

УДК 004.428.4

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СИСТЕМЫ УПРАВЛЕНИЯ УСТРОЙСТВАМИ ПО ПРОТОКОЛУ MQTT И СИСТЕМЫ УПРАВЛЕНИЯ СОСТОЯНИЕМ ВЕБ-ПРИЛОЖЕНИЯ

Лазарев И. С. (Университет ИТМО)

Научный руководитель – доцент, к.п.н, Государев И. Б.
Университет ИТМО

В докладе производится анализ работ по данной тематике, описывается текущее состояние понятий, применимых к системе управления состоянием веб-приложения и системе управления устройствами, с использованием протокола MQTT, выявляются как отличительные, так и общие черты исследуемых систем, определяется возможная сфера применения объединённой системы. Исследование проводится на основе системы управления состоянием фреймворка Svelte и библиотеки MQTT.js, реализующей протокол обмена сообщениями MQTT на языке программирования JavaScript.

Введение. Сложность веб-приложений постоянно возрастает. Так, за последние несколько лет появилось большое количество фреймворков, позволяющих разрабатывать сложные веб-приложения. Постепенно становится стандартным использование вместе с этими фреймворками библиотек, реализующих так называемое «состояние» (state) приложения. В некоторых фреймворках, таких как Svelte, данная функциональность является встроенной, в то время как в других требуется подключение сторонних библиотек, например часто встречающимся сочетанием таких фреймворка и библиотеки является связка React и Redux. В то же время за последнее время фактическим стандартом отрасли интернета вещей (IoT) становится протокол MQTT (Message Queuing Telemetry Transport). Данный протокол предлагает лёгкую встраиваемость в системы, небольшую нагрузку на канал связи и простоту в использовании. Изначально, данный протокол был ориентирован на работу с телеметрией от различных датчиков. Обмен сообщениями в рассматриваемом протоколе осуществляется между клиентом, играющим роль издателя или подписчика и брокером сообщений. Необходимо отметить, что в рассматриваемых понятиях есть некоторые общие черты. Так, и MQTT-брокер и объект состояния являются реализацией поведенческого паттерна проектирования «Посредник».

Теоретической основой проводимого исследования выступили работы зарубежных (Eugster P.T., Rahimian F., Berstel B. и другие) авторов, посвященных теме состояния веб-приложений и теме управления устройствами интернета вещей.

Для решения задач исследования использовался комплекс теоретических методов исследования, таких как сравнительный анализ, обобщение, синтез, методы индукции и дедукции.

Основной информационной базой проводимого исследования послужили поисковая система по текстам научных публикаций Google Scholar и сервис ResearchGate.

Основная часть. Большая часть библиотек, реализует функциональность состояния веб-приложения с помощью поведенческого паттерна проектирования «Посредник». В таких случаях, в качестве посредника выступает объект состояния, в то время как клиенты (например, поля веб-формы) выступают в роли издателя и/или подписчика.

Парадигма «Издатель/Подписчик» используется в качестве мощного коммуникационного протокола для разработки широкого спектра приложений. В этой парадигме коммуникационные объекты слабо связаны друг с другом с точки зрения времени, пространства и синхронизации. Таким образом, парадигма может поддерживать такие полезные и гибкие характеристики как анонимность, связи «многие-ко-многим» и асинхронность, которые являются критичными для распределенных систем.

MQTT – это чрезвычайно простой и легкий протокол обмена сообщениями с парадигмой «Издатель/Подписчик», разработанный для устройств с ограниченными возможностями и сетей с низкой пропускной способностью, высокой задержкой или недостаточной надёжностью. Принципы проектирования данного протокола заключаются в том, чтобы минимизировать пропускную способность сети и требования к ресурсам устройства, одновременно пытаясь обеспечить надёжность и некоторую степень гарантии доставки. Эти принципы также делают протокол идеальным для устройств типа «машина-машина» (M2M) или «Интернета вещей», а также для мобильных приложений, где пропускной канал и заряд батареи имеют первостепенное значение. Так традиционные системы, использующие парадигму «Издатель/Подписчик» имеют централизованный брокер, для управления топиками, издателями и подписчиками. Наиболее популярные решения, такие как mosquitto, сохраняют сообщения, пришедшие от издателей. Это означает, что даже после отправки сообщения всем текущим подписчикам, данное сообщение не будет удалено из системы. В случае создания новой подписки, соответствующей топике сохраненного приложения, сообщение будет отправлено новому подписчику.

Факт сохранения данных брокером позволяет рассматривать брокер по аналогии с хранилищем веб-приложения. По определённому действию издателя брокер изменяет состояние всех подписчиков данного топика. При наличии клиента, сохраняющего все сообщения приложений, данный клиент может рассматриваться в качестве аналога состояния. Всё взаимодействие с данным клиентом происходит через брокер, что соответствует тому, как взаимодействие с состоянием происходит через хранилище (редьюсер).

Для сравнения управления устройствами по протоколу MQTT и управления состоянием веб-приложения будет рассматриваться реализация протокола MQTT для языка программирования JavaScript в виде библиотеки MQTT.js и система управления состоянием фреймворка Svelte. Данный выбор обусловлен, во-первых, удобством сравнения двух различных инструментов, реализованных с помощью одного языка программирования, и, во-вторых, наличием у автора определённого опыта работы с обоими системами.

Несмотря на то, что управление устройствами по протоколу MQTT не полностью аналогично управлению состоянием веб-приложения, созданного с помощью фреймворка Svelte их схожесть, позволяет предположить о возможности использования состояния веб-приложения в качестве клиента, сохраняющего все сообщения приложений.

Выводы. В ходе работы были исследованы такие понятия как состояние веб-приложения, управление этим состоянием, протокол передачи данных MQTT и, в частности, MQTT-брокер. Несмотря на то, что до сих пор применение протокола MQTT не рассматривалось применительно к веб-приложениям, эта возможность заслуживает дальнейшего подробного изучения.

Также, был проведён сравнительный анализ процесса управления объектом состояния веб-приложения и управления устройствами по протоколу MQTT.

Таким образом, следует отметить, что несмотря на определённые различия между MQTT-брокером и объектом состояния веб-приложения, концепции подхода к реализации управления данными системами близки к идентичным. Обе эти системы используют поведенческий паттерн проектирования «Посредник». Схожесть исследуемых систем позволяет в дальнейшем изучать возможность создания веб-приложений, не требующих использования веб-сервера в классическом понимании, для управления устройствами по протоколу MQTT.