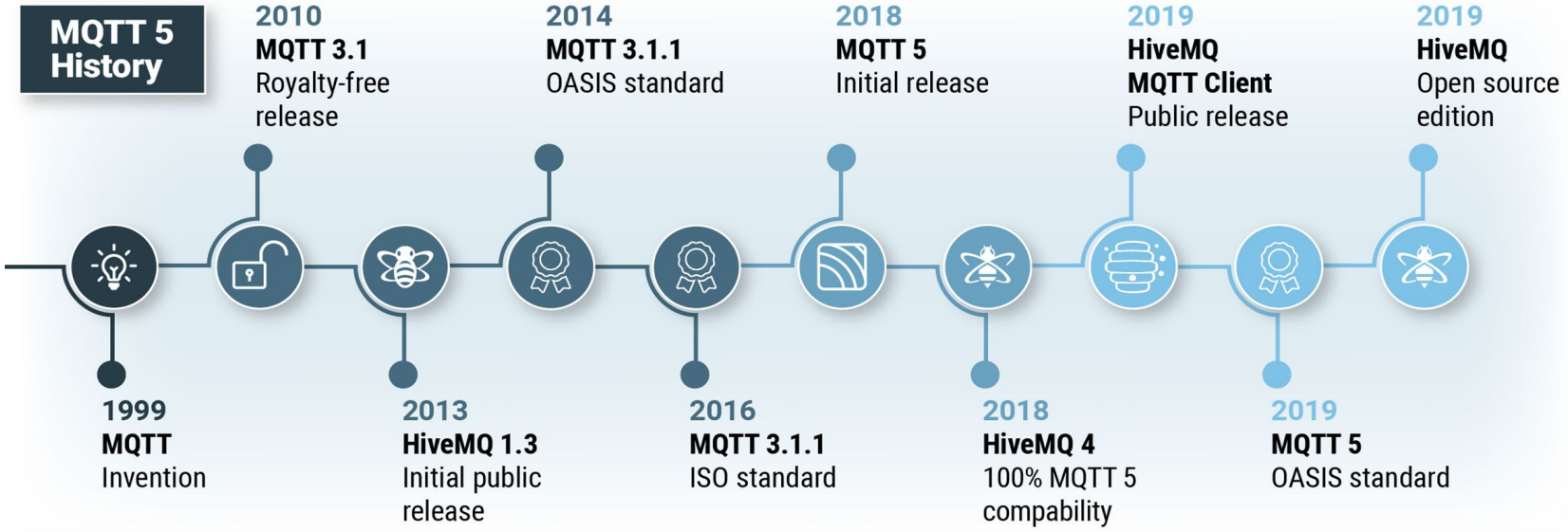
MQTT on the edge

MQTT

- Lightweight protocol on top of TCP/IP
- Publish / Subscribe pattern
- De-coupling of sender and receiver
- Instant message delivery
- Open standard



MQTT History

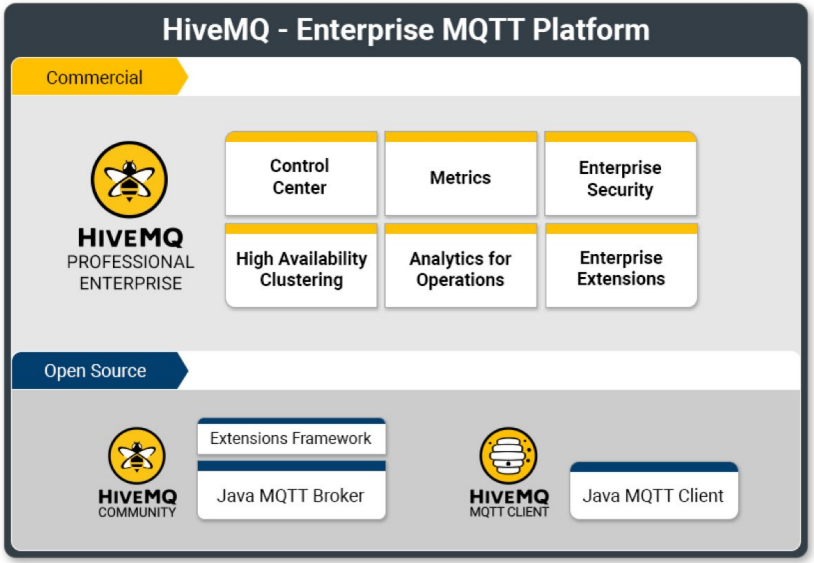


MQTT 5 Features

- **Compatibility and Portability improvements**
- **Session & Message Expiry**
- **Shared Subscriptions**
  - → Load balancing for clients
  - → Multiple clients share the same subscription
- **User Properties**
  - → User defined metadata
  - → Reduce bandwidth and costs
- **Improved Client Feedback**
  - → More descriptive reason codes and messages
  - → Negative acknowledgements
- **Request / Response pattern**

HiveMQ ecosystem

- **Enterprise MQTT platform**
- **Different Editions**
  - → Commercial with enterprise features and up to 24/7 support
  - → Open Source “Community” edition
- **Open Source client library**
  - → built for high-performance
- **Enterprise integrations**
  - → Security
  - → Kafka
  - → ....



Why HiveMQ on the Edge?

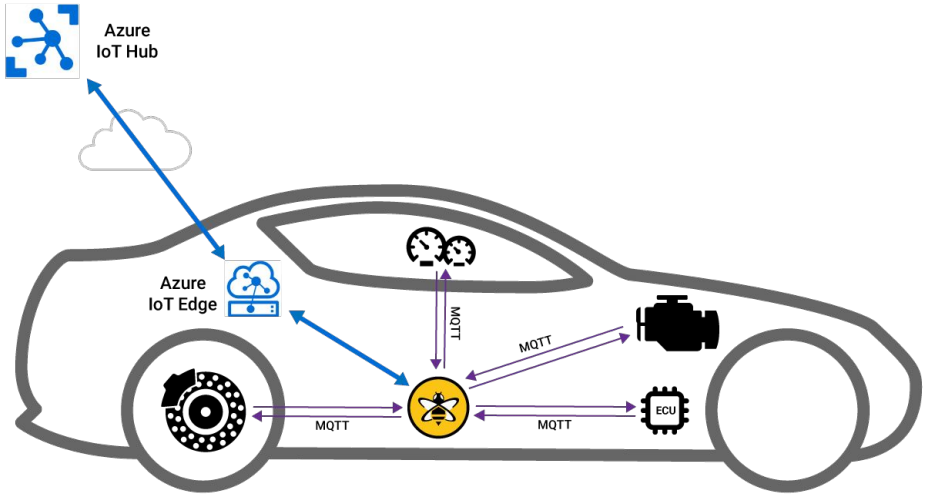
- **MQTT as open standard**
  - → Interoperability
  - → Vendor neutrality
  - → HiveMQ supports 100% MQTT v5
  - → Azure has partial MQTT v3.1.1 support
- **Independent edge**
  - → Reliable intra-edge communication, without dependency on internet connectivity
  - → Vendor neutrality
- **Control data flow and cost**
  - → Control which part of the IoT data is sent to the cloud or analyzed on the edge
  - → Reduce bandwidth and costs

Why HiveMQ and IoT Edge?

- **Bidirectional messaging**
  - → Direct Communication of components / machines at the edge with each other
- **Simple deployment and maintenance**
  - → Managed through Azure Cloud UI
  - → Simple concepts
- **Intelligent Edges**
  - → (Pre-)analyze data at the edge
  - → Utilize Azure AI technologies
- **High throughput on the Edge**
  - → Maintain high-throughput of messages on the edge, not every message has to go to the cloud
  - → Reduce bandwidth and costs

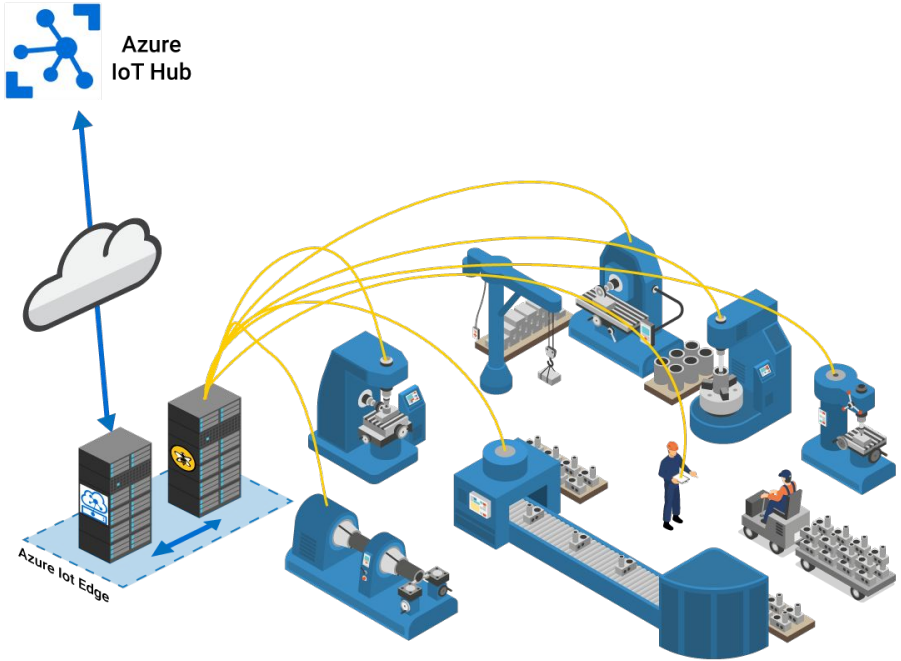
Use Cases

Intra Machine Communication



- **Easy to manage and operate**
  → IoT Devices can be provisioned and managed in the cloud
- **Efficient intra machine communication**
  → Low latency
  → Low overhead
- **Early data analysis**
  → Machine learning and AI for each machine
- **Cost effective**
  → Only selected and pre-analyzed data is streamed to the cloud

Inter Machine Communication

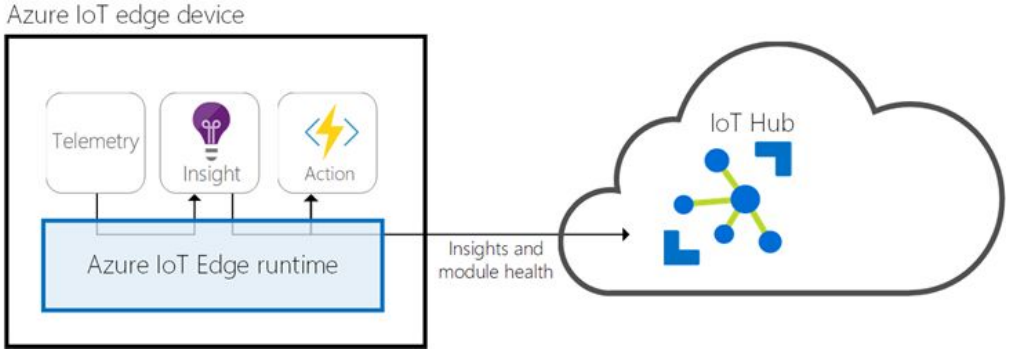


- **Reliable**
  → Factory stays independent of internet connectivity
  → Each building / location can have own communications hub
- **Fast**
  → Lower latencies in local network
- **Interoperable**
  → between machines and vendors
  due to standard protocol MQTT v5
- **Secure**
  → Separate security layers for on-premise and cloud
- **Intelligent**
  → Computing and analysis right at the edge

Azure IoT Edge Concepts

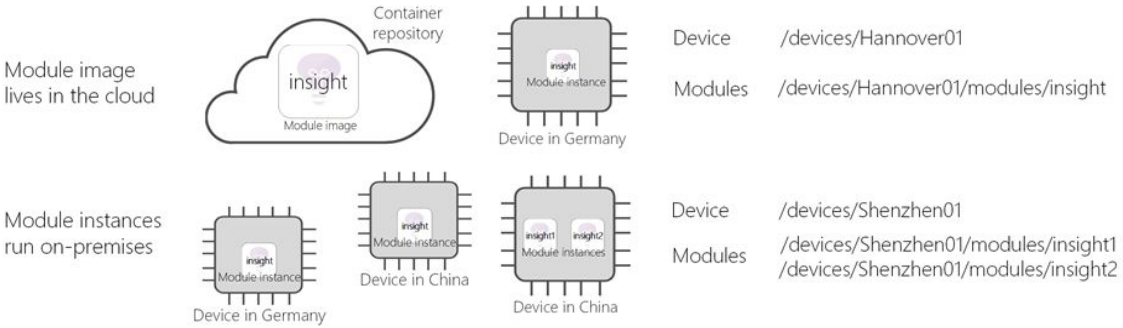
Concept - Azure IoT Edge Runtime

- Installs and updates workloads on the device.
- Maintains Azure IoT Edge security standards on the device.
- Ensures that IoT Edge modules are always running.
- Reports module health to the cloud for remote monitoring.



- Facilitates communication between downstream leaf devices and the IoT Edge device.
- Facilitates communication between modules on the IoT Edge device.
- Facilitates communication between the IoT Edge device and the cloud

Concept - Module



- **Module image** is a package containing the software that defines a module.
- **Module instance** is the specific unit of computation running the module image on an IoT Edge device.

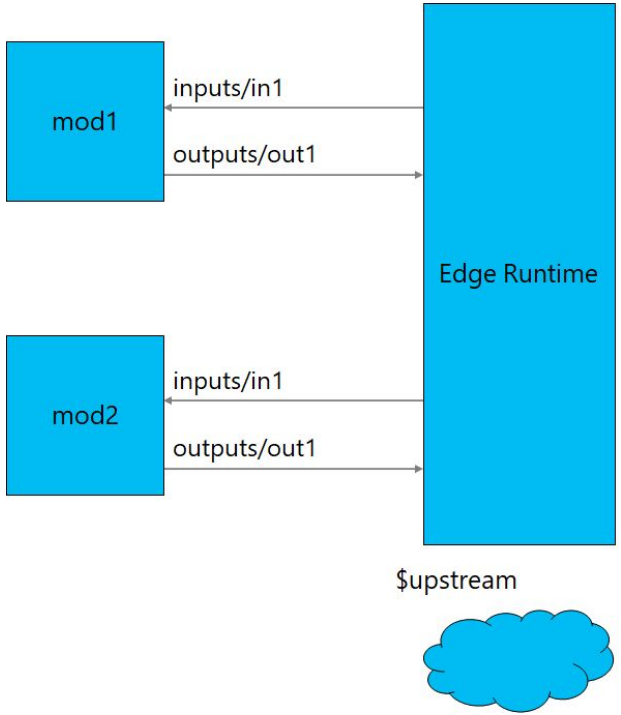
  → The module instance is started by the **IoT Edge** runtime

- **Module identity** is a piece of information (including security credentials) stored in IoT Hub, that is associated to each module instance.
- **Module twin** is a JSON document stored in IoT Hub, that contains state information for a module instance.
- SDKs to develop custom modules in multiple languages (C#, C, Python, Java, Node.JS)

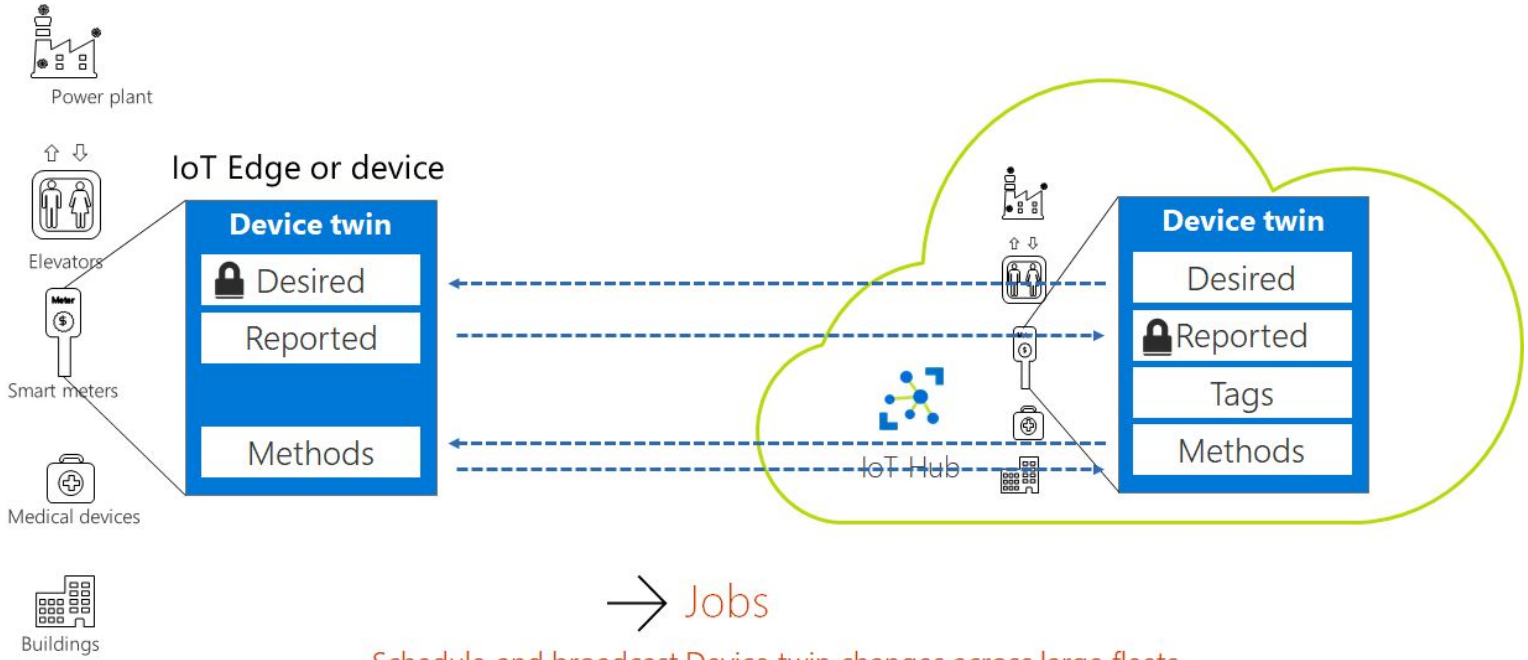
Concept - Routing

- **Query Language**

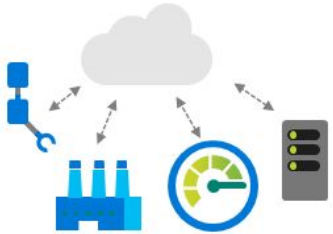
  FROM /messages/modules/TelemetrySubscriberModule/outputs/ INTO $upstream*



Concept - Device Management



IoT in the Cloud vs Edge

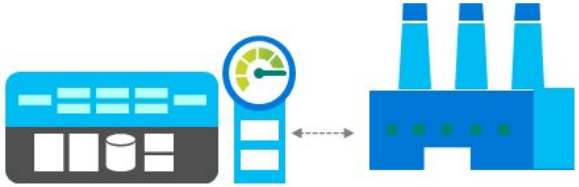


IoT in the Cloud

Remote monitoring and management

Merging remote data from multiple IoT devices

Infinite compute and storage to train machine learning and other advanced AI tools



IoT on the Edge

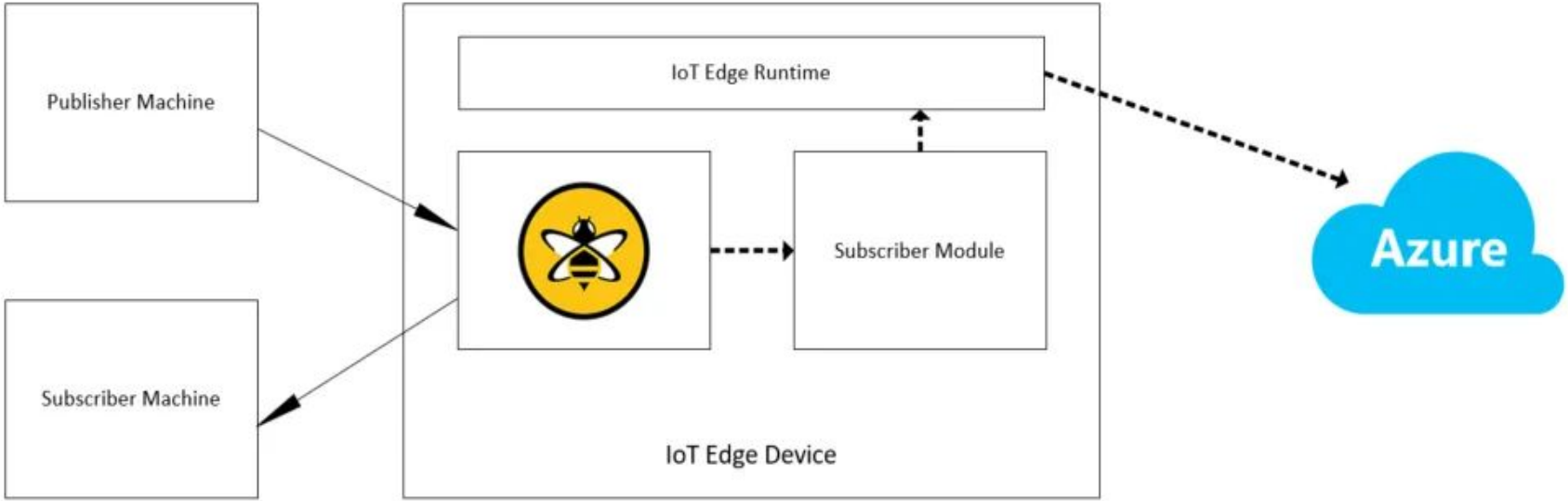
Low latency tight control loops require near real-time response

Protocol translation & data normalization

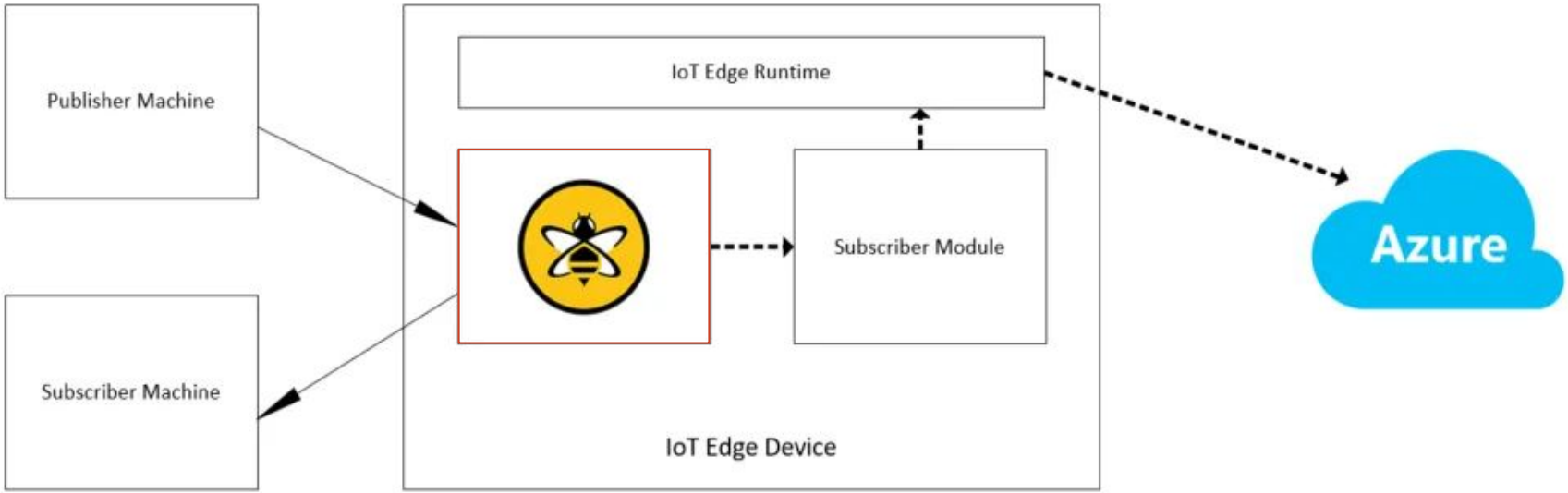
Identity translation

Symmetry

HiveMQ on Azure IoT Edge



HiveMQ on Azure IoT Edge

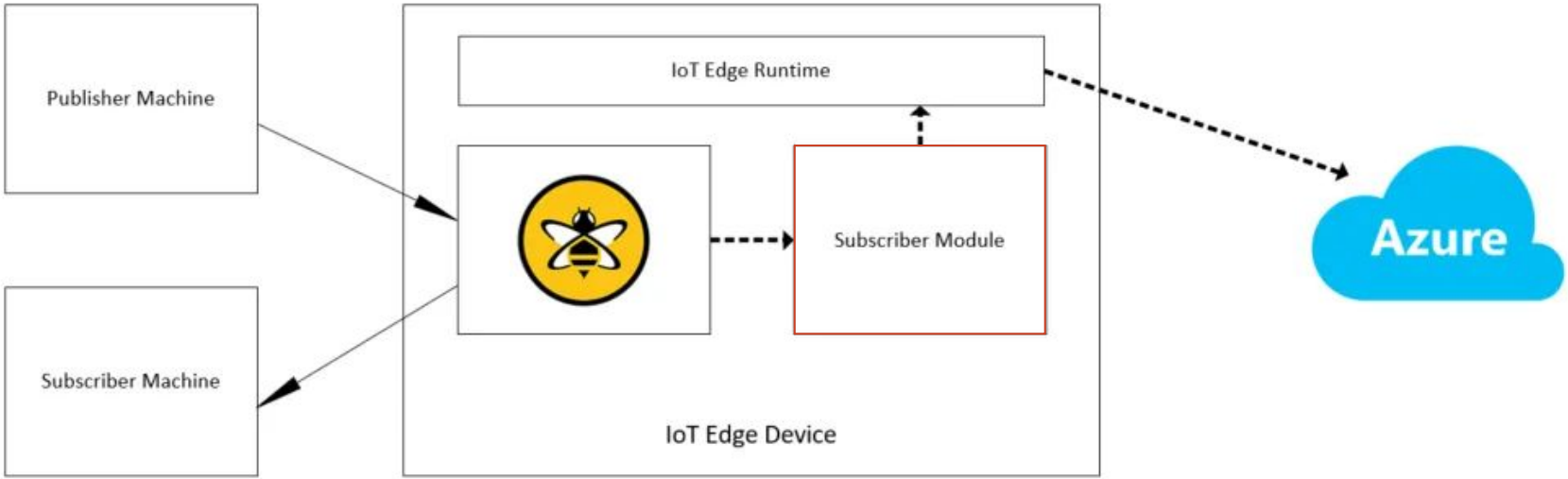


Deploying HiveMQ Broker from the Cloud

- Deployment manifest
- Deploying at scale

```
"HiveMQModule":{
   "settings":{
      "image":"docker.io/hivemq/hivemq4:latest",
      "createOptions":{
         "HostConfig":{
            "PortBindings":{
               "1883/tcp":[
                  {
                     "HostPort":"1883"
                  }
               ],
               "8080/tcp":[
                  {
                     "HostPort":"8080"
                  }
               ]
            }
         }
      }
   },
   "type":"docker",
   "version":"1.0",
   "status":"running",
   "restartPolicy":"always"
}
```

HiveMQ on Azure IoT Edge

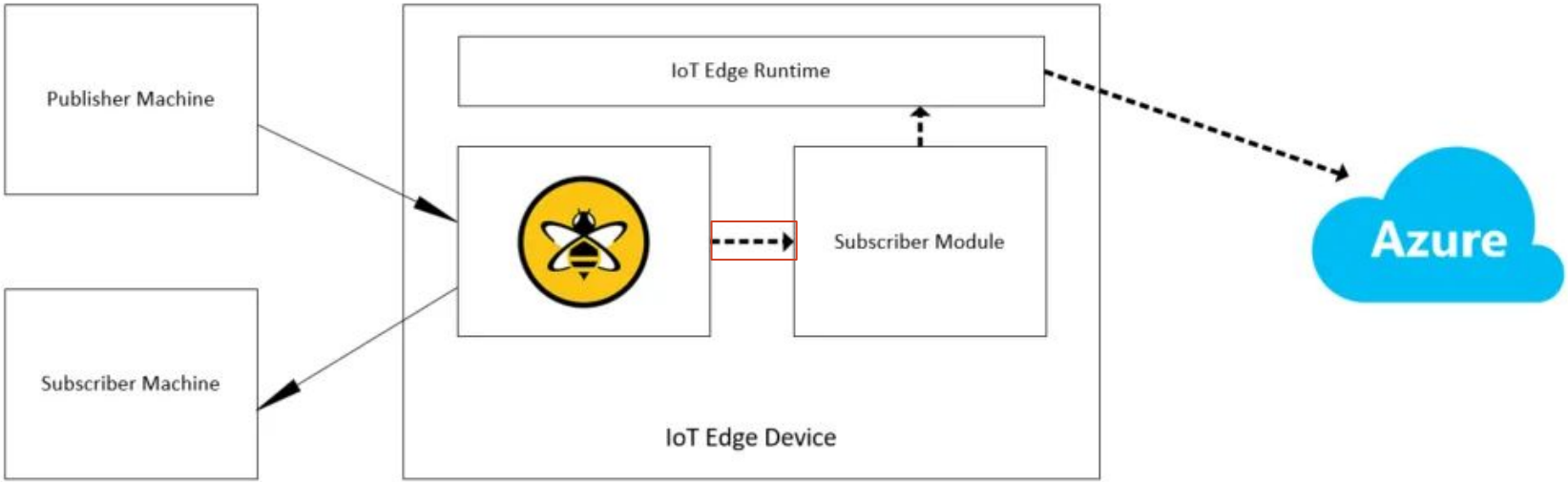


Deploying Custom Module from the Cloud

- Deployment manifest
- Deploying at scale

```
"TelemetrySubscriberModule": {
    "version": "1.0",
    "type": "docker",
    "status": "running",
    "restartPolicy": "always",
    "settings": {
      "image": "${MODULES.TelemetrySubscriberModule}",
      "createOptions": {
        "HostConfig": {
          "ExtraHosts": [
            "hivemq.test.iotlab:ip_address_of_iot_edge_host"
            ]
        }
      }
    }
}
```

HiveMQ on Azure IoT Edge



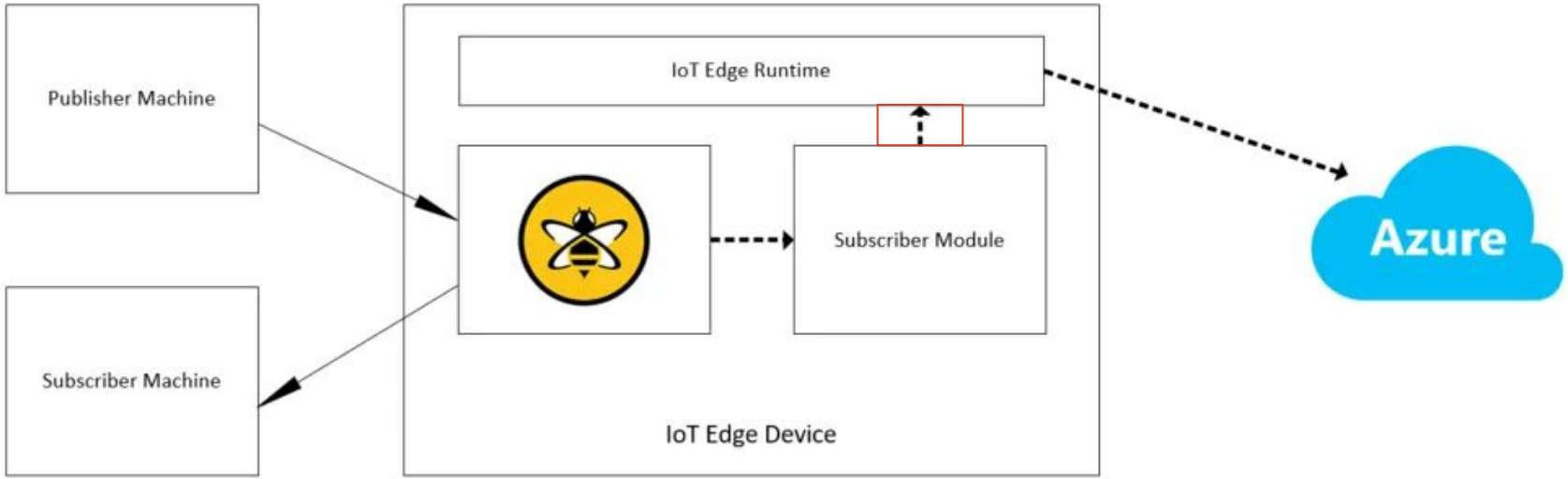
Deploying Custom Module from the Cloud

- Topic name can be a part of Module Configuration (Module Twin)
- Using HiveMQ Client Library

```
public static void ConnectToMqttBroker(String topicName)
{
    mqttClient.connectWith()
    .send()
    .whenComplete((connAck, throwable) -> {
        if (throwable != null) {
            System.out.println("Authentication failed. Please check your credentials!");
        } else {
            System.out.println("Connected to HiveMQ broker, subscribing to the topic '" + topicName + "'.");
            mqttClient.subscribeWith()
            .topicFilter(topicName)
            .callback(messageConsumer)
            .send()
            .whenComplete((subAck, throwable2) -> {
                if (throwable2 != null) {
                    System.out.println("Failed to subscribe to the topic '" + topicName + "'.");
                } else {
                    System.out.println("Subscribed to the topic '" + topicName +"'");
                }
            });

        }
    });
}
```

HiveMQ on Azure IoT Edge



Routing Messages and Sending Results to the Cloud

- Sending the message to the module output
- Azure IoT Edge routes decide where the message goes next (another module, cloud - IoT Hub)
- "TelemetrySubscriberModuleToIoTHub": "*FROM /messages/modules/TelemetrySubscriberModule/outputs/* INTO $upstream*"

```
protected static class MessageConsumer implements Consumer<Mqtt5Publish> {
    private ModuleClient moduleclient;

    @Override
    public void accept(Mqtt5Publish t) {

        byte[] payload = t.getPayloadAsBytes();

        String str = new String(payload, StandardCharsets.UTF_8);

        System.out.println("Received message from the broker: " + str);

        Message moduleMessage = new Message(t.getPayloadAsBytes());

        this.moduleclient.sendEventAsync(moduleMessage, eventCallback, moduleMessage, App.OUTPUT_NAME);
    }

    public void setModuleClient(ModuleClient moduleClient) {
        this.moduleclient = moduleClient;
    }
}
```

Machine to Machine Communication

- Clients/Machines can leverage HiveMQ client library or any other MQTT library
- Using HiveMQ broker to leverage MQTT features
- One example is ‘LastWill’ message to determine whether machine got disconnected for triggering the alarms through the cloud
- Broker for machines with additional logic embedded for preprocessing before alarming another machine
- Full code available on GitHub: https://github.com/kgalic/IoTEdgeM2MWithHiveMQ

What’s next?

What's next?

- **AI / ML at the edge and in the cloud**
  → (Pre-)analyze data at the edge
  → Utilize Azure AI technologies
- **Even deeper integration**
  → HiveMQ extensions that directly integrate with IoT Edge
- **High availability at the edge**
  → Deploy HiveMQ clusters
  → Multiple IoT Edge Devices at the same location
- **HiveMQ Cloud <-> Azure Cloud**
  → Managed MQTT brokers through HiveMQ Cloud
  → Big Data analytics with Azure

Resources

Get Started with MQTT

MQTT Essentials Series

HIVEMQ

Evaluate HiveMQ Broker

HIVEMQ CLOUD

Try HiveMQ Cloud

ANY QUESTIONS?

Reach out to community.hivemq.com

THANK YOU