

**Pflichtübung 1**

Studiengang Unternehmens- und Wirtschaftsinformatik

Sommersemester 2016

Verteilte Architekturen und Technologien (VAR)

1432464 Lucas Kneis

1423444 Johannes Heiler

1428699 Tobias Jünemann

29. April 2016

**Einleitung**

Unser Unternehmen heißt Contoso GmbH und wir planen eine Client- Serversoftware für die Verwaltung unserer Hardware zu schreiben. Wir planen den Einsatz einer Message Oriented Middleware (MOM) um die Anbindung der Clients an die Server zu vereinfachen. Dies hat den Vorteil, dass die Kommunikation asynchron erfolgt und der Client oder Server nicht immer verfügbar sein muss. Außerdem ist und die Nebenläufigkeit und Parallelisierung sehr wichtig. Als Programmiersprache für Client und Server wollen wir Java verwenden.

**Aufbau**

Der Client besteht Hauptsächlich aus einer grafischen Benutzeroberfläche. Mit dieser können Geräte hinzugefügt, geändert, gefunden und entfernt werden. Im Server befinden sich die eigentliche Programmlogik und die Anbindung zur Datenbank. Die Programmlogik kontrolliert, ob der Benutzer plausible Eingaben gemacht hat und die entsprechenden Berechtigungen besitzt. Die Middleware stellt die Verbindung der Clients mit dem Server her.

**Middleware Auswahl**

Wir haben uns entschieden drei Middleware- Frameworks genauer anzusehen. Diese wären xmlBlaster, RabbitMQ und WebsphereMQ. (Apache ActiveMQ?)

**xmlBlaster**: xmlBlaster ist eine Message Oriented Middleware, welche durch XML-Nachrichten kommuniziert. Der Inhalt der Nachricht kann alles sein z.B. Text, Bilder, Binärdateien oder Java-Objekte. Die Kommunikation mit dem Server kann über eine große Zahl an Protokollen erfolgen, z.B. HTTP, RMI oder CORBA. Wenn der Server nicht verfügbar ist unterstützen die Clients das Zwischenspeichern der Nachrichten. Da xmlBlaster unter der LGPL vertrieben wird darf es auch kommerziell kostenlos eingesetzt werden. Einziges Negatives an dieser Middleware ist, dass sie zwar eine Benutzer/Passwort-Abfrage zur Verfügung stellt aber gleichzeitig keine verschlüsselten Verbindungen unterstützt. Möchte man den Datenverkehr vor Lauschangriffen schützen muss man dies sehr aufwendig über SSL realisieren. Ein Einsatz in einem internen geschlossenen Netzwerk ist so gerade noch akzeptabel. Wenn aber auch unautorisierte Personen Zugriff zu diesem Netzwerk haben sollte besser eine andere Middleware verwendet werden.[[1]](#footnote-1)

**RabbitMQ:** RabbitMQ ist eine weitere Nachrichtenorientierte Middleware, die sich vor allem durch die Unterstützung von sehr vielen Programmiersprachen sowie Protokollen auszeichnet. Ein weiterer positiver Aspekt ist, dass RabbitMQ sehr benutzerfreundlich und leicht zu handhaben ist. RabbitMQ ist also ein Message Broker welcher in der Programmiersprache Erlang geschrieben wurde. Oft wird diese Open Source Middleware als ein Übergangswerkzeug eingesetzt, um verschiedene Applikationen programmiersprachenunabhängig zu entkoppeln oder auch um Informationen und Aufgaben asynchron zu Bearbeiten. RabbitMQ kann für jede gesendete Information entscheiden, ob der Erhalt vom Empfänger bestätigt werden muss oder nach dem „Fire und forget“ Prinzip abgesendet wird. Damit es einen Schutz vor Ausfällen gibt, werden oft mehrere Server zusammengeschlossen und eine Art Cluster gebildet. Die Nachrichten können dann auf mehrere Server gleichzeitig gespeichert werden. Die Wahrscheinlichkeit Nachrichten zu verlieren wird hiermit deutlich verringert, auch bei Ausfall eines Servers. Es wird das AMQP Protokoll für die Kommunikation zwischen Broker, Client und Verbraucher genutzt.[[2]](#footnote-2)

**WebsphereMQ:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | xmlBlaster | RabbitMQ | WebsphereMQ |
| Lizenzkosten | keine(Open Source) | keine(Open Source) |  |
| Anschaffungskosten | keine | keine |  |
| Programmiersprachen | PHP, Perl, Python, C, C++, C#, Visual Basic.net, Flash, J2ME, Java | Java, Ruby, Python, .NET, PHP, Perl, C/C++, weitere….. |  |
| Betriebssysteme | Abhängig von der Programmiersprache (Windows, Linux, MacOS) | Windows, Linux/Unix, MacOS |  |
| Komplexität | Leicht zu erlernen | Benutzerfreundlich, viele Anleitungen vorhanden |  |
| Sicherheit | Passwortabfrage vorhanden, Nachrichten unverschlüsselt. | Nutzername/Passwortabfrage für Benutzer, Nutzer haben verschiedene Rechte |  |
| Sonstiges |  |  |  |
| Sonstiges |  |  |  |

**Bewertungsmatrix**

**[[3]](#footnote-3)**

**Fazit**

Letztendlich sind wir zum Entschluss gekommen, in unserer Firma xxxxxxx verwenden. Zu dieser Entscheidung sind wir gekommen weil…………

1. https://www.techfak.uni-bielefeld.de/~swrede/xml-isy/talks/mom-xmlblaster.pdf,

   http://www.xmlblaster.org, http://www.xmlblaster.org/FAQ.html#SSH [↑](#footnote-ref-1)
2. https://entwickler.de/online/rabbitmq-zur-verteilten-bearbeitung-von-aufgaben-und-entkopplung-von-applikationsteilen-5-136604.html,

   http://www.rabbitmq.com/features.html [↑](#footnote-ref-2)
3. Tabelle angelehnt an [↑](#footnote-ref-3)