

# DAIMLERCHRYSLER

## **Propagierung leichtgewichtiger Kontextinformation in SOAP**

Mario Jeckle

DaimlerChrysler Forschungszentrum Ulm

[mario.jeckle@daimlerchrysler.com](mailto:mario.jeckle@daimlerchrysler.com)

[mario@jeckle.de](mailto:mario@jeckle.de)

[www.jeckle.de](http://www.jeckle.de)

## Übersicht

- Begriffsklärung:
  - *Kontext vs. Transaktion*
- Motivation:
  - Zur Notwendigkeit und Anwendung kontextgesicherter Kommunikation
- Technisches Umfeld:
  - Verwandte Lösungsätze
  - W3Cs SOAP v1.2
- Problemstellung:
  - Propagierung programmiersprachen- und webinfrastrukturneutraler (ausschließlich) SOAP-basierter Kontextinformation
- Lösung:
  - SOAPs Erweiterungsmechanismus
  - Syntax
  - Ausführungsmodell
- Anwendungsszenarien und Beispiele

## Begriffsklärung

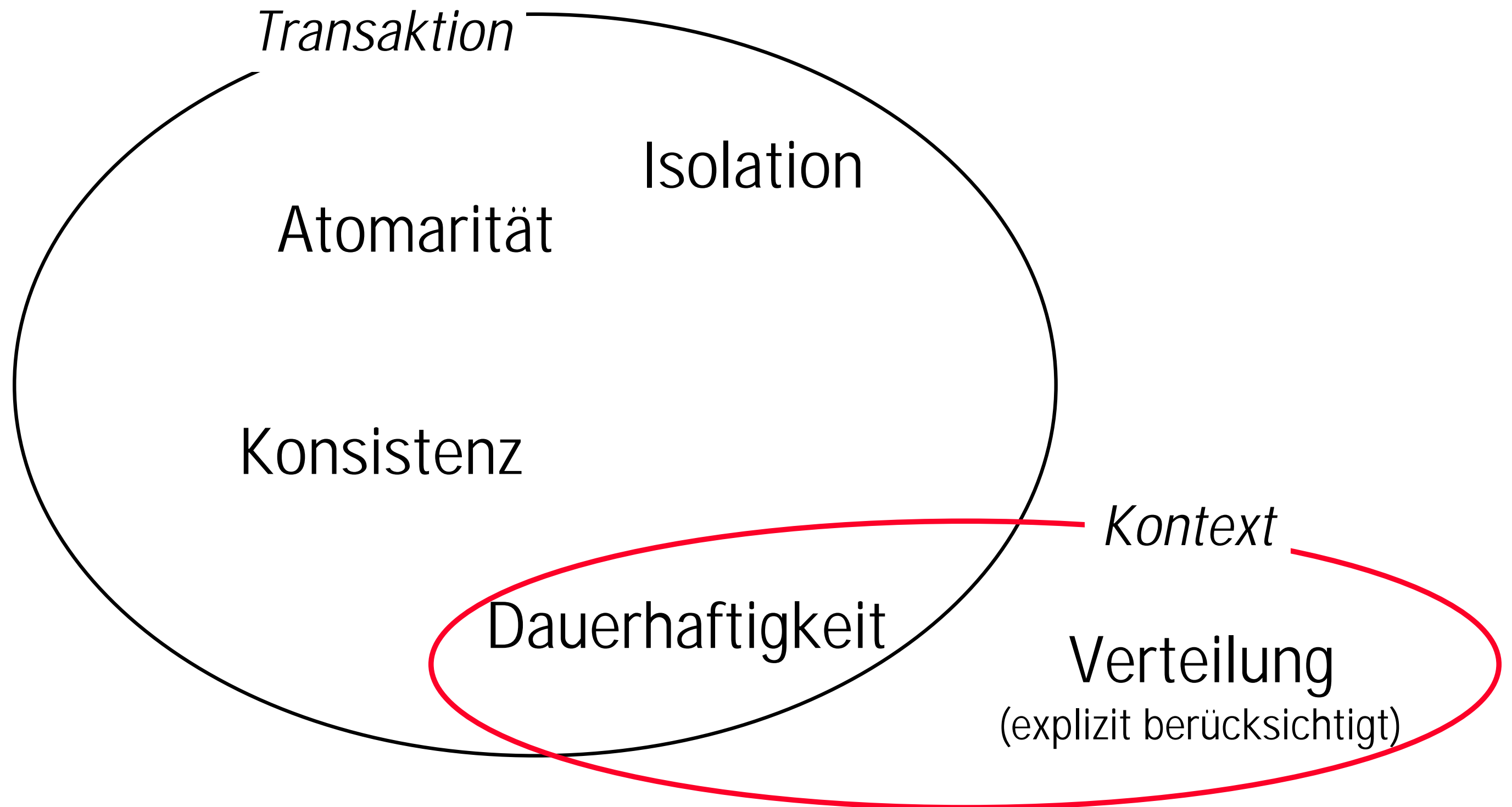
- Transaktion

Aus Integritätsgründen notwendige Behandlung einer Menge von Arbeitsschritten (etwa: Datenbankschreiboperationen) als abgeschlossene Einheit.  
(-> ACID-Prinzipien)

- Kontext (Session)

Verbindung zwischen zwei Partnern in deren Kontext, im Rahmen einer Menge von Interaktionen, Daten ausgetauscht werden.

# Begriffsklärung



## Motivation

Zur Notwendigkeit und Anwendung kontextgesicherter Kommunikation ...

- Typische technische- und business-Kommunikationen bestehen aus mehreren Einzelkommunikationen
  - Zusammenhang zwischen den Einzelkommunikationen muß entweder explizit in jedem Applikationsdatenpaket wiedergegeben werden oder durch zustandsorientierte Applikation verwaltet werden
  - Kontext bildet eine semantische Klammer um eine Menge einzelner Kommunikationsvorgänge
- Explizite Kontextinformation reduziert die übertragene Datenmenge
- Oftmals ist transaktionales Verhalten (d.h. Einhaltung der ACID-Prinzipien) nicht unbedingt notwendig
- Einsatzfelder:
  - Warenkorb-Szenario
  - Login-in, gefolgt von einer Reihe (interaktiver) Operationen, Log-out
  - (Partielle-)Delegation von (Dienstleistungs-)Anfragen

## Technisches Umfeld

Verwandte und existierende Lösungsansätze:

- Transaktionsverarbeitung
  - *OASIS Business Transaction Protocol (BTP)*  
An klassischer (verteilter) transaktionaler Verarbeitung (d.h. 2PC) orientiert
  - *Java Transaction Service (JTS)*,  
Spezifikation eines *Java Transaction API (JTA)* konformen *Transaction Managers*  
Java Mapping auf CORBAs Object Transaction Service (OTS)
  - *XML Transaction Authority Markup Language (XAML)*  
... wird nicht weiterentwickelt

## Technisches Umfeld

Verwandte und existierende Lösungsansätze:

- Kontextverarbeitung
  - HTTP/1.1 (IETF 2068, 2616)
    - keep alive gestattet lediglich den Client über das Verbindungsende zu informieren.
  - Cookies (IETF RFC 2109, 2964)
    - Lediglich (kleine textwertige) Namen-Wert-Paare für HTTP.
  - URL „Anreicherung“ (IETF RFC 1945, 2616; URL Rewrite)
    - Lediglich für HTTP definiert.
    - Unter Sicherheitsgesichtspunkten höchst fragwürdig.
    - Für Web Services nicht möglich, da Service URL nicht dynamisch ausgehandelt werden kann.

## Technisches Umfeld

### W3Cs SOAP v1.2

- Gegenwärtig durch W3C Arbeitsgruppe *XML Protocol* standardisiert.
- Baut auf dem *Simple Object Access Protocol* auf.
- Definiert ein erweiterbares Rahmenwerk und ein Verarbeitungsmodell zur XML-Encodierung beliebiger Nachrichten und entfernter Methodenaufrufe (RPC) über Internetmechanismen.
- Erweiterbar durch anwenderdefinierte SOAP Header Elemente.
- Aktive SOAP Knoten (*Intermediaries*) prozessieren aktiv Nachrichten (-anteile) zwischen Sender und endgültigem Empfänger.
- Häufigste Anwendung: *SOAP RPC Repräsentation* (SOAP Part2, Kap. 4).
- Häufigst verwendetes Transportprotokoll: HTTP.

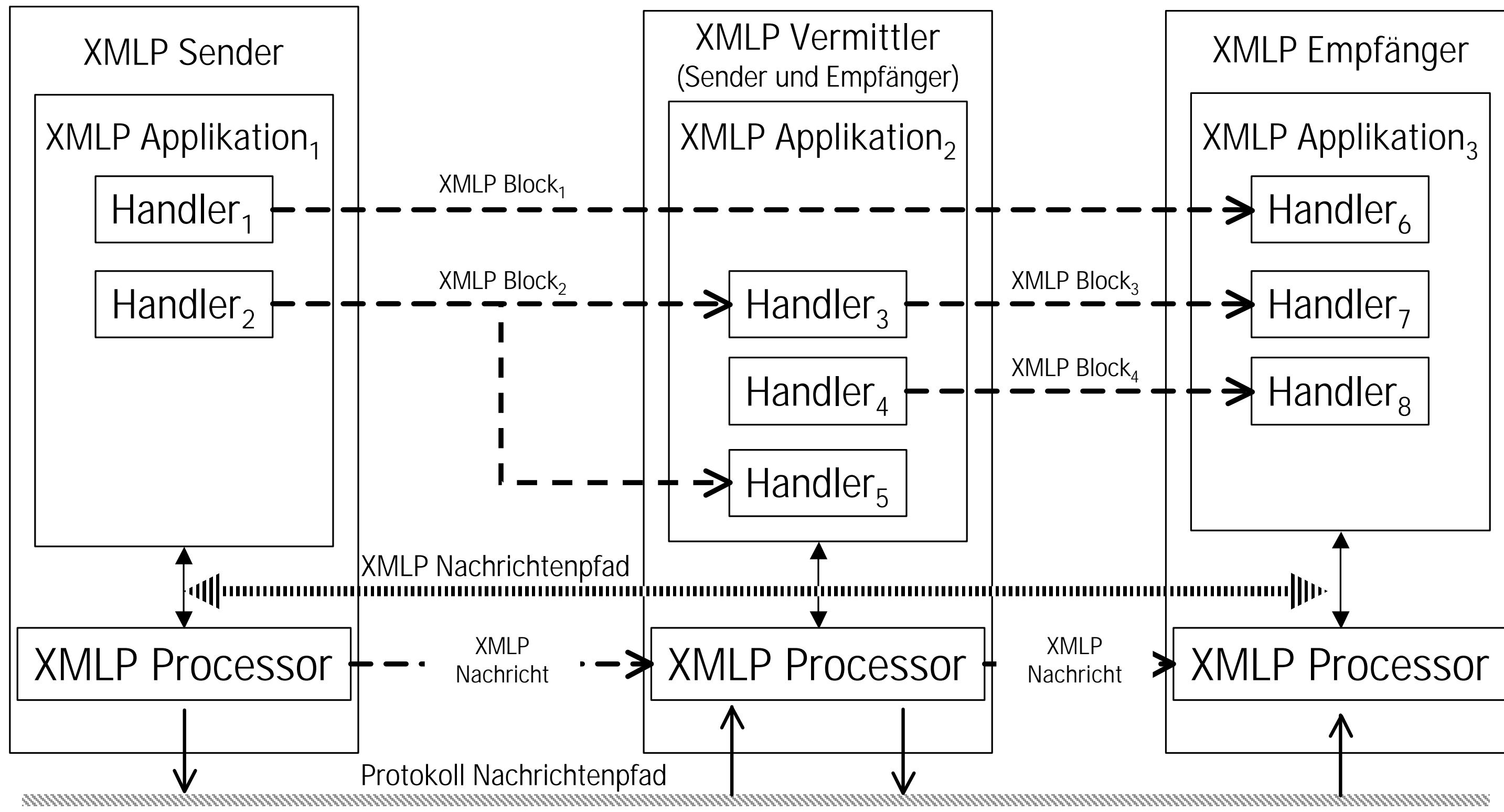


# Technisches Umfeld

XMLP-Knoten

XMLP-Knoten

XMLP-Knoten



## Technisches Umfeld

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:m="http://example.org/2001/06/quotes">
  <env:Header>
    <m:doSomething >42</m:doSomething>
  </env:Header>
  <env:Body>
    <m:GetLastTradePrice
      env:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
      <m:Symbol>DCX</m:Symbol>
    </m:GetLastTradePrice>
  </env:Body>
</env:Envelope>
```

# Technisches Umfeld

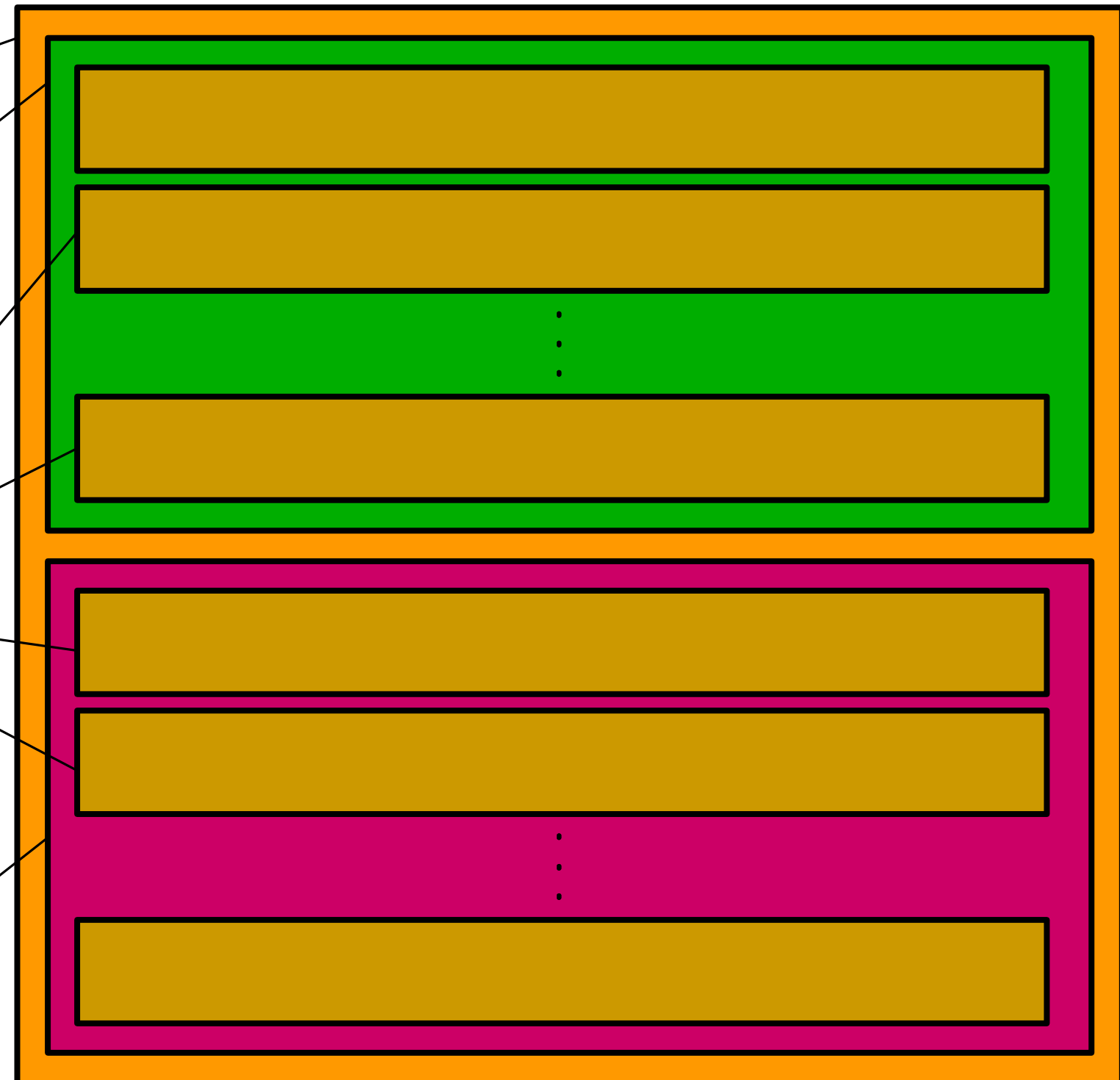
## Anatomie einer SOAP v1.2-Nachricht

SOAP-Nachricht

SOAP-Header

SOAP-Block

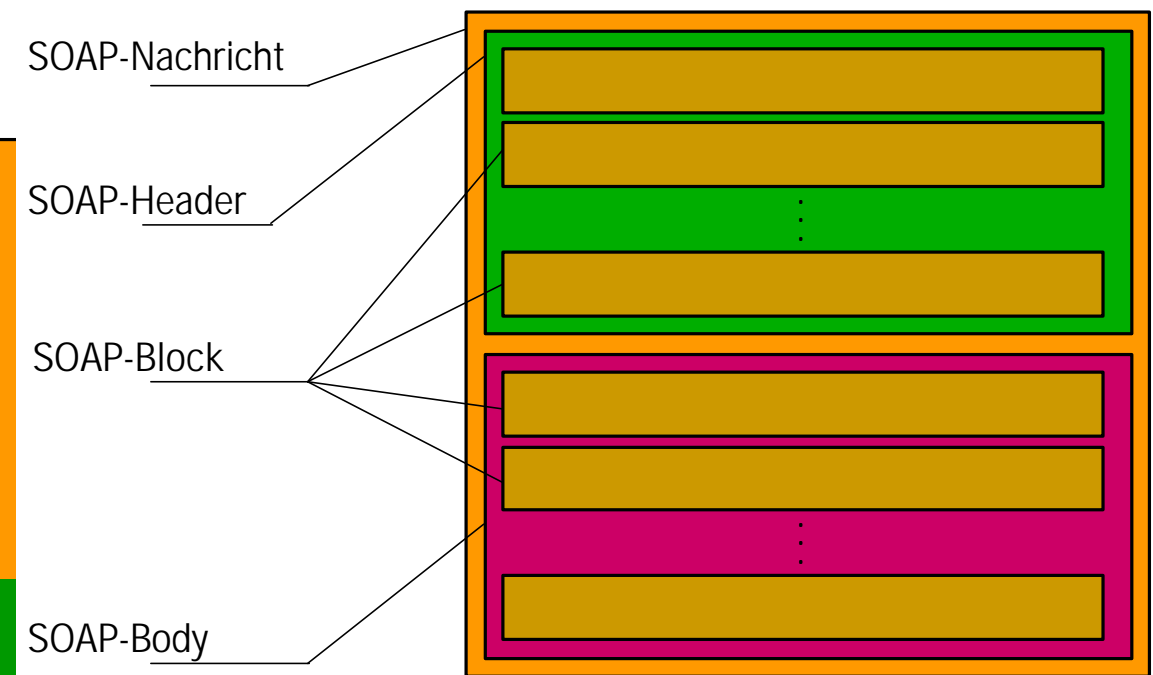
SOAP-Body



# Technisches Umfeld

```

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:m="http://example.org/2001/06/quotes">
  <env:Header>
    <m:doSomething>42</m:doSomething>
  </env:Header>
  <env:Body>
    <m:GetLastTradePrice
      env:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
      <m:Symbol>DCX</m:Symbol>
    </m:GetLastTradePrice>
  </env:Body>
</env:Envelope>
  
```



## Technisches Umfeld

- SOAP Header (SOAP v1.2, Part1, Sect. 5.2)
  - ... sind (metainformations-)Anteile einer SOAP-Nachricht.
  - Können durch Anwender wahlfrei und in SOAP-Nachrichten eingebettet werden.
  - Sind einem eigenen (d.h. nicht-SOAP-Namensraum) zugeordnet.
  - Verarbeitungspflicht (mustUnderstand-Attribut) durch *SOAP Knoten* kann durch Anwender festgelegt werden.
- SOAP Knoten (SOAP v1.2, Part1, Sect. 2.1)
  - ... sind aktive Elemente entlang eines Protokollnachrichtenpfades, die SOAP-codierte Inhalte verarbeiten können.
  - Können SOAP Header konsumierend verarbeiten.
  - Können explizite Adressaten eines Headers sein.

## Problemstellung

Propagierung programmiersprachen- und webinfrastrukturneutraler (ausschließlich) SOAP-basierter Kontextinformation

- Gewünschte Lösungseigenschaften:
  - Kontextpropagierung
    - => keine Transaktionalität.
    - => Business Transaction Protocol, XML Transaction Authority Markup Language scheiden aus.
  - Programmiersprachenneutralität
    - => Applikationsprachen-basierte Ansätze (Java Transactions) scheiden aus.
  - Webinfrastrukturneutralität:
    - => HTTP-basierte Ansätze (HTTP/1.1, Cookies, URL „Anreicherung“) scheiden aus.
  - (ausschließliche) SOAP-Basiertheit
    - => Nutzung der SOAP-inhärente Erweiterungsmechanismen.

→ Nutzung der SOAP-Header

## Lösung

### SOAPs Erweiterungsmechanismus

- Definition eines eigenen Headers zur Aufnahme von Kontextinformation
  - Jeder Kontext wird durch einen eindeutigen Identifier identifiziert.
  - Jeder Kontext ist mit einer begrenzten Gültigkeitszeit.
  - Jeder Kontext trägt die Identifikation seines Erzeugers.
  - Kontexte können geschachtelt werden.

## Lösung

### SOAPs Erweiterungsmechanismus

- Schritte zur Definition eines eigenen Headers zur Aufnahme von Kontextinformation
  - Definition eines geeigneten Namensraumes.
  - Definition des notwendigen Vokabulars.
  - Definition des Ausführungsmodells.



## Lösung

### SOAPs Erweiterungsmechanismus

- Schritte zur Definition eines eigenen Headers zur Aufnahme von Kontextinformation
  - Definition eines geeigneten Namensraumes
  - Definition des notwendigen Vokabulars
  - Definition des Ausführungsmodells

Prinzipiell jede syntaktisch korrekte URI möglich, sinnvollerweise jedoch eine über längere Zeit unveränderliche, die idealerweise sogar Rückschlüsse auf den Verwendungskontext gestattet.

<http://www.example.com/SOAPContext>

## Lösung

### SOAPs Erweiterungsmechanismus

- Schritte zur Definition eines eigenen Headers zur Aufnahme von Kontextinformation
    - Definition eines geeigneten Namensraumes
    - **Definition des notwendigen Vokabulars**
    - Definition des Ausführungsmodells
1. Ein Element Information Item zur Kapselung des Headers
  2. Die Nutzinformation
    1. Über Zeit und Raum eindeutige Kontextidentifikation.
    2. Verfallszeitpunkt.
    3. Erzeugeridentifikation.

## Lösung

### SOAPs Erweiterungsmechanismus

- Schritte zur Definition eines eigenen Headers zur Aufnahme von Kontextinformation
  - Definition eines geeigneten Namensraumes
  - **Definition des notwendigen Vokabulars**
  - Definition des Ausführungsmodells

### 1. Ein Element Information Item zur Kapselung des Headers

```
<ctx:Context xmlns:ctx="http://www.example.com/SOAPContext">
```

```
</ctx:Context>
```

## Lösung

### SOAPs Erweiterungsmechanismus

- Schritte zur Definition eines eigenen Headers zur Aufnahme von Kontextinformation
  - Definition eines geeigneten Namensraumes
  - **Definition des notwendigen Vokabulars**
  - Definition des Ausführungsmodells

## 2. Die Nutzinformation

### 1. Über Zeit und Raum eindeutige Kontextidentifikation

```
<ctx:Context xmlns:ctx="http://www.example.com/SOAPContext">  
  <ctx:ID>f2f7c5ec-5123-40dc-be64-2b9953e60201</ctx:ID>  
  
</ctx:Context>
```

## Lösung

### SOAPs Erweiterungsmechanismus

- Schritte zur Definition eines eigenen Headers zur Aufnahme von Kontextinformation
  - Definition eines geeigneten Namensraumes
  - **Definition des notwendigen Vokabulars**
  - Definition des Ausführungsmodells

## 2. Die Nutzinformation

### 2. Verfallszeitpunkt

```
<ctx:Context xmlns:ctx="http://www.example.com/SOAPContext">  
  <ctx:ID>f2f7c5ec-5123-40dc-be64-2b9953e60201</ctx:ID>  
  <ctx:Expires>2002-06-28T08:25:00:00Z</ctx:Expires>  
  
</ctx:Context>
```

## Lösung

### SOAPs Erweiterungsmechanismus

- Schritte zur Definition eines eigenen Headers zur Aufnahme von Kontextinformation
  - Definition eines geeigneten Namensraumes
  - **Definition des notwendigen Vokabulars**
  - Definition des Ausführungsmodells

### 2. Die Nutzinformation

### 3. Erzeugeridentifikation

```
<ctx:Context xmlns:ctx="http://www.example.com/SOAPContext">  
  <ctx:ID>f2f7c5ec-5123-40dc-be64-2b9953e60201</ctx:ID>  
  <ctx:Expires>2002-06-28T08:25:00:00Z</ctx:Expires>  
  <ctx:Issuer>http://www.example.com/someIssuer</ctx:Issuer>  
</ctx:Context>
```

## Lösung

### SOAPs Erweiterungsmechanismus

- Schritte zur Definition eines eigenen Headers zur Aufnahme von Kontextinformation
  - Definition eines geeigneten Namensraumes
  - Definition des notwendigen Vokabulars
  - **Definition des Ausführungsmodells**
- **Kontext-Header müssen durch jeden SOAP-Knoten verarbeitet werden  
(mustUnderstand="true" und role="...next")**

## Lösung

### SOAPs Erweiterungsmechanismus

- Schritte zur Definition eines eigenen Headers zur Aufnahme von Kontextinformation
  - Definition eines geeigneten Namensraumes
  - Definition des notwendigen Vokabulars
  - **Definition des Ausführungsmodells**
    - **Verfallszeitpunkt darf nur eingeschränkt, nicht ausgeweitet werden**



## Lösung

### SOAPs Erweiterungsmechanismus

- Schritte zur Definition eines eigenen Headers zur Aufnahme von Kontextinformation
  - Definition eines geeigneten Namensraumes
  - Definition des notwendigen Vokabulars
  - **Definition des Ausführungsmodells**
- **SOAP-Knoten (excl. SOAP-Sender) verwaltet Kontext-Information**  
**=> Keine Datenstrukturen im Client**

## Lösung

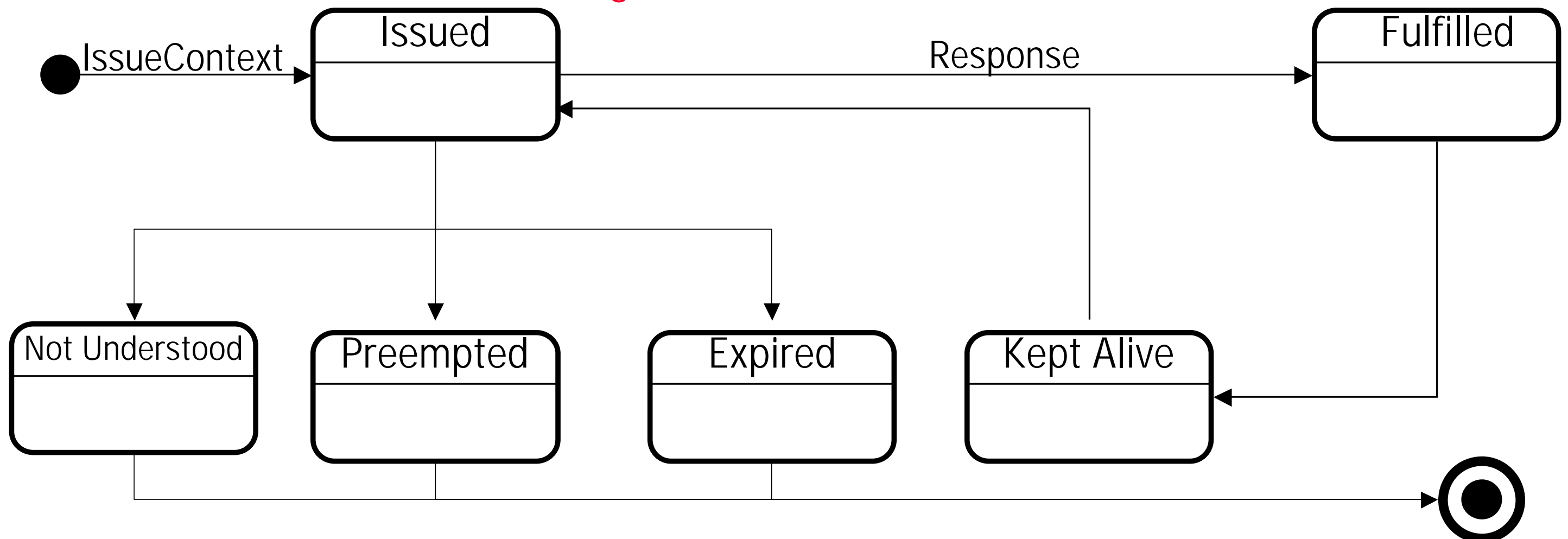
### SOAPs Erweiterungsmechanismus

- Schritte zur Definition eines eigenen Headers zur Aufnahme von Kontextinformation
  - Definition eines geeigneten Namensraumes
  - Definition des notwendigen Vokabulars
  - **Definition des Ausführungsmodells**
    - **Kann ein Kontext nicht erfüllt werden, d.h. kann die Anfrage bis zum Eintritt der Verfallzeitpunkts oder früher nicht beantwortet werden, erhält der sendende SOAP-Knoten eine Fehlermeldung**

## Lösung

### SOAPs Erweiterungsmechanismus

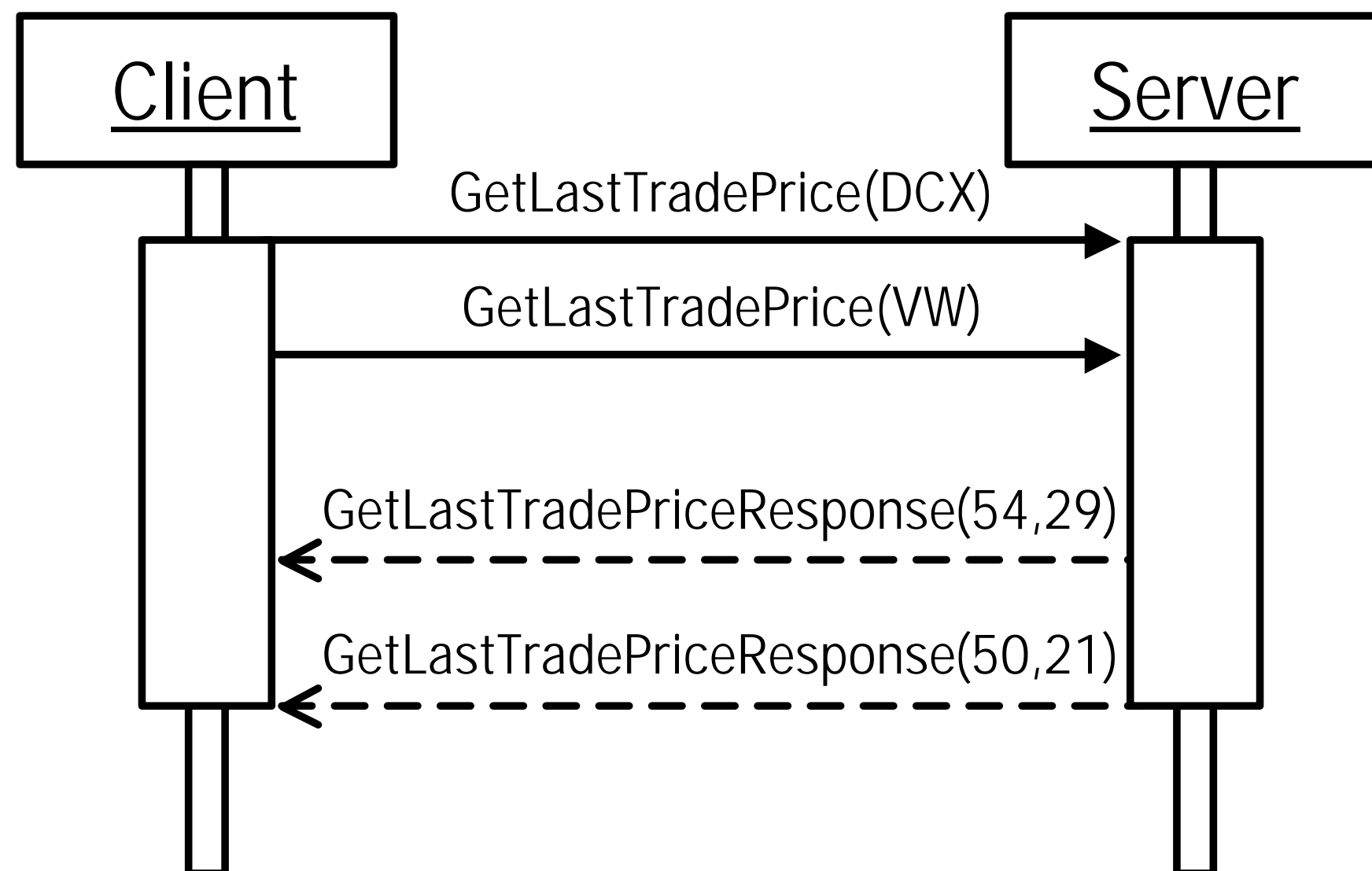
- Schritte zur Definition eines eigenen Headers zur Aufnahme von Kontextinformation
  - Definition eines geeigneten Namensraumes
  - Definition des notwendigen Vokabulars
  - **Definition des Ausführungsmodells**



# Anwendungsszenarien und Beispiele

## Nachrichten-Korrelation

- Grundidee:  
Herstellung der Sender-Empfänger-Korrespondenz bei asynchroner Dienstleistung
- Beispiel:  
Aktienkursanfrage



## Anwendungsszenarien und Beispiele

### Nachrichten-Korrelation

- Beispiel:  
Aktienkursanfrage
- Problem:  
Asynchrone Kommunikation  
ohne Kontextinformation  
nicht durchführbar,  
da Ergebnis nicht (mehr)  
der Anfrage zugeordnet  
werden kann

```
<env:Envelope
  xmlns:env="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <m:GetLastTradePrice
      env:encodingStyle="http://www.w3.org/2001/12/soap-encoding"
      xmlns:m="http://example.org/2001/06/quotes">
      <m:Symbol>DCX</m:Symbol>
    </m:GetLastTradePrice>
  </env:Body>
</env:Envelope>
```

```
<env:Envelope
  xmlns:env="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <m:GetLastTradePriceResponse
      env:encodingStyle="http://www.w3.org/2001/12/soap-encoding"
      xmlns:m="http://example.org/2001/06/quotes">
      <m:Price>50,21</m:Price>
    </m:GetLastTradePriceResponse>
  </env:Body>
</env:Envelope>
```

# Anwendungsszenarien und Beispiele

## Nachrichten-Korrelation

- Lösung:
  - Anfrage(n)

```
<env:Envelope
  xmlns:env="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <ctx:Context xmlns:ctx="http://www.example.com/SOAPContext">
    <ctx:ID>f2f7c5ec-5123-40dc-be64-2b9953e60201</ctx:ID>
    <ctx:Expires>2002-06-28T08:25:00:00Z</ctx:Expires>
    <ctx:Issuer>URN:pin:bs4321234</ctx:Issuer>
  </ctx:Context>

  <env:Body>
    <m:GetLastTradePrice
      env:encodingStyle="http://www.w3.org/2001/12/soap-encoding"
      xmlns:m="http://example.org/2001/06/quotes">
      <m:Symbol>DCX</m:Symbol>
    </m:GetLastTradePrice>
  </env:Body>
</env:Envelope>
```

# Anwendungsszenarien und Beispiele

## Nachrichten-Korrelation

- Lösung:
  - Antwort(en)

```
<env:Envelope
  xmlns:env="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

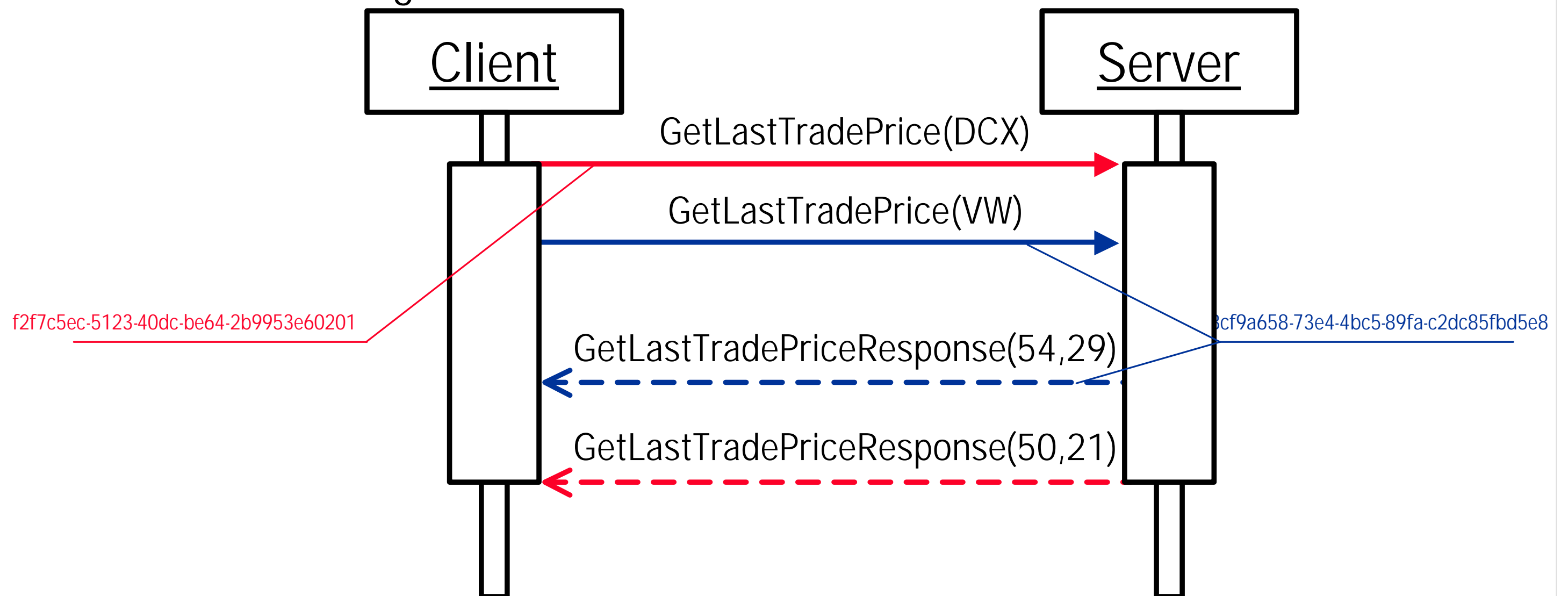
  <ctx:Context xmlns:ctx="http://www.example.com/SOAPContext">
    <ctx:ID>f2f7c5ec-5123-40dc-be64-2b9953e60201</ctx:ID>
    <ctx:Expires>2002-06-28T08:25:00:00Z</ctx:Expires>
    <ctx:Issuer>URN:pin:bs4321234</ctx:Issuer>
  </ctx:Context>

  <env:Body>
    <m:GetLastTradePriceResponse
      env:encodingStyle="http://www.w3.org/2001/12/soap-encoding"
      xmlns:m="http://example.org/2001/06/quotes">
      <m:Price>50,21</m:Price>
    </m:GetLastTradePriceResponse>
  </env:Body>
</env:Envelope>
```

# Anwendungsszenarien und Beispiele

## Nachrichten-Korrelation

- Lösung:
  - Kontextbildung

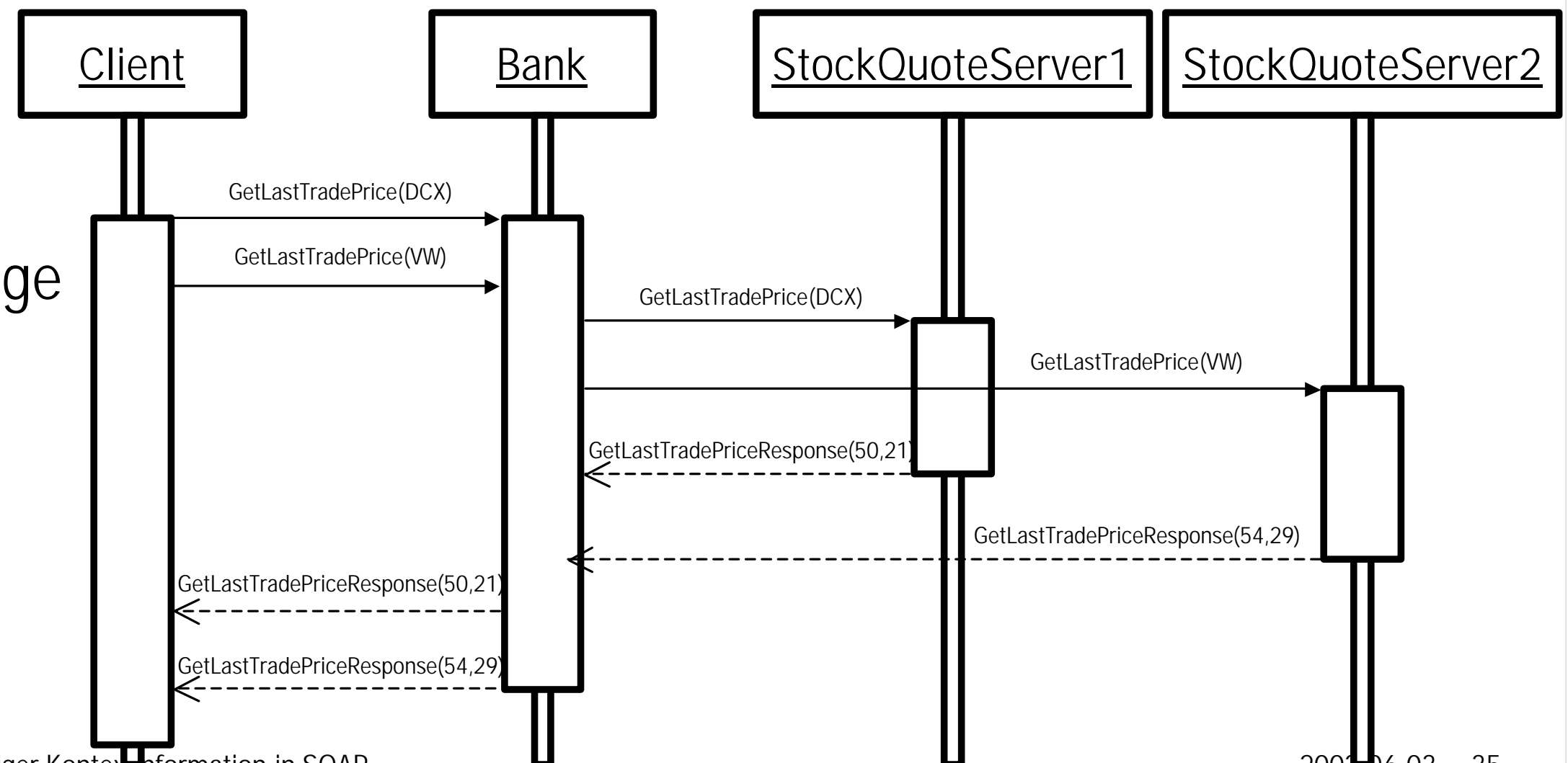




# Anwendungsszenarien und Beispiele

## Delegative Dienstleistung

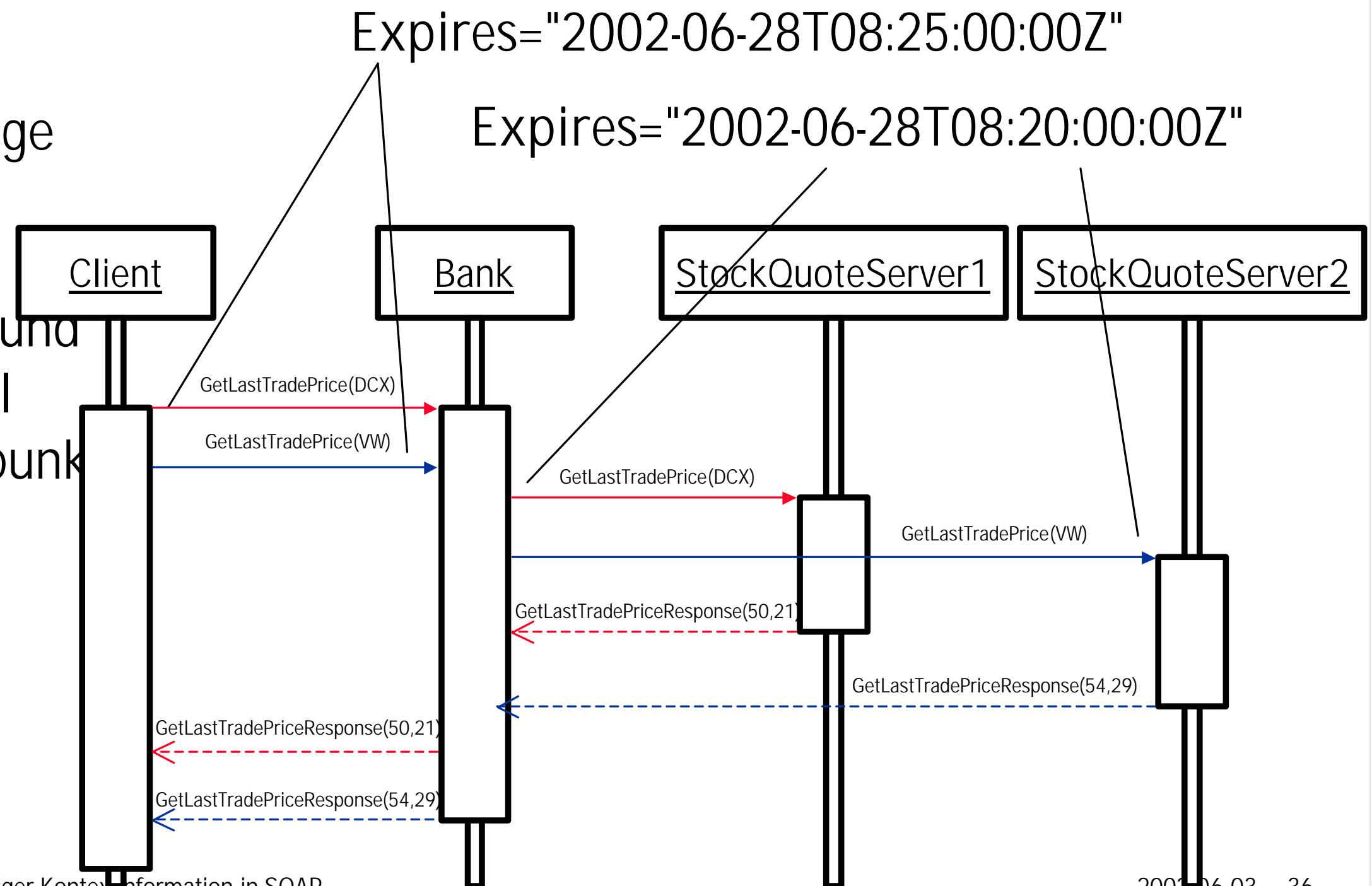
- Grundidee:
  - Garantierte Antwortzeit
  - Asynchrone parallele Aufrufe
- Beispiel:  
Aktienkursanfrage



# Anwendungsszenarien und Beispiele

## Delegative Dienstleistung

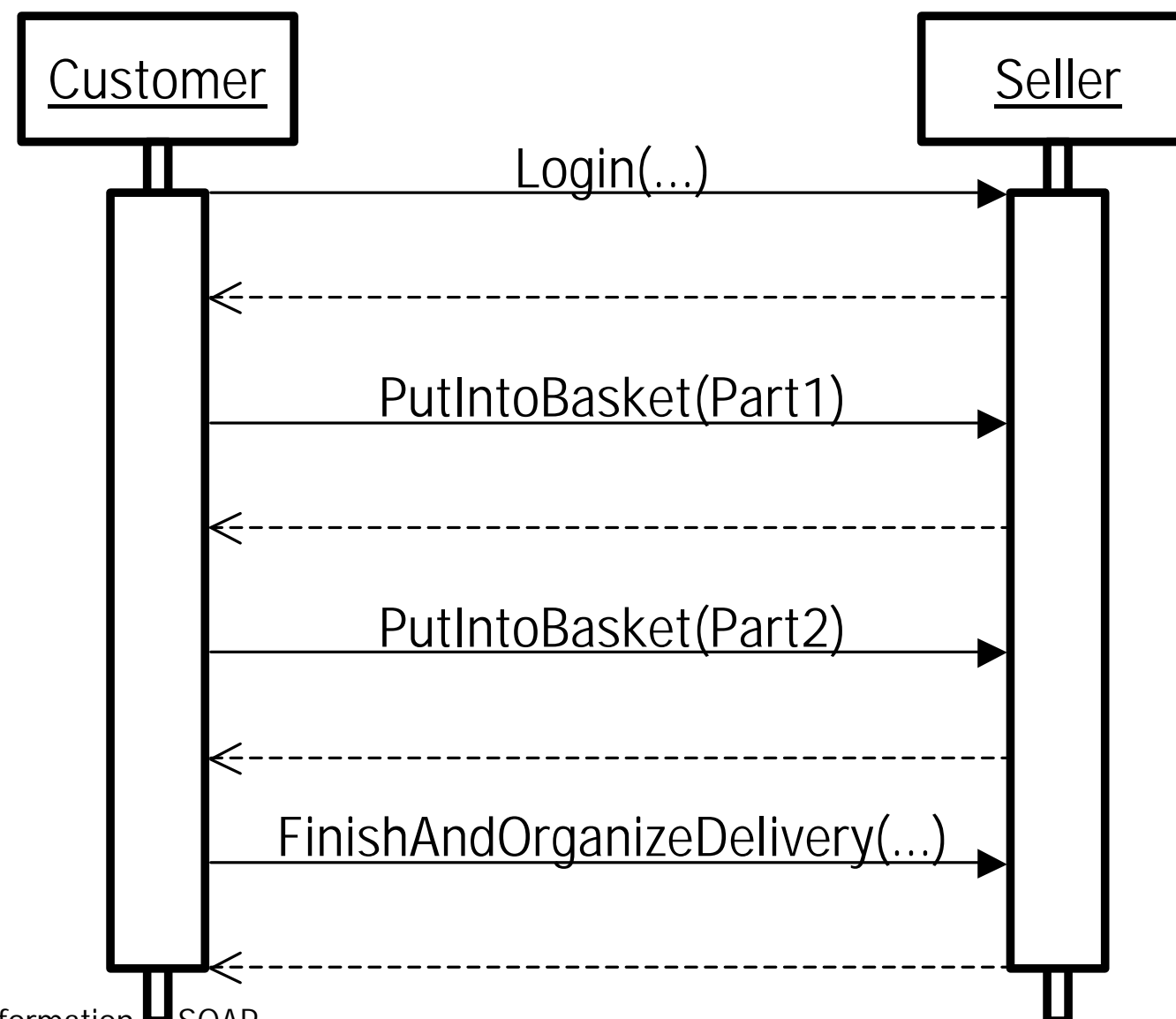
- Beispiel:  
Aktienkursanfrage
- Lösung:  
Kontextbildung und geschickte Wahl der Verfallszeitpunkt



# Anwendungsszenarien und Beispiele

## Eingebettete Aktionen

- Grundidee:
  - Kontexte innerhalb von Kontexten
- Beispiel:  
Warenkorbfüllung



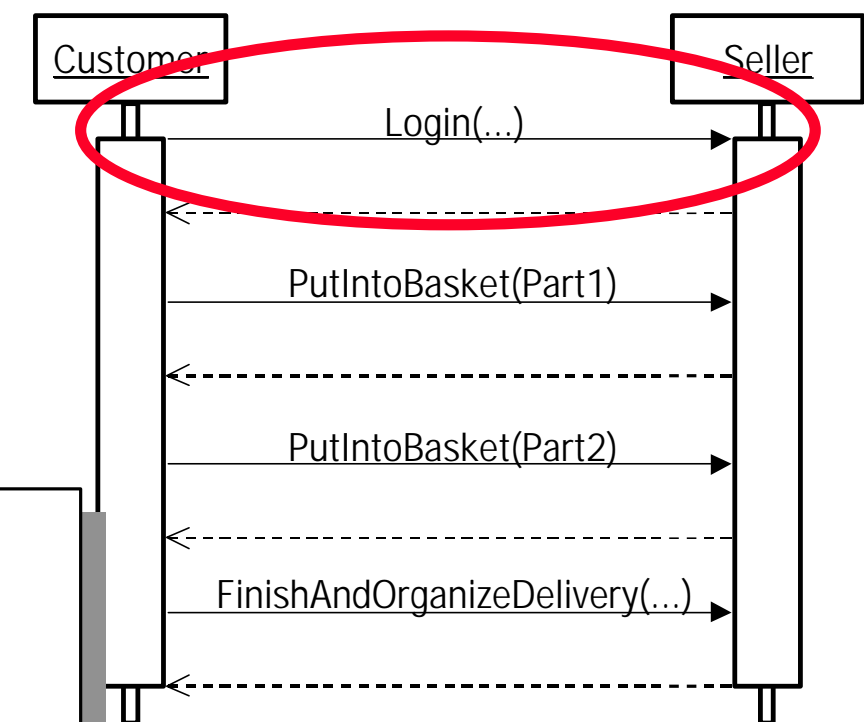
# Anwendungsszenarien und Beispiele

## Eingebettete Aktionen

```
<env:Envelope
  xmlns:env="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <ctx:Context xmlns:ctx="http://www.example.com/SOAPContext"
    keepAlive="true">
    <ctx:ID>1165e806-cc02-4a68-9345-92f932fbd1dd</ctx:ID>
    <ctx:Expires>2002-06-30T10:00:00:00Z</ctx:Expires>
    <ctx:Issuer>URN:pin:bs4321234</ctx:Issuer>
  </ctx:Context>

  <env:Body>
    <m:CreateBasket
      env:encodingStyle="http://www.w3.org/2001/12/soap-encoding"
      xmlns:m="http://example.org/eCommerceSample">
      <m:Username>John Doe</m:Username>
    </m:CreateBasket>
  </env:Body>
</env:Envelope>
```



# Anwendungsszenarien und Beispiele

## Eingebettete Aktionen

```
<env:Envelope
  xmlns:env="http://www.w3.org/2001/09/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <ctx:Context xmlns:ctx="http://www.example.com/SOAPContext">
    <ctx:ID bindTo="1165e806-cc02-4a68-9345-92f932fbd1dd"
      da7e49f4-deab-4e5a-9d61-589fa901df59</ctx:ID>
    <ctx:Expires>2002-06-30T10:00:00:00Z</ctx:Expires>
    <ctx:Issuer>URN:pin:bs4321234</ctx:Issuer>
  </ctx:Context>

  <env:Body>
    <m:CreateBasket
      env:encodingStyle="http://www.w3.org/2001/12/soap-encoding"
      xmlns:m="http://example.org/eCommerceSample">
      <m:Username>John Doe</m:Username>
    </m:CreateBasket>
  </env:Body>
</env:Envelope>
```

