

# HANDBUCH DEMONSTRATOR ZUR EVALUIERUNG EINES DEZENTRALEN EVENTSYSTEMS

Version 1.0 vom 28.12.2018  
Bernd Prünster – [bernd.pruenster@a-sit.at](mailto:bernd.pruenster@a-sit.at)

## Inhaltsverzeichnis

Inhaltsverzeichnis	1
1. Einleitung	1
2. Systemvoraussetzungen	1
3. Installation	1
3.1. Kompilieren von Source Code	1
3.2. Bereitgestelltes Paket	2
4. Inbetriebnahme	2
4.1. Konfiguration	2
4.2. Starten	2
4.3. Testen	2
5. Handhabung	2
6. Lizenz	2

## 1. Einleitung

Beim vorliegenden Demonstrator handelt es sich um ein Framework – eine Programmbibliothek – zur Umsetzung eines verteilten Eventsystems. Im Gegensatz zu Frameworks, wie z.B. *RabbitMQ*<sup>i</sup> ist diese Umsetzung vollständig dezentral, d.h. es gibt keine zentrale Instanz wie z.B. einen Broker. Im Kern basiert die Umsetzung auf einer Implementierung des Kademia-Protokolls. Die Programmbibliothek muss im konkreten Semantik von Events erweitert werden, um tatsächlich einsatzbereit zu sein. Da es sich hierbei jedoch um Applikationsspezifika handelt, wurde bewusst darauf verzichtet. Details hierzu sind Abschnitt 3.2 zu entnehmen.

## 2. Systemvoraussetzungen

Die Implementierung ist in *Kotlin* umgesetzt und verwendet *Apache Maven* als Buildsystem. Konkret wird Apache Maven 3.5 und Java 8 vorausgesetzt.

## 3. Installation

Da es sich beim bereitgestellten Paket um eine Programmbibliothek handelt, ist keine Installation vorgesehen.

### 3.1. Kompilieren von Source Code

Die Kompilation erfolgt über Apache Maven durch Ausführen von `mvn compile`.

## 3.2. Bereitgestelltes Paket

Das bereitgestellte Paket ist in Kotlin umgesetzt und versteht sich als Basis um ein dezentrales, verteiltes Eventsystem umzusetzen. Das betreffende Modul befindet sich im Ordner `evt`. Die dafür notwendigen Mechanismen, sowie der Unterbau in Form eines strukturierten P2P-Netzwerks sind vorhanden und einsatzfähig. Konkret muss die `Action`-Klasse um Aktionen mit applikationsspezifischer Semantik erweitert werden. Ebenso sind entsprechende Serialisierungs- und Deserialisierungsfunktionen als Extension-Functions der `ByteBuffer`-Klasse umzusetzen (siehe `Events.kt`).

## 4. Inbetriebnahme

Da es sich um eine Programmbibliothek handelt, ist keine eigenständige Inbetriebnahme möglich.

### 4.1. Konfiguration

Das bereitgestellte Paket verfügt über keinerlei nicht-programmatische Konfigurationsparameter. Netzwerkknoten werden über eine Referenz auf eine Instanz der `Config`-Klasse des Moduls `h-kad` gestartet.

### 4.2. Starten

Da es sich um eine Programmbibliothek handelt, ist kein eigenständiger Start möglich.

### 4.3. Testen

Eine JUnit-Test-Suite liegt bei.

## 5. Handhabung

Tests im Modul `evt` veranschaulichen die Handhabung der Programmbibliothek.

## 6. Lizenz

Das bereitgestellte Projekt wird unter den Bedingungen der Open-Source-Lizenz für die Europäische Union (EUPL) V1.1 bereitgestellt (siehe `Lizenz.pdf`) und verwendet Bibliotheken der Stiftung SIC (siehe `SIC_LICENSE.txt`).

---

<sup>i</sup> <https://rabbitmq.com/>