

## Nachdenkzettel Logging

Firaz Ilhan, Jens Schlegel, Maximilian Dolbaum

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: die Anwendung stürzt ab
- info: Benutzer hat ein Button angeklickt
- debug: Ein neues Objekt wird erstellt

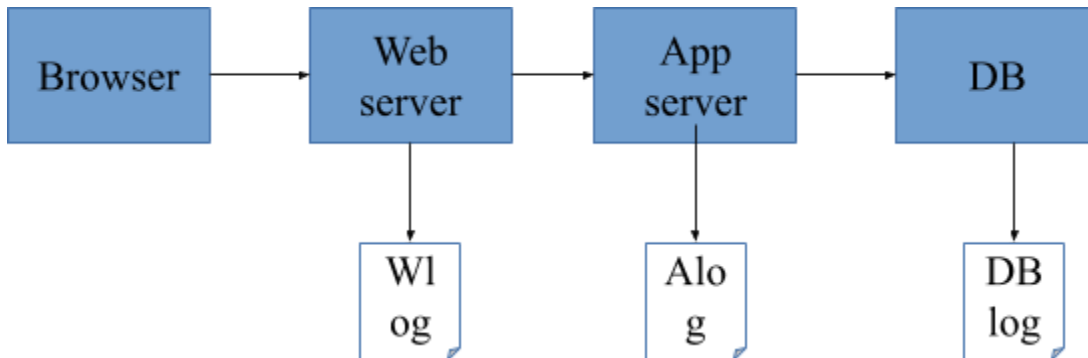
3. Sie verwenden einen FileAppender für das Logging. Jetzt soll Ihre Application im Datacenter laufen. Was machen Sie mit dem FileAppender?

Die Dateigröße der Loggingdatei begrenzen, sodass die Datei nicht so groß wird. Möglicherweise möchte man sowieso kein log von vor 2 Wochen haben.

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

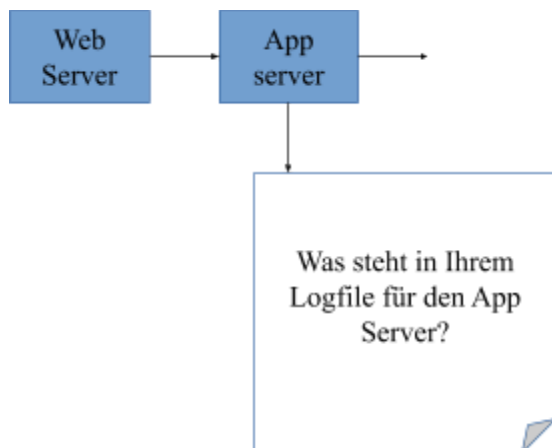
Logging macht die Anwendung langsamer, aber durch logging ist die Wartung einfacher. Man kann den logging ein eigenes Thread geben

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?



Man muss schauen wo was wie geloggt wurde. Also einmal im Web server, einmal im App server und 1X in DB

6. Was sollten Sie pro Komponente/Tier loggen?



Man kann Anzahl der Verbindungen, IP-Adressen, Protokolle, Verbindungsabbrüche loggen.

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?

Man kann die Zeit loggen, wie lange eine Verbindung da war