

## Nachdenkzettel Logging

.....  
Vorname, Name, Matrikelnummer

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>
```

1.2 Wie würde man erreichen, dass für alle Klassen innerhalb eines Packages ein spezieller Loglevel gelten würde? Könnte man auch alle Klassen eines Packages in ein anderes File loggen?

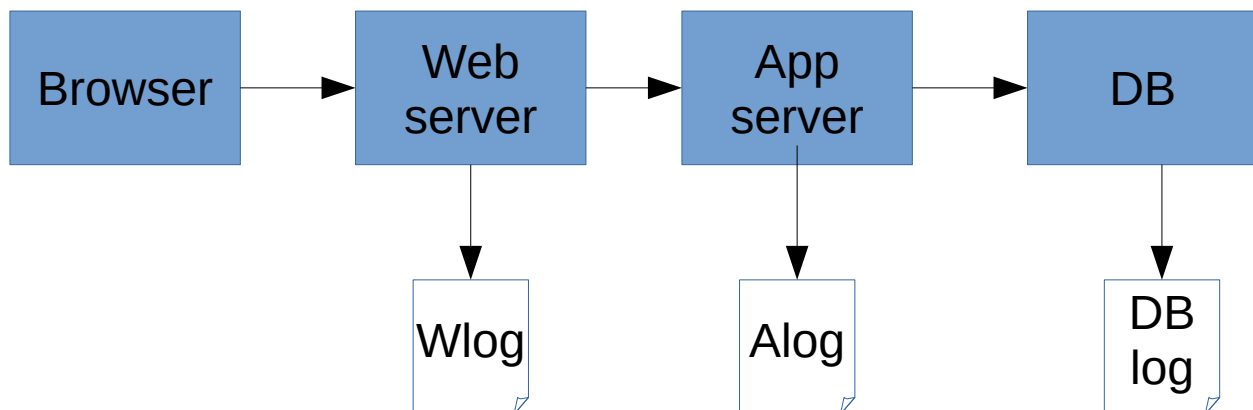
2. Geben Sie je ein Beispiel wann Sie den loglevel

- error      Exceptions
  - info      whenever anything happens
  - debug      whenever anything interesting happens
- verwenden

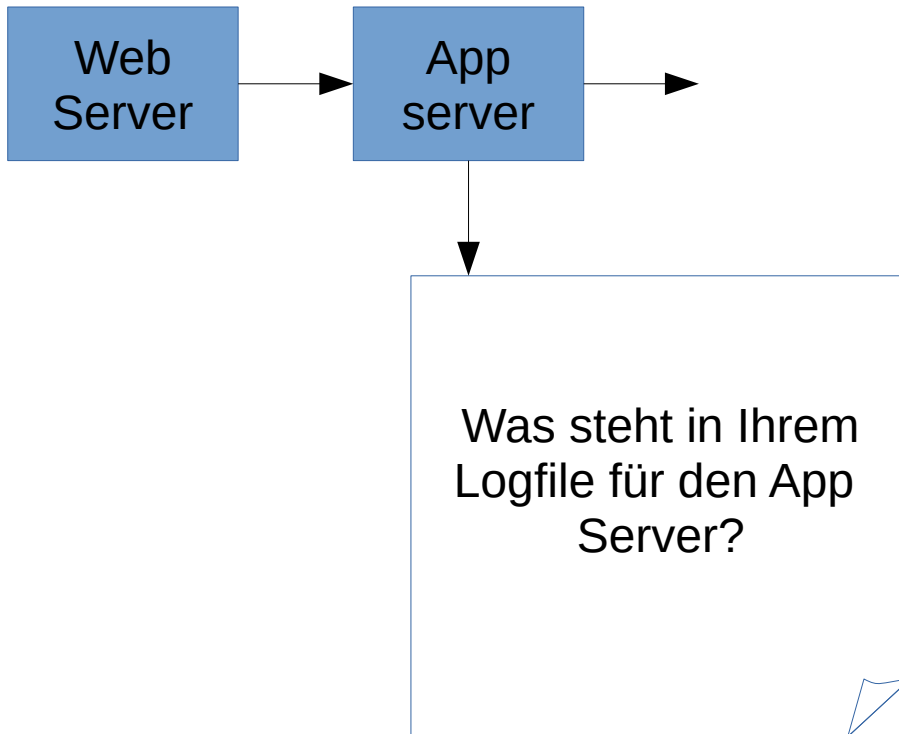
3. Sie verwenden einen FileAppender für das Logging. Jetzt soll Ihre Application im Datacenter laufen. Was machen Sie mit dem FileAppender?

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

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?



6. Was sollten Sie pro Komponente/Tier 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?

8. Sie fügen log-statements in die Login-Klasse ein. Was müssen Sie unbedingt beachten???  
Tipp: Denken Sie über Userverhalten nach. Und über Mitarbeiter....

7) • wer wann eine connection aufbaut

• wann er sie zurück gibt

• Differenz der beiden Zeiten: wie wie lange eine connection lebt

• wie viele connections sind noch da?

=> warten: muss jemand auf eine Verbindung warten?

=> jeder fünfte connection muss warten

/  
mehr connection

\  
Datenbank aufteilen

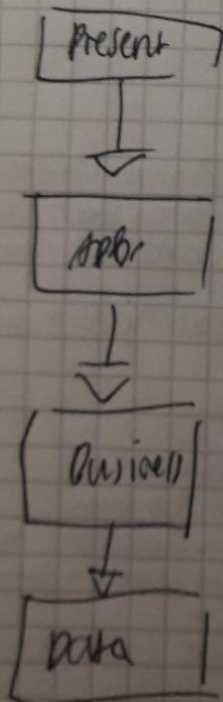
• verliert jemand connections?

=> Kontrolle schaffen

8) Problem: UserID-Feld -> PW eingegeben

logfile: mitreiter könnten PW sehen

## DESIGN PATTERNS



von oben nach unten greifen  
auf die Tiefenliegenden ein  
zu



3) Rolling File Appender

↳ sonst füllt das die Disks

4) Bei extremen Fällen ja, bei den normalen Anwendungen nicht spürbar

Realtime Anwendungen im Embedded-Bereich unter Umständen

5) Typisches Multitier-System / Layer-Architektur

→ wo ist der Zusammenhang?

→ man will die Teile eines Request korrelieren können  
Unique ID vergeben

"metapher" Starbuckes Name auf Becher

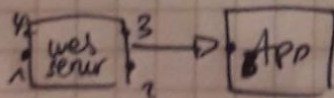
6) - Request-Typ

- Cookies / Namen

- Zeitstempel

- Zeit wann rein + wann raus → Protokollieren: wie  
lange hat der Request  
gestraucht hat

- auch möglich: Warnung wenn Verarbeitungszeit über bestimmte  
Zeit liegt  
vier Zeitstempel



1: wenns reinkommt

2: wann zur App geht

3: wann zum Webserver zurück kommt

4: wanns zum User geht

→ messen, damit man weiß, wo wie viel Zeit  
draufgeht