Webinar

# How to Monitor and Observe IoT and MQTT Applications with HiveMQ

Hosted by 


# IoT Observability

Presentation by David Guschakowski

# Speaker


**David Guschakowski**

Sales Engineer

david.guschakowski@hivemq.com

https://www.linkedin.com/in/david-guschakowski/


- David is Sales Engineer at HiveMQ and serves customers by identifying their needs and providing them with technical support.
- He has provided project and sales consulting services for many years in the areas of data and application integration, data management and messaging. Using this experience, he is helping HiveMQ customers to achieve best possible outcomes with their IoT solutions.
- His main focus is maintaining a high level of customer satisfaction at HiveMQ.

# Agenda

- Differences of observability in classic IT and IoT
- Challenges that come with IoT
- Example production issue
- HiveMQ’s capabilities that help customers reduce time to solution
- Live Demo

# IT Vs. IoT Observability

**IT Applications are:**

- Distributed applications at large scale
- Network of well known actors
- Operate over reliable networks
- Mostly persistent access
- Mostly visibility into data flows
- Mostly single node applications


**IoT Applications are:**

- Distributed applications at massive scale
- Network of black boxes
- Operate over unreliable networks
- No persistent access
- No visibility into data flows
- Crucial components are distributed

# IoT Challenges: Identifying Errors in Noise

**Many IoT applications are based on messaging**

- Lost messages can create problems for an application
- Messages that are late to arrive can also create problems

**IoT devices can lose messages**

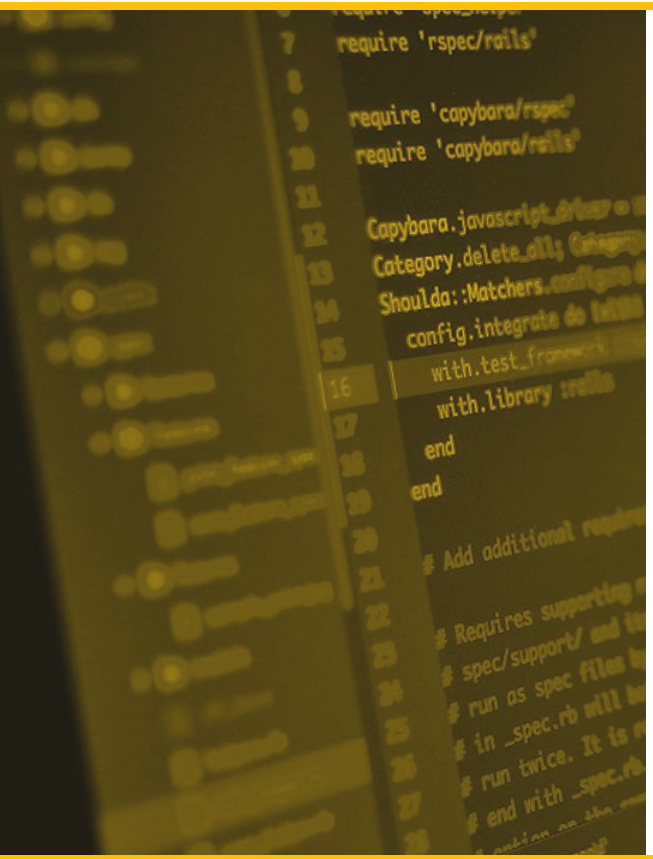
- Programming errors
- Networking error
- Hardware errors

# IoT Challenges: Too much data

**IoT applications can generate too much data for traditional analysis**

- Network Monitoring tools such as Wireshark can be overwhelmed with the amount of IoT Data
- Analysis of live data has impact on performance
- Filters are needed to limit performance impact and reduce time to identify issues in clients or topics

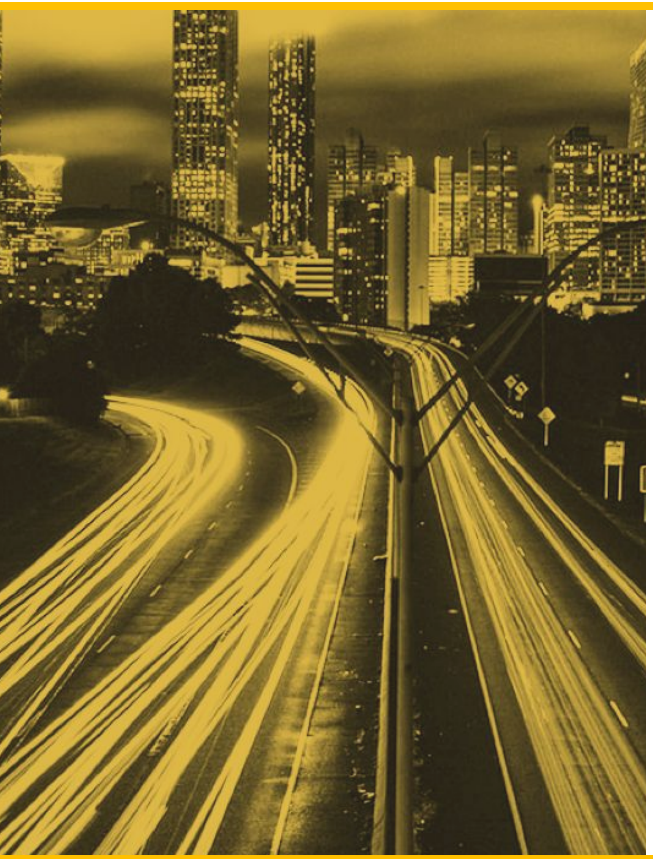
# IoT Challenges: System of Systems




**IoT applications are made up of many systems**

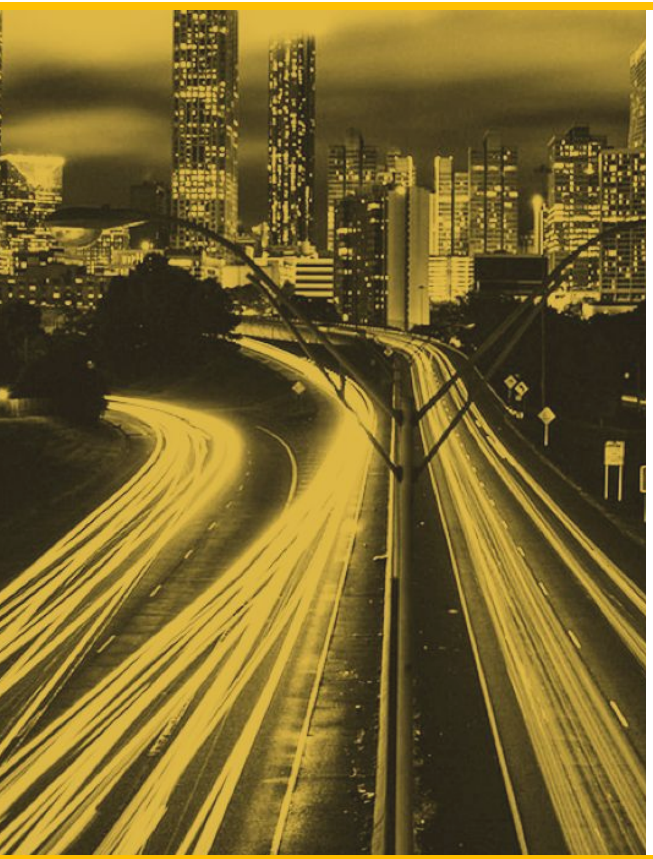
- Need to understand health of each service
- Services should generate metrics that can be used for a single application specific monitoring view
- Expose system related data e.g. memory, CPU, disk usage
- Expose MQTT related data e.g. connections, inbound publish rate, outbound publish rate

# Production Issue in Car Sharing Service



- One day 8,000 cars in Frankfurt did not work
- Call center overwhelmed with support calls
- Overall system appears as a black box for the Call center
- They only see that the system is not working and is slow

# Production Issue in Car Sharing Service

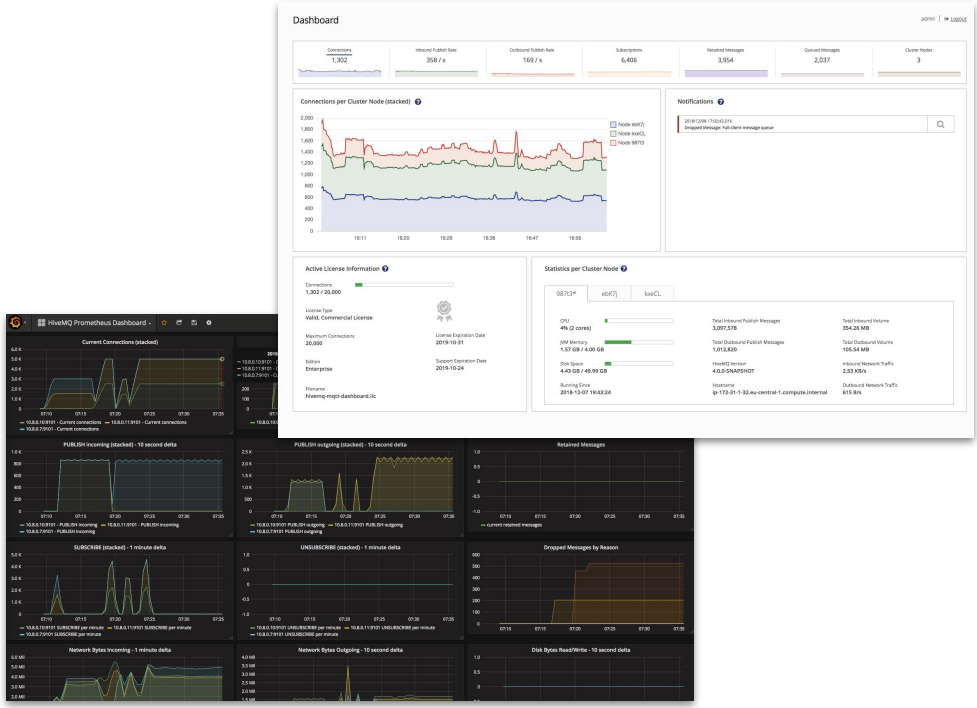


- HiveMQ, the MQTT broker, a central part of the system was able to provide visibility
- Used HiveMQ Control Center to discover the issue was with the cellular network provider not in the application layer

# HiveMQ Observability Capabilities

## System Monitoring

- HiveMQ Control Center allows for monitoring of system health of the HiveMQ broker
- JMX endpoint and extensions enable integration into any existing application performance monitoring tool



# HiveMQ Observability Capabilities

## Logging

- HiveMQ uses Logback and therefore supports multiple logging appenders
- Easy to integrate logging consolidation like Syslog
- Multiple log files
- Easy to create separate files for custom extensions

```
<configuration>
...

    <appender name="SYSLOG" class="ch.qos.logback.classic.net.SyslogAppender">

        // IP-Address of your syslog server
        <syslogHost>$IP-Address</syslogHost>

        <facility>user</facility>
        // replace X with the actual node
        <suffixPattern>[nodeX] %-30(%d %level)- %msg%n%ex</suffixPattern>
    </appender>

    <root level="DEBUG">
        <appender-ref ref="SYSLOG" />
    </root>

'''

</configuration>
```

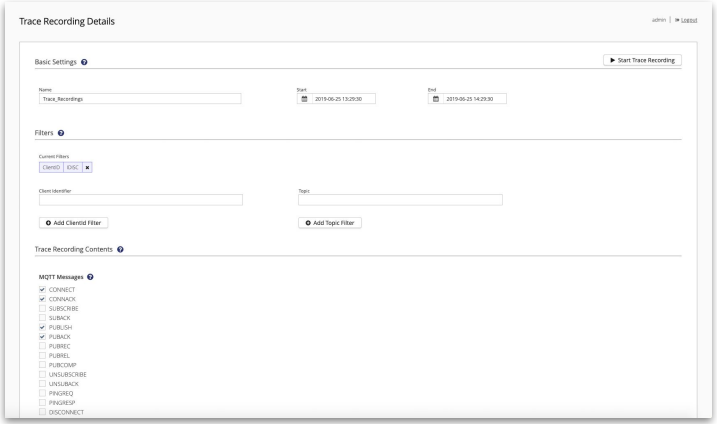
# HiveMQ Observability Capabilities

## Client & Topic Tracing

HiveMQ Trace Recording allows to:

- Specify time frame to record
- Specify range of devices to record
- Set filters for MQTT topics to trace
- Set MQTT Messages to trace

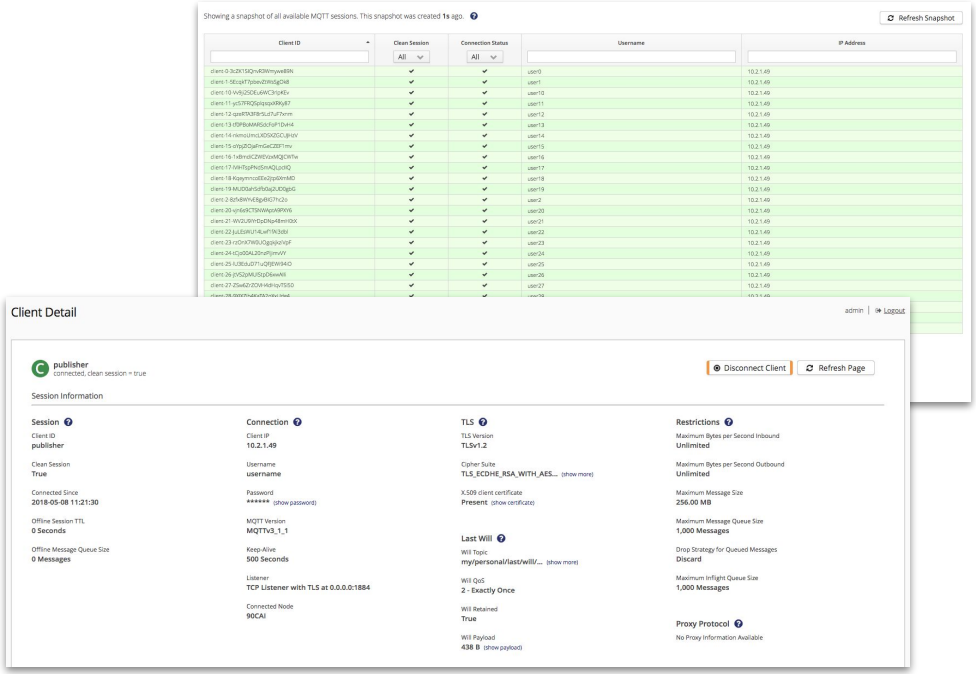
➔ Enables fine grained diagnostics and debugging of irregular behaviour while dealing with millions of actors and events per second



# HiveMQ Observability Capabilities

## Device Health

- HiveMQ Control Center allows to query and list all present devices
- Drill down into each to show the health and connection information of a specific device
- Client Event history

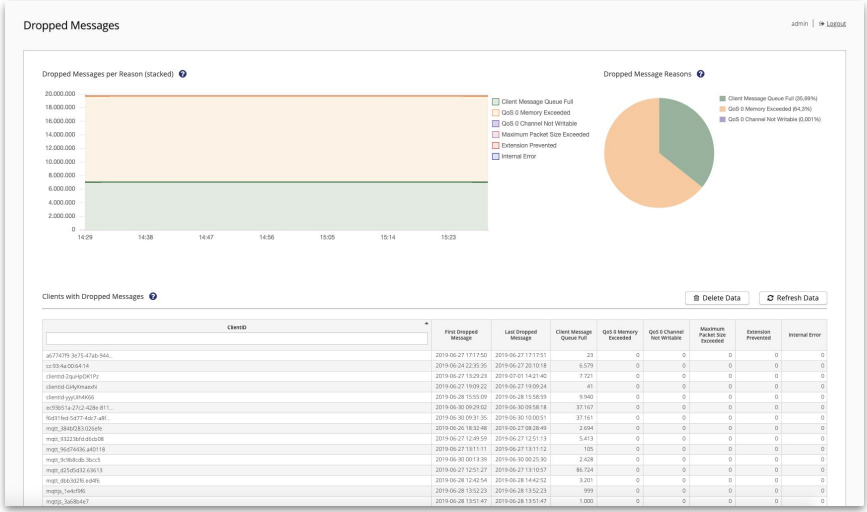


# HiveMQ Observability Capabilities

## Dropped Message

HiveMQ Control Center offers analytics functionality of messages not published by a broker, called dropped messages. It can provide information about:

- Reason of dropped message
- Affected clients or shared subscriptions
- Relevant timestamps

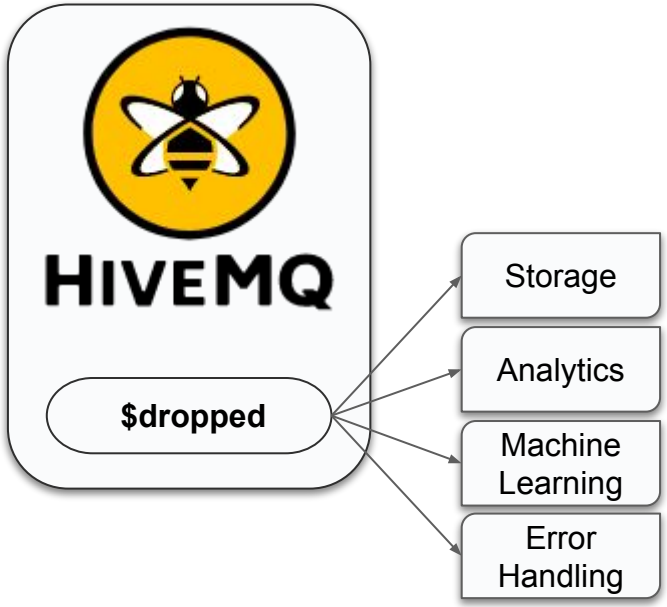


# HiveMQ MQTT-Addons

**$dropped Topic**

Captures all dropped messages on the system

- Client's message queue full
- QoS 0 message not delivered due to inability to write to client socket or memory being exceeded
- Message size bigger than max size of the receiving client
- Dropped messages due to internal errors or prevented by a Publish Inbound Interceptor

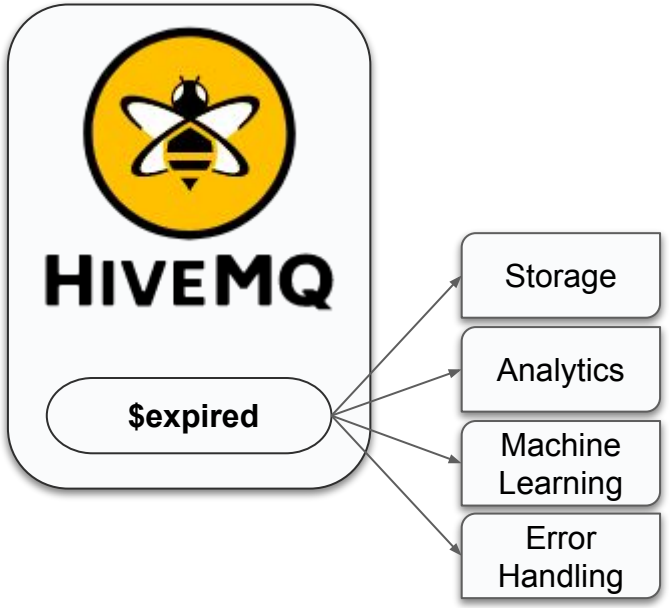


# HiveMQ MQTT-Addons

**$expired Topic**

Captures all expired messages on the HiveMQ system

- A client takes too long to consume the message
- A message expires before an offline client can consume it
- A retained message that is stored on the broker expires

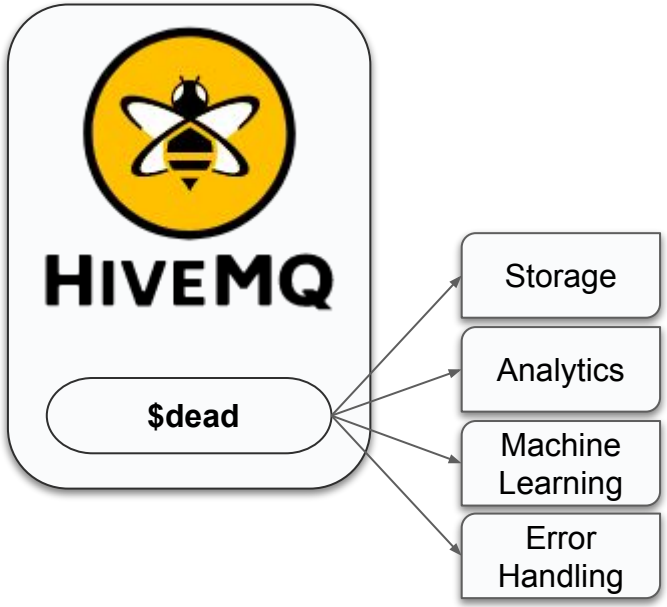


# HiveMQ MQTT-Addons

**$dead Topic**

Captures all dead messages on the HiveMQ system

- A published message without subscribers is considered a dead message



# Live Demo

# Resources

Try HiveMQ!
https://www.hivemq.com/downloads/

Visit HiveMQ Control Center
https://www.hivemq.com/docs/hivemq/4.6/control-center/introduction.html

Try HiveMQ Cloud!
https://www.hivemq.com/cloud/



# Poll and Q&A

# THANK YOU

Contact

David Guschakowski

Sales Engineer


david.guschakowski@hivemq.com


https://www.linkedin.com/in/david-guschakowski/