

Wochenbericht 10 - RPC Übertragung, Watchdog

David Dao

July 2, 2025

1 Mitglieder des Projektes

Mitglied des Projektes	Aufgabe	Fortschritt	Zeiteinsatz	Check
Manh-An David Dao	Controller RPC Übertragung, Arc42 Überarbeitung	80%	4h	in review
Jannik Schön	Watchdog, Arc42 Überarbeitung	80%	4h	in progress
Marc Siekmann	ITS-Board View, RPC Übertragung in C	90%	5h	in progress
Philipp Patt	RPC Übertragung in JAVA, DNS	80%	6h	in progress

2 Bearbeitete Themen und Schlüssel Erkenntnisse

2.1 Arc42 Überarbeitung

Die Kapitel 4-7 wurden Überarbeitet¹.

2.2 Controller RPC Übertragung

Der Controller wurde implementiert, und erste Schritte zur RPC-Übertragung wurden durchgeführt. Ursprünglich war gRPC geplant, jedoch stellte sich heraus, dass es auf TCP basiert, welches verbindungsorientiert ist.² Diese Eigenschaft führt zu Problemen bei der Echtzeitkommunikation. Daher haben wir uns entschieden, eine eigene RPC-Lösung zu entwickeln, die speziell auf Echtzeitkommunikation ausgerichtet ist und UDP nutzt.

2.3 Watchdog/Heartbeat

In der ersten Iteration wurde sich auf einen zentralen Hotspot, bzw. eine zentralisiertes Heartbeat Protokoll verständigt³. Nach weiter Diskussion sind wir zu dem Schluss gekommen, dass ein solcher Ansatz ein zu großes Risiko in Hinsicht auf einem Single Point of Failure bietet⁴. Um dieses Risiko zu mindern, soll im nächsten Schritt eine Hierarchisches Heartbeat Protokoll⁵ implementiert werden. Dabei soll eine Out-of-Band Kommunikation genutzt werden. Jeder Prozess bekommt seinen eigenen Watchdog.

2.4 Fertigstellung der Applikation

Die Applikation wurde nach dem arc42 ⁶ fertiggestellt.

¹https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sole/-/merge_requests/29

²<https://grpc.io/blog/grpc-load-balancing/>

³https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sole/-/merge_requests/31

⁴https://github.com/scimbe/vs_script/blob/main/vs-script-first-v01.pdfS.198

⁵https://github.com/scimbe/vs_script/blob/main/vs-script-first-v01.pdfS.199

⁶https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sole/-/merge_requests/29

2.5 RPC Übertragung in C und JAVA

Für das ITS-Board wurde das IO nach der vorgeschriebenen FSM im arc42 Dokument ⁷ implementiert. Ebenfalls wurde ein Client der Middleware in C implementiert und beispielhaft eine JAVA Implementierung des Modells via RPC beispielhaft angesprochen. Bzgl. Watchdog muss noch ein Server implementiert werden. Zudem wurde ein Client und ein Server in JAVA umgesetzt. Dieses soll für die Applikationsstubs eingesetzt werden, die in JAVA umgesetzt sind. Das Marshalling wird mit JSON umgesetzt und mit UDP versendet.

2.6 DNS

DNS ist nach dem konzept der Service-Registrierung⁸ implementiert. Das Naming⁹ gruppiert mehrere RPCs in Services und der DNS löst die Kombination Service & Funtionsname auf. Als nächstes gilt es diesen in den diversen clients einzubinden.

3 Nächste Schritte

- Vervollständigung arc42 Applikation und Middleware
- Fertigstellung Watchdog und DNS
- Testen der Applikation mit RPC

⁷https://git.haw-hamburg.de/infwgi246/vs_praktikum-2025-sole/-/merge_requests/29

⁸https://github.com/scimbe/vs_script/blob/main/vs-script-first-v01.pdf S.213

⁹https://github.com/scimbe/vs_script/blob/main/vs-script-first-v01.pdf S.214