

**Verteilte Systeme
WS 2015/16**

**Übungsblatt 7
Praktische Übungen**

Ziel dieses Blattes ist es, ein grundlegendes Verständnis zu gewinnen für:

- die Entwicklung verteilter objektorientierter Java-Anwendungen auf der Basis entfernter Methodenaufrufe mittels Remote Method Invocation (RMI)
- die Entwicklung verteilter Anwendungen, bei denen die Kommunikationsschicht durch eine plattformunabhängige Beschreibungssprache und ein zugehöriges Werkzeug generiert werden. Dazu wird hier Apache Thrift verwendet, ein Framework, das ursprünglich bei Facebook entwickelt wurde und die nahtlose Kommunikation von Services in unterschiedlichen Programmiersprachen ermöglicht.

Zur Einführung in Java RMI können Sie die auf der Veranstaltungsseite unter Materialien bereitgestellte RMI-Zusammenfassung der ETH Zürich (rmi.pdf) nutzen. Beachten Sie, dass seit Java 1.5 die manuelle Erstellung von Stubs nicht mehr notwendig ist; die in der Zusammenfassung erwähnten Aufrufe von `rmic` können also weggelassen werden. Ausführlichere Informationen sind z.B. erhältlich unter

- <http://download.oracle.com/javase/tutorial/rmi/>
- <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html>

Die offizielle Webseite für Thrift ist <http://thrift.apache.org/>.

Ein Tutorial findet sich unter <http://thrift.apache.org/tutorial/>, wo die Datei `tutorial.thrift` verlinkt ist, die zeigt, wie die Syntax der Interface-Definition aussieht. Eine Referenz mit weiterer Beschreibung finden Sie unter <http://diwakergupta.github.io/thrift-missing-guide/>.

Sie finden den Thrift-Compiler als vorkompilierte Binary (Linux, x86_64 Format, zu benutzen in den Linux-Pools) in Ihrem Projektverzeichnis.

Es werden sechs Eclipse-Projekte in Ihrem SVN bereitgestellt:

1. „VFileSystemServerAPI“ enthält das bekannte Java-Interface aus der Aufgabe 5. Hier müssen Sie für RMI Änderungen/Anpassungen vornehmen.
2. „VFileSystemEmulator“ enthält eine Java-Implementierung des Java-Interfaces. Dieses können Sie für dieses Aufgabenblatt nutzen. Hier in den Source-Folder müssen für den Thrift-Teil die neuen generierten Dateien kopieren. Die `*.thrift` Datei soll in den „thrift“ Ordner abgelegt werden. Alternativ können Sie auch Ihre letzte Abgabe verwenden. Dann ist es allerdings notwendig, das Eclipse-Projekt aus Aufgabe 5 in den Ordner 7 Ihres SVN zu kopieren und hinzuzufügen. In diesem Fall müssen Sie auch den thrift-Ordner anlegen.
3. „VFileSystemLibs“ enthält alle notwendigen Java-Bibliotheken für diese Aufgabe. Die Projekte importieren dieses Projekt bereits. Sie müssen hier nichts ändern oder anpassen.
4. „VFileSystemClientRMI“ enthält das Gerüst für den RMI-Client. Ergänzen Sie die enthaltene Klasse um Ihre Lösung, bzw. fügen Sie wenn notwendig weitere Klassen hinzu.
5. „VFileSystemClientThrift“ enthält das Gerüst für die Umsetzung eines Thrift-Clients. Ergänzen Sie die enthaltene Klasse um Ihre Lösung und fügen Sie die generierten Klassen dem Projekt hinzu.
6. „VFileSystemServerCLI“ enthält den bekannten Client aus der vorherigen Aufgabe. Sie können alternativ auch die letzte Version verwenden (sind identisch).

Wichtig:

Denken Sie daran, dass wenn Sie neue Dateien erzeugen oder anlegen, dass diese mit `svn add <filename>`, bzw. über das Subversive-Eclipse-Plugin dem SVN-Repository hinzugefügt werden müssen. Wenn Sie dies nicht machen, bzw. vergessen, dann sind die Dateien nur auf Ihrer lokalen Arbeitskopie enthalten, aber können von den Lehrbeauftragten nicht bewertet werden!

Aufgabe 7.1 (Einarbeitung RMI):

Machen Sie sich mit den Grundlagen von RMI vertraut. Sollte das Kurzdokument der ETH Zürich nicht reichen, können Sie z.B. die o.a. Tutorials durcharbeiten. Am Ende sollte Ihnen klar sein,

- wie man ein Remote Interface festlegt
- wie man ein Remote Interface implementiert
- was die Bedeutung der RMI Registry ist
- wie man Dienste/Objekte auf der Server-Seite registriert
- wie man Dienste/Objekte von entfernter Client-Seite benennt und aufruft
- wie man mit Referenzen auf entfernte Objekte umgeht

Aufgabe 7.2 (RMI-Zugriff auf das VFileSystem)

Erweitern Sie die Projekte `VFileSystemServerAPI` und `VFileSystemEmulator` so, dass das `VFileSystem` mittels RMI exportiert wird. Zeichnen Sie dazu die Interfaces so aus, dass sie gültige RMI Remote Interfaces werden, und passen Sie die Implementierung entsprechend an (Klassen per `UnicastRemoteObject` exportieren). Implementieren Sie das RMI-Clientprojekt (`VFileSystemClientRMI`) für einen Zugriff über RMI.

Aufgabe 7.3 (Einarbeitung Thrift)

Machen Sie sich mit den Grundlagen von Thrift vertraut. Lesen dazu zunächst die `tutorial.thrift` durch. Lesen ggf. weiterführend das „Thrift: The Missing Guide“ Dokument (s.o.).

Aufgabe 7.4 (Thrift-Zugriff auf das VFileSystem)

In dieser Aufgabe soll der Zugriff auf das `VFileSystem` unter Nutzung von Thrift stattfinden. Thrift folgt dem Prinzip des RPC, die Services sind also im Gegensatz zu RMI nicht objektorientiert.

- a) Thrift-Datei
Obwohl aus einer Thrift-Interface-Beschreibung Stubs für unterschiedliche Programmiersprachen erzeugt werden kann, sollen hier Client und Server in Java implementiert werden. Der Server soll in das Projekt `VFileSystemEmulator` integriert werden und der Client in dem Projekt `VFileSystemClientThrift` implementiert werden.

Schreiben Sie zunächst eine `.thrift`-Datei, die einen Service „`VFileSystem`“ enthält, mit Methoden, die der ursprünglichen API ähneln.

- b) Verwenden Sie das „`thrift`“ Programm mit der Option „`--gen java`“ um sich die Thrift-Stubs aus dem Interface zu generieren
- c) Implementieren Sie für den Server das Interface `VFileSystem.Iface` in einer Klasse `VFileSystemThriftImpl` (neu anlegen!) Verwenden Sie für die Implementierung das Interface des `VFileSystemServerAPI`-Projekt, das der Emulator implementiert.

- d) Implementieren Sie eine Klasse, die eine neue Instanz von `VFileSystemThriftImpl` erzeugt, sowie ein Thrift-Server startet (`TSimpleServer`)
- e) Implementieren Sie in dem Projekt `VFileSystemClientThrift` den Thrift-Client, der sich zum Thrift-Server verbindet. Als GUI soll das bereits bekannte `VFileSystemServerCLI`-Projekt verwendet werden. Achten Sie darauf, dass die CLI das Java-Interface aus dem `VFileSystemServerAPI`-Projekt benötigt! Sie müssen daher eine weitere Abbildung von Thrift zu dem Java-Interface realisieren.
- f) Denken Sie daran, die erzeugten Dateien mit `svn add <filename>` Ihrem SVN hinzuzufügen. Anderenfalls sind diese bei einem Commit nicht enthalten und können nicht bewertet werden! Dies trifft **insbesondere** für die von Ihnen angelegte `.thrift` Datei zu!