



# **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

<b>1</b>	<b>Vision</b>	<b>2</b>
<b>2</b>	<b>Concepts</b>	<b>3</b>
2.1	Definitions . . . . .	3
2.2	Requirements . . . . .	4
	<b>List of Figures</b>	<b>6</b>
	<b>List of Tables</b>	<b>7</b>

# 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 Concepts

To understand the workings of Varroa, we will have to take a look at the different parts that make up the system.

### 2.1 Definitions

**Varroa Distributed System** An orchestration of multiple Varroa Instances consisting of one Commander and at least one Agent.

**Commander** The Commander is a part of the Varroa Distributed System, that parses the scenario, generates chunks and distributes them to the Agents. Only one Commander exists in a Varroa distributed system.

**Agent** The Agent is part of the Varroa Distributed System. It receives Chunks from the Commander and passes them to its MQTT-Agents. A Varroa distributed system contains at least one Agent.

**Varroa Instance** A running Varroa process in a single JVM, can be either Commander or Agent.

**MQTT Agent** Part of the load structure, that takes chunks and executes them with MqttBee clients.

**MQTT Client** A MqttBee client used to execute the Commands that are defined in the Chunks.

## 2.2 Requirements

**Scenario** A scenario is an abstract representation of a real MQTT-Use-Case. It defines the topology of all participating MQTT clients and brokers. The scenario enables the simulation of a large amount of MQTT clients.

**Client group** A group of Clients that share similar behaviour and properties.

**Command** A command is an abstract representation of a work step that must be executed by a MQTT client.

**Chunk** The scenario is split in Chunks by the Commander and then those Chunk are distributed to the agents.

## 2.2 Requirements

## 2.2 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

## List of Figures

## List of Tables