**Team:** 09, Alexander Sawadski & Wlad Timotin

**Aufgabenaufteilung:**

Aufgabe wurde gemeinsam bearbeitet

**Quellenangaben:**

Folien von Klauck

**Begründung für Codeübernahme:**

-

**Bearbeitungszeitraum:**

| | |
|---|---|
| 22.11.2014 | 7 std |
| 23.11.2014 | 6 std |
| 26.11.2014 | 3 std |
| 27.11.2014 | 10 std |
| 28.11.2014 | 8 std |
| 29.11.2014 | 11 std |
| 30.11.2014 | 6 std |
| Insgesamt | 51 std |

Aufgabe wurde gemeinsam bearbeitet

**Aktueller Stand:**

| | |
|---|---|
| Entwurf: | fertig |
| Implementierung: | noch nicht fertig |

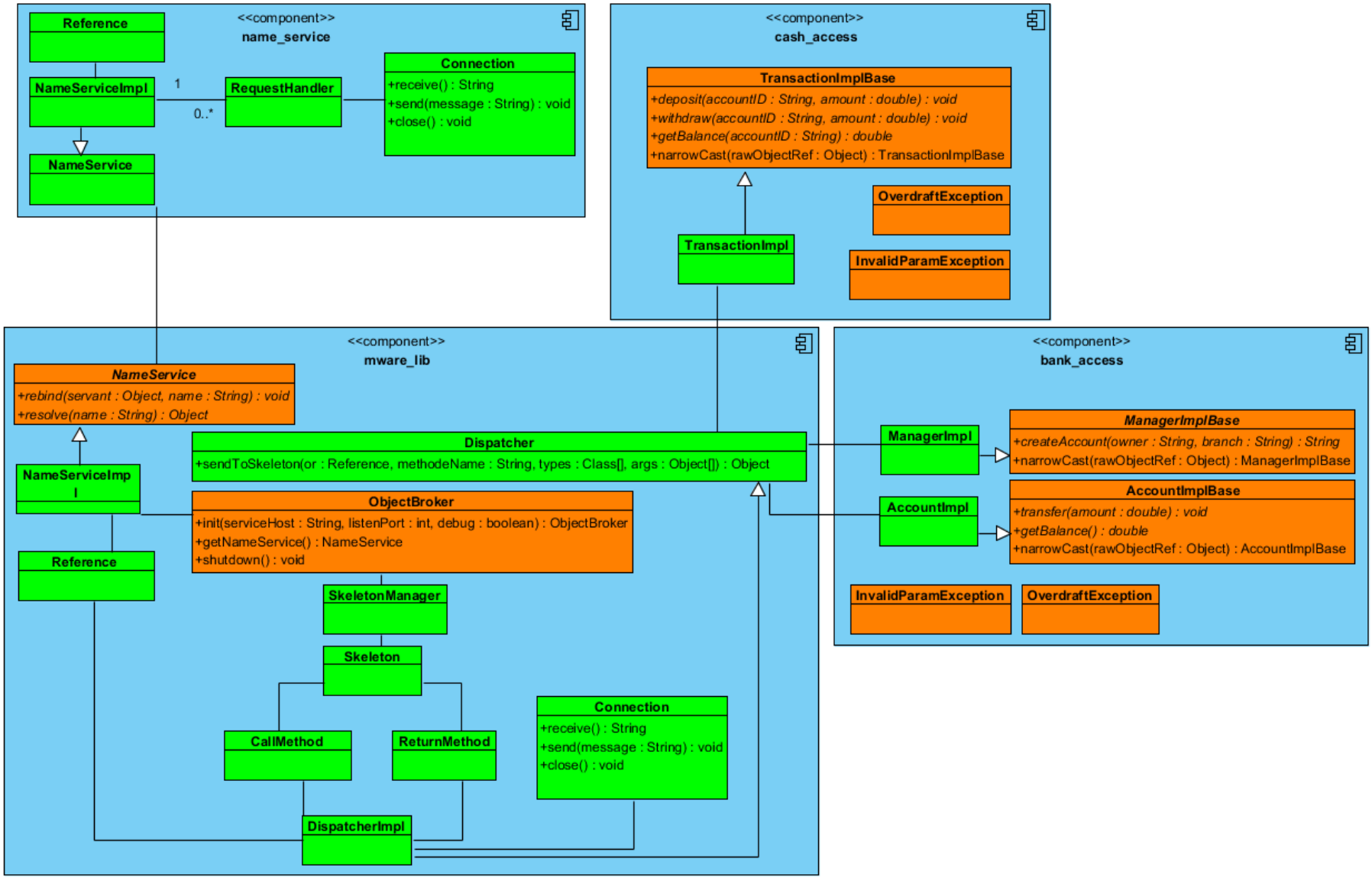
**Änderungen im Entwurf:**

-

**Entwurf:**

version 1

## Komponentendiagramm



## Komponenten

Die Middleware wird in vier Komponenten unterteilt um Flexibilität zu gewärleisten.

**Cash_Access & Bank_Access:**

Cash_Access und Bank_Access sind spezialisierte Teile der Middleware, die Schnittstellen der Anwendungen repräsentieren und dafür sorgen dass die Schnittstellen korrekt übertragen werden.

Ein **Stub** ist der Stellvertreter einer Schnittstelle. Seine Aufgabe sind die Aufrufe einer Methode auf ein entferntes Objekt zu ermöglichen. In unserer Aufgabe sollen 3 festkodierte Stubs implementiert werden.

**ManagerImplBase & ManagerImpl**

Stellt abstrakte Methoden zur Verfügung
createAccount
*narrowCast:* Erzeugt das Stubobjekt für den Anwender

**AccountImplBase & AccountImpl**

Stellt abstrakte Methoden zur Verfügung
transfer, getBalance
*narrowCast:* Erzeugt das Stubobjekt für den Anwender

**TransactionImplBase & TransactionImpl**

Stellt abstrakte Methoden zur Verfügung
deposit, withdraw, getBalance
*narrowCast:* Erzeugt das Stubobjekt für den Anwender

**MWare_lib**

**NameService & NameServiceImpl**

Der NameService ist der Stellvertreter für den NameService aus der name_service -Komponente. Die Servants werden hier verwaltet. Meldet ein Objekt beim Namensdienst an. Liefert einen generischen Objektreferenz zurück.

**Dispatcher & DispatcherImpl**

Nimmt Anfrage von Stub entgegen und sendet CallMethod an zugehörigen Skeleton.

**Reference**

Die Reference eines Objektes. Diese besteht aus folgen Elementen:

*Type*: Der Datentyp des referenzierten Objekts.
*IP*: Die IP, wo sich das Objekt befindet
*Port*: Der Port, wo sich das Objekt befindet
*Name*: Der Klassenname des referenzierten Objekts.

**ObjectBroker**

Der ObjectBroker ist der zentrale Einstiegspunkt der Middleware aus Anwendungssicht. Seine Aufgaben sind das Liefern des Namensdienst und das Beenden der Middleware.

**SkeletonManager**

Der SkeletonManager erstellt und startet mehrere Skeletons.

**Skeleton**

Führt die erhaltenen Aufrufe auf und antwortet mit der ReturnMethod.

**CallMethod**

Wird vom Skeleton und dem Stub verwendet um entfernte Methoden auszuführen.

**ReturnMethod**

Wird vom Skeleton und dem Stub verwendet um die Rückgabe zu erhalten.

**Connection**

Die Connectionklasse ist für die Kommunikation zuständig.

**Name_Service**

**NameService & NameServiceImpl**

Im NameService werden Referenzen zu einem Namen abgelegt.

**Reference**

Die Reference eines Objektes. Diese besteht aus folgen Elementen:

*Type*: Der Datentyp des referenzierten Objekts.
*IP*: Die IP wo sich das Objekt befindet
*Port*: Der Port wo sich das Objekt befindet
*Name*: Der Klassenname des referenzierten Objekts.

**RequestHandler**

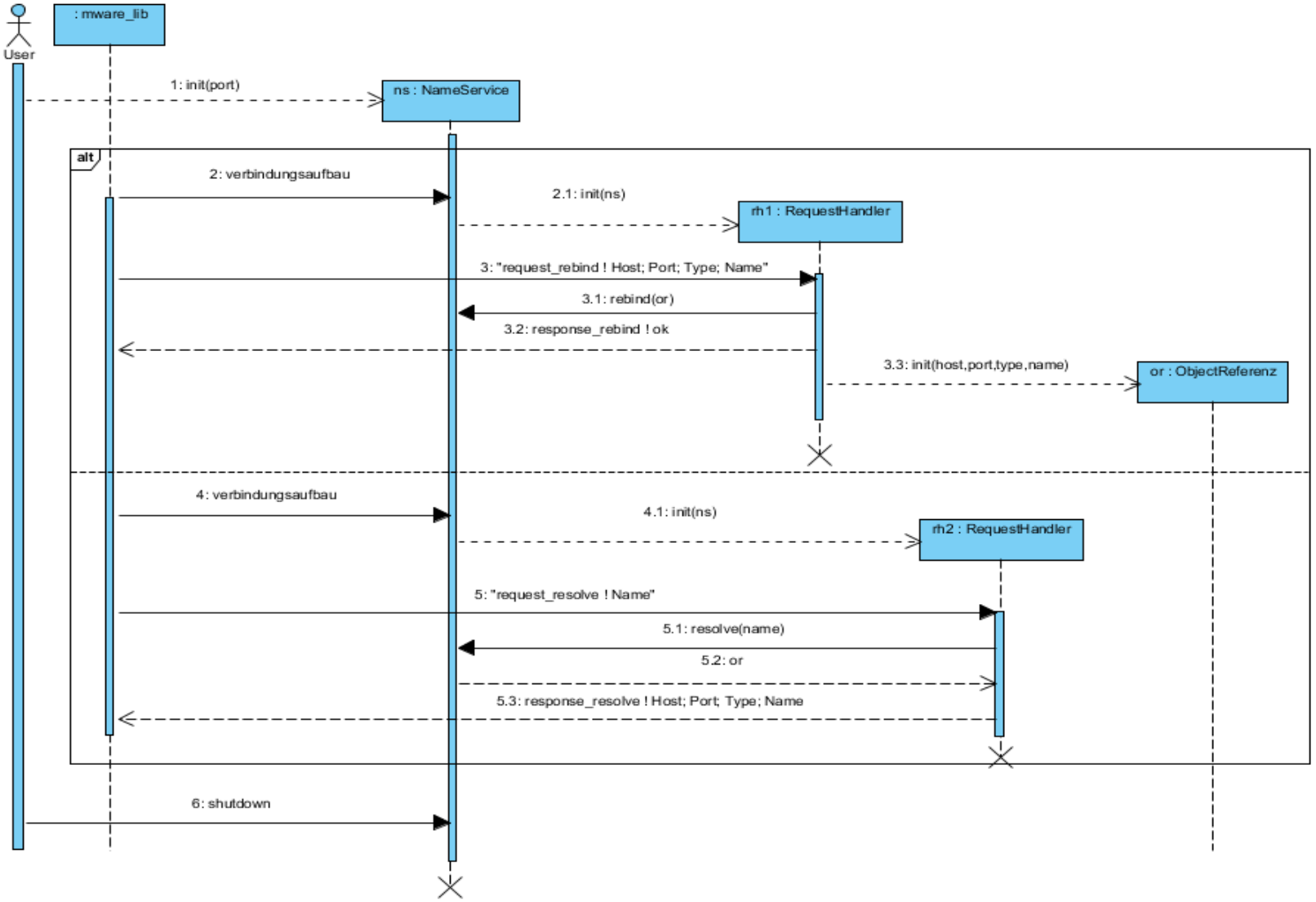
Empfängt/Sendet die Nachrichten von/nach der Middleware.

**Connection**

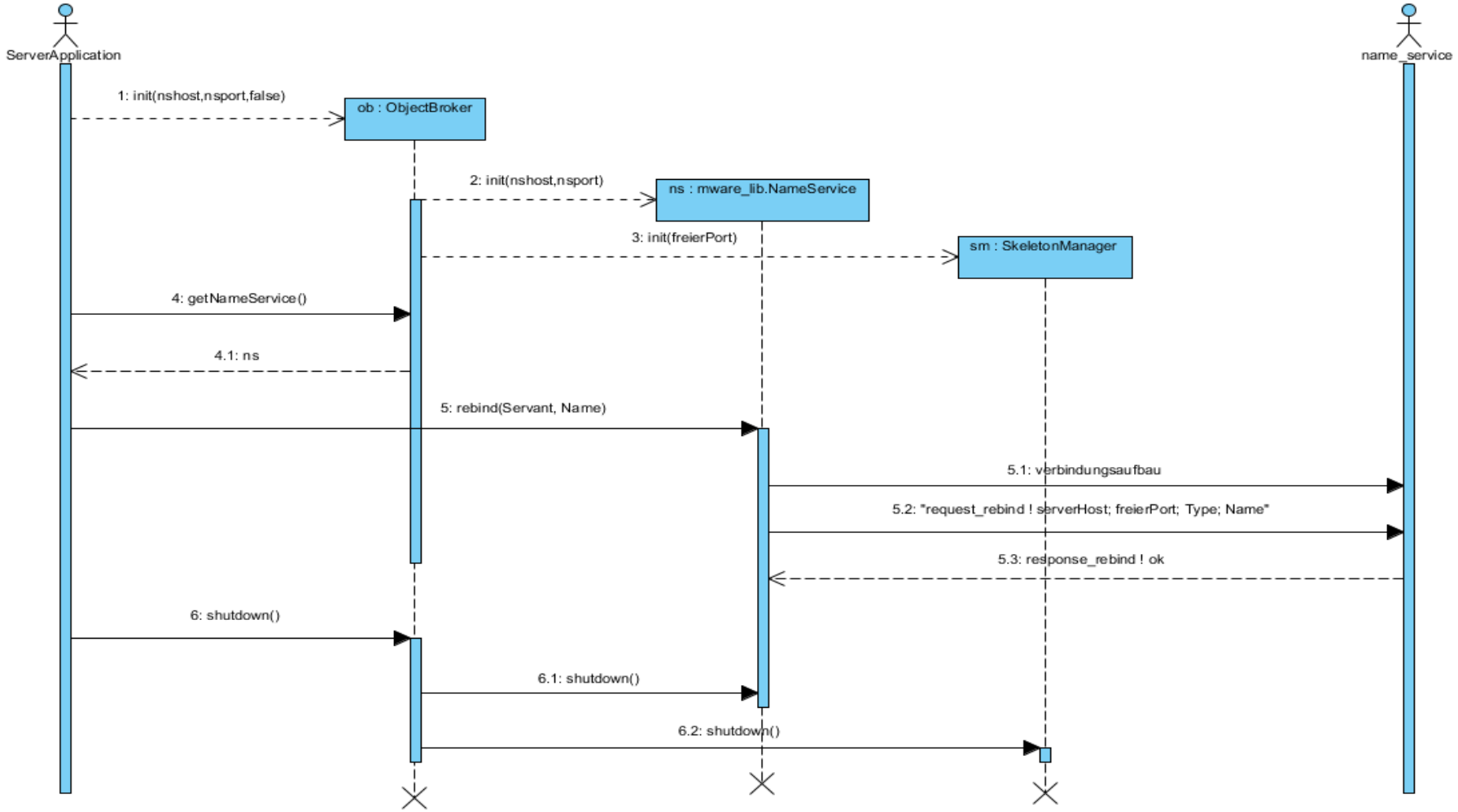
Die Connectionklasse ist für die Kommunikation zuständig.

**Sequenzdiagramm**

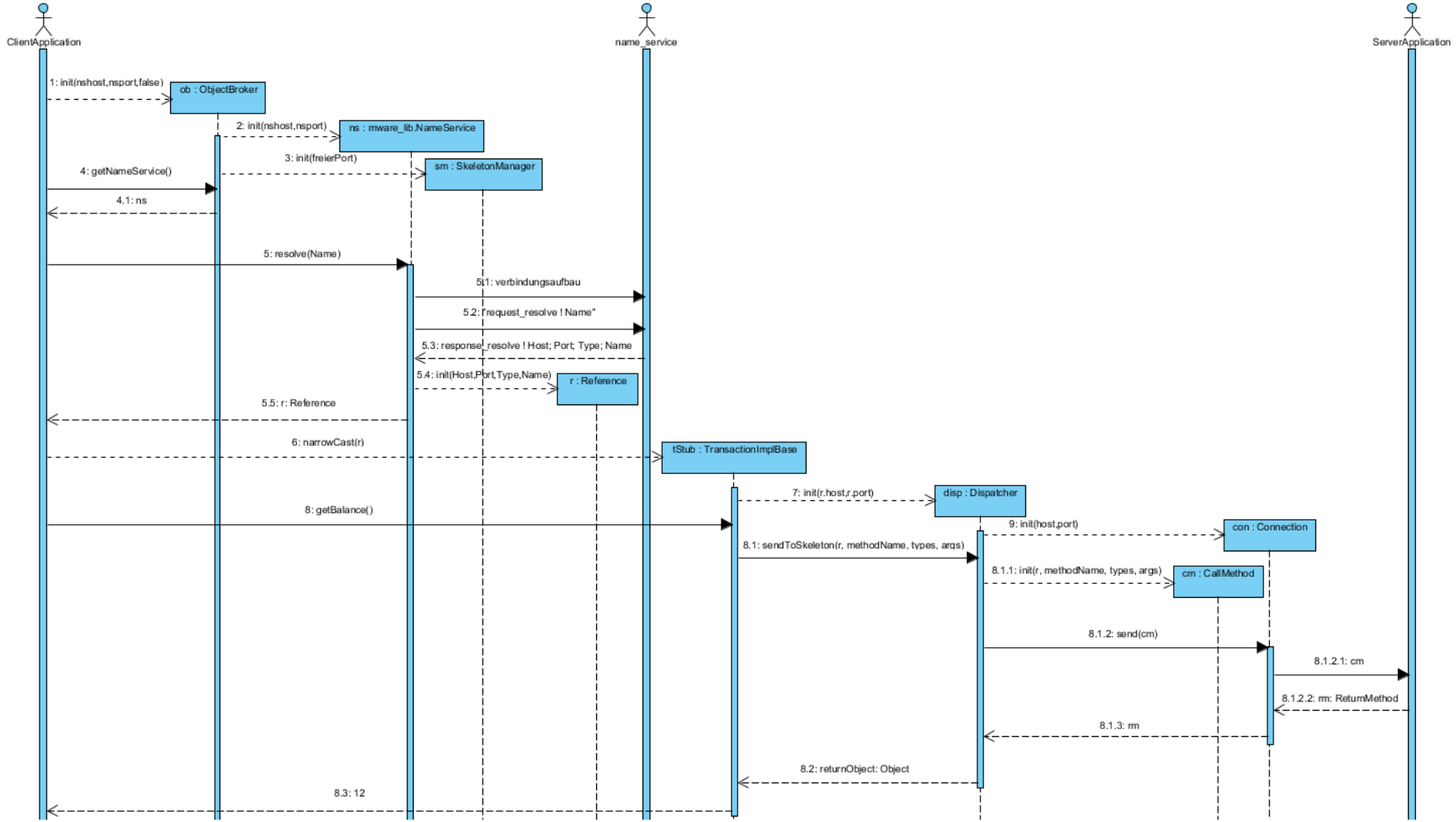
**Nameservice**



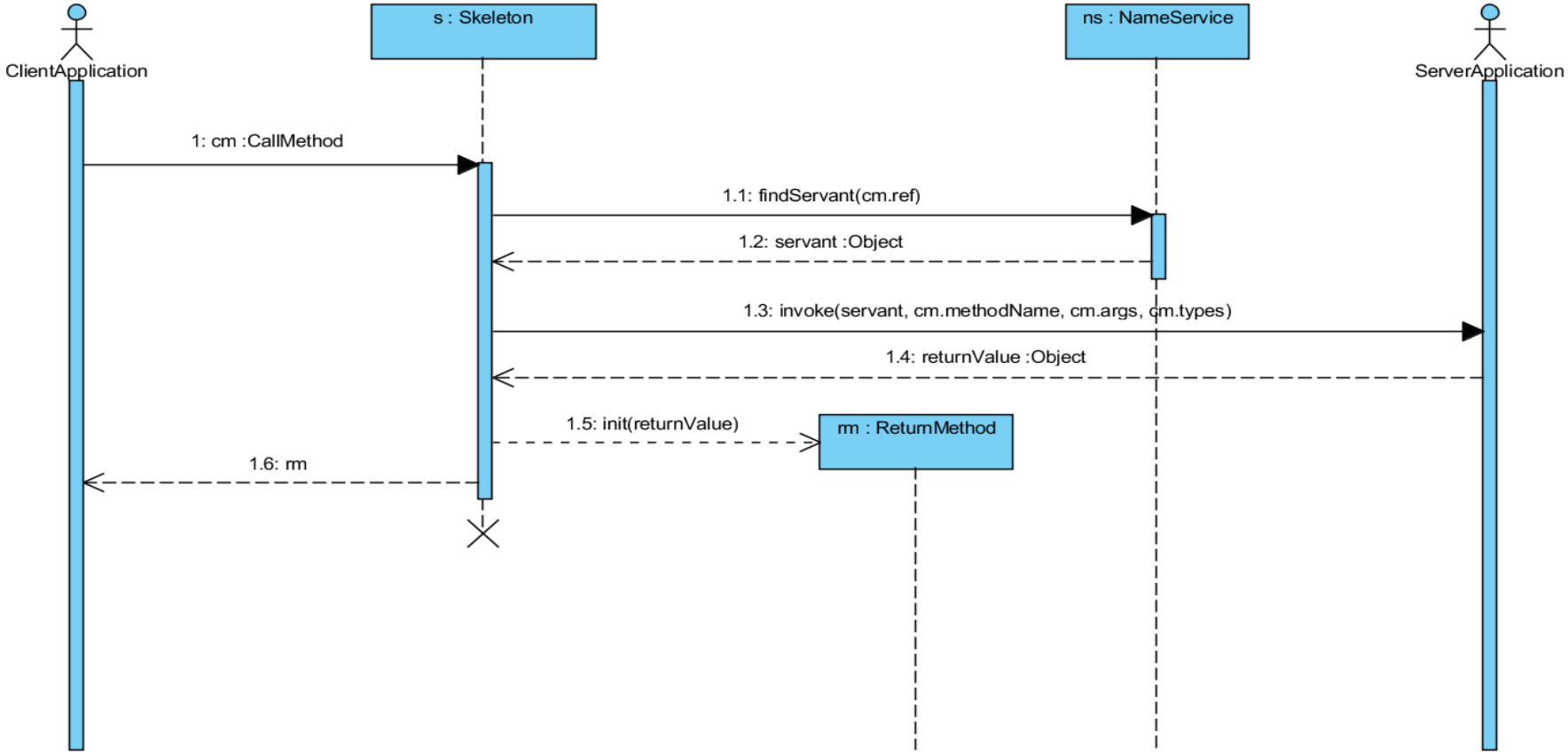
**Middleware_rebind_shutdown**



## Middleware_resolve



**Middleware_Skeleton**



## Protokoll

### Middleware und Nameservice

Zwischen der Middleware und dem NameService werde Zeichenketten rübergeschickt!

Rebind Anfrage:

request_rebind ! Host; Port; Type; Name

Rebind Antwort:

response_rebind ! ok

response_rebind ! error

Resolve Anfrage:

request_resolve ! Name

Resolve Antwort:

response_resolve ! Host; Port; Type; Name

response_resolve ! error

### bank_access und mware_lib

Zwischen bank_access und mware_lib werden Objekte verschickt,
(ReturnMethod und CallMethod)