

Entwurfsdokument

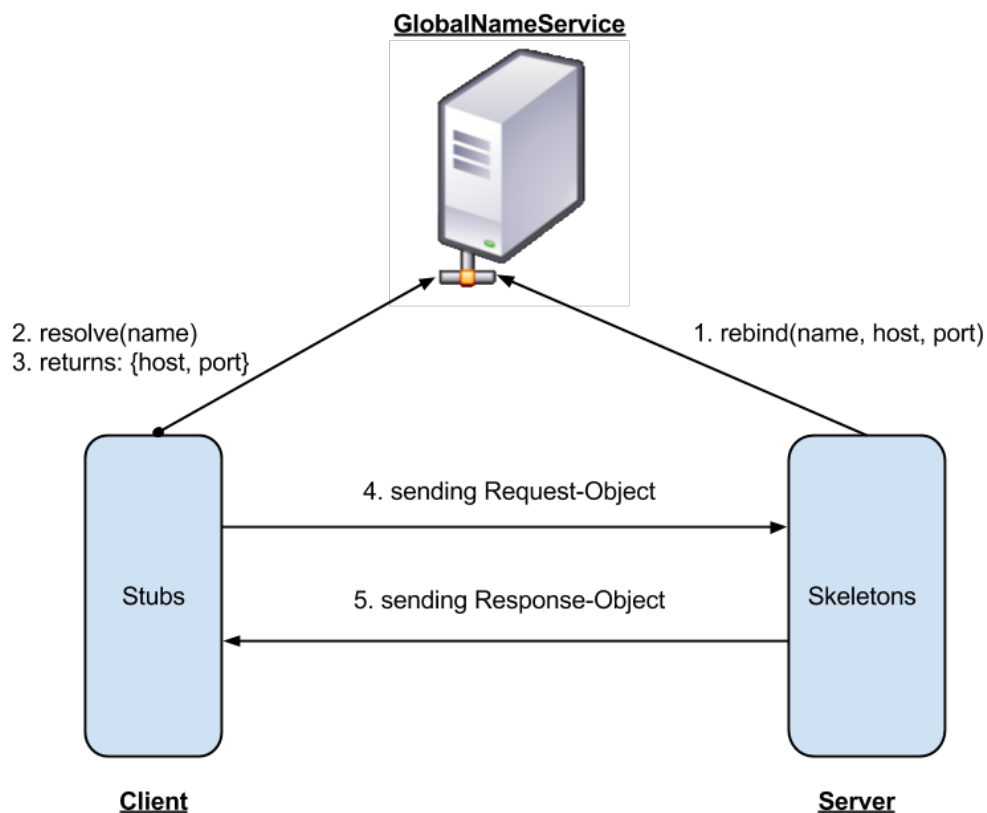
Middleware

Ziel dieser Aufgabe ist es, eine vereinfachte Middleware für ein verteiltes Kontensystem zu entwickeln. Die Implementierungssprache ist Java.

Eigenschaften der Middleware

- Kommunikation über TCP - Request-Response
- ermöglicht Interprozesskommunikation über verteilte Methodenaufruf
- Namensservice
 - lokal: Schnittstelle zum globalen Nameservice-Server
 - global: Nameservice-Server

allgemeiner Kommunikationsablauf



Implementierungsdetails

TCP Nachrichtenformat - Globaler Nameservice

Die Kommunikation zu dem globalen Nameservice erfolgt über die im folgenden genannten Nachrichten. Jede Nachricht ist ein String und wird beim empfangen entsprechend geparst. Als Trennungsoperator zwischen den jeweiligen Atomen wird ';' benutzt.

Rebind

REBIND	<i>name</i>	<i>hostname</i>	<i>port</i>
--------	-------------	-----------------	-------------

- **REBIND**
 - Keyword zur identifizierung der Message beim globalen Nameservice
- **name**
 - eindeutige ID des Objektes
- **hostname**
 - Hostname | IP des Servers
- **port**
 - Port des Servers

Als Bestätigung wird "OK" zurückgeschickt. Falls ein Fehler auftreten sollte wird "ERROR" zurückgesendet.

Resolve

RESOLVE	<i>name</i>
---------	-------------

- **RESOLVE**
 - Keyword zur identifizierung der Message beim globalen Nameservice
- **name**
 - eindeutige ID des Objektes

Returnmessage vom Server

OK ERROR	<i>hostname</i>	<i>port</i>
------------	-----------------	-------------

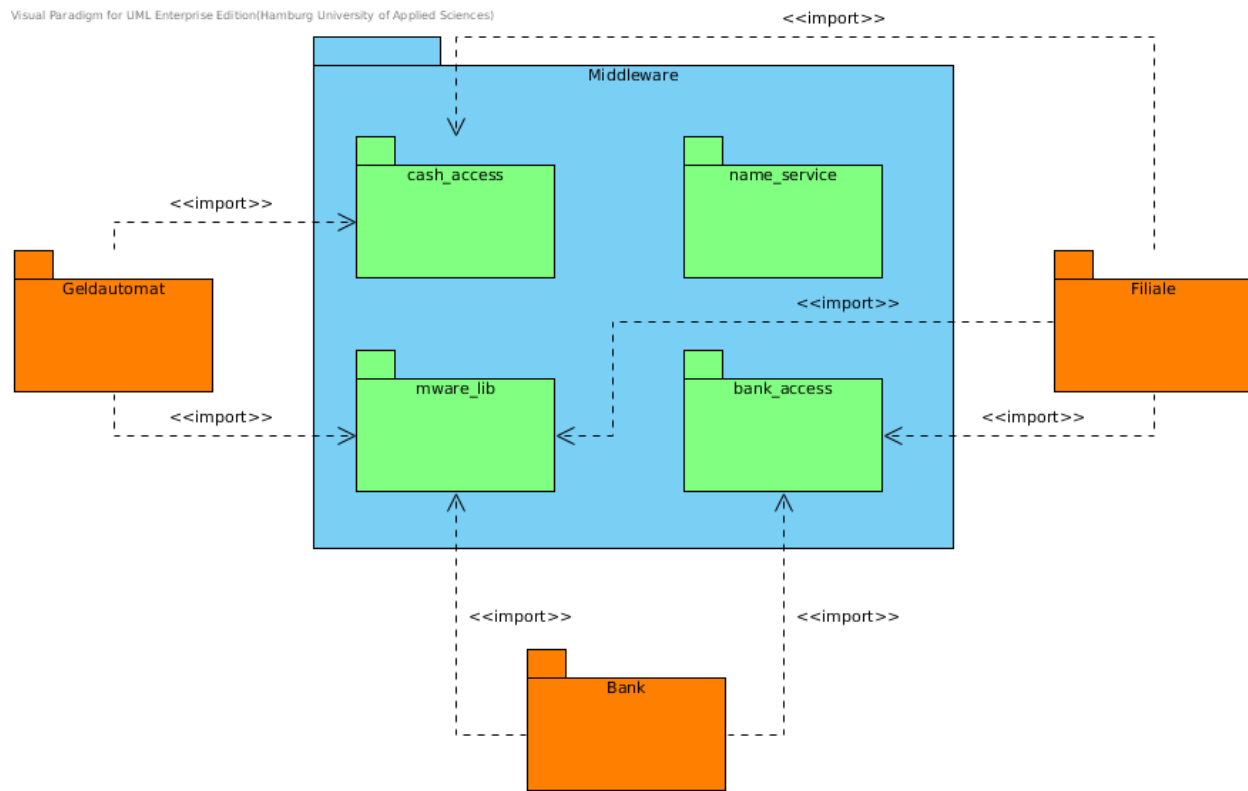
- **OK | ERROR**
 - Keyword zur Verifizierung des Aufrufs
 - *OK*
 - gültige Anfrage
 - *hostname* und *port* sind gesetzt
 - *ERROR*
 - ungültige Anfrage
 - *hostname* und *port* sind *null*
- **hostname**
 - *hostname* des Objekt-Servers
- **port**
 - *port* des Objekt-Servers

Kommunikation zwischen den Anwendungsclienten und -server

Für die Kommunikation zwischen den Anwendungen werden Request- und Response-Objekte versendet. Beide Objekte werden mittels Java-Objektserialisierung über das Netzwerk verschickt.

Das Request-Objekt wird von den Clienten in den Stubs erstellt und kann vom Server in den Skeletonobjekten ausgewertet werden. Dieser ruft dann die gewünschte Methode auf und sendet den Returnwert als Response-Objekt zurück zum Clienten. Die Response-Objekte werden dann von den Stubs ausgewertet und geben den Returnwert an die Anwendung zurück.

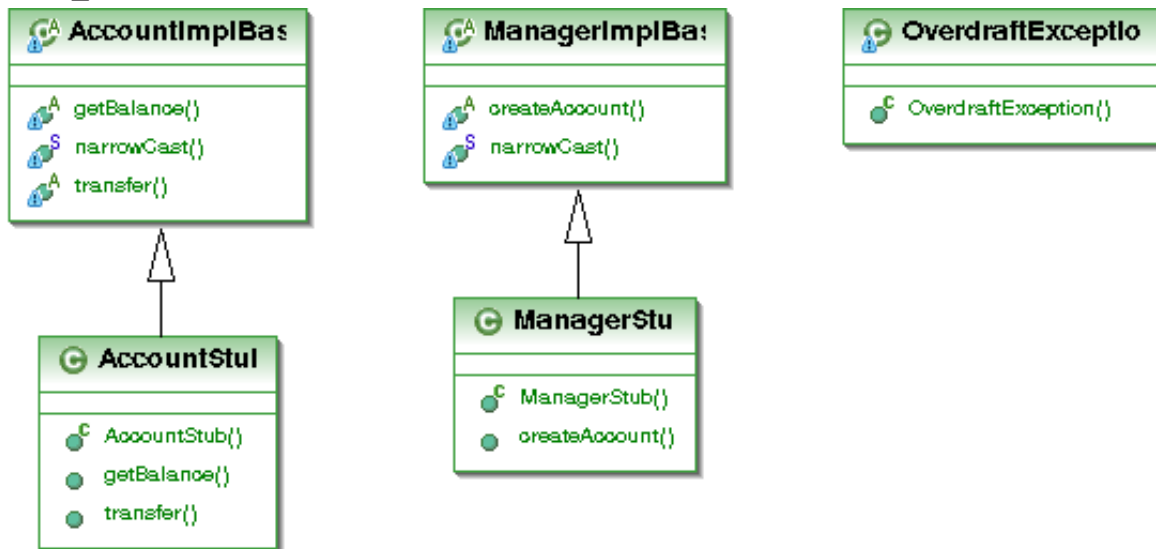
Paketübersicht



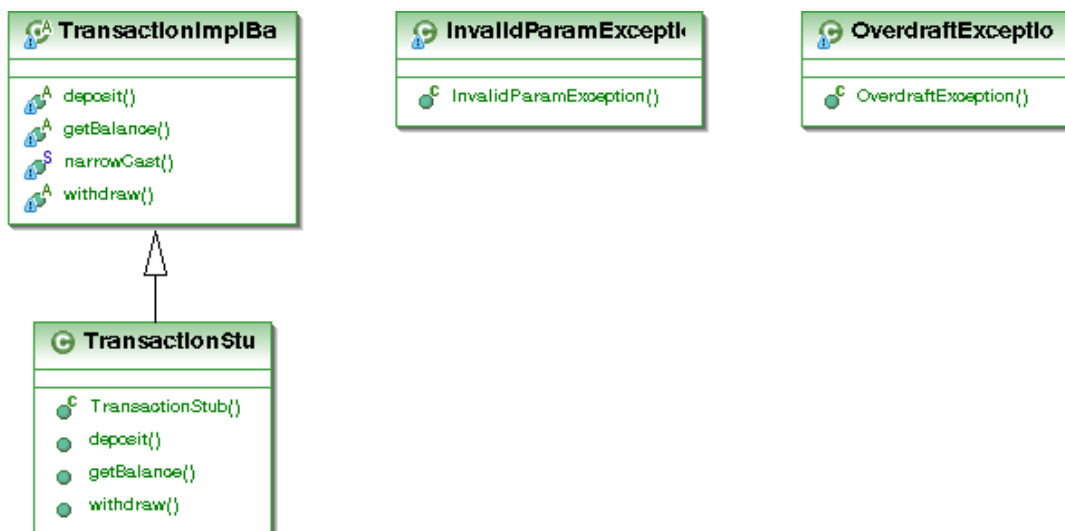
- Die Middleware besteht aus den Komponenten cash_access, name_service, mware_lib und bank_access
- Die Applikation “Geldautomat” muss die Komponenten cash_access und mware_lib verwenden
- Die Applikation “Bank” verwendet die Komponente mware_lib als auch der Komponente bank_access
- Die Applikation “Filiale” verwendet die Komponenten: mware_lib, bank_access und cash_access
- Die Komponente mware_lib beinhalten die Methoden für den lokalen Nameservice, die notwendig sind um auf die Komponente name_service zuzugreifen
- mware_lib ermöglicht sowohl die Registrierung von Remote-Objects als auch den lokalen Zugriff auf entfernte Objects.
- mware_lib ist der Kern der Middleware
- Die Komponenten bank_access, cash_access stellen sicher, dass die Verteilten Anwendugen über die selben Methoden verfügen

Klassendiagramme

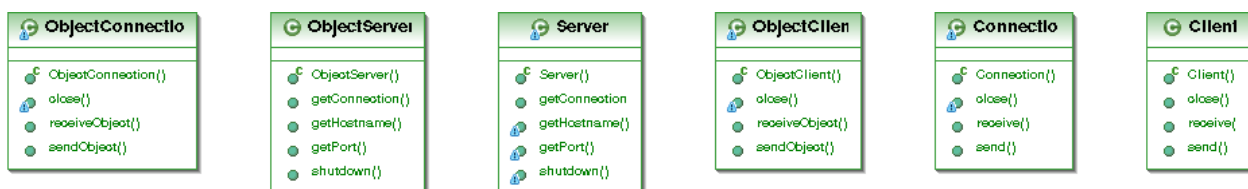
bank_access



cash_access



communication



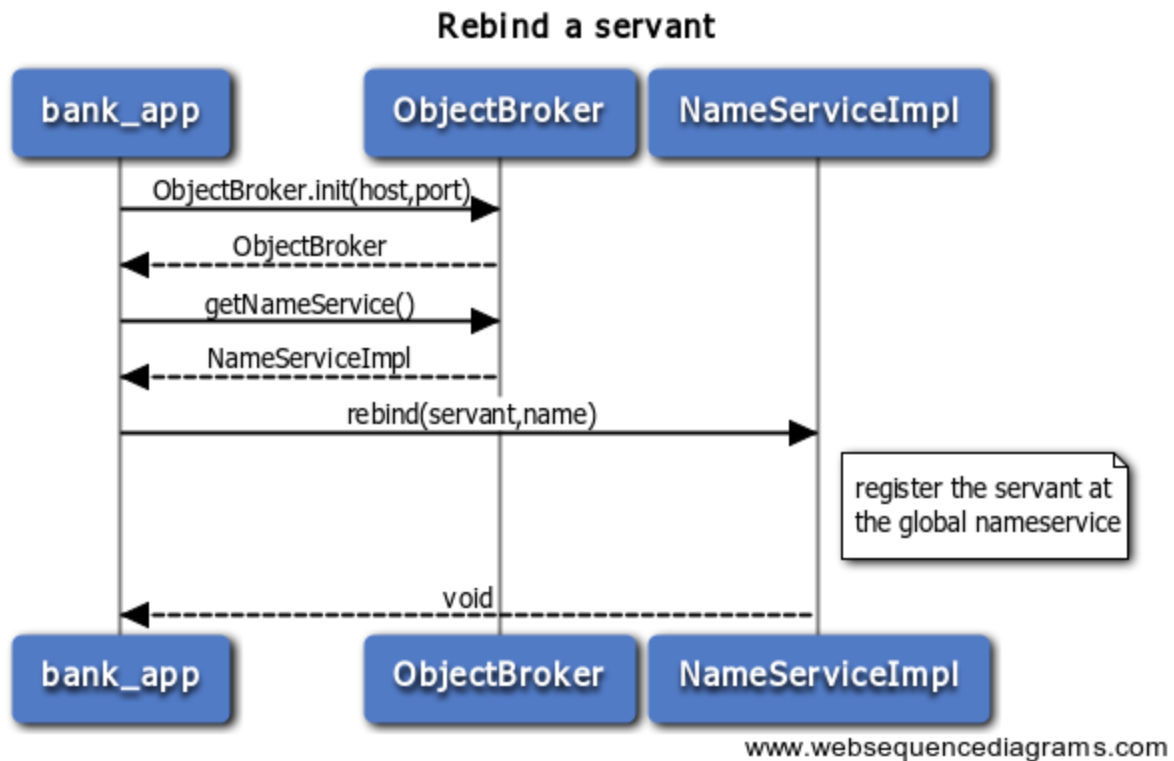
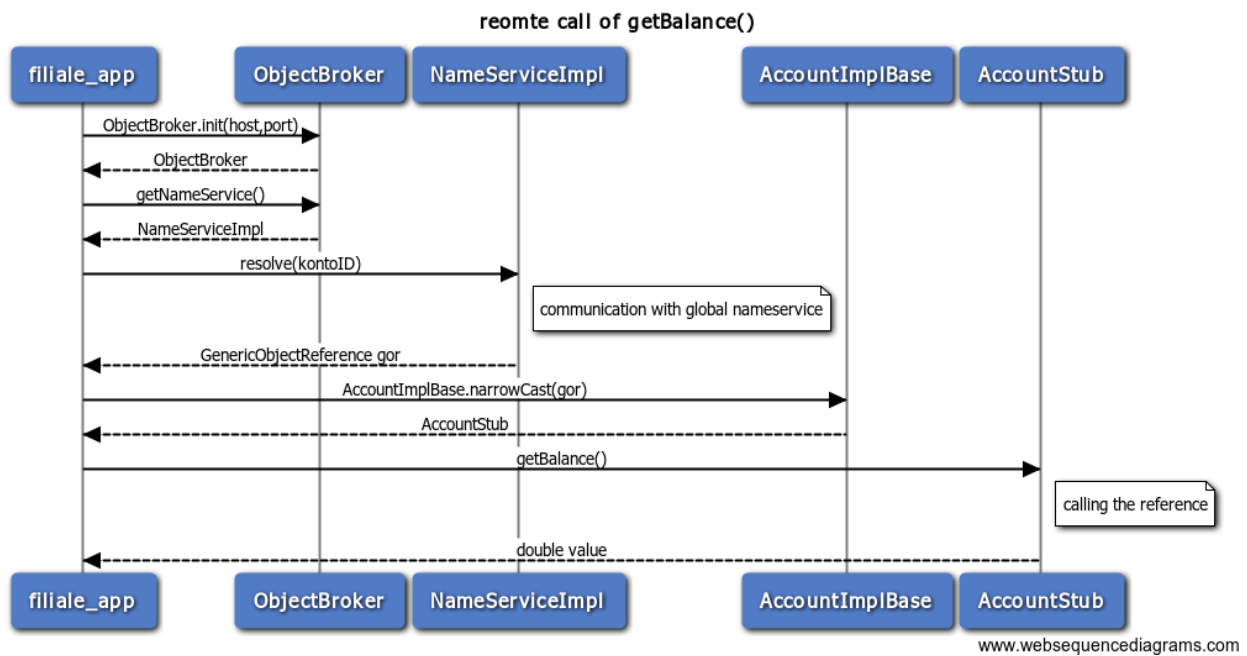
mware_lib



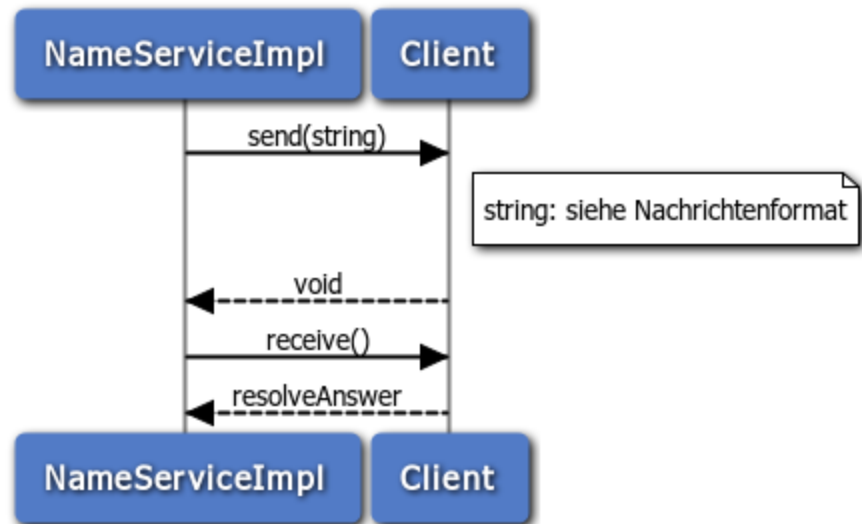
name_service



Laufzeitsichten

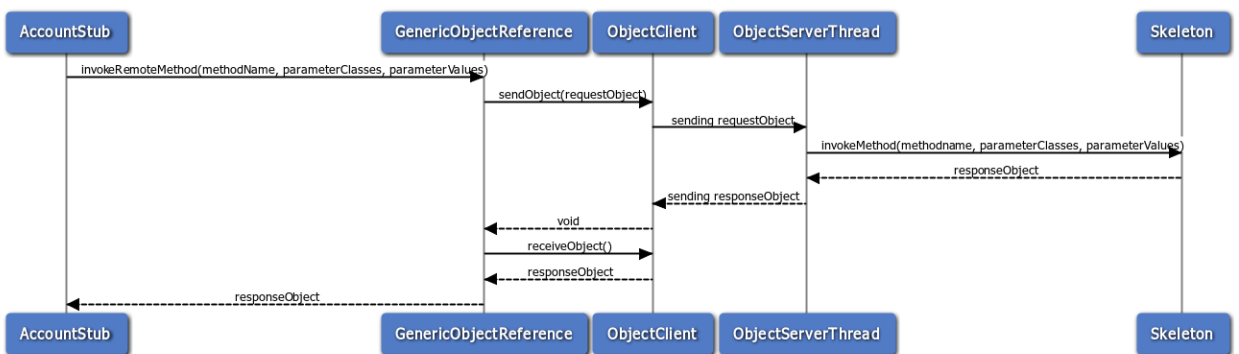


Communication with global nameservice



www.websequencediagrams.com

Communication between Client and Server



www.websequencediagrams.com