

Uniform representations are needed since if an auction is won the auction house directly creates a task in the winner's task list and therefore the representations must match.

Disadvantages

All groups are required to adapt their existing representations.

Bids are sent directly over HTTP (WebSub implementation)

Bids for open auctions are sent directly to the receiving group via HTTP request opposed to sending them over Websub.

Advantages

Each group only receives bid which are meant for their auctions

Disadvantages

Additional endpoints need to be provided

Placing shadow task directly in the winner's task list

Once an auction is won the auction house which launched the auction directly places a new task in the winner's task list.

Advantages

The winning group does not have to handle fetching a task once it has won an auction.

Small changes need to be made in the workflow since a shadow task is handled the same way as an internal task.

Disadvantages

Only the winning group is notified when an auction has ended.

Changing the AuctionId to a UUID

Auction Ids have been changed from Integers (increasing numbers) to Strings (UUIDs)

Advantages

Since in the MQTT implementation every Group receives all bids, unique ids simplify the determination if a bid is meant for an internal auction.

Disadvantages

All groups need to implement UUIDs and there is a (very) small chance of collisions

Comparison of W3C WebSub and MQTT

As noted in the exercise sheet, WebSub is in fact the W3C standardization of PubSubHubbub. It has started out as an extension of the Atom and RSS protocols, which are used to generate real-time change notifications¹. MQTT is designed for connections devices that have resource constraints or limited network bandwidth.

A bit of background

Both protocols are bi-directional protocols that follow the publish-subscribe design pattern. From the perspective of the OSI model, WebSub provides notifications through HTTP, whereas MQTT typically runs over a transport protocol only (TCP/IP). More generally, as both MQTT and WebSub are following the publish-subscribe pattern, they are favorable when it comes to scalability comparing to a polling approach. The major benefit is that this pattern implements loose coupling². If the subscriber fails, the publisher still functions

Transmission of information

The transmission of information differs when comparing both protocols. WebSub forms a network of publishers, subscribers, and hubs. WebSub is based on webhooks, which means you need to have an HTTP server able to handle requests coming from the web³. In MQTT Clients only interact with a broker and the interaction relies on the TCP protocol for data transmission. Due to the lightweight design the overhead can be much smaller compared to WebSub, which implements application layer protocols. A minimal MQTT control message can be as little as two bytes of data⁴.

The issue of selecting the protocol really boils down to effectiveness. For larger projects that are decentralized, WebSub can provides the additional functionality while maintaining the overall benefits of REST. For IOT devices that require minimal overhead MQTT might be more robust.

Use Cases

MQTT really shines when one needs the functionality of persistent sessions. Having a central entity as an event broker allows us to easily manages and tracks all client connection states⁵. In that sense MQTT is great for monitoring IOT devices in industry 4.0 environments, such as a smart factory. Consequently, an MQTT empowered device can act

¹ Web Sub – Common Cases and Implementations: <https://nordicapis.com/websub-common-cases-and-implementations/>

² Stack overflow Coupling Discussion about Publish-Subscribe Design Pattern: <https://stackoverflow.com/questions/58572415/how-does-publish-subscribe-design-pattern-establish-loose-coupling-between-compo>

³ How to implement PubSubHubbub: <https://blog.superfeedr.com/howto-pubsubhubbub>

⁴ MQTT – Conceptual Deep Dive: <https://ably.com/topic/mqtt>

⁵ <https://en.wikipedia.org/wiki/MQTT>

as an Industrial internet of things (IIoT) Gateway, bridging smart field devices to cloud servers⁶. It occurs minimal overhead during communication and aims to reduce network bandwidth, allowing it to scale more easily compared to WebSub.

In contrast, WebSub's design to generate real-time change notifications is great for news and forum updates. Additionally, the context of application might demand WebSub in situations where subscribers are dynamically changing, and their availability cannot be assured.

⁶ MQTT and Industry 4.0: <https://reonix.com/mqtt-industry-4-0-and-the-future-of-automation/>