

Praktikum Architektur von Informationssystemen

Sommersemester 2013 – 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 5: Redundantes HES

Nachdem ihr Geschäft kontinuierlich gewachsen ist, müssen Sie das HES robust gegen Ausfall gestalten. Sie entscheiden sich daher, das komplette HES-System redundant auszulegen, und somit das Pattern „**Aktive Redundanz**“ zu implementieren. Die beteiligten Elemente sind auf dem folgenden Skizzen schematisch dargestellt:

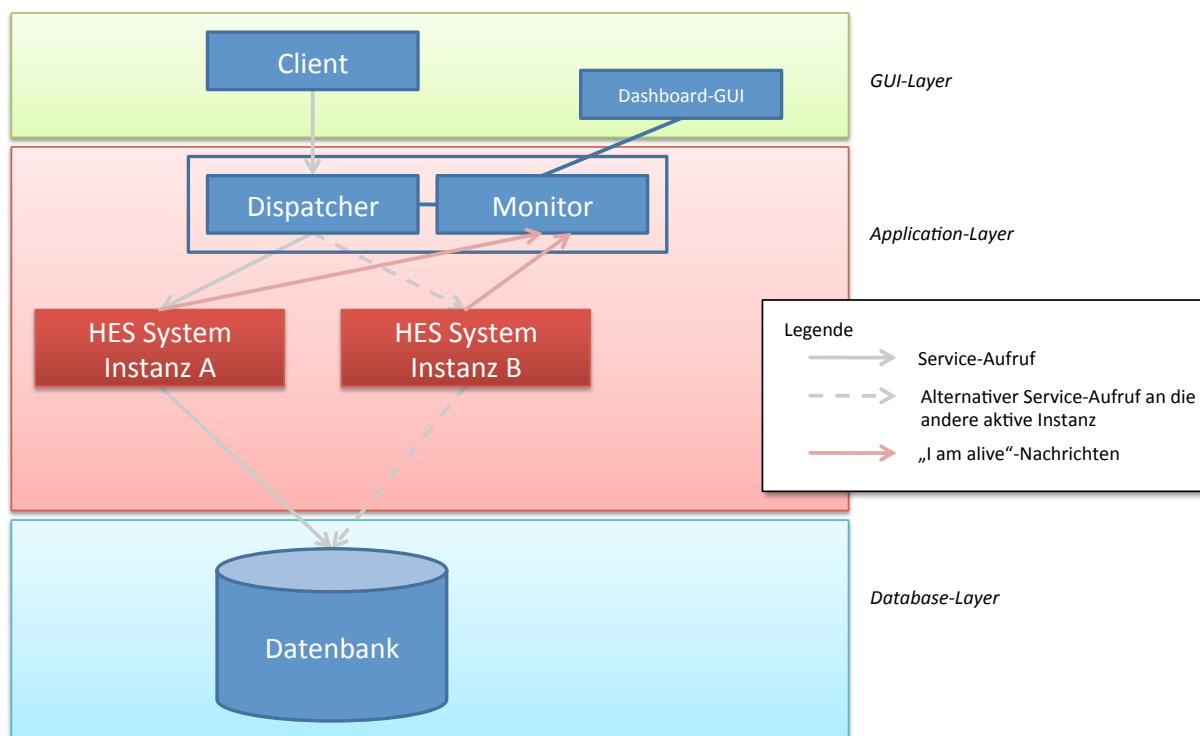


Abbildung 1: Bausteine und Verteilung

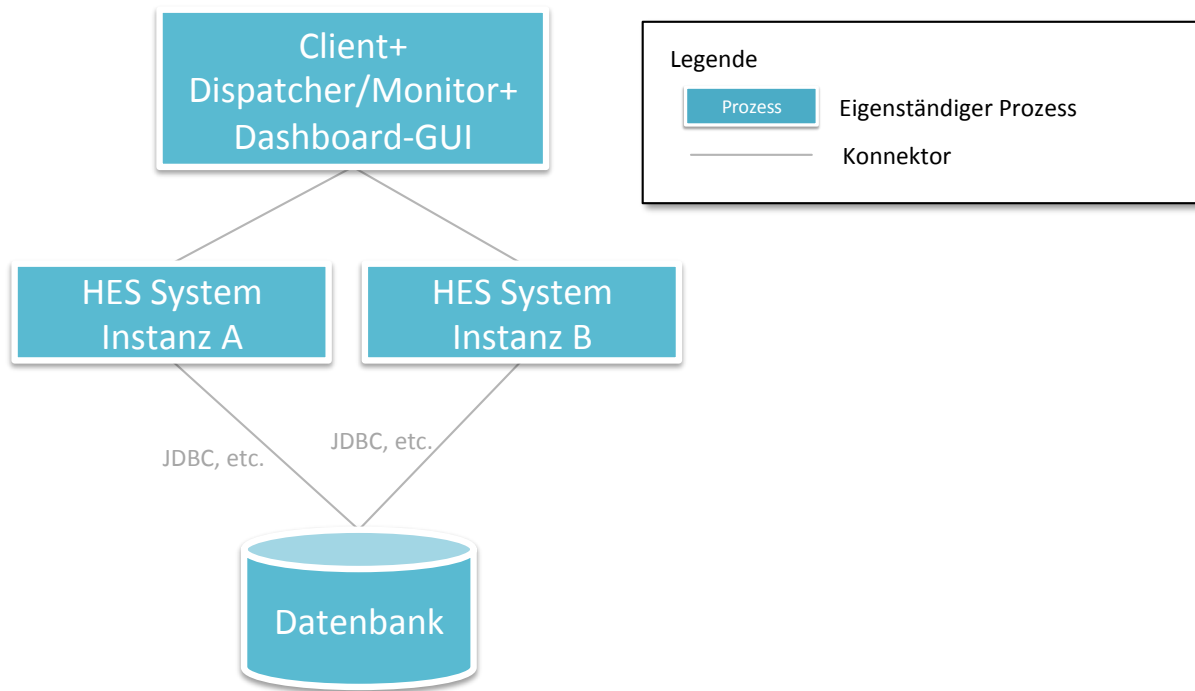


Abbildung 2: Prozesse

Der Softwareprozess, der die HES-Logik implementiert, wird doppelt ausgelegt und je eine aktive Instanz läuft auf einem separaten Rechner. Da das Datenbanksystem als sehr zuverlässig gilt, wird dieses nicht zusätzlich redundant ausgelegt, sondern nur eine einzige Instanz verwendet. Der Ablauf einer Auftragsstellung und -abwicklung durch den Klienten läuft wie folgt:

- Der Klient stellt alle Anfragen (Funktionsaufrufe) 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 HES-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

- Zur Verbindung Client(s)+Dispatcher+Dashboard-GUI↔HES-Serverprozesse benötigen Sie einen prozessübergreifend arbeitenden **Konnektor**.
- 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.

Präsentation beim Praktikumstermin

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

- Die Architektur (Bausteinsicht, Laufzeitsicht, Verteilungssicht) der Dispatcher- und Monitorfunktionalität für das in der Spezifikation beschriebene Szenario als UML-Diagramme.
- Codeteile, die Ihre Dispatcher- und Monitorfunktionalität umsetzen.
- Eine Live-Demonstration Ihrer Testszenarien.

Die **Präsentation und der Code** sind zum Praktikumstermin über emil 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.