

# Testat VS 2019

## Titel: sicheres verteiltes Schreiben

### Voraussetzung

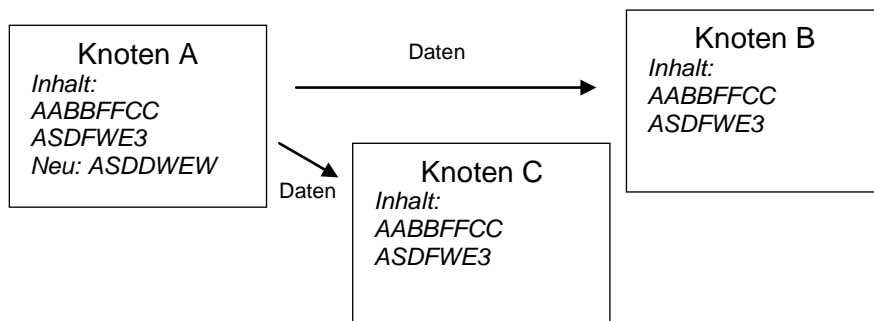
Als Voraussetzung für dieses Testat dienen die Laborübungen im Fach "Verteilte Systeme". Der Quellcode hieraus kann als Basis für die Rahmenbedingungen verwendet werden.

### Ziel

Ziel dieser Arbeit ist es, Daten im verteilten System sicher zu schreiben, sodass auf jedem Knoten ein konsistenter Stand der geschriebenen Daten entsteht.

Bei dem System kann jeder Knoten Daten zum Schreiben in das System eingeben und diese werden konsistent gehalten. Ein neuer Knoten kann jederzeit hinzukommen oder den Verbund verlassen und hat (solange Teil des Systems) eine konsistente Sicht auf die Daten.

Das Schaubild 1 zeigt einen exemplarischen Aufbau.



Beispielhafter Ablauf:

1. Knoten A will Daten bekanntgeben und sendet diese an B und C
2. Alle Knoten haben einen konsistenten Stand, nachdem diese den Daten „haben“

Das System soll im Fehlerfall immer noch lauffähig sein und sollte mit min. folgenden Fehlerfällen definiert umgehen können:

- Knoten bricht bei der Kommunikation an
- Knoten fällt aus
- Knoten tritt dem System nicht korrekt bei/bricht dabei ab
- Daten werden nicht korrekt kodiert geliefert

Das System soll in kurzer Zeit in der Lage sein, auch im Fehlerfall einen konsistenten Stand wiederherzustellen.

Die Studenten wählen eine Verteilungsstrategie aus, dokumentieren und implementieren diese. Die Studenten testen ihre Implementierung und reflektieren das Ergebnis hinsichtlich der Skalierbarkeit und der Performance anhand der Response-Zeiten unter Last.

## Abgrenzung

Für das Testat sollen folgende Punkte als Grundlage dienen.

### **Betriebssystem und Laufzeitumgebung**

Das Betriebssystem ist freigestellt, jedoch soll die Anwendung auf einem Raspberry PI mit einer JRE Laufzeitumgebung  $\geq$  JRE-1.8.0 laufen, die als Abnahmeumgebung zu berücksichtigen ist.

### **Serverumgebung und Knoten**

Die Anwendung soll min. 100 Knoten unterstützen können. Als Knoten zählt eine Knotenanwendung. Jedoch sind die Anwendungen auf min. 2 Rechner verteilt werden. (Abnahme erfolgt auf min. 2 Raspberry PIs).

## Daten

Um die Daten identifizierbar zu machen, muss jeder Datensatz min. folgende Inhalte aufweisen.

- Zeitstempel
- Quell-ID
- Daten (in Textform)

Weitere Datenelemente sind je nach Implementierung notwendig und werden entsprechend gewählt.

Die Daten können entweder zufällig erzeugt werden, oder manuell eingetragen werden. Für Performance und Lasttest-Szenarien wird ein Automatismus empfohlen.

### **Zeitpunkt**

Als spätester Abgabetermin ist der 31. Dezember 2019 23:00 Uhr festgelegt. Als Eingangskriterium dient der Zeitstempel des Empfangens. (!)

### **Form**

Die Abgabe findet in elektronischer Form (Email) statt. Die Email wird an folgende Adressen geschickt:

to: [patrick.jungk@boxwork.eu](mailto:patrick.jungk@boxwork.eu)  
cp: [harkle@gmx.net](mailto:harkle@gmx.net)

Der Anhang darf eine maximale Größe von 10 MB nicht überschreiten, da die Email sonst nicht zugestellt wird. Die Dateien im Anhang sind gepackt als ZIP-Datei (eine für alle Anhänge) anzufügen.

Im Betreff jeder Email ist folgende Struktur anzugeben:

*TESTAT19\_<KURS>\_<GUPPE>\_<MATRIKELNUMMER>.zip*

z.B.:

***TESTAT19\_TINF17A\_01\_1535485.zip***

## ***Inhalt***

Der Inhalt des Testates umfasst:

<b>Bezeichnung</b>	<b>Beschreibung</b>	<b>Umfang</b>
Quellcode	der Gesamte Quellcode samt verwendeter Ressourcen und Kommentaren	-
Lauffähige Ausführung	Ausführbare Version des Programms samt Startroutine	-
Analyse	Aufgabenanalyse	Max. 1 Seite
Grobkonzept	Konzeptionelle Darstellung des Ablaufes und beschreibender Text (Ablaufplan, Grobarchitektur)	Max. 5-8 Seiten (Kurz halten)
Testplan	Abnahme- und Testplan für die "Abnahme", Protokoll und Ergebnisse.	10 Testfälle reichen aus
Performance-Auswertung	Hinsichtlich der Response – Zeit für 2,10,50,100 Client, inkl. Bewertung	1 Auswertung inkl. Text, max. 1 Seite