

# Verteilte Systeme Testdokumentation

## Bearbeiter:

- Matthias Adrian (Mtrk. 752237)
- Jan Zipprich (Mtrk. 757956)

## Aufgabe 1a:

Funktional / Nicht funktional	Beschreibung des Tests	Durchführung	Erwartete Ergebnis	Tatsächliches Ergebnis
Funktional	10 Sensordatenpakete an die Zentrale senden, und überprüfen ob auch tatsächlich 10 Pakete angekommen sind	Man kann mittels der Portainer Oberfläche die Ausgaben einsehen, und sehen ob die Daten entsprechend ankommen und korrekt ausgegeben werden	Es ist davon auszugehen, dass kein Paketverlust auftritt, weil Server und Client auf dem localhost laufen und wir auf der Senderseite ausschließlich Daten konform des Übertragungsformats senden	Daten werden konsistent ausgegeben und es scheint kein Paketverlust aufzutreten
Nicht Funktional	Es soll geprüft werden, wie hoch die Latenz zwischen dem Empfangen der Daten an der Zentrale bis zur Ausgabe der Daten auf der Konsole.	Es wird der Zeitpunkt erfasst, wann die Daten als byte Stream von der Zentrale empfangen wurden. Es wird der Zeitpunkt erfasst, wann die Daten auf der Konsole ausgegeben wurden (nach dem die print Methode aufgerufen wurde) und mit dem vorherigen Zeitpunkt verglichen. Das ganze wird für 10 Pakete gemessen.	Die Latenz liegt unter 1ms da der Dienst nach dem Empfangen und konvertieren der Daten diese sofort ausgibt.	Die Latenz liegt nach 10 Versuchen jeweils unter 1ms, womit diese Latzen zu ignorieren ist.

## Aufgabe 1b:

Funktional / Nicht funktional	Beschreibung des Tests	Durchführung	Erwartete Ergebnis	Tatsächliches Ergebnis
Funktional	Der Webserver liefert valide HTML Seiten mit Sensordaten. Auf der Seite <a href="https://validator.w3.org/">https://validator.w3.org/</a> kann man den von der zentrale zurückgegebenen html Code auf Korrektheit validieren	Es werden Daten vom TCP Webserver über GET Anfragen abgefragt und mithilfe eines HTML Validator überprüft.	Die zurückgelieferten Seiten halten sich an den HTML Standard und geben keine Fehlermeldung der Analyse.	Als Warnung wurde gemeldet, dass ein lang Attribut fehlt (zur Festlegung der Sprache der html Seite) und das head Element fehlte zur Deklaration des Titel. Diese Fehler wurden behoben.
Nicht Funktional	Die Webseiten des HTTP Servers sind übersichtlich aufgebaut und auch bei großen Datenmengen ist eine gute Lesbarkeit sichergestellt.	Man lässt den Server eine Weile laufen und ruft die entsprechenden Webseiten auf.	Da die Sensordaten untereinander ausgegeben werden, ist eine gute Lesbarkeit wahrscheinlich NICHT sichergestellt.	Schon nach relativ kurzer Zeit werden zu viele Sensordaten angezeigt und die Lesbarkeit ist eingeschränkt. Vor allem bei der Webseite für alle Sensordaten ist dies der Fall. Man könnte die einzelnen Typen von Sensoren gruppieren (z.b. in einen Spoiler packen) oder mittels CSS kann man ein dropdown Menu realisieren, dadurch kann die Lesbarkeit verbessert werden.

## Aufgabe 2:

Funktional / Nicht funktional	Beschreibung des Tests	Durchführung	Erwartete Ergebnis	Tatsächliches Ergebnis
Funktional	Es werden alle Daten die von der Zentrale empfangen wird an den Anbieter weitergeleitet und in einer Datenbank persistiert.	Es wird die Logausgabe von der Zentrale über einen Zeitraum von 20 Sekunden betrachtet und mit dem Inhalt der Datenbank abgeglichen.	Die von der Zentrale empfangenen Daten sind in der Datenbank vollständig und identisch vorhanden.	Es sind alle von der Zentrale Empfangenen Sensordaten in der Datenbank des Anbieters vorhanden.
Nicht Funktional	Die Zuverlässigkeit des Thrift Servers wird getestet.	Der Server wird eine Weile laufen gelassen wird und es wird geschaut ob Fehlermeldungen/Exceptions ausgegeben werden.	Es kann durchaus vorkommen, dass der Docker Container für den Anbieter (d.h. der Thrift Client) VOR der Zentrale (dem Thrift Server) gestartet wird.	Dies ist tatsächlich vorgekommen. Dies kann man beheben, indem man einen Delay implementiert (keine geschickte Lösung), oder indem mehrere Verbindungsversuche seitens des Anbieters (=Thrift Client) unternommen werden.

### Aufgabe 3:

Funktional / Nicht funktional	Beschreibung des Tests	Durchführung	Erwartete Ergebnis	Tatsächliches Ergebnis
Funktional	Bei einem Ausfall der Zentrale werden die Messdaten auf dem MQTT Client zwischengespeichert , bis die Zentrale wieder verfügbar ist, da der Quality of Service Level auf 1 eingestellt ist. Das heißt der Sender müsste die Daten zwischenspeichern, bis er ein PUBACK Paket von dem MQTT Broker erhält.	Die Zentrale wird für einen Zeitraum von einer Minute aus und wieder eingeschaltet. Nach dem Einschalten werden die Logs mit den Sensoren verglichen um zu sehen, ob alle Daten nachträglich Verarbeitet wurden.	Alle Logs die während des Ausfalls generiert wurden, werden von der Zentrale abgearbeitet und sind vollständig vorhanden.	Während des Tests wurden von den 4 Sensoren jeweils 60 Datensätze (240 Insgesamt) generiert welche nach dem Wiedereinschalten der Zentrale verarbeitet wurden.
Nicht Funktional	Es können von außerhalb Verfälschte Daten in das System eingespeist werden, da die IP Adresse des Senders von der Zentrale nicht überprüft werden kann.	Es wird ein weiterer vermeintlicher Sensor hinzugefügt, der den gleichen Sensor Name sowie Typen eines bestehenden Sensors hat. Dieser sendet wie alle anderen Sensoren Daten an den MQTT Broker.	Die Daten, die an der Zentrale ankommen können nicht mehr zwischen dem Fake Sensor und dem echten Sensor unterschieden werden, weil die IP-Adressen der Clients nicht überprüft werden.	Die Sensordaten können nicht voneinander unterschieden werden. Der einzige auffallende Unterschied ist, dass mehr Daten von einem Sensor empfangen werden.