

Wochenbericht VI - Praktikum "Verteilte Systeme": Middleware und Watchdog

Phillip Patt

June 3, 2025

1 Mitglieder des Projektes

Mitglied des Projektes	Aufgabe	Fortschritt	Zeiteinsatz	Check
Manh-An David Dao	Sequenzdiagramm Watchdog	80%	2h	in review
Jannik Schön	Einarbeitung Namensauflösung und Service Discovery	60%	2h	in progress
Marc Siekmann	Blockdiagramm u. Sequenzdiagramme Middleware	60%	2h	in progress
Philipp Patt	Proof of concept des middleware interfaces	20%	2h	in progress

2 Zusammenfassung der Woche

Diese Woche fand die Ausarbeitung der Bausteinsicht und die erste Implementierung der Laufzeitsicht der Middleware statt. Ausserdem wurde sich mit Watchdog, Namensauflösung und Proof of concept bzgl RPC auseinandergesetzt.

Wesentliche pull-requests/commits sind in den Fußnoten hinterlegt.

3 Bearbeitete Themen und Schlüssel Erkenntnisse

3.1 Sequenzdiagramm Watchdog

Eine erste Version des Sequenzdiagramms für den Watchdog wurde entworfen¹. Dabei wurde festgestellt, dass der Watchdog außerhalb des überwachten Systems sein muss². Hierbei ist uns aufgefallen, dass geklärt werden muss, an welchen Stellen der Watchdog in dem Gesamtsystem implementiert werden muss. Diese Darstellung dient als Grundlage für weiterführende Gespräche und die technische Abstimmung³. Ziel ist es, ein gemeinsames Verständnis für das Verhalten der beteiligten Applikationen im Falle eines Verbindungsverlusts einer Node zu schaffen.

Das Diagramm bildet initiale Überlegungen zur Überwachung und Reaktion bei Kommunikationsausfällen ab und soll im weiteren Verlauf verfeinert und um spezifische Anwendungsfälle ergänzt werden.

¹https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sole/-/merge_requests/23

²https://github.com/scimbe/vs_script/blob/main/vs_script-first-v01.pdf S. 115

³https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sole/-/blob/7121a6166e17963d88de5c1cf893c6a51036d4e5/Docs/diagrams/Watchdog_01062025.png

3.2 Blockdiagramm u. Sequenzdiagramm Middleware

Auf Basis der Lösungsstrategie wurden die einzelnen Komponente und die Aufgaben der Komponenten für die Middleware bestimmt ⁴. Es wurde bestimmt, dass der einfache Forward-Proxy von jedem Client umgesetzt wird.⁵ Dafür wird es zunächst eine zentrale Namensauflösung geben.⁶ Weiterhin wurden Sequenzdiagramme eines einfachen RPC Aufrufs mit Namensauflösung entwickelt⁷, sowie der Verlauf der Registrierung. Hierfür werden die Funktionen in der Lösungsstrategie angepasst. Das Marshalling wird überarbeitet und insofern vereinfacht, dass es zukünftig keinen dedizierten Datentypen mehr gibt, sondern eine Listenstruktur übergeben wird, sodass nur primitive Datentypen gemarshallt werden.⁸

3.3 Einarbeitung Namensauflösung und Service-Discovery

Es wurde sich mit den verschiedenen Möglichkeiten der Namensauflösung beschäftigt. ⁹ ¹⁰ Ziel dabei war es ein grundlegendes Verständnis der Namensauflösung, aber auch der Service Discovery zu erhalten. Dabei sollen in der nächsten Woche die verschiedenen Möglichkeiten diskutiert werden, um die passende Lösung für unsere Aufgabenstellung zu finden.

In der Bearbeitung wurde der Bereich der Namensauflösung und von der Service Discovery vermischt, in der nächsten Woche soll hier wieder genauer getrennt werden.

3.4 Proof of concept des middleware interfaces

Um besser einschätzen zu können inwiefern die Schnittstelle zwischen Applikation und Middleware Sinnvoll gewählt ist wollten wir einen RPC mit den entsprechenden interfaces exemplarisch programmieren. Für diese haben wir uns, um weniger selbst implementieren zu müssen, überlegt grpc zu verwenden und in Wrapper zu hüllen die unseren Interfaces entsprechen. Auch wollten wir im Rahmen dessen mehrere Sprachen (java & c)¹¹ verwenden, um sprachspezifische Problem frühstmöglich festzustellen.

Das Verwenden von grpc hat sich in C als aufwändiger als gedacht herausgestellt da bspw. das Serialisieren von protobuf Formaten selbst implementiert werden müsste¹². Inwiefern der Umfang des Proof of Concept Sinnvoll gewählt ist oder ob er zu umfangreich ist sollte in den kommenden Wochen nochmal hinterfragt werden. Die Umsetzung in mehreren Sprachen erhöht bspw. den Aufwand massiv und bringt geringen Mehrwert.

4 Nächste Schritte

- Konsens über den Watchdog
- Entwickeln eines MVP der Applikation, um Middleware MVP, insbesondere RPC und Marshalling, ohne Namensauflösung zu testen zwischen 2 Nodes.
- Diskussion Service Discovery und Namensauflösung

⁴https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sole/-/merge_requests/21

⁵https://github.com/scimbe/vs_script/blob/main/vs-script-first-v01.pdf S. 95

⁶https://github.com/scimbe/vs_script/blob/main/vs-script-first-v01.pdf S. 213

⁷https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sole/-/merge_requests/22

⁸https://github.com/scimbe/vs_script/blob/main/vs-script-first-v01.pdf S.124 f.

⁹https://github.com/scimbe/vs_script/blob/main/vs-script-first-v01.pdf S. 212 ff.

¹⁰Maarten van Steen, Andrew S. Tanenbaum: *Distributed Systems*, January 2025, p. 347 ff.

¹¹<https://grpc.io/docs/languages/>

¹²<https://stackoverflow.com/questions/50953651/how-to-write-a-grpc-client-server-in-c>