



**WEBINAR**

# **MQTT 5 - Why You Need It and Potential Pitfalls**



**HIVEMQ**

# WELCOME

**Dominik  
Obermaier**

 @dobermai



- HiveMQ CTO
- Strong background in distributed and large scale systems architecture
- OASIS MQTT TC Member
- Author of „The Technical Foundations of IoT“
- Conference Speaker
- Program committee member for German and international IoT conferences



# MQTT REFRESHER

# MQTT Overview



- IoT Messaging Protocol
- 3 QoS Levels
- Retained Messages
- Persistent (offline) sessions
- Binary with minimal overhead

# MQTT Use Cases



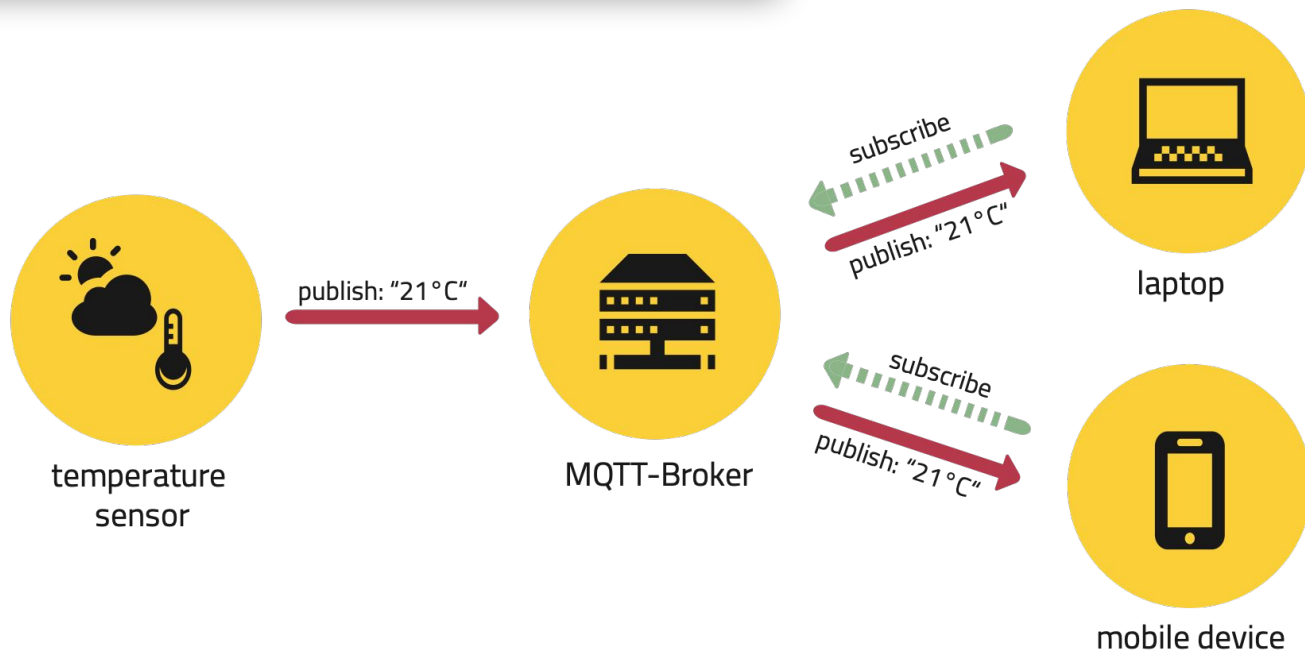
- Push Communication
- Reliable Communication over unreliable networks
- Constrained Devices
- Low Bandwidth and High Latency
- Enterprise backends to mobile communication

# MQTT Use Cases



- Connected Car
- Industry 4.0 / IIoT
- Logistics / Transportation
- (IoT) Messaging Middleware
- Telecommunications

# Publish / Subscribe





# MQTT HISTORY





**1999**  
**MQTT invented**  
for oil pipeline  
monitoring



**2012**  
**Mosquitto 1.0**  
released



**2010**  
**MQTT 3.1** opened as  
royalty free protocol





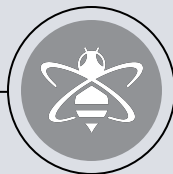
**2013**  
OASIS TC formed



**2018**  
MQTT 5 officially released



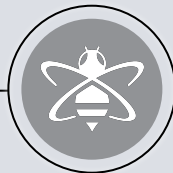
**2013**  
HiveMQ 1.3 released



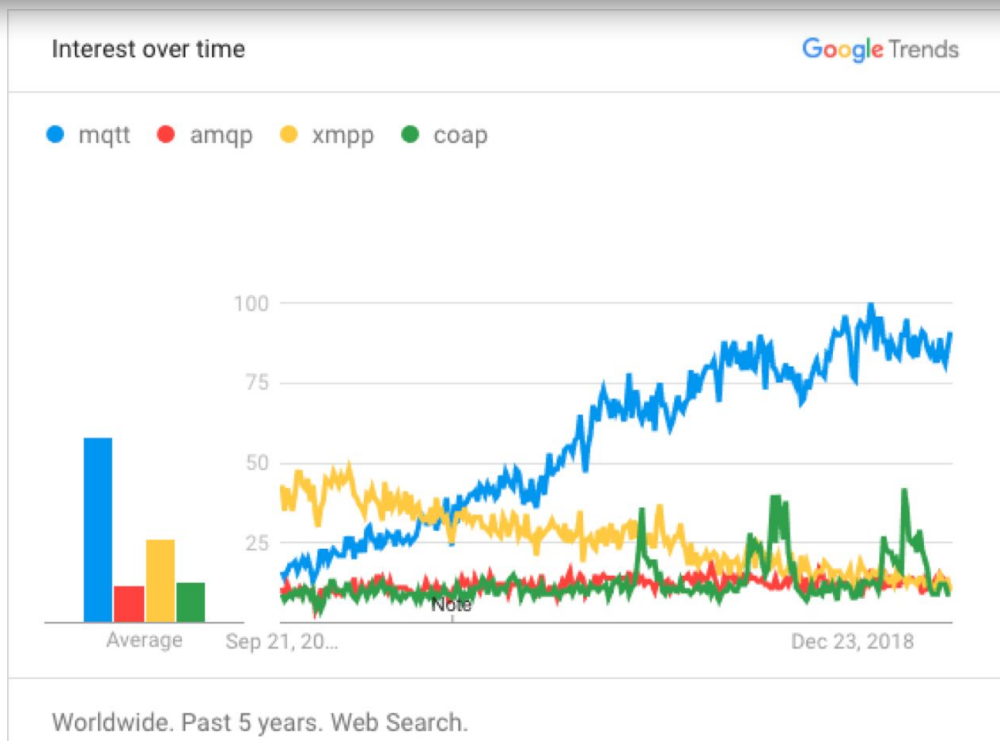
**2014**  
MQTT 3.1.1  
officially released



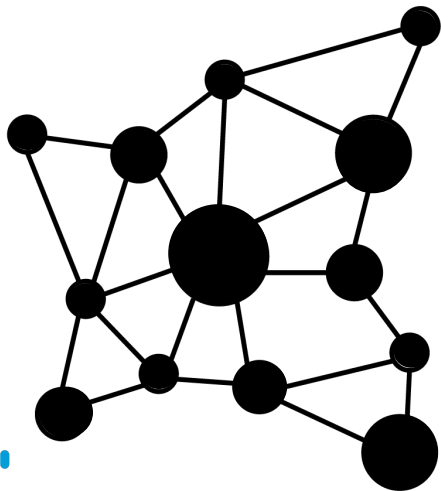
**2018**  
HiveMQ 4 has  
MQTT5 support



# MQTT is quite popular nowadays



# THE MQTT TC



5

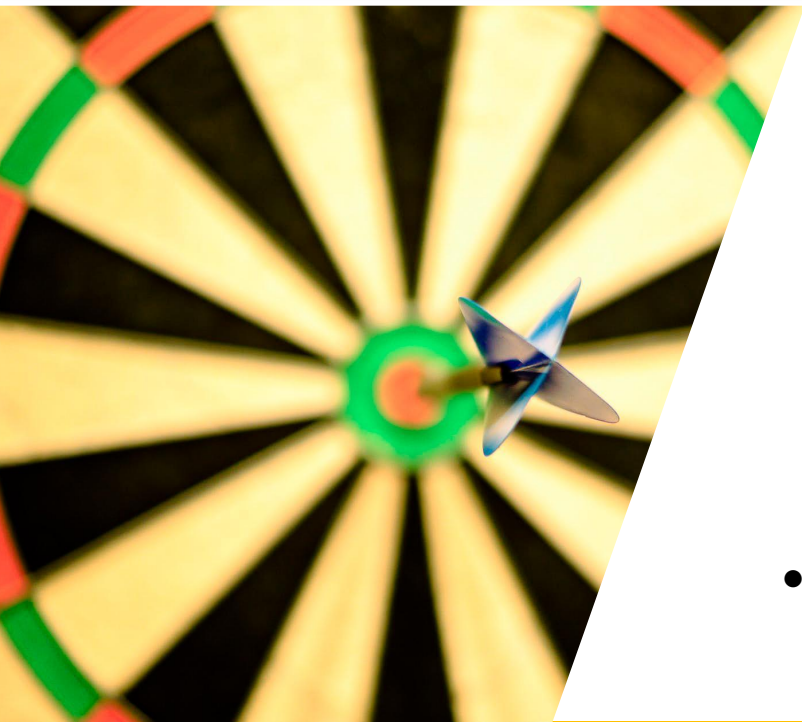
**MQTT 5**

# MQTT 5 - Overview



- Successor of MQTT 3.1.1
- Non-backward compatible
- First public release in January 2018, official release in March 2019
- Many new features
- Clarifications of the 3.1.1 specification

# MQTT 5 - Goals



- Enhancements for scalability and large scale systems
- Improved error reporting
- Formalize common patterns including capability discovery and request response
- Extensibility mechanisms including user properties
- Performance improvements and support for small clients



# FOUNDATIONAL CHANGES

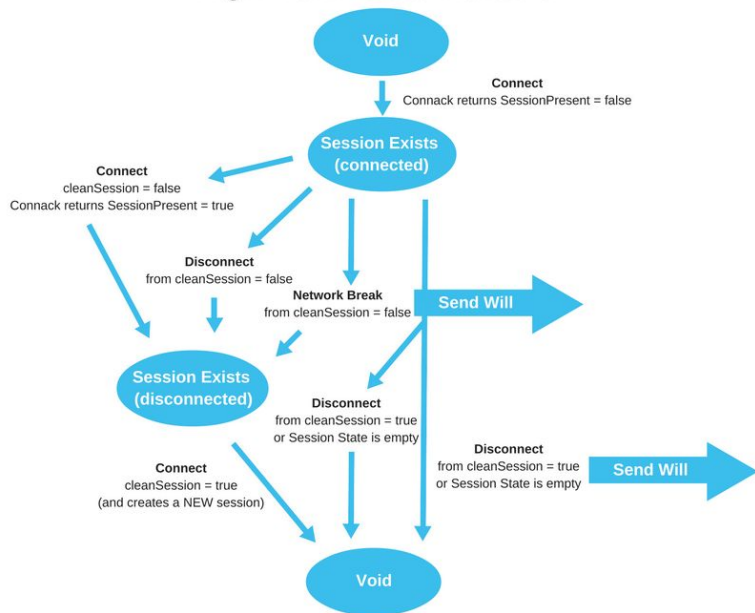


# Properties in MQTT Header & Reason Codes

- ACKs can contain **Negative acknowledgements**
- Return Codes for Unsupported Features
  - Retain Available
  - Maximum QoS
  - Wildcard Available
  - Subscription Identifiers available
  - Shared Subscriptions available
  - Maximum Message Size
  - Server Keep Alive

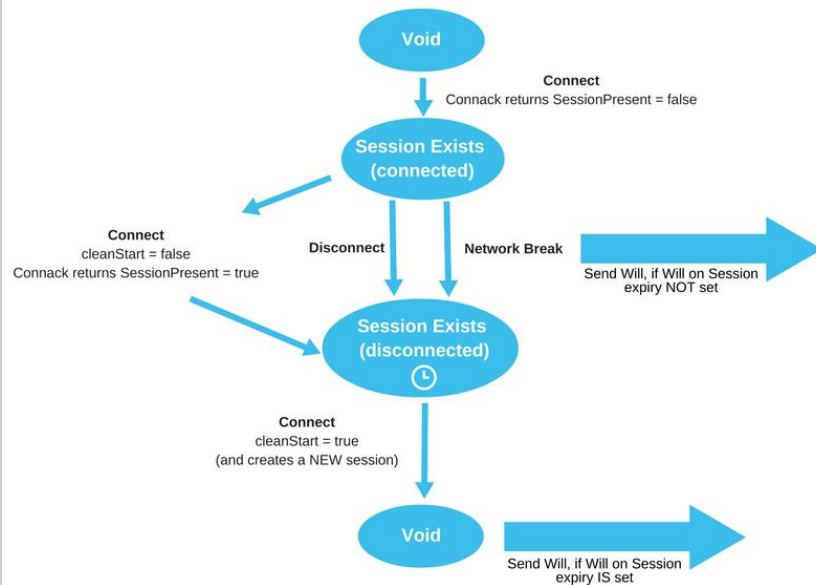
# Streamlined Session State / Lifecycle

## MQTT V3.1.1 State Transition



Source:

## MQTT V5 State Transition



# Other Notable Changes



- No QoS 1 & 2 retry
- Passwords without usernames allowed
- Bi-directional DISCONNECT packets
- New Data Type: UTF-8 String pairs
- New Protocol version on the wire



# NEW FEATURES



# Session & Message Expiry

# Session & Message Expiry



- Session Expiry is an optional part of the CONNECT message
- Session Expiry Interval in Seconds
- Broker expires session after the given interval as soon as the client disconnects
- Publication Expiry interval is an optional part of a PUBLISH message
- Applies to online and queued messages



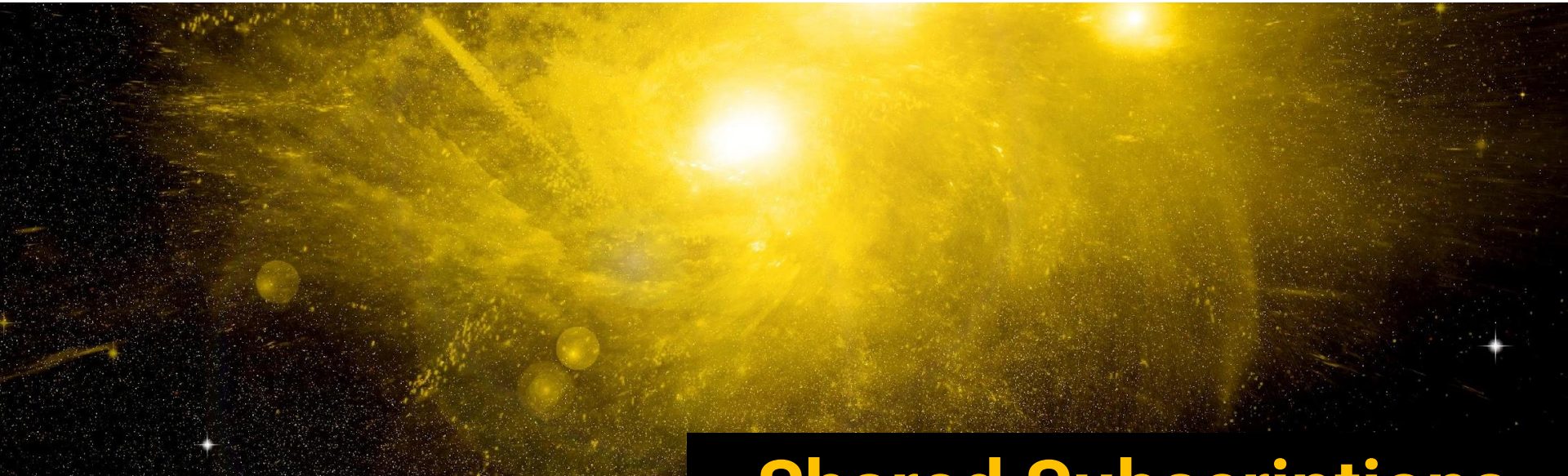
# User Properties

# User Properties



- User Defined Metadata Headers
- Can be part of most MQTT packets
- UTF-8 encoded Strings
- An unlimited number of user properties can be added





# Shared Subscriptions

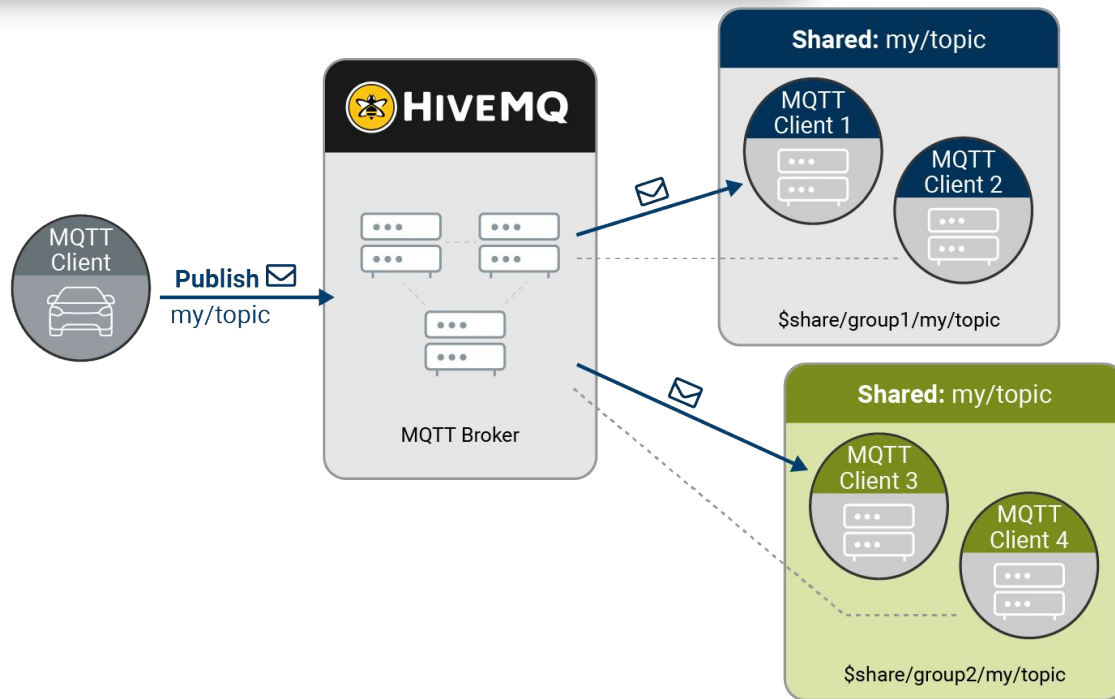
# Shared Subscriptions

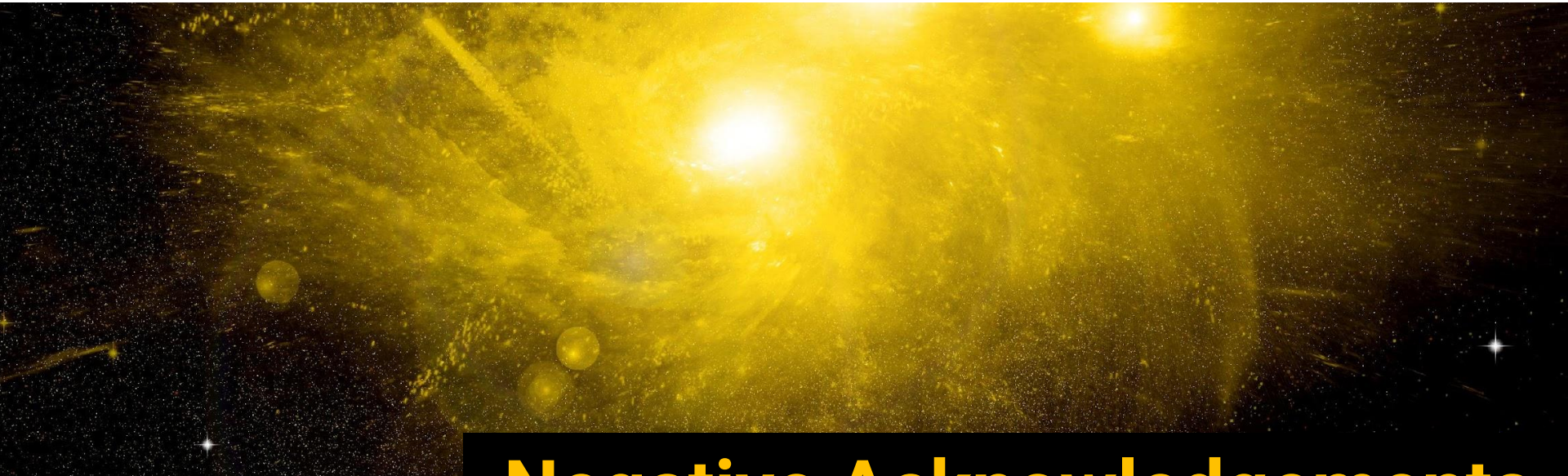


- Client Load Balancing. Multiple clients share the same subscription
- Special Syntax: **\$share/{ID}/my/topic**
- Useful for scaling out backend subscribers
- Also supported by HiveMQ for MQTT 3.1 and MQTT 3.1.1
- Up-/Downscaling of clients at runtime possible. Perfect for cloud native scenarios (Kubernetes, ...)
- Optional feature, not supported by all vendors\*

*\* HiveMQ fully supports all optional features, including this feature*

# Shared Subscriptions



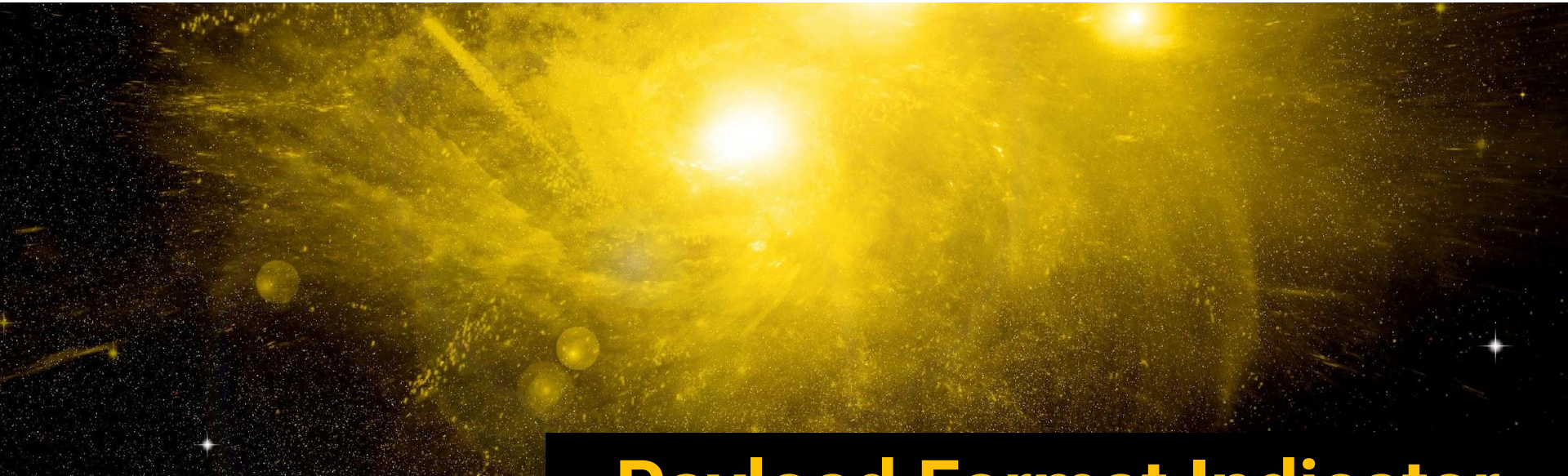


# Negative Acknowledgements

# Return Codes & Negative Acknowledgements



- Most MQTT packets can carry error codes
- Reason Strings are optionally available
- Return codes optionally, server can choose for security reasons to just disconnect clients
- Return Codes from protocol level errors to application level errors
- DISCONNECT messages allow clients to figure out why they were disconnected

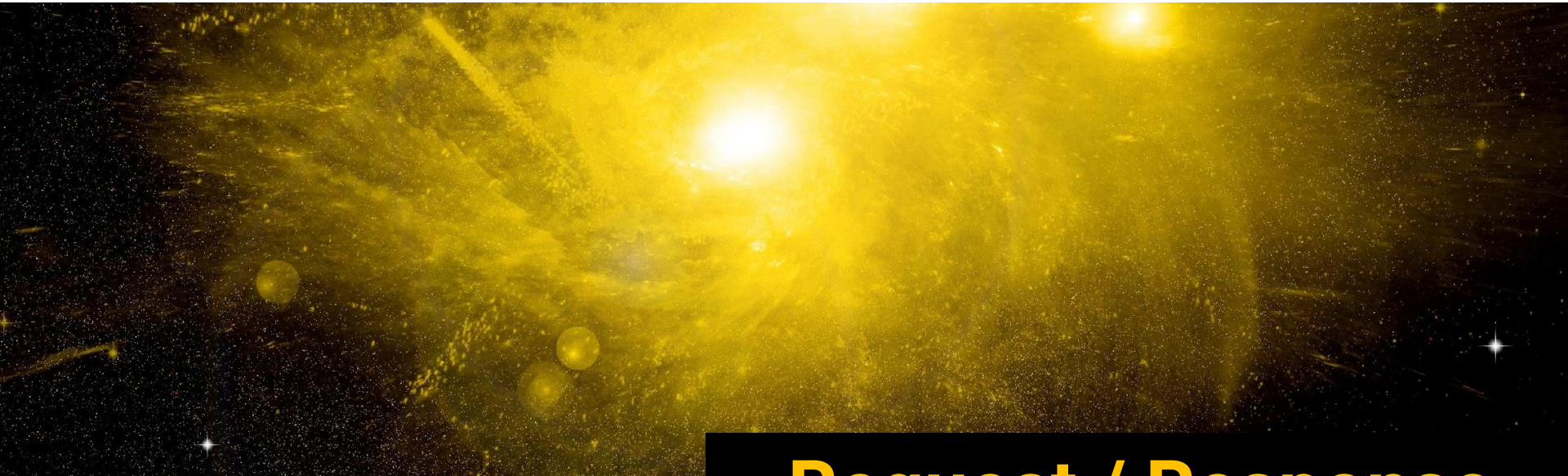


# Payload Format Indicator

# Payload Format Indicator & Content Type



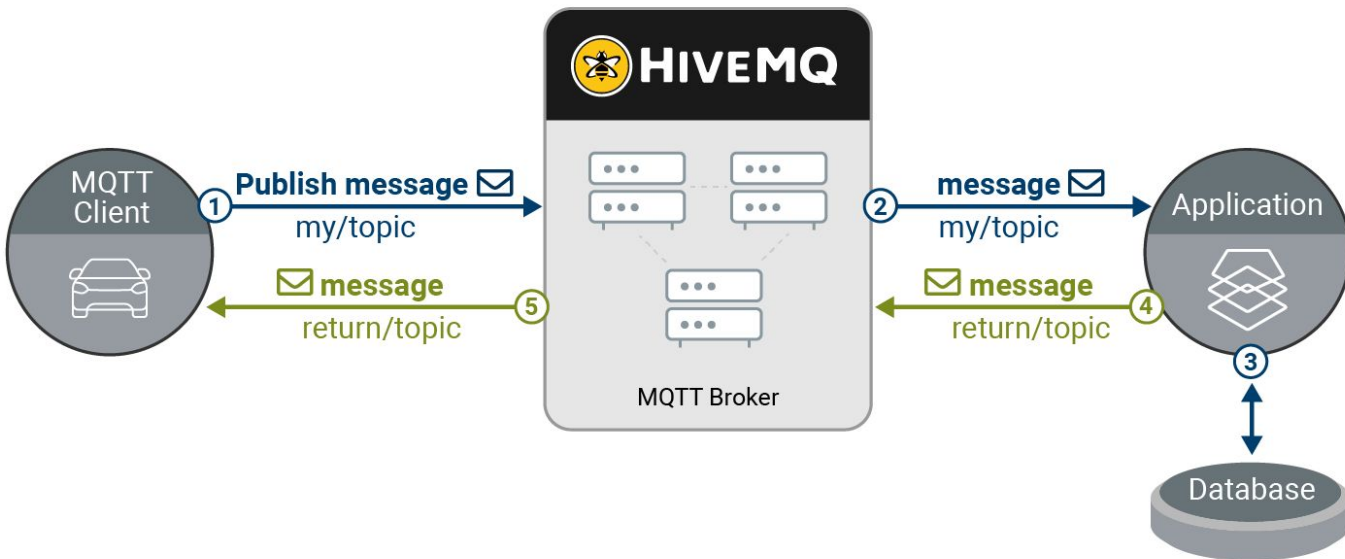
- Optional part of a PUBLISH message
- Return codes for ACK messages available if payload format is invalid
- Receiver may validate the format indicator
- “Content Type” optional header can carry a MIME type
- “Payload Format Indicator” can be binary or UTF-8



# Request / Response



# Request / Response



# Request / Response



- Pattern for “business ACKs”
- Request / Response is a pattern that is also available for MQTT 3.1.1
- Publishing Client must subscribe to a response topic prior to sending data
- “Request Response Information” Header for response topic
- “Correlation Data” header for correlation of request/response



# Topic Aliases

# Topic Aliases



- Clients can send numeric aliases instead of the whole topic for further MQTT PUBLISH messages
- Reduces the size of messages significantly for large topics
- Client -> Server as well as Server ->Client
- Optional feature, not supported by all vendors\*

*\* HiveMQ fully supports all optional features, including this feature*

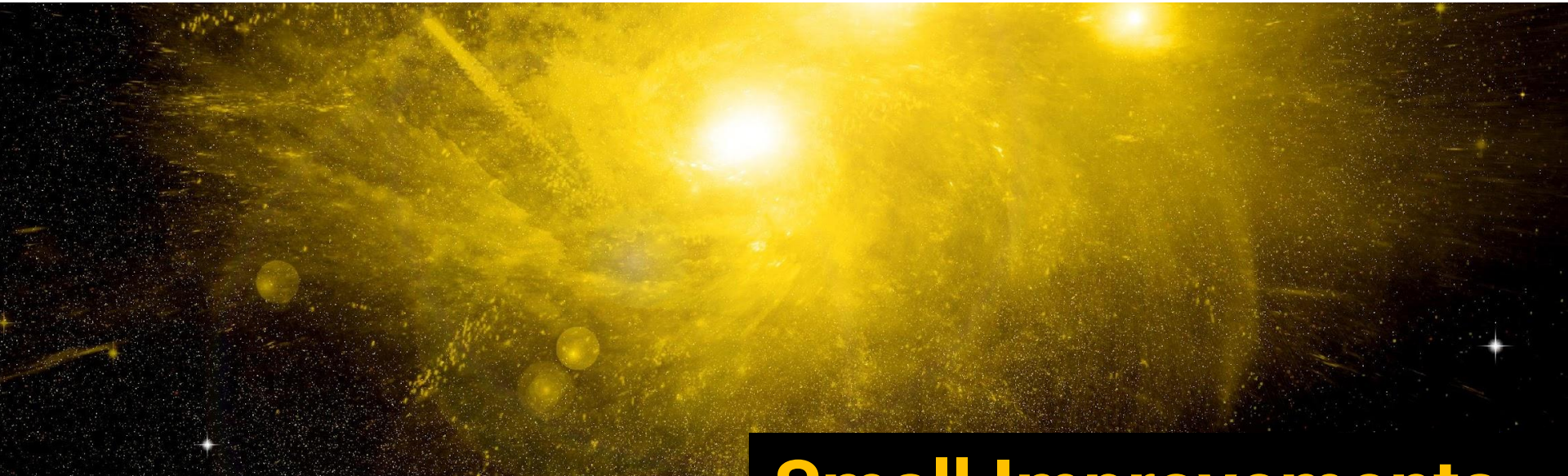


# Flow Control

# Flow Control



- Flow Control allows MQTT clients (and servers) to indicate the maximum messages that can be processed concurrently
- Only applies to QoS 1 and 2 messages
- Prevents overload of constrained clients
- Replaces the MQTT 3 “in-flight window” concept



# Small Improvements

# Small Improvements



- Subscription ID
- Passwords without usernames are now allowed
- Server can now also send a DISCONNECT packet
- Maximum Message Size Indication
- Streamlined Session State



# Small Improvements



- Subscription Options
- Will Delay
- Server Keep Alive
- Assigned Client ID
- Server References



# Pitfalls

# #1 Optional Features



- MQTT 5 has optional features\*
- Some mandatory and beloved MQTT 3 features are now optional, so beware!
- Server can indicate optional features but this requires business logic on the client side!
- So MQTT 5 != MQTT 5 for different vendors
- Many sophisticated standalone products (like HiveMQ or mosquitto) fully support MQTT 5 with all optional features

*\* HiveMQ fully supports all optional features*

# #1 Optional Features



List of optional features a server can indicate

- Retain Available
- Maximum QoS
- Wildcard Available
- Subscription Identifiers available
- Shared Subscriptions available
- Maximum Message Size
- Server Keep Alive

# #2 Compatibility



- MQTT 5 is not compatible with MQTT 3
- Reality: Many MQTT 5 projects also have MQTT 3 devices in the mix
- Brokers like HiveMQ support hybrid MQTT 3 and 5 operation
- MQTT 3 and MQTT 5 features are mapped for compatibility and can be used simultaneously with HiveMQ
- Talk to your vendor about the migration path

# #3 Client Ecosystem



- Vast amount of MQTT 3 clients available
- MQTT 5 clients for some language still in early days
- Production ready MQTT 5 clients are available
- See next slides for an overview of MQTT 5 software

# #4 Additional Security Options



- MQTT 5 introduces new CONNECT Headers
  - *Authentication Method*
  - *Authentication Data*
- MQTT 5 introduces a new Packet: AUTH packet
- Decide on what you really require or use a secure and opinionated state-of-the-art solution like the [HiveMQ Enterprise Security Extension](#)

# #5 Don't overuse User Properties



- User properties of MQTT 5 are similar to HTTP headers and allow to create custom protocols on top of MQTT
- Clearly decide beforehand what is metadata and what is payload
- Hint: Most things you'd like to intercept (e.g. with the [extension system](#) of your broker) should probably be a user property
- Antipattern: No payload but lots of user properties





**SHOULD WE UPGRADE YET?**



# Broker Compatibility

# Mosquitto



- MQTT 5 support since April 2019
- Supports optional features
- Version 1.6 added MQTT 5 support

# HiveMQ



- MQTT 5 Support with HiveMQ 4
- 100% MQTT 5 support since 2018, including all optional features
- Hybrid MQTT 3 and 5 possible at scale (up to 10,000,000 concurrent MQTT connections)
- Extension system allows to use all MQTT 5 functionality
- Open Source Community Edition also supports 100% of all MQTT 5 features

# Cloud Platforms

- AWS IoT Core: **No** MQTT 5 Support
- Azure IoT Hub: **No** MQTT 5 Support
- Google Cloud IoT Core: **No** MQTT 5 support



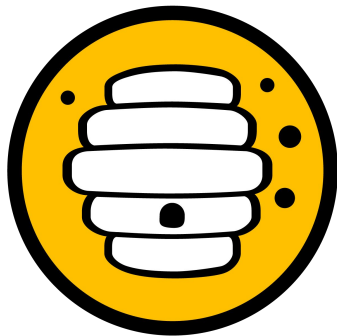
# Client Compatibility

# Eclipse Paho MQTT 5 Support



- C: **YES**
- C++: **No**
- C Embedded: **No**
- Java: **No**
- Android: **No**
- Javascript: **No**
- Python: **No**
- Go: **No**
- Rust: **YES**
- C#.NET: **No**

# HiveMQ MQTT Client



**HIVEMQ**  
MQTT CLIENT

- Java MQTT 5 (and 3.1.1) Open Source Client library. Designed for production use
- Reactive Design with built-in backpressure handling
- Built for high scalability and embeddability. Supports tens of thousands messages per second!
- Available on [Github](#) under Apache 2 license
- Created and maintained by the HiveMQ and BMW CarIT folks





**SHOULD WE UPGRADE YET?**



+



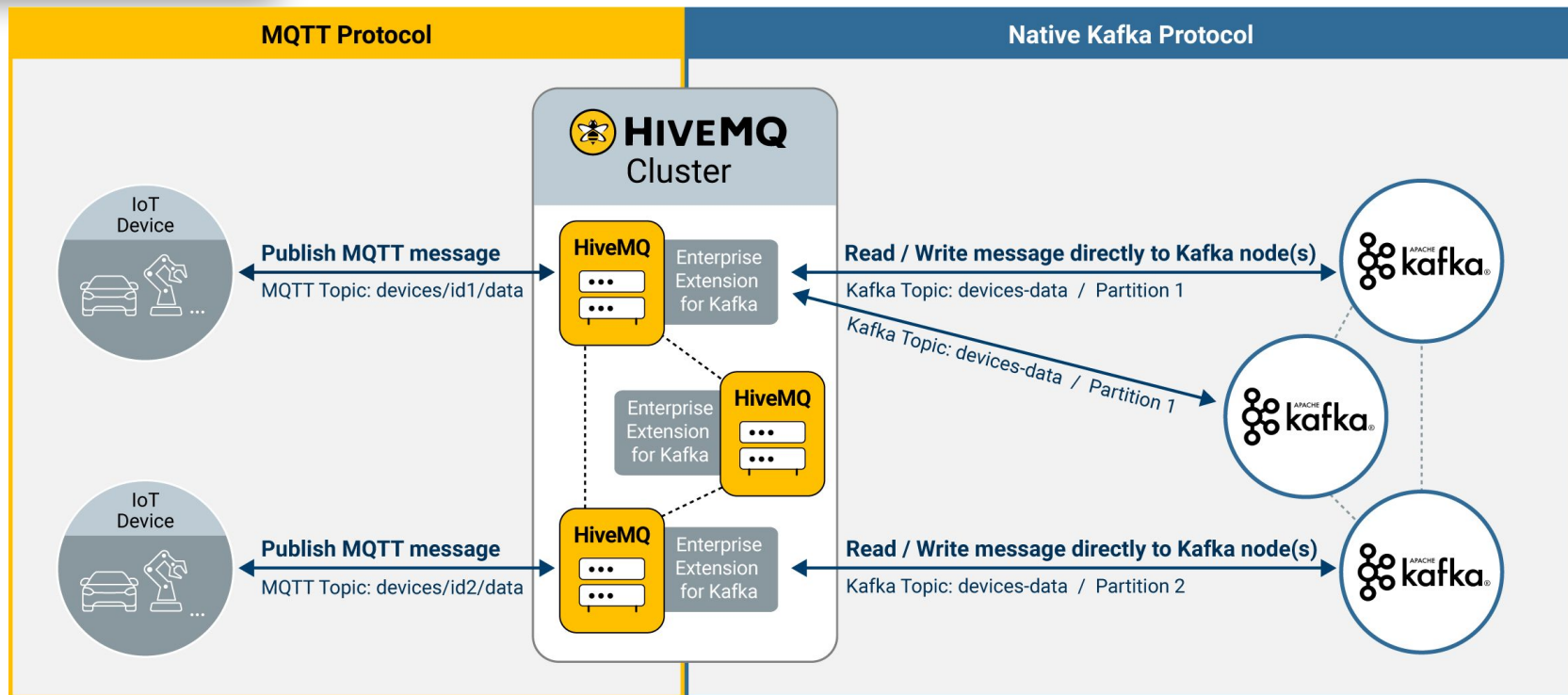
**Bonus!**

# HiveMQ Kafka Extensions

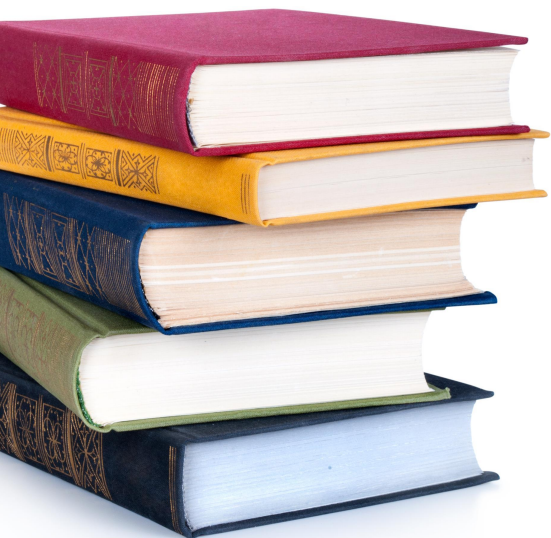


- Bi-Directional Integration of MQTT to Kafka (& vice versa)
- Already supports MQTT 5
- Allows to parse Kafka Messages and extract data for MQTT 5 headers (+ User Properties)
- Extreme throughput and high scale
- Hybrid mode with MQTT 3 and MQTT 5 possible

# Kafka



# Summary



- MQTT 5 has new features you shouldn't miss for new projects (especially Industry 4.0!)
- Be careful, MQTT 5 has optional features. Talk to your vendor to make sure you get all features you require
- MQTT 5 is used in production for mission critical use cases today
- Vendors like HiveMQ allow seamless transition and hybrid mode for MQTT 5

# Next Steps



# HIVEMQ

- **MQTT + Kafka** - Creating state of the art, highly scalable and fault tolerant messaging pipelines from device to application and back
- **Monitoring and observability**: How to find the needle in the haystack for IoT production deployments?
- **Cloud native MQTT**: How to run large scale deployments on Kubernetes and integrate with the cloud native ecosystem
- **Security**: How to integrate existing third-party systems like databases, REST APIs, Active Directory and OAuth?

Reach out to me directly to schedule a conversation: [\*dominik@hivemq.com\*](mailto:dominik@hivemq.com)

# Resources



[Get Started with MQTT](#)



[MQTT Essentials Series](#)



[Get HiveMQ](#)



# ANY QUESTIONS?

Reach out to [community.hivemq.com](https://community.hivemq.com)





# THANK YOU

Stay updated on upcoming webinars

[Subscribe to our Newsletter!](#)

