

Praktikum Architektur von Informationssystemen

Sommersemester 2011 – Aufgabenblatt 3

Prof. Dr. Stefan Sarstedt <stefan.sarstedt@haw-hamburg.de>, Raum 1085, Tel. 040/42875-8434

Mitarbeiter: Norbert Kasperczyk-Borgmann <nkb@informatik.haw-hamburg.de>

Aufgabe 3: SuperKiGaV XE (eXtremeEdition)

Nachdem nun in der gesamten Bundesrepublik Ihre Software zum Einsatz kommen soll, müssen Sie diese robust gegen Ausfall gestalten. Als Architekturstil sehen Sie nun aus Performanz- und Skalierbarkeitsgründen vor, unabhängige Applicationserver für die Implementierung der Geschäftslogik einzusetzen. Die Clients, Geschäftslogik, Datenbank und Rechteverwaltung können somit auf verschiedenen Rechner laufen.

Sie entscheiden sich außerdem, den sehr kritischen Anmeldeprozess eines Kindes an eine Gruppe redundant ausulegen, und somit das Pattern „**Aktive Redundanz**“ zu implementieren. Die beteiligten Elemente sind in einer logischen Sicht und einer Prozess-/Verteilungssicht auf dem folgenden Skizzen schematisch dargestellt:

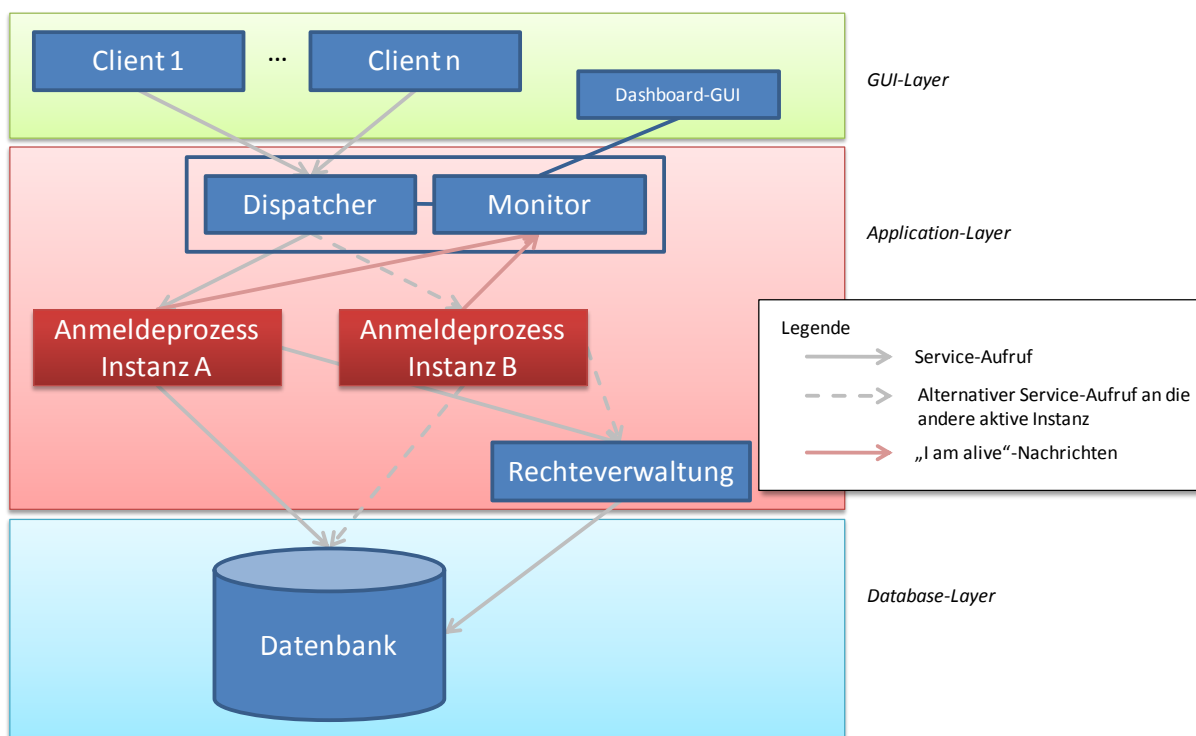


Abbildung 1: Logische Sicht der Elemente und Verbindungen

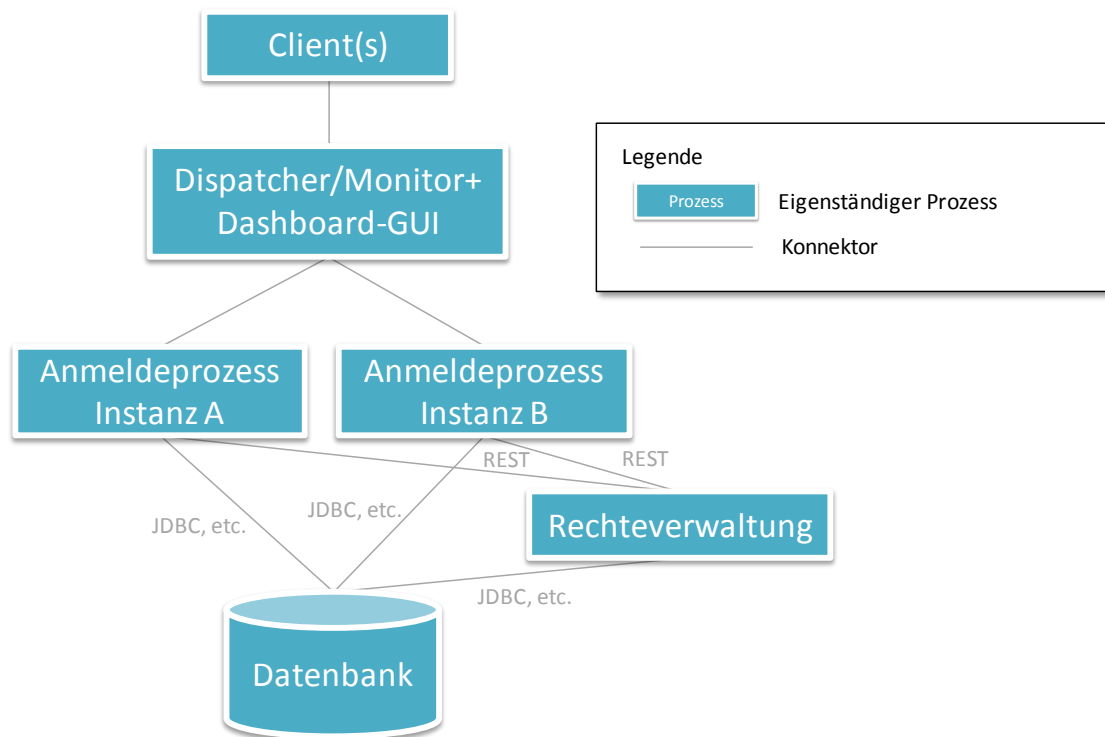


Abbildung 2: Prozess-/Verteilungssicht

Der Softwareprozess, der die Anmeldung der Kinder implementiert, wird doppelt ausgelegt und je eine aktive Instanz läuft auf einem separaten Rechner. Da das Datenbank- und Rechteverwaltungssystem aus Aufgabe 2 als sehr zuverlässig gelten, werden diese nicht zusätzlich redundant ausgelegt, sondern jeweils eine einzige Instanz verwendet. Der Ablauf einer Anmeldeanfrage durch den Klienten läuft wie folgt:

- Der Klient stellt alle Anfragen an einen zentralen Dispatcherprozess.
- Der Dispatcherprozess leitet die Anfrage an **eine** der beiden aktiven Anmeldeprozess-Instanzen zur Bearbeitung weiter. Die Auswahl erfolgt „Round-Robin“.
- Eine Monitor-Komponente überwacht die aktiven Instanzen und teilt dem Dispatcher-Prozess den Ausfall bzw. eine Wiederverfügbarkeit einer aktiven Instanz mit. Hierzu empfängt der Monitor sogenannte „I am alive“-Nachrichten von den aktiven Instanzen. Bleiben diese Nachrichten für einen bestimmten (konfigurierbaren) Zeitraum aus, geht der Monitor davon aus, dass dieser Prozess momentan nicht zur Verfügung steht.
- Über eine „Dashboard-GUI“ kann das SuperKiGaV-Betriebsteam jederzeit den Verfügbarkeitszustand der Prozesse ansehen (als „Ampel-Darstellung“). Des Weiteren kann das Team auch manuell die Instanzen offline und wieder online schalten. Außerdem sehen sie die Anzahl der durch die jeweiligen Instanzen bislang verarbeiteten Serviceanfragen und die jeweilige „Uptime“ und „Downtime“.

Zusätzliche Hinweise

- Der bzw. die Clients, der Dispatcher (inkl. der Monitor-Teilkomponente), sowie die Anmelde-Instanzen sollen in unterschiedlichen Prozessen (nicht Threads) laufen.
- Die Dashboard-GUI soll in demselben Prozess wie der Dispatcher/Monitor laufen.
- Die beiden Anmeldeprozesse sollen auf physikalisch verschiedenen Rechnern laufen.

- Zur Verbindung Client(s)↔Dispatcher, Dispatcher↔Anmeldeprozess(e) benötigen Sie entsprechende prozessübergreifend arbeitende Konnektoren.
- Testen Sie Ihr System sowohl durch manuelles offline/online-Schalten der Anmeldeinstanzen über die Dashboard-GUI, als auch durch „Abschießen“ und Neustarten der Anmeldeprozesse auf den jeweiligen Rechnern. Beobachten Sie dabei die Status auf Ihrem Dashboard.

Hinweise für Zweierteams

Die folgenden Punkte des Aufgabenblattes gelten für Zweierteams:

- Client+Dispatcher/Monitor können in demselben Prozess laufen. Ein Konnektor ist somit hier nicht nötig.
- Für die Dashboard-GUI entfallen die Funktion „offline/online-Schalten der Anmeldeprozesse“ und das Monitoring der Uptime/Downtime/Anzahl der Serviceaufrufe.
- Der Dispatcher/Monitor braucht nicht die Wiederverfügbarkeit einer Anmeldeinstanz zu überwachen. Falls eine Instanz somit als „ausgefallen“ markiert wurde, bleibt sie es.

Präsentation beim Praktikumstermin

Halten Sie beim zugeordneten Praktikumstermin im Team einen Vortrag, in dem Sie folgendes zeigen:

- Die technische Architektur (Statisch, Verteilung und Dynamik/Ablauf der Dispatcher- und Monitorfunktionalität im Anwendungsfall „Kind an Gruppe anmelden“ als UML-Sequenzdiagramm)
- Codeteile, die Ihre Dispatcher- und Monitorfunktionalität umsetzen.
- Eine Live-Demonstration Ihrer Testszenarien.

Die **Präsentation und der Code** sind zum Praktikumstermin über Moodle abzugeben.

Zusatzaufgaben (*)

- a) Erweitern Sie den Dispatcher und Monitor derart, dass beliebig viele Anmelde-Prozesse verwendet werden können. Diese sollen durch das Betriebsteam über das Dashboard an- bzw. abgemeldet werden können.
- b) Entwerfen und implementieren einen Load-Balancer in Ihrem Dispatcher. Dazu soll der Dispatcher für einen Serviceaufruf diejenige Instanz auswählen, die auf einem Rechner mit der geringsten CPU-Last läuft. Hierzu müssen die aktiven Instanzen periodisch Informationen über die aktuelle Last an den Dispatcher senden. Die Last der jeweiligen Instanz soll in der Dashboard-GUI dargestellt werden.