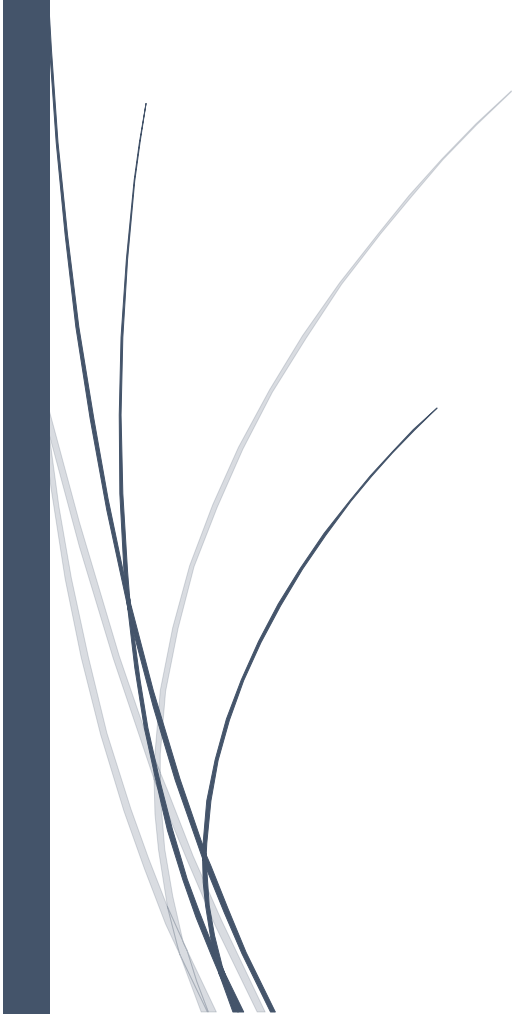
2/10/2014

DEZSYS01

REMOTING PATTERNS



Christian Janeczek, Wolfgang Mair
5AHITT

Contents

Task Description

Das Framework für Remoting Patterns finden sie unter dem Thema "Resources"!

Gruppenarbeit: 2 Mitglieder (Server/Client)

Analysieren Sie in einer Gruppe von 2 Leuten die mitgelieferte Implementation der verteilten LeelaApplikation. Identifizieren Sie dabei alle verwendeten Elemente der "Basic Remoting Patterns" und erstellen Sie UML-Klassendiagramme für die Pakete comm, comm.socket, comm.soap, evs2009 und evs2009.mapping

Schließen Sie die unfertigen Tests ab, und dokumentieren Sie etwaige Schwierigkeiten.

Was ist zu tun?

- UML Klassendiagramm
- Erweitern der Testfälle (mind. einen Testfall erweitern)
- Kritik und Verbesserungsvorschläge

Punkte (16):

Identifikation von Basic Remoting Patterns ... 1Pkt
Beschreibung der Applikation ... 4Pkt
UML-Diagramme ... 3Pkt
Schreiben von einem neuen Testfall ... 2Pkt
konstruktive Verbesserungsvorschläge / Kritikpunkte ... 6Pkt

Design consideration

REQUIREMENT: Identification of Remoting Patterns Usage

The usage of Basic RemotingPatterns in this software design cannot be denied. As mentioned in the documentation, the application was built upon the Broker-Pattern, which doesn't seem to be entirely true. To us it seemslike, there has been some kind of merging going on. Not only has the Broker-Pattern been utilized, but also initial parts of the RequestHandler Pattern.

In this case, we have a newborn child with genes of the Father(Broker) as well as the Mother (RequestHandler). The name-choosing is your special task.

REQUIREMENT: Description of the application

„Die Applikation wird durch die Auswahl eines Peers (Name) gestartet. Da kein NamingService verwendet wird, sondern eine statische Liste an teilnehmenden Peers (**peers.csv**), welche durch einen **PeerReader** ausgelesen werden kann, muss dieser Name auch vorher definiert werden.
Die Anforderung der ACID-Implementierung wird durch die Instanzen des **TransactionManager** und des **SessionPeer** gelöst. Ein entfernter Peer meint, seine Anfragen sofort an den Peer zu leiten, jedoch fängt diese der SessionPeer ab. Dieser kommuniziert dann mit dem TransactionManager, der Befehle erst nach erfolgreichem Endstatus an die Resources weiterleitet. Die Befehle werden in einer Liste zwischengespeichert.“

Die Kommunikation schließt nun folgende Klassen und Interfaces ein:

➢**AbsoluteObjectReference** hat notwendige Informationen eines Peers, wie Protokoll und Bestimmungsort. Das AOR wird vom Requestor verwendet.

➢**Lookup** liefert das AOR eines Peers zurück, das durch dessen Name indentifizierbar ist.

➢**Requestor** bietet ein dynamisches Interface zum Aufruf von Methoden über den RequestHandler an.

➢**RequestorHandler** arbeitet als Schnittstelle zwischen dem lokalen Peer und den Anfragen von entfernten Peers. Dabei nutzt der RequestHandler den Invoker für die einzelnen Server Instanzen.

➢**Invoker** bietet die Methode handleRequest(byte[]), welche eingehende Anfragen abarbeitet.

➢**ProtocolPluginServer** wird als Interface in den einzelnen Plugins implementiert und bearbeitet die eingehenden Aufrufe. Die einzelnen Protokolle werden beim Aufruf des Invokers instanziert und konfiguriert.

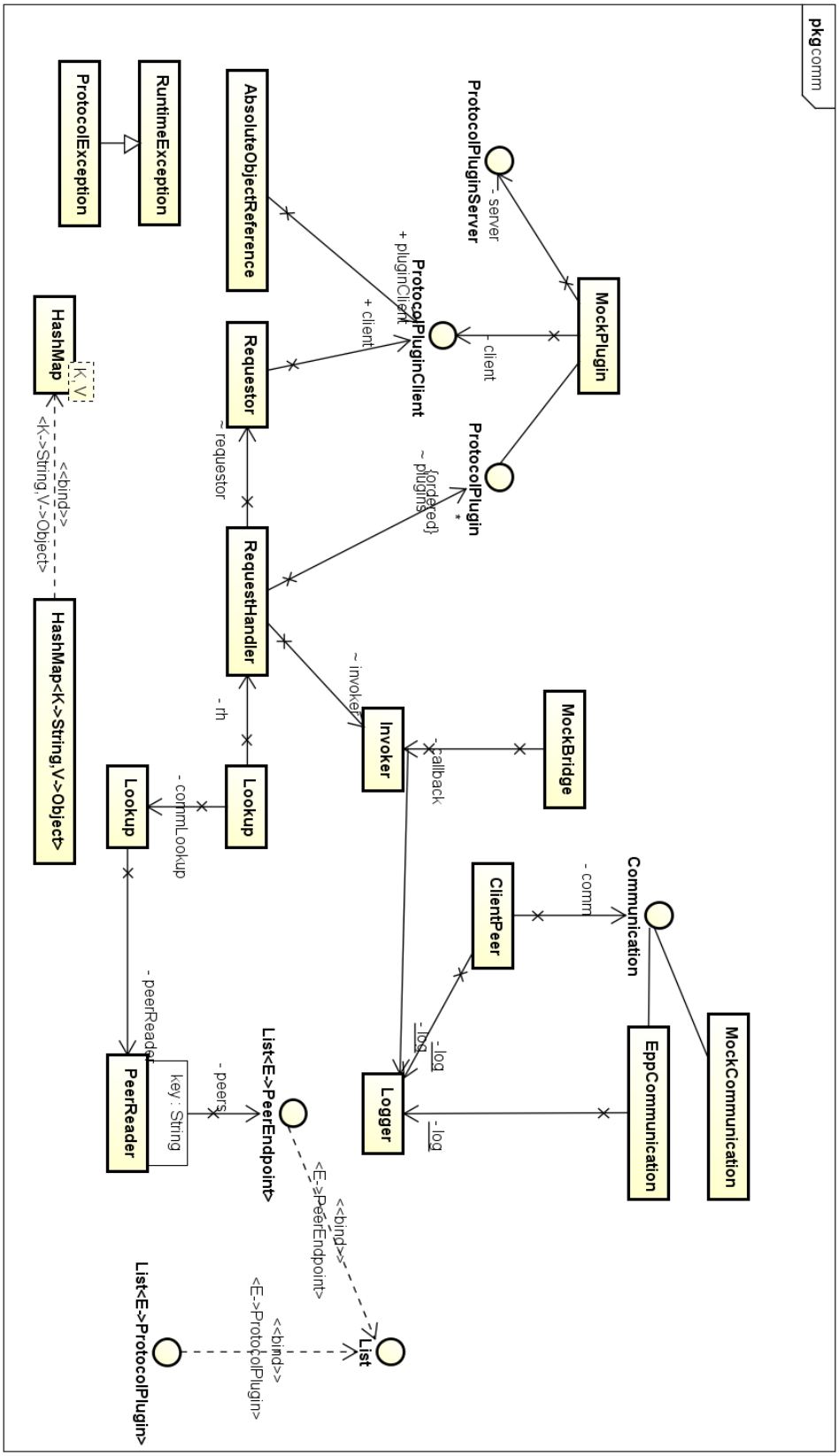
➢**ProtocolPluginClient** ist als Interface in den Protokollen als Schnittstelle nach außen vorgesehen. Durch das AOR wird der richtige Requestor ausgewählt und verwendet um eine Anfrage an einen entfernten Peer zu senden.

The German explanations have been taken from the **evs028.pdf** documentation presented to you by:

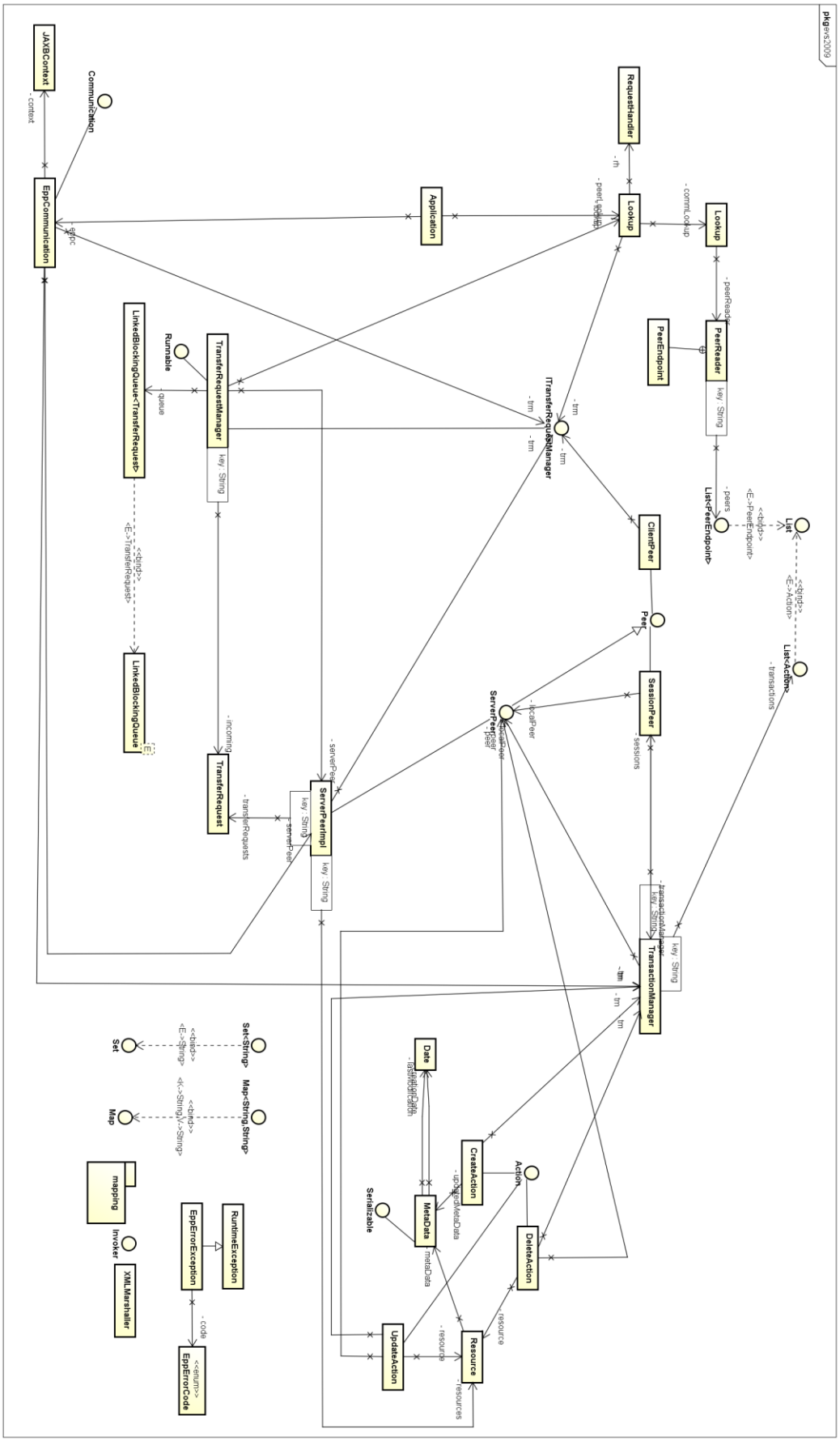
© Borko, Greifeneder, Motlik Seite 3 von 9 07.06.2009
Entwurfsmethoden für Verteilte Systeme evs028

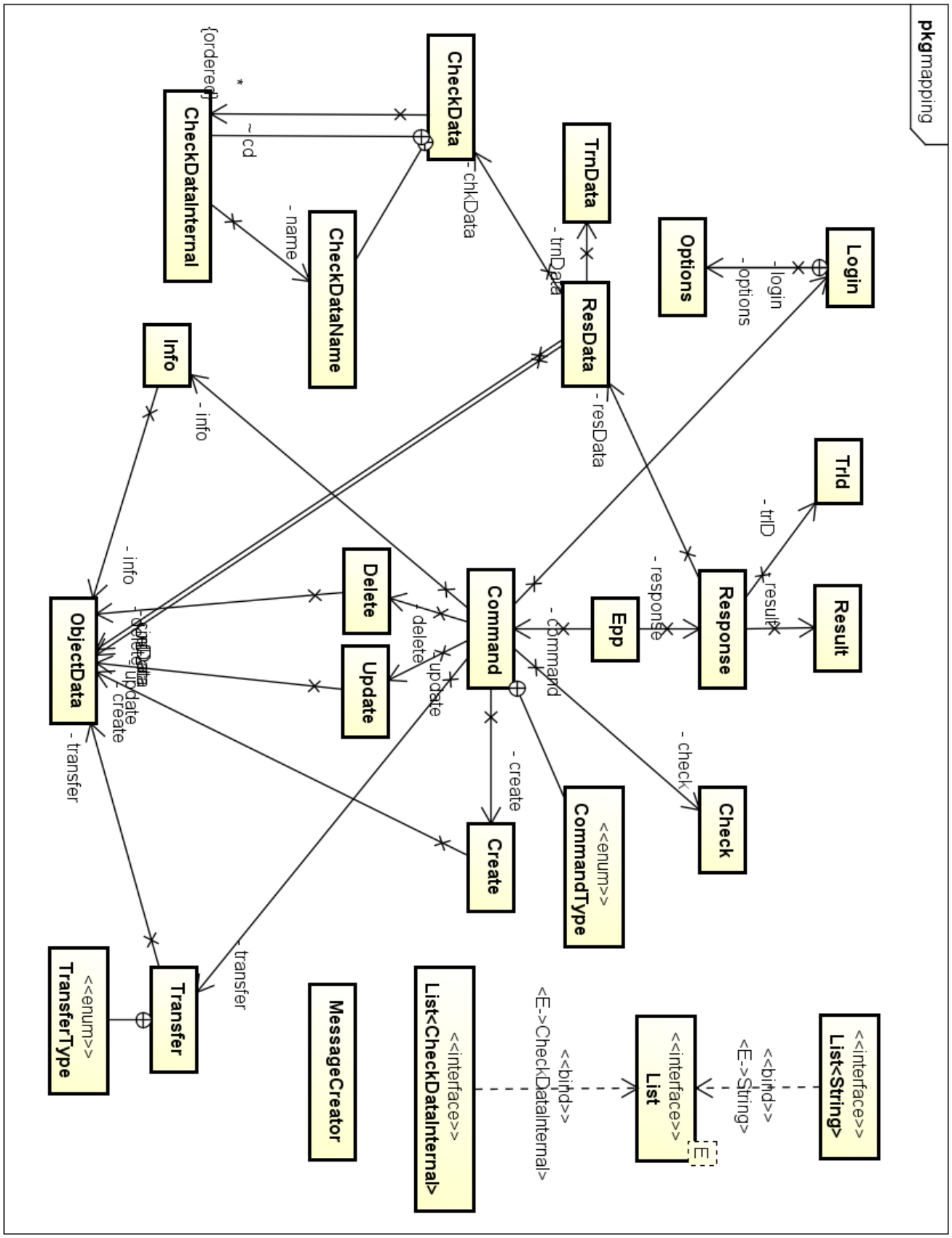
REQUIREMENT: UML-Diagrams

The UMLs are located right here



This UML is way too big I apologize for the nuisance ...




pkg mapping
CheckData
CheckDataInternal
CheckDataName
TrnData
Options
Login
ResData
Info
Trid
Result
Response
Epp
Command
Delete
Update
Create
Check
ObjectData
Transfer
<<enum>>
CommandType
<<enum>>
TransferType
MessageCreator
<<interface>>
List<String>
<<interface>>
List
<E>
<<interface>>
List<CheckDataInternal>
<<bind>>
<E->String>
<<bind>>
<E->CheckDataInternal>
{ordered}
*
~ cd
- chkData
- name
- trnData
- login
- options
- resData
- info
- trID
- result
- response
- command
- delete
- update
- create
- check
- transfer

REQUIREMENT: Writing a new Test Case

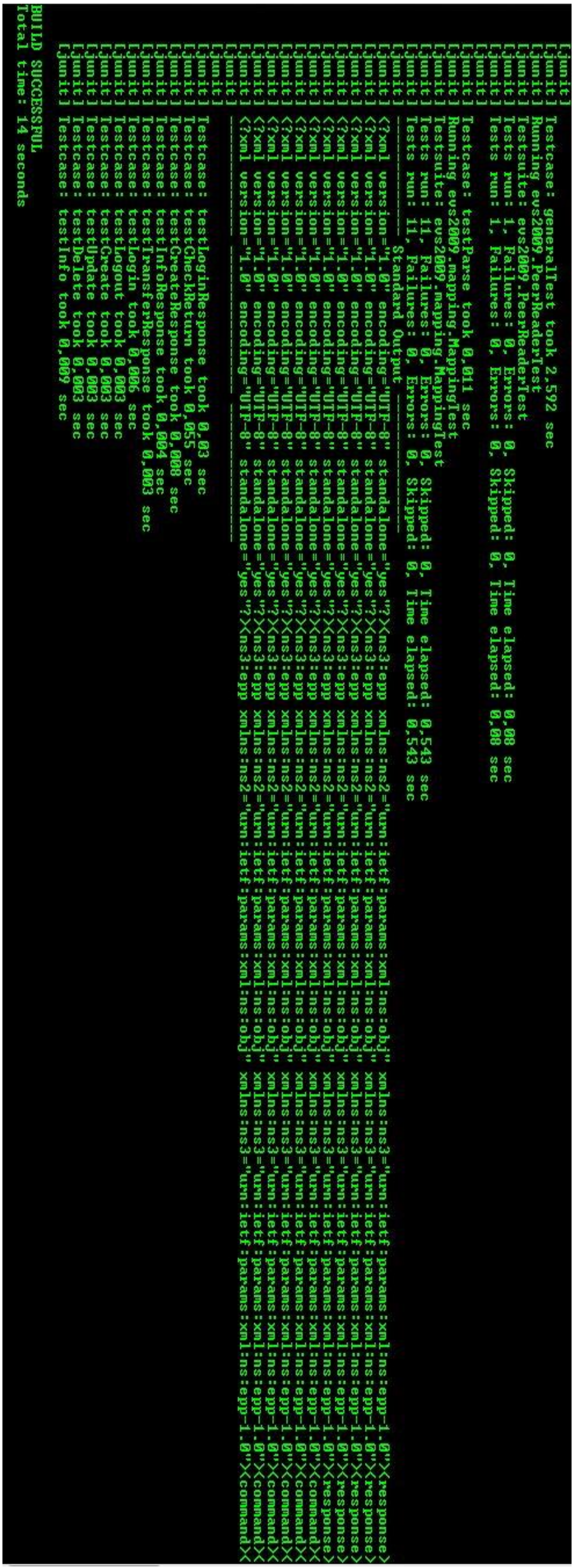
This Test was tested in the AuthenticationTests class.

```
/*
 * A method which tests what happens when there is someone going to login twice
*/
@Test
publicvoidcorrectDoubleLogin() throws Exception {
	peer.login(Helper.correctPassword, Helper.correctPassword);
	peer.login(Helper.correctPassword, Helper.correctPassword);
	peer.logout();
}
```

These Tests were tested in the CRUDTests class.

```
/*
 * A method to test the reaction of the server when he receives a null object
*/
@Test
publicvoid correctCreationAndReadZero() {
	String identifier = null;
	insertObject(identifier);
	byte[] readBytes = serverPeer.read(identifier);
	assertEquals(getBytes().length, readBytes.length);
	assertEquals(testString, new String(readBytes));
}

/*
* A method to test the reaction of the server when he receives a lot of read tasks
at once
*/
@Test
publicvoid correctCreationAndReadloop() {
	String identifier = "Wow";
	insertObject(identifier);
	byte[] readBytes = null;
	for(inti = 0; i<500; i++){
		readBytes = serverPeer.read(identifier);
	}
	assertEquals(getBytes().length, readBytes.length);
	assertEquals(testString, new String(readBytes));
}
```

```
test-junit-present:

test-junit-status:

junit-missing:

test:
    [junit] Running comm.CommunicationTest
    [junit] Testsuite: comm.CommunicationTest  - Skipped: 0, Time elapsed: 6,995 sec
    [junit] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 6,995 sec
    [junit]
    [junit] ------------- Standard Output ---------------
    [junit] 0   [main] INFO  comm.socket.SocketPlugInClient  - SPC :: Configuring clientpeer plugin socket to host: localhost port: 12300
    [junit] 111 [pool-2-thread-1] DEBUG comm.socket.SocketPlugInServer  - SRV :: Running listener on port 12300
    [junit] 114 [pool-2-thread-1] INFO  comm.socket.SocketPlugInServer  - Server sends bytes:32
    [junit] 115 [main] DEBUG comm.socket.SocketPlugInClient  - Client receives bytes:32
    [junit] 115 [main] DEBUG comm.socket.SocketPlugInClient  - Got bytes32
    [junit] 115 [main] DEBUG comm.socket.SocketPlugInClient  - Read is 32
    [junit]
    [junit] Testcase: testSocket took 1,687 sec
    [junit] Testcase: testSOAP took 5,195 sec
    [junit] Running evs2009.ApplicationTest
    [junit] Testsuite: evs2009.ApplicationTest
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2,662 sec
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2,662 sec
    [junit]
    [junit] ------------- Standard Output ---------------
    [junit] 0   [main] INFO  comm.socket.SocketPlugInClient  - SPC :: Configuring clientpeer plugin socket to host: localhost port: 12345
    [junit] 18  [main] INFO  comm.socket.SocketPlugInClient  - SPC :: Configuring clientpeer plugin socket to host: localhost port: 12300
    [junit] 58  [pool-1-thread-1] DEBUG comm.socket.SocketPlugInServer  - SRV :: Running listener on port 12345
    [junit] 99  [pool-1-thread-1] INFO  comm.socket.SocketPlugInServer  - Server sends bytes:379
    [junit] 99  [main] INFO  comm.socket.SocketPlugInClient  - Client receives bytes:379
    [junit] 99  [main] DEBUG comm.socket.SocketPlugInClient  - Got bytes379
    [junit] 99  [main] DEBUG comm.socket.SocketPlugInClient  - Read is 379
    [junit] 104 [pool-1-thread-1] DEBUG comm.socket.SocketPlugInServer  - SRV :: Running listener on port 12345
    [junit] 119 [pool-1-thread-1] INFO  comm.socket.SocketPlugInServer  - Server sends bytes:452
    [junit] 119 [main] INFO  comm.socket.SocketPlugInClient  - Client receives bytes:452
    [junit] 119 [main] DEBUG comm.socket.SocketPlugInClient  - Got bytes452
    [junit] 119 [main] DEBUG comm.socket.SocketPlugInClient  - Read is 452
    [junit] 123 [pool-1-thread-1] DEBUG comm.socket.SocketPlugInServer  - SRV :: Running listener on port 12345
    [junit] 139 [pool-1-thread-1] INFO  comm.socket.SocketPlugInServer  - MetaData size:256
    [junit] 139 [pool-1-thread-1] DEBUG evs2009.EppCommunication  - Server sends bytes:838
    [junit] 139 [main] INFO  comm.socket.SocketPlugInClient  - Client receives bytes:838
    [junit] 139 [main] DEBUG comm.socket.SocketPlugInClient  - Got bytes838
    [junit] 139 [main] DEBUG comm.socket.SocketPlugInClient  - Read is 838
    [junit] 144 [pool-1-thread-1] DEBUG comm.socket.SocketPlugInServer  - SRV :: Running listener on port 12345
    [junit] 147 [pool-1-thread-1] INFO  comm.socket.SocketPlugInServer  - Server sends bytes:510
    [junit] 147 [main] DEBUG comm.socket.SocketPlugInClient  - Client receives bytes:510
    [junit] 147 [main] DEBUG comm.socket.SocketPlugInClient  - Got bytes510
    [junit] 150 [pool-1-thread-1] DEBUG comm.socket.SocketPlugInServer  - Read is 510
    [junit] 152 [pool-1-thread-1] INFO  comm.socket.SocketPlugInServer  - SRV :: Running listener on port 12345
    [junit] 153 [main] INFO  comm.socket.SocketPlugInClient  - Client receives bytes:375
    [junit] 153 [main] DEBUG comm.socket.SocketPlugInClient  - Got bytes375
    [junit] 153 [main] DEBUG comm.socket.SocketPlugInClient  - Read is 375
    [junit] 155 [pool-1-thread-1] DEBUG comm.socket.SocketPlugInServer  - SRV :: Running listener on port 12345
```



```
    [junit]
    [junit] Testcase: generalTest took 2,592 sec
    [junit] Running evs2009.PeerReaderTest
    [junit] Testsuite: evs2009.PeerReaderTest
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0,08 sec
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0,08 sec
    [junit]
    [junit] Testcase: testParse took 0,011 sec
    [junit] Running evs2009.mapping.MappingTest
    [junit] Testsuite: evs2009.mapping.MappingTest
    [junit] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0,543 sec
    [junit] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0,543 sec
    [junit] ------------- Standard Output ---------------
    [junit] <?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns3:epp xmlns:ns2="urn:ietf:params:xml:ns:obj" xmlns:ns3="urn:ietf:params:xml:ns:epp-1.0"><response>
    [junit] <?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns3:epp xmlns:ns2="urn:ietf:params:xml:ns:obj" xmlns:ns3="urn:ietf:params:xml:ns:epp-1.0"><response>
    [junit] <?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns3:epp xmlns:ns2="urn:ietf:params:xml:ns:obj" xmlns:ns3="urn:ietf:params:xml:ns:epp-1.0"><response>
    [junit] <?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns3:epp xmlns:ns2="urn:ietf:params:xml:ns:obj" xmlns:ns3="urn:ietf:params:xml:ns:epp-1.0"><response>
    [junit] <?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns3:epp xmlns:ns2="urn:ietf:params:xml:ns:obj" xmlns:ns3="urn:ietf:params:xml:ns:epp-1.0"><response>
    [junit] <?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns3:epp xmlns:ns2="urn:ietf:params:xml:ns:obj" xmlns:ns3="urn:ietf:params:xml:ns:epp-1.0"><command><
    [junit] <?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns3:epp xmlns:ns2="urn:ietf:params:xml:ns:obj" xmlns:ns3="urn:ietf:params:xml:ns:epp-1.0"><command><
    [junit] <?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns3:epp xmlns:ns2="urn:ietf:params:xml:ns:obj" xmlns:ns3="urn:ietf:params:xml:ns:epp-1.0"><command><
    [junit] <?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns3:epp xmlns:ns2="urn:ietf:params:xml:ns:obj" xmlns:ns3="urn:ietf:params:xml:ns:epp-1.0"><command><
    [junit] <?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns3:epp xmlns:ns2="urn:ietf:params:xml:ns:obj" xmlns:ns3="urn:ietf:params:xml:ns:epp-1.0"><command><
    [junit] <?xml version="1.0" encoding="UTF-8" standalone="yes"?><ns3:epp xmlns:ns2="urn:ietf:params:xml:ns:obj" xmlns:ns3="urn:ietf:params:xml:ns:epp-1.0"><command><
    [junit] ------------- ---------------- ---------------
    [junit]
    [junit] Testcase: testLoginResponse took 0,03 sec
    [junit] Testcase: testCheckReturn took 0,055 sec
    [junit] Testcase: testCreateResponse took 0,008 sec
    [junit] Testcase: testInfoResponse took 0,004 sec
    [junit] Testcase: testTransferResponse took 0,003 sec
    [junit] Testcase: testLogin took 0,006 sec
    [junit] Testcase: testLogout took 0,003 sec
    [junit] Testcase: testCreate took 0,003 sec
    [junit] Testcase: testUpdate took 0,003 sec
    [junit] Testcase: testDelete took 0,003 sec
    [junit] Testcase: testInfo took 0,009 sec

BUILD SUCCESSFUL
Total time: 14 seconds
```

REQUIREMENT: Improvement Suggestions / Critique

- Insufficient documentation/comments inside the code (Zeilenkommentare)
  - It is hard to understand what exactly is going on, easily getting overwhelmed

- If there should be any libraries, which aren't provided in a library folder, there should be some kind of notification
  - Maybe some kind of Documentation in the zip-file would be helping

- AuthenticationTests.java is not completely bug-free

- Unfortunately there happen to be unhandled warnings (more than just once!)

Apportionment of work with effort estimation

Competent person(s)	**Task**	**Description**	**Estimated time in h**
Janeczek	Design consideration	Which Basic Remote Patterns have been used?	2
Mair	Design consideration	Creating the UML diagrams	2
Janeczek, Mair	Description of the application	Describing the application	1
Mair	Test Case	Implementing a new Test Case	2
Janeczek, Mair	Brainstorming: Improvement Suggestions, Critique	Implementing the State Pattern Solution	2

Final Time Apportionment

Competent person(s)	Task	Estimated time in h	Actual time in h	Comment
Janeczek	Design consideration	2	1	Janeczek
Mair	Design consideration	1	2	Mair
Janeczek, Mair	Description of the application	1	1	Janeczek, Mair
Mair	Test Case	1	2	Mair
Janeczek, Mair	Brainstorming: Improvement Suggestions, Critique	2	2	Janeczek, Mair

Bibliography

Title: Communication Framework
Author: Borko Michael, Greifeneder Michael, Motlik Florian
Source: evs028.pdf
Last modified:2009/06/07
Last seen: 2014/10/02