

.NET Remoting

André Frimberger

30.11.2004

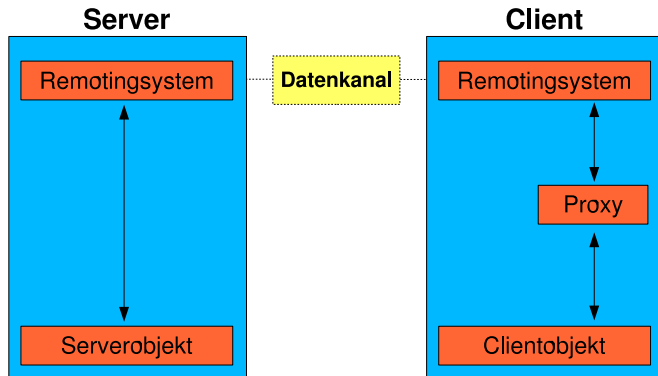
Gliederung

- 1 Einleitung
 - Was ist .NET Remoting?
- 2 Die Interprozess Kommunikation
 - Grundkonzept
 - der Datenkanal
 - Parameterübergabe
 - Instanziierung von Serverobjekten
 - Lebensdauer von Objekten
 - Versionierung
 - WSDL (Web Service Description Language)
 - Sicherheit
 - Beispiele
- 3 Zusammenfassung

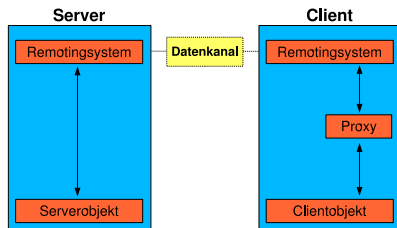
Was ist .NET Remoting?

- .NET Technologie zur Realisierung von verteilten Komponenten
- ermöglicht die Kommunikation zwischen Applikationsdomänen
- plattformunabhängige Interprozesskommunikation
- .NET pendant zu:
 - Remote Procedure Call (RPC) → Andreas Kern
 - Remote Method Invocation (RMI) (Java) → Mario Kiefer

Ein Übertragungsbeispiel

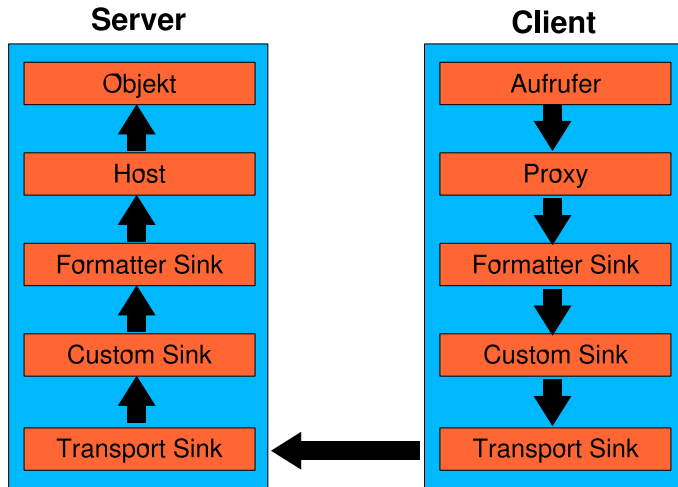


- Clientobjekt stellt Serveranfrage an Proxy
- Proxy kommuniziert mit Remotingsystem und prüft, ob Serverobjekt lokal oder remote



- Remotingsystem auf der Clientseite übergibt Daten zum Versand an den Datenkanal
- Datenkanal verpackt, überträgt und entpackt Daten, um sie anschließend an das Serverremotingsystem zu übergeben
- Remotingsystem auf Serverseite stellt Anfrage an Serverobjekt

Der Datenkanal



- prinzipiell Protokollunabhängig
- einzelne Sinks können ausgetauscht werden
- es werden zwei Übertragungsmethoden angeboten:
 - binäre Datenübertragung per TCP
 - sehr schnell im Vergleich zu SOAP
 - wird defaultmäßig verwendet
 - SOAP über HTTP
 - wegen hohem Protokolloverhead langsamer als binäre Übertragung
 - sehr flexibel, da Übertragung per HTTP erfolgt
 - Kommunikation auch zwischen Netzen möglich, die durch Proxies getrennt sind
 - komplett Plattform- und Programmiersprachenunabhängig

- Ein Beispiel zu SOAP:

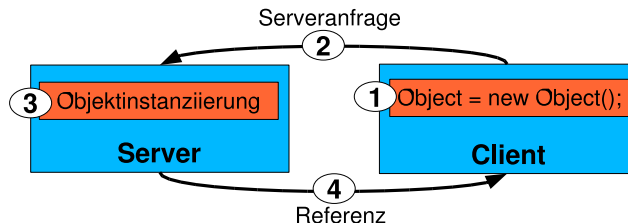
Listing 1: SOAP

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap
    /envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.
    org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>MyParameter</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


Parameterübergabe

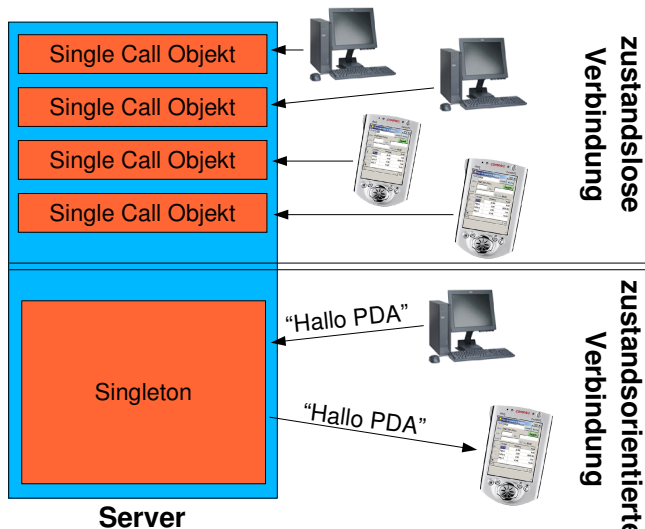
- Marshaling by Reference
 - Objekte werden per Referenz angesprochen
 - Instanziierung auf dem Server, Client bekommt Referenz zurückgeliefert
 - Änderungen vom Client betreffen auch Server
 - für große Datenmengen geeignet
- Marshaling by Value
 - Werte werden zur Verarbeitung kopiert
 - Daten werden serialisiert, bzw. deserialisiert
 - Objekt muss serialisierbar sein
 - Es können nicht alle Klassen serialisiert werden

Client Aktivierung



- Client stellt Anfrage an den Server welches Objekt er in welcher Version haben will
- Server instanziiert Objekt
- Server liefert Referenz auf Objekt zurück

Server Aktivierung



Server Aktivierung

- SingleCall Objekt
 - eine Instanz fertigt genau eine ankommende Anfrage ab
 - besonders gut für Objekte, die nur wenig Arbeit verrichten müssen
 - Zustandslos
- Singleton Objekt
 - eine Instanz fertigt mehrere Anfragen ab
 - besonders gut für Objekte, die viel Arbeit verrichten müssen
 - Zustandsorientiert

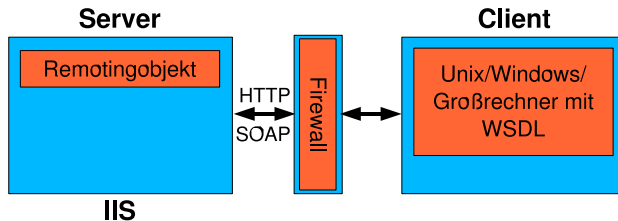
LeaseTime von Objekten

- Timeout kann bei Objektaktivierung als Parameter mit angegeben
- wird nach dem Timeout auf das Objekt zugegriffen, so wird eine Ausnahme geschmissen
- es entsteht kein Netzwerktraffic
- "Lebenserhaltungsmaßnahme" durch Ping Keep-alive-Mechanismus
 - wird über Sponsor Objekt realisiert

Versionierung

- Serveraktivierte Objekte
 - Server entscheidet, welche Version instantiiert wird
 - wird dem Server keine Versionsnummer mitgegeben, so verwendet dieser stets die neueste
- Clientaktivierte Objekte
 - Client entscheidet, welche Version instantiiert wird
 - eine fehlende Versionsangabe impliziert die neueste Version

WSDL (Web Service Description Language)



- XML Sprache zum beschreiben von Webservices für Nachrichtenaustausch
- eine WSDL Datei beschreibt Objekte und Methoden des Webservices
 - kann mit SOAPSUDS.EXE erstellt werden
- besonders interessant in heterogenen Netzwerken



WSDL (Web Service Description Language)

- jeder Client, der diese WSDL Datei verarbeiten kann, kann auch mit dem Service über SOAP kommunizieren
- plattform-
- protokoll-
- programmiersprachen-
- protokollunabhängig
- offener Standard (W3C)

Sicherheit

- .NET Remoting selbst bietet keinerlei Sicherheitsmechanismen an
- Mögliche Authentifizierungsmöglichkeiten
 - Authentifizierung des Internet Information Server (IIS)
 - SSL des IIS
 - von Windows bereitgestellte Authentifizierungsmechanismen
 - Implementierung eines eigenen Datenkanals

Beispiel

Client Server Beispiel...

Zusammenfassung

- yet another Interprocess Communication Model
- bietet kein Sicherheitsmodell
- weites Einsatzgebiet, vom mobilen Endgerät bis zum heterogenen Computernetzwerk
- teils offengelegte Schnittstellen
- modernes Remotingframework
- sehr flexibel

Fragen ?