

Ausschreibung Abschlussarbeit (Bachelor/Master Informatik)

Performance- und Feature-Analysen nachrichtenbasierter Protokolle

Motivation. Anwendungen in den Domänen wie Ubiquitous Computing, Internet of Things oder Sensornetzwerke nutzen häufig Geräte aus verteilten, dynamischen Ensembles. Die genaue Zusammensetzung der erreichbaren Systemumgebung ist meist erst zur Laufzeit bekannt. Jedoch müssen Entwickler ihre Anwendung zur Entwurfszeit umsetzen. Hier unterstützen Programmierabstraktionen bei der Implementierung. So können bspw. Richtlinien in Form von Metadaten in der Anwendung spezifiziert werden, welche zur Laufzeit durch eine Ausführungsumgebung in konkrete Instruktionen überführt werden.

Grundlage. Eine Programmierabstraktion sind *Dynamische Mengen*, die es erlauben gegen ein Interface zu programmieren, welches eine einzelne Instanz repräsentiert, jedoch durch die Laufzeitumgebung mehrere Instanzen verwalten kann¹. Entsprechend der Selektionskriterien werden Funktionsaufrufe repliziert und mehrere Rückgabewerte aggregiert. Zur Reduzierung der Netzwerklast wird adaptiv bei Schnittstellenmethoden, die nicht den Zustand des Gerätes verändern (Getter-Methoden), zwischen pull- und push-basierter Kommunikation umgeschaltet basierend auf Anfrageraten der Anwendung und Aktualisierungsraten des Gerätes. Hierbei werden neben klassischen Kommunikationsmustern wie Request/Reply auch Publish/Subscribe angewendet. Das Erstgenannte lässt sich auch auf letzteres abbilden.

Ziel der Arbeit. Für die Evaluierung bietet sich eine Auswahl aus nachrichtenbasierten Protokollen zum Einsatz in ereignisbasierten Systemen (Message-oriented Middleware) an. Beispiele: Advanced Message Queuing Protocol (AMQP), Data Distribution Service (DDS), Java Messaging Service, OpenMAMA, OpenWire, Message Queue Telemetry Transport (MQTT), Streaming Text Oriented Messaging Protocol (STOMP), Extensible Messaging and Presence Protocol (XMPP), RESTful Messaging Service (RestMS), Web Services Eventing (WS-Eventing).

Die eingehende Recherche stellt Gemeinsamkeiten und Unterschiede der jeweiligen Spezifikationen gegenüber und zeigt verfügbare Implementierungen auf. Hervorzuhebende Merkmale der Spezifikationen bzw. Implementierungen sind bspw. Paradigmen: Abbildung von request/reply, publish/subscribe; Datenstrukturen: dead-letter queue, peek queue; Transportprotokolle; Routing: broadcast, multicast, distribution list; Sicherheit/Verschlüsselung; Nachrichten: Dienstgüte, Prioritäten, Bestätigungen, Zustellgarantien, Transaktionen; Plattformen. Die abschließende Evaluierung beinhaltet Performance-Analysen. Hierbei werden aussagekräftige und vergleichbare Testreihen konzipiert, umgesetzt und grafisch dargestellt (R, gnuplot,...).

Die vorgenannten Ausführungen stellen lediglich eine Diskussionsgrundlage dar. Den genauen Inhalt einer Abschlussarbeit erarbeiten wir in einem persönlichen Gespräch.

¹Eigene Veröffentlichungen zur Programmierabstraktion *Dynamische Mengen*: <https://doi.org/10.1145/2834965.2834973>, <https://doi.org/10.1145/3007203.3007213>

Matthias Prellwitz, M.Sc.

Fon +49 381 498-0
+49 381 498-7638
Fax +49 381 498-7622

Mail matthias.prellwitz@uni-rostock.de
www.wava.informatik.uni-rostock.de/prellwitz.html

www.informatik.uni-rostock.de

Architektur von
Anwendungssystemen (AVA)

Chair:
Prof. Dr.-Ing. habil Gero Mühl

Sitz Albert-Einstein-Straße 22
Konrad-Zuse-Haus
18059 Rostock
Fon +49 381 498-7630
Fax +49 381 498-7622

Mail ava.office@uni-rostock.de
www.wava.informatik.uni-rostock.de