

Nachdenkzettel Logging

Sara Tietze (40800), Pia Schilling (40905), Merve Özdemir (41709)

1. Kennzeichnen Sie in der Config die Stellen wo über das

- was geloggt wird
- wieviel geloggt wird
- wo geloggt wird
- wie geloggt wird

entschieden wird

```
<Configuration>
  <Appenders>
    <File name="A1" fileName="A1.log" append="false">
      <PatternLayout pattern="%t %-5p %c{2} - %m%n"/>
    </File>
    <Console name="STDOUT" target="SYSTEM_OUT">
      <PatternLayout pattern="%d %-5p [%t] %C{2} (%F:%L) - %m%n"/>
    </Console>
  </Appenders>
  <Loggers>

    <!-- You may want to define class or package level per-logger rules -->
    <Logger name="se2examples.core.businessLogic.VehicleManager" level="debug">
      <AppenderRef ref="A1"/>
    </Logger>
    <Root level="debug">
      <AppenderRef ref="STDOUT"/>
    </Root>
  </Loggers>
</Configuration>
```

2. Geben Sie je ein Beispiel wann Sie den loglevel

- **error**: Exceptions, Datenzugriffe bei Daten die es nicht gibt
- **info**: zum Programmstart, bei falschen Eingaben
- **debug**: Analyse von Sachverhalten von Fehlern oder Bugs (bei gezielter Überwachung) verwenden

3. Sie verwenden einen FileAppender für das Logging. Jetzt soll Ihre Application im Datacenter laufen. Was machen Sie mit dem FileAppender?

- Die Log- Dateien verschlüsseln, damit Dritte diese nicht lesen können.
- Rolling FileAppender, damit es zu keinem Speicherproblem kommt

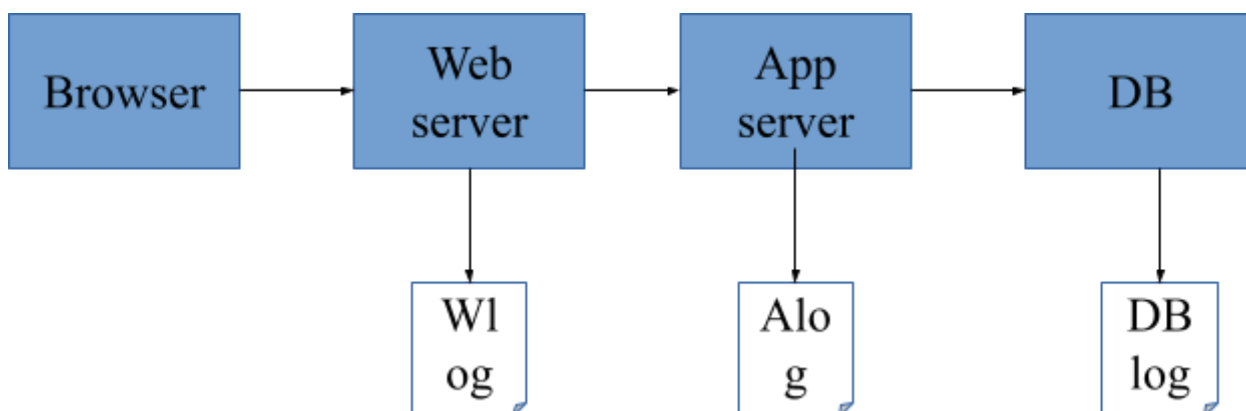
4. Macht logging Ihre Application langsamer? Was passiert wenn Sie `log.debug(„foobar“)`; aufrufen? Wie sollte sich das Logging Subsystem verhalten?

Ja, aber insignifikant, außer bei Games.

Wenn foobar aufgerufen wird, wird das Programm auch verlangsamt aber nur minimal.

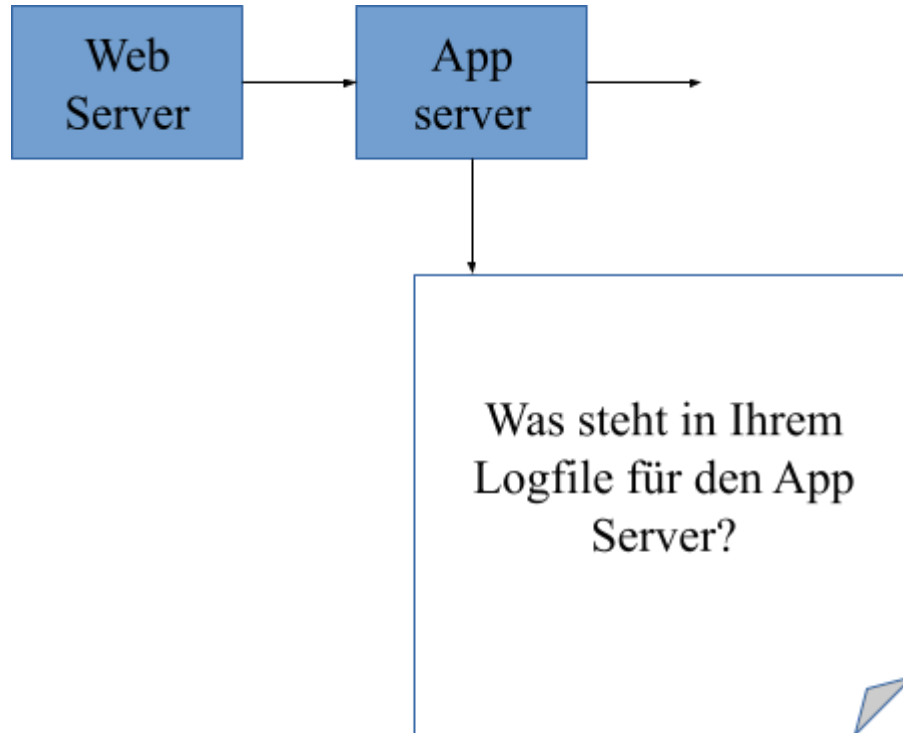
Man muss auf Übersichtlichkeit achten → wird wild gemischt

5. Ein Request an Ihre Application durchläuft einen Proxy Server, dann einen Web Server, dann einen Application Server und dann die Datenbank. Auf jedem Server loggen Sie die Requests. Welches Problem tritt auf?



Wir vergeben eine Korrelations-ID (ein Token) und dieser wird an jede Maschine weitergereicht, so weiß man um welchen Log es sich handelt (von welchem Programm/Server der Log ist).

6. Was sollten Sie pro Komponente/Tier loggen?



- Zeitstempel:
 - Start und Ende von Aufrufen
 - Wann geht es raus?/Wann kommt es zurück?
 - Request- Laufzeiten
- Status
 - CPU-Auslastung
 - wie viele Threads

7. Aus Geschwindigkeitsgründen halten Sie teure DB-Connections auf Vorrat in einem Pool. Jeder Request vom Client braucht dann eine Connection. Der Pool hat die Methoden:

```
DB Connection con = ConnectionPool.getConnection();  
ConnectionPool.freeConnection( DBConnection dbCon);
```

Was loggen Sie in Ihrem App Server? Oder anders gefragt: Was wollen Sie beim Umgang mit dem Pool als Software-Architektin wissen?

- Ob man sich connections sparen kann.
- Wie viele Connections nach jedem getConnection frei sind.