



Varroa

MQTT-Scenario-Testing-Tool

Masters Level Study Project, Prof. PhD. Siebert
SS 2018 - WS 2018/2019

R. Atherton, S. Baier, S. Giebl, G. Held, Y. Weber, T. Weiden

Contents

1	Vision	2
2	Requirements	3
3	Scenario Concept	5
4	Architecture	6
4.1	Varroa Distributed System Architecture	6
4.2	Commander	7
4.3	Chunk Handshake	7
	List of Figures	9
	List of Tables	10

1 Vision

The Name of our MQTT-Testing-Tool (Varroa) is inspired by the varroa mite, which is a species of mite that infects honey bee colonies. This name has been chosen due to it working in a similar way but instead of infesting a hive, it tries to infest a broker. The inspiration for this name came from the broker 'HiveMQ' and it's branding. The basic use-case of Varroa is testing the resilience of brokers by creating load. Hereby load is defined by a number of MQTT-clients sending different sequences of MQTT-messages to the broker. Which sequences get carried out in which order is determined by a Scenario. A scenario defines the temporal execution as well as the amount of actions across a MQTT-network and the topology of the network. The motivation for the creation of this project was the lack of testability of MQTT-systems.

Varroa is organized as a distributed system, due to the impossibility of creating enough MQTT-clients on a single machine to overload a MQTT-broker, especially if the broker is also a distributed system.

2 Requirements

2 Requirements

#	Title	User Story	Importance
1	Transparency	Varroa has to be comprehensible for the user.	Must have high
2	10.000.000 MQTT Clients	Varroa has to be able to generate a large amount of clients.	Must have high
3	Scalability	Varroa should scale vertically with relatively low scaling costs.	Must have
4	Determinism	Varroa has to work in deterministic ways, meaning it should produce the same result for a Scenario every time.	Must have
5	Distributed	Varroa is a distributed System.	Must have low
6	Usability	Varroa has to be easily usable.	Very important
7	Code Quality	Varroa's coding quality should be very high.	Important
8	Stability	Varroa has to run in a stable manner.	Important
9	Resource efficiency	Varroa has to use the available computation and memory resources efficiently.	Important
10	User / Developer Guide	Varroa needs a User / Developer Guide.	Somewhat important
11	Automation capacity	Varroa should be automatable	Somewhat important

3 Scenario Concept

4 Architecture

Varroas architecture is organised as a distributed system, whereas there are two main roles in the system: the Agents and the Commander. The Commander is the central unit that passes work packages of the Scenario to the Agents. In contrast the Agents process the passed packages and create MQTT clients to execute the testing process.

4.1 Varroa Distributed System Architecture

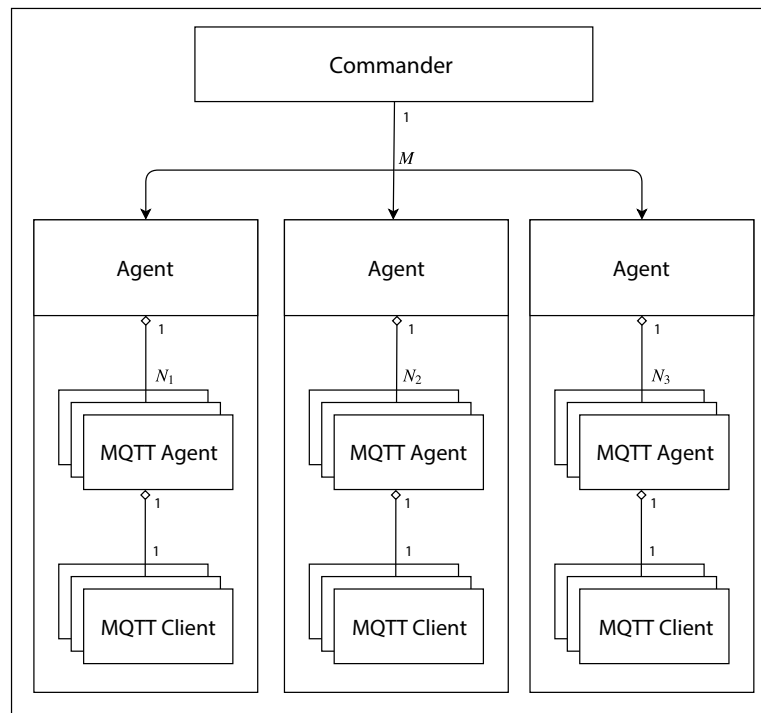


Figure 4.1: Varroa Distributed System Architecture

A Varroa Distributed System is composed of a Commander and multiple Agents. The Commander and all Agents are executed in separate Varroa Instances. Every Agent holds a number of MQTT Agents and each MQTT Agent manages one MQTT Client.

4.2 Commander

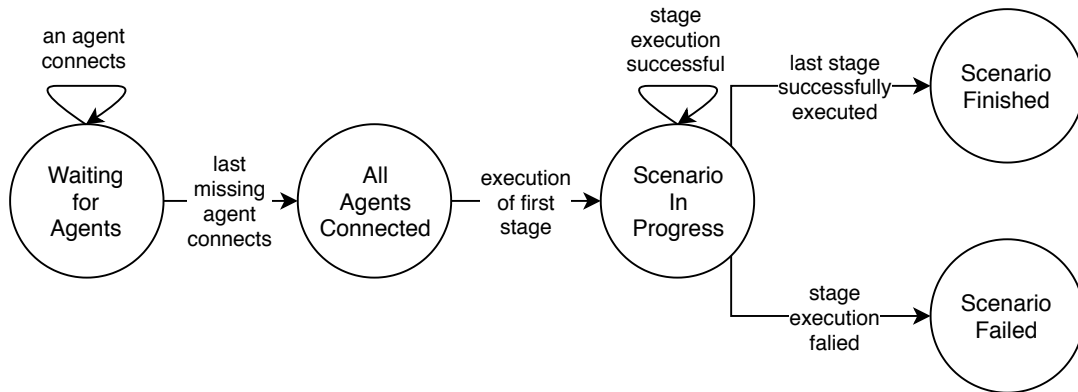


Figure 4.2: Commander States

4.3 Chunk Handshake

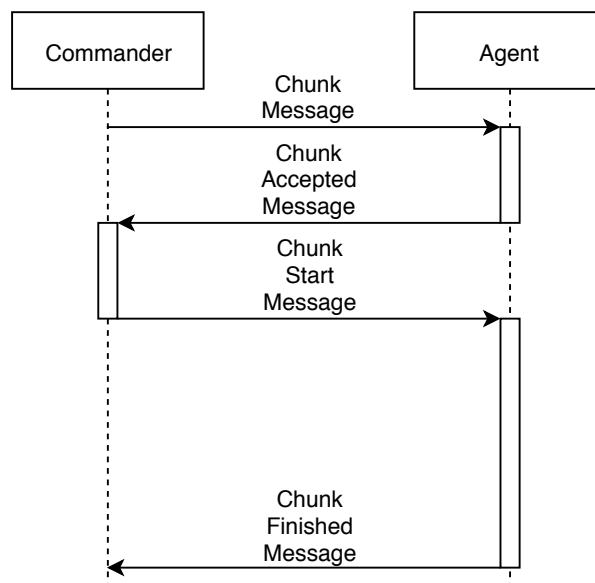


Figure 4.3: Chunk Handshake with Accept

4.3 *Chunk Handshake*

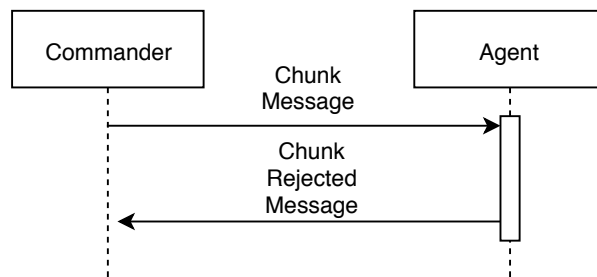


Figure 4.4: Chunk Handshake with Reject

List of Figures

4.1	Varroa Distributed System Architecture	6
4.2	Commander States	7
4.3	Chunk Handshake with Accept	7
4.4	Chunk Handshake with Reject	8

List of Tables