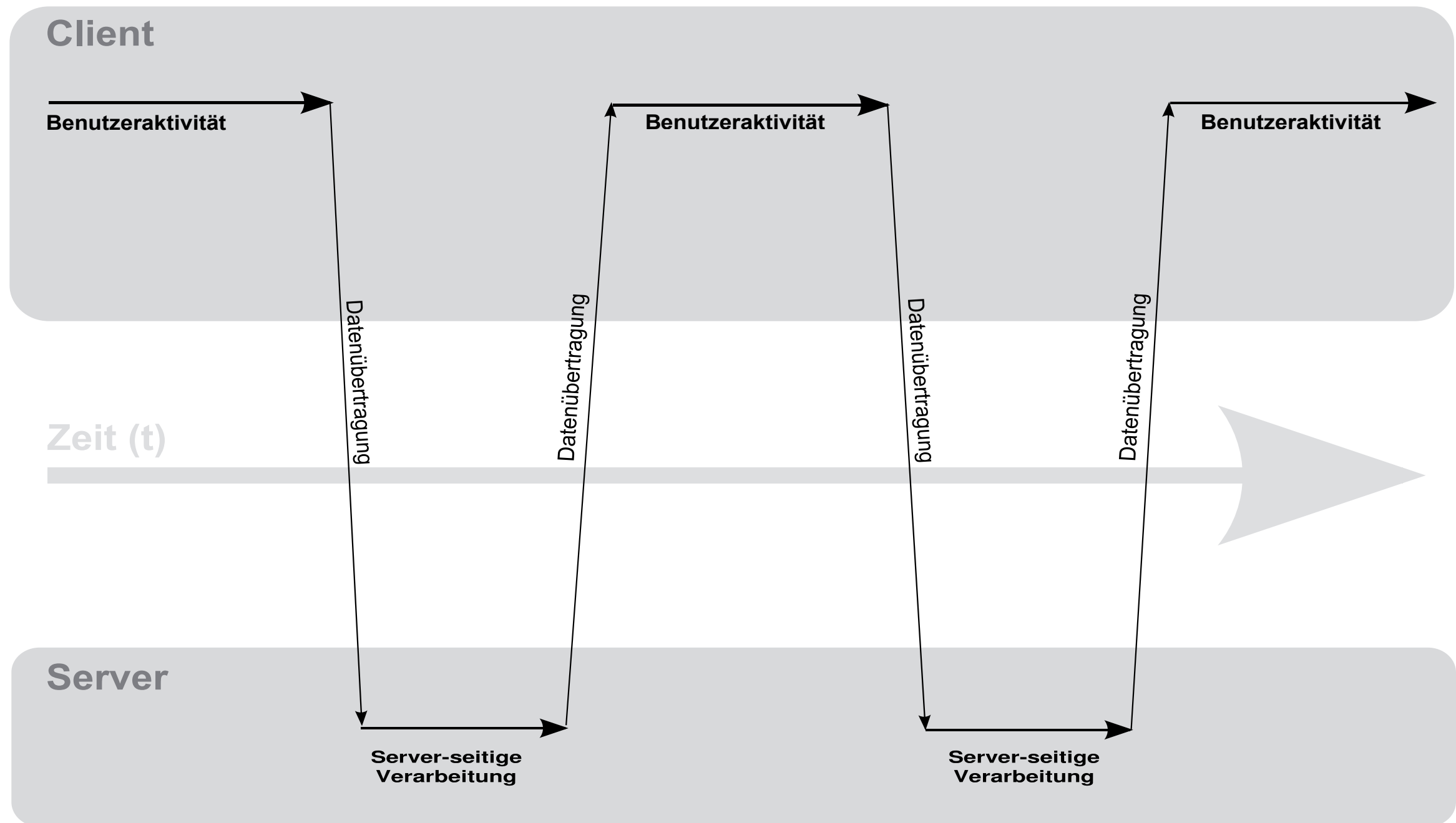


Asynchrone Kommunikation - Eine erste Vertiefung

Asynchrone Kommunikation Vertiefung - Ziele

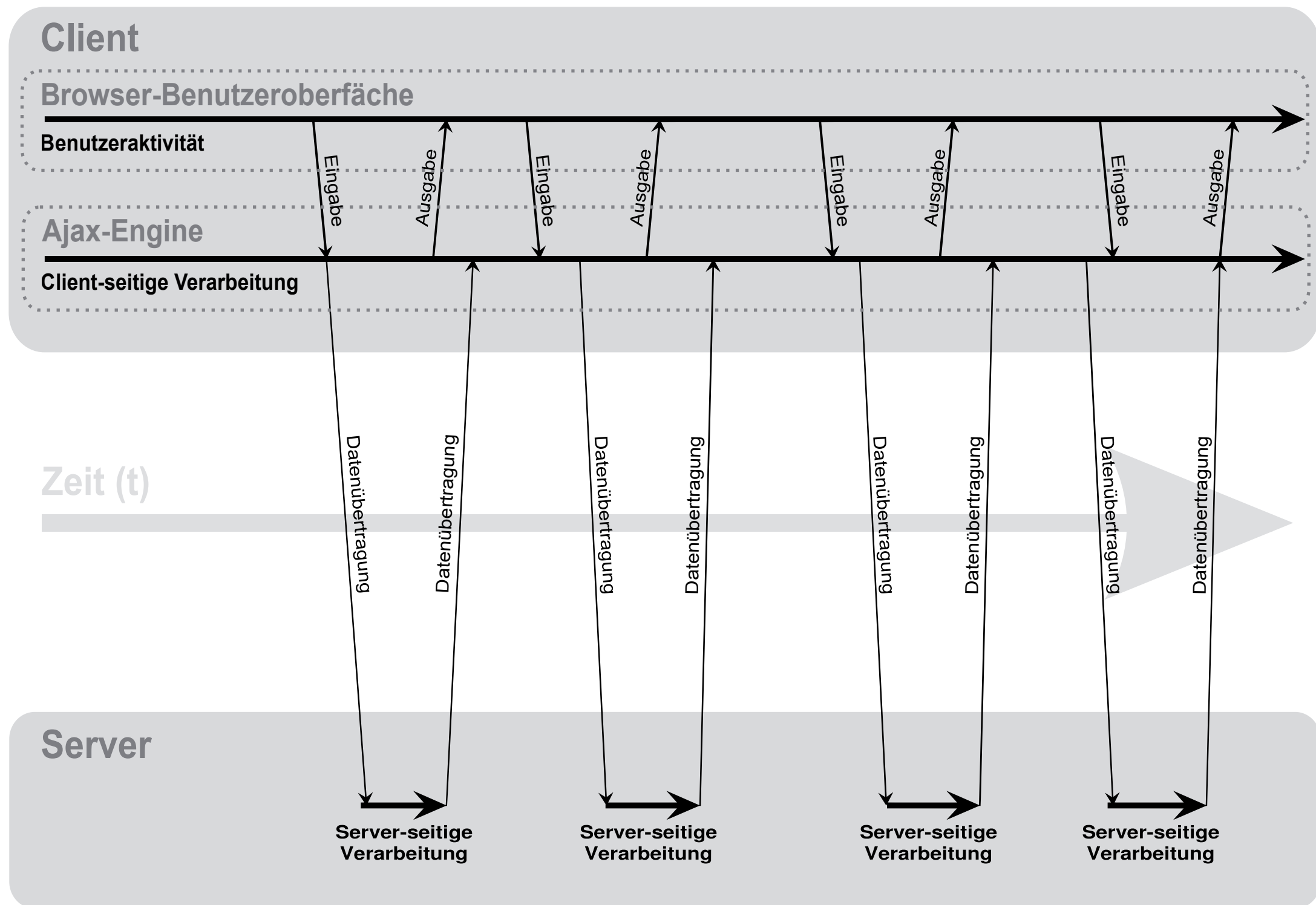
- Die Asynchronität in Browser basierten Anwendungen auf der Basis von Ajax erklären können.
- Die asynchrone Kommunikation auf Basis des Atom Standards einordnen und erklären können.
- Die zeitliche und die referenzielle Entkopplung als Möglichkeiten der Realisierung von loser Kopplung erklären können.
- Wesentliche Konzepte der asynchronen Kommunikation auf der Basis von Middleware Systemen am Beispiel JMS klären können.

Aktivitätsfluss: WWW Anwendung



Quelle: www.wikipedia.de/ajax

Aktivitätsfluss: AJAX / WWW Anwendung



Quelle: www.wikipedia.de/ajax

Taxonomie der Formen der Koordination

Beispiele	zeitlich gekoppelt	zeitlich entkoppelt
refereziell gekoppelt	synchroner Dienstaufruf	asynchroner Dienstaufruf
refereziell entkoppelt	Multicast	generative Kommunikation

Really Simple Syndication (RSS)



Subscription

RSS Reader (Polling)

XML Format: Inhalt + Metadaten

Weiterentwicklung: Atom

Atom Syndication Format ->
registrierte Medientypen

Atom Publishing Protocol (AtomPub) --> standardisierte
REST-Schnittstelle zum Publizieren

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<feed xmlns="http://www.w3.org/2005/Atom">
```

```
  <title>Example Feed</title>
```

```
  <link href="http://example.org/feed/" rel="self" />
```

```
  <link href="http://example.org/" />
```

```
  <id>urn:uuid:60a76c80-d399-11d9-b91C-0003939e0af6</id>
```

```
  <updated>2003-12-13T18:30:02Z</updated>
```

```
  <entry>
```

```
    <title>Atom-Powered Robots Run Amok</title>
```

```
    <link href="http://example.org/2003/12/13/atom03" />
```

```
    <link rel="alternate" type="text/html" href="http://example.or
```

```
    <link rel="edit" href="http://example.org/2003/12/13/atom03/ec
```

```
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
```

```
    <updated>2003-12-13T18:30:02Z</updated>
```

```
    <summary>Some text.</summary>
```

```
    <author>
```

```
      <name>John Doe</name>
```

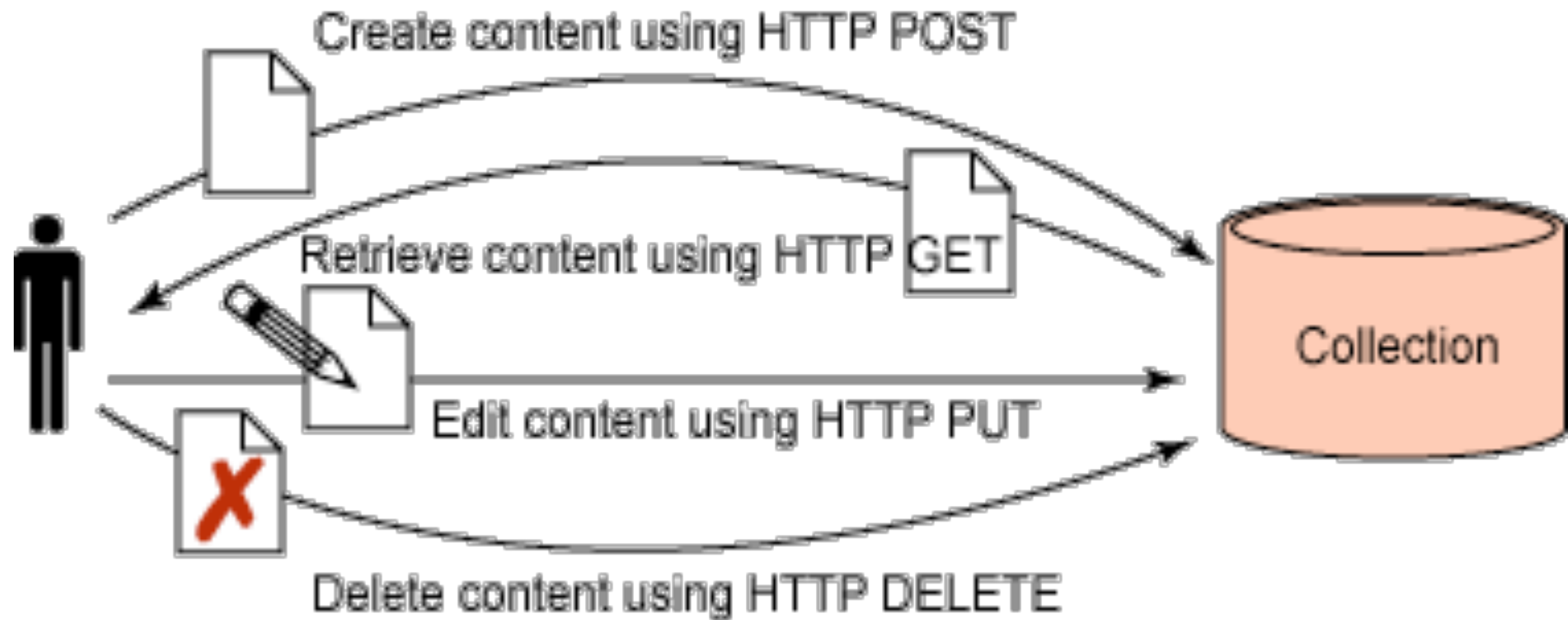
```
      <email>johndoe@example.com</email>
```

```
    </author>
```

```
  </entry>
```

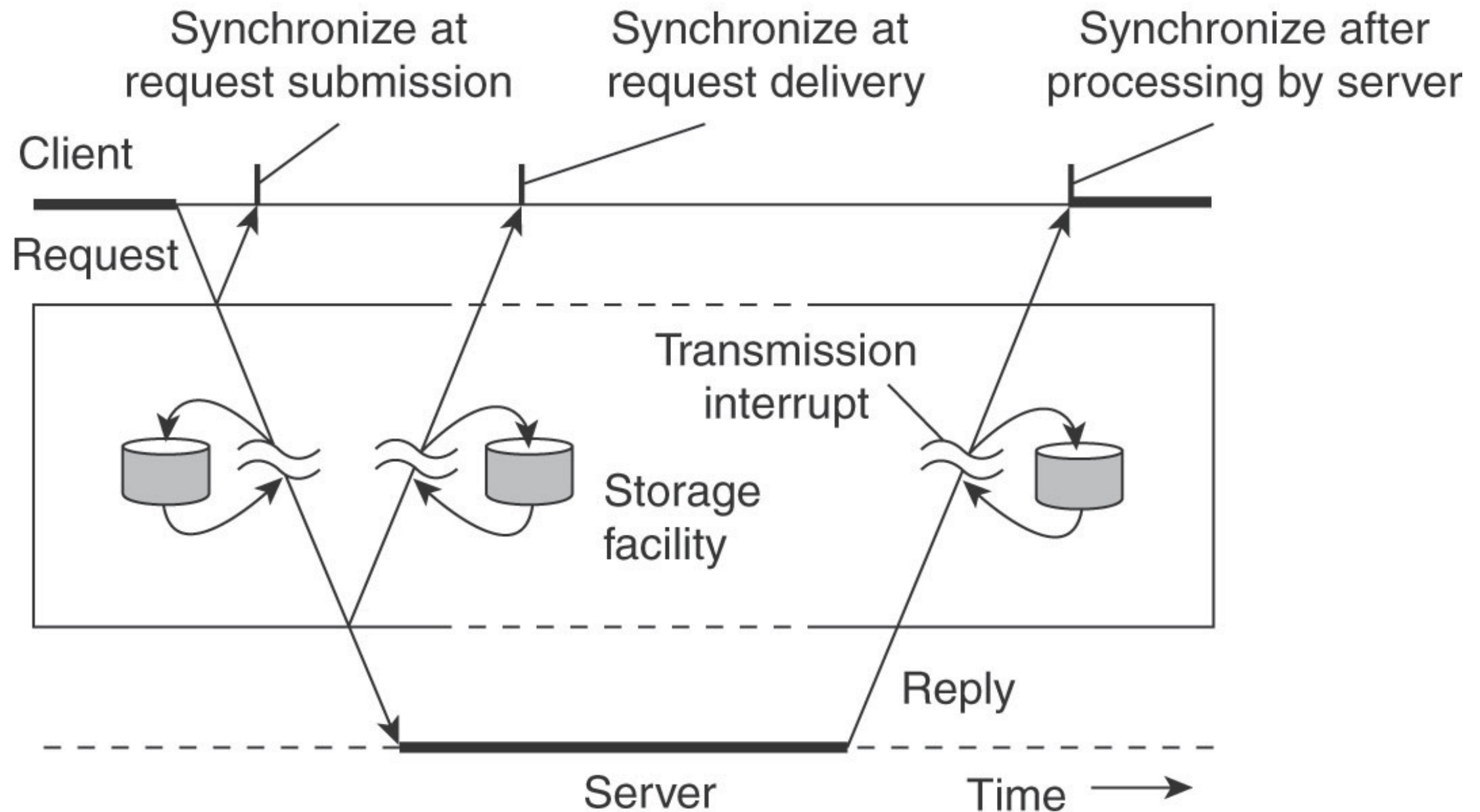
```
</feed>
```


Atom Pub Schnittstelle



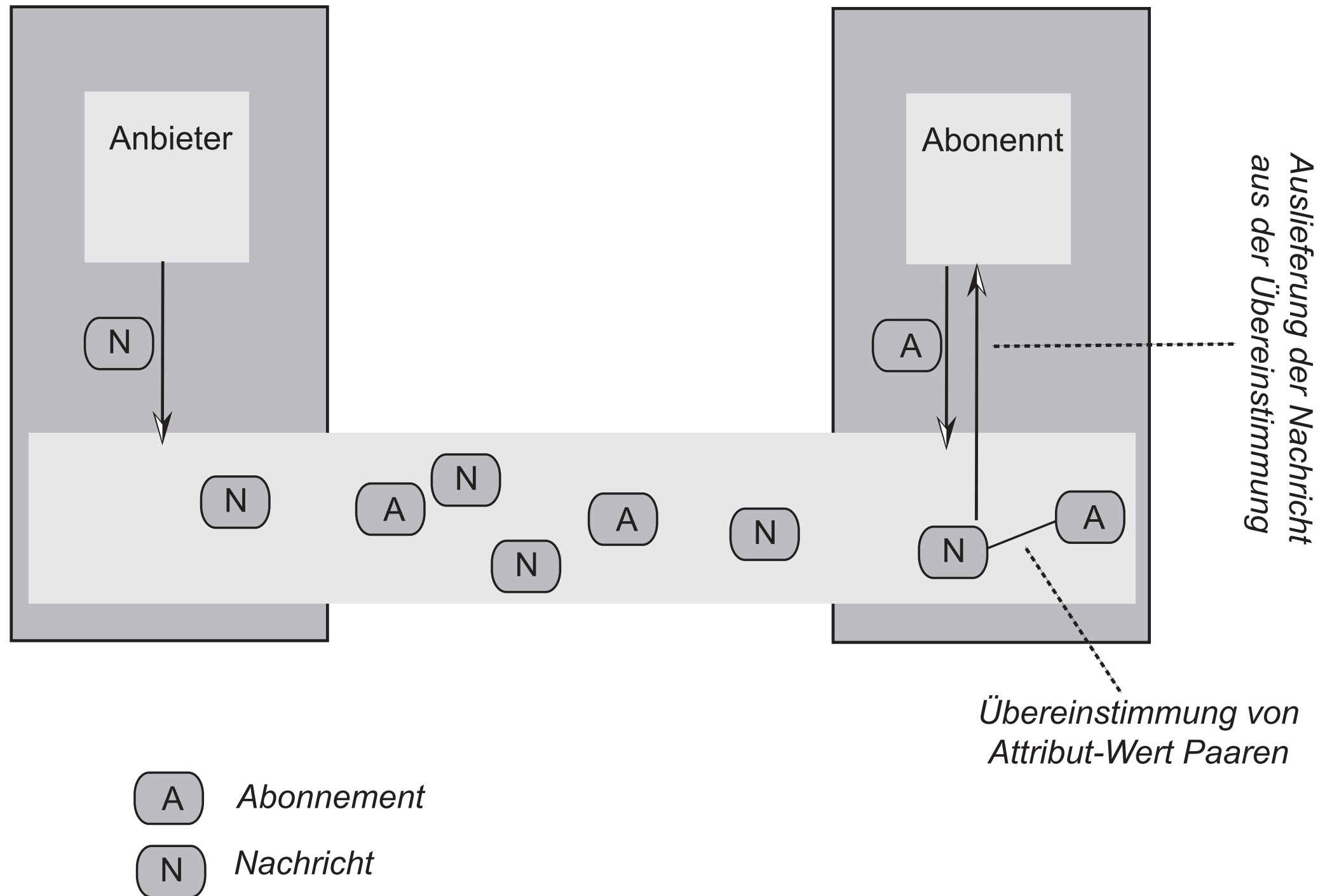
Quelle: <http://www.ibm.com/developerworks/library/x-atompp1/>, letzter Abruf 12.5.2013

Nachrichten orientierte Middleware als zwischenspeichernder Dienst



Quelle: Tanenbaum et al.: Distributed Systems

Referenzielle Entkopplung: Publish - Subscribe



Schnittstelle einer Warteschlange

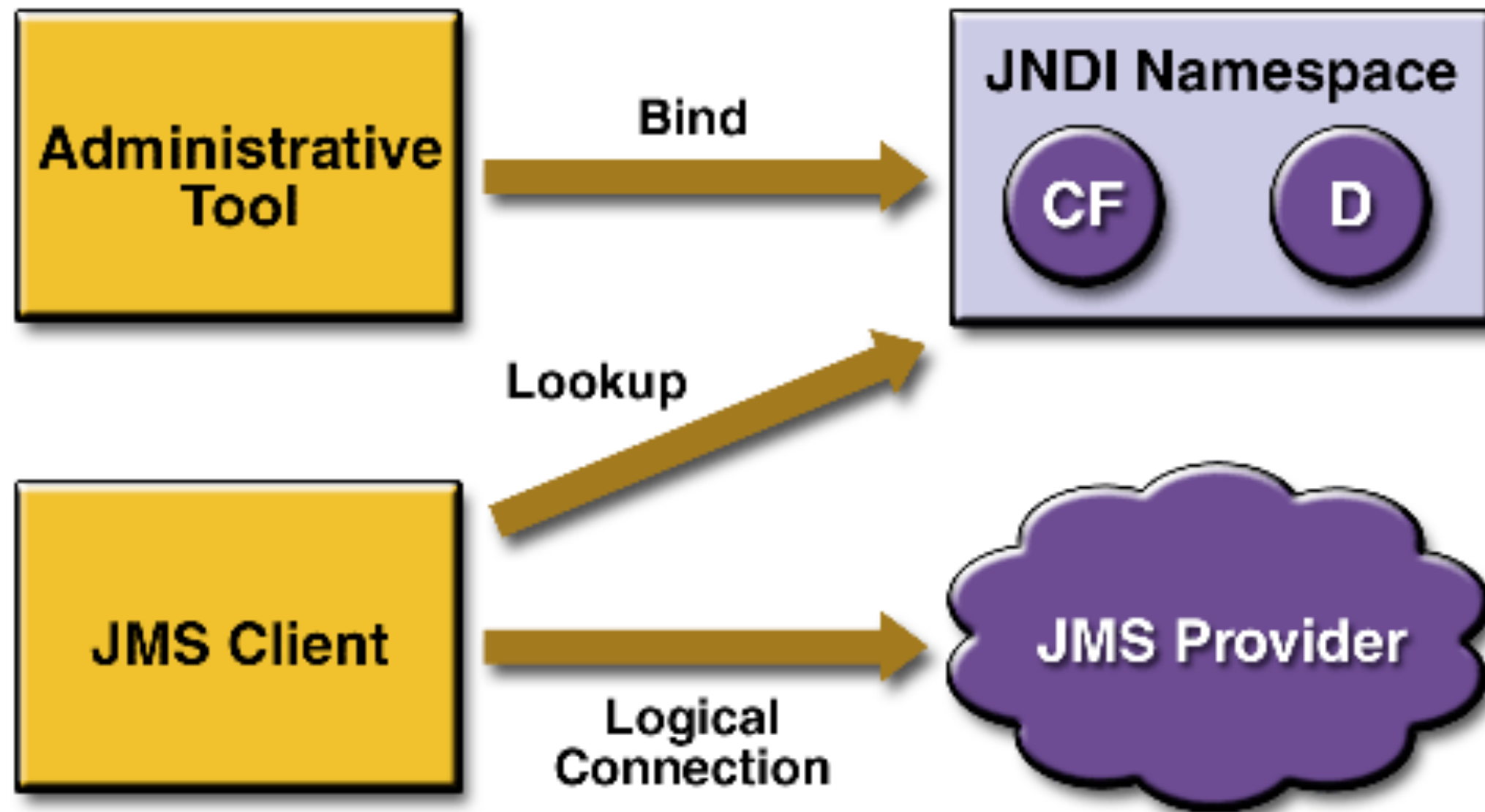
Primitive	Meaning
Put	Append a message to a specified queue
Get	Block until the specified queue is nonempty, and remove the first message
Poll	Check a specified queue for messages, and remove the first. Never block
Notify	Install a handler to be called when a message is put into the specified queue

Quelle: Tanenbaum et al.: Distributed Systems

JMS (1.3) Begriffe

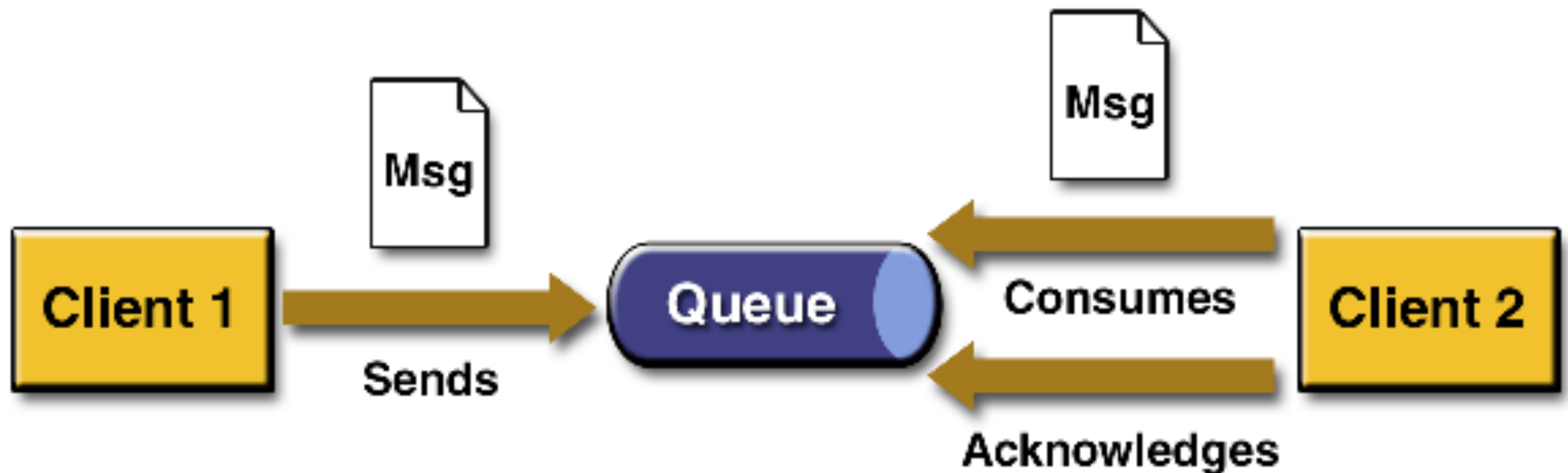
- Ein **JMS provider** ist ein messaging system, das JMS interfaces und administrative Funktionen bietet. Java EE 5 Implementierungen wie Weblogic, Webspere oder JBoss beinhalten einen JMS provider.
- **JMS client** ist ein in Java geschriebenes Programm, das JMS Nachrichten produziert und/oder konsumiert.
- **Administered objects** sind vorkonfigurierte Objekte eines JMS providers, die von eine JMS Client genutzt werden können .
- Die Name/Adresse Zuordnung wird über das Java Naming and Directory Interface (**JNDI**) vorgenommen

JMS (1.3) API Architektur



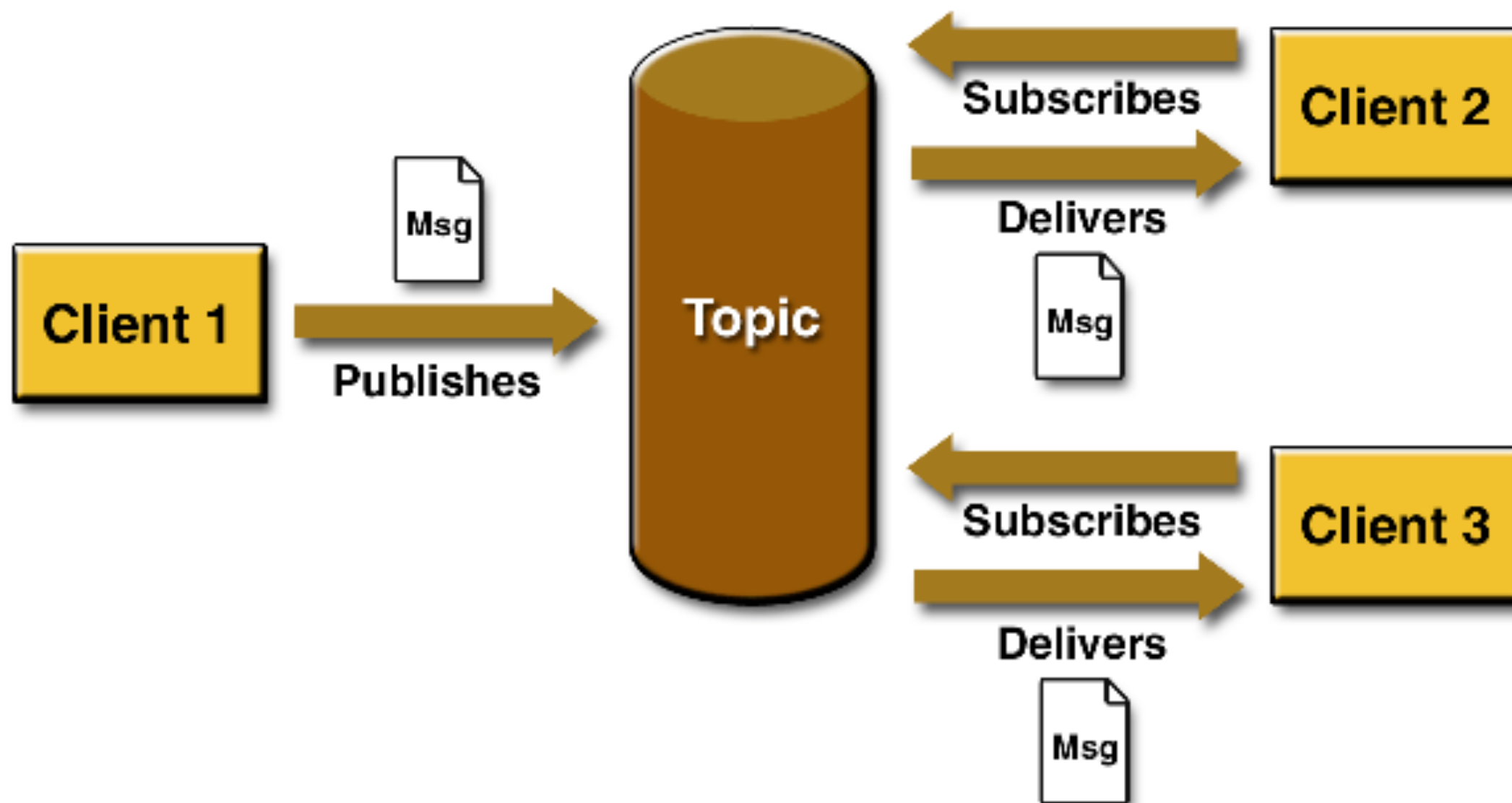
Quelle: [http:// java.sun.com](http://java.sun.com)

Point-to-Point Domäne



Quelle: [http:// java.sun.com](http://java.sun.com)

Publish/Subscribe Domäne



Quelle: [http:// java.sun.com](http://java.sun.com)

Nachrichtentypen in JMS (1.3)

Message Type	Body Contains
<code>TextMessage</code>	A <code>java.lang.String</code> object
<code>MapMessage</code>	name/value pairs, with names as Strings and values as primitive types in Java
<code>BytesMessage</code>	A stream of uninterpreted bytes.
<code>StreamMessage</code>	A stream of primitive values in Java
<code>ObjectMessage</code>	A <code>Serializable</code> object in Java
<code>Message</code>	Nothing. header fields and properties only.

Quelle: [http:// java.sun.com](http://java.sun.com)

Asynchrone Kommunikation Vertiefung - Zusammenfassung

- Asynchronität in Browser basierten Anwendungen auf der Basis von Ajax
- Die asynchrone Kommunikation auf Basis des Atom Standards
- Zeitliche und die referenzielle Entkopplung
- Asynchronen Kommunikation auf der Basis von JMS