# 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

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. Its own architecture is loosely based on the concepts of MQTT, whereas there are two main parts of the system: the Agent and the Commander. The Commander is the central unit that passes the data of the Scenario to the Agents. In contrast the Agents processes the passed data and creates MQTT-clients to start the testing process. Due to the distributed Nature of Varroa there were alot of architectural challenges, for example the even distribution of load across the agents.

# 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 | Usabillity | 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