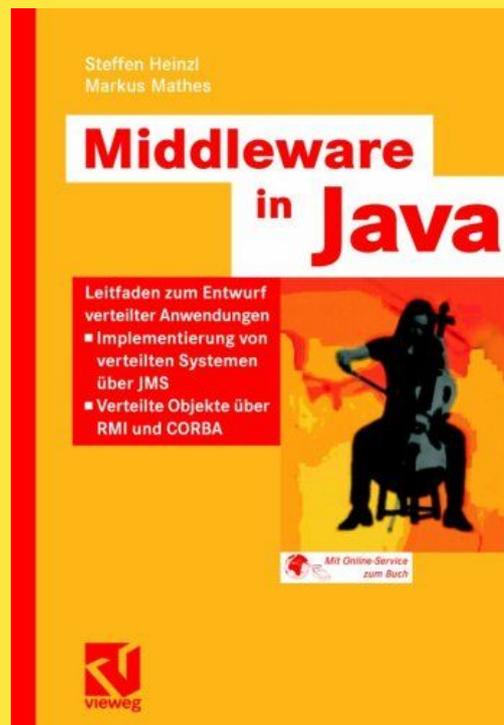


Kapitel 8: Nachrichtenbasierte Kommunikation mit JMS



Middleware und nachrichtenorientierte Middleware

- Eine Software heißt **Middleware** genau dann, wenn sie die Entwicklung und den Betrieb eines verteilten Systems ermöglicht und Funktionen anbietet, die über einfache Netzwerkkommunikation hinausgehen.
- Eine Middleware heißt **nachrichtenorientierte Middleware (MOM)** genau dann, wenn die Kommunikation zwischen den beteiligten Komponenten durch den Austausch von Nachrichten über eine Zwischeninstanz erfolgt.

Vor- und Nachteile einer MOM

Vorteile:

- Kommunikation auf hohem Abstraktionsniveau
- lose Kopplung zwischen einzelnen Komponenten
- Entwickler können sich auf Anwendungslogik konzentrieren

Nachteile:

- nicht „out-of-the-box“ verwendbar
- zusätzlicher Overhead
- ungeeignet für Echtzeitsysteme
- zusätzliche Lizenzkosten

Java Message Service (JMS)

- Der Java Message Service (JMS) ist eine **Schnittstellenspezifikation** von Sun Microsystems zum Zugriff auf nachrichtenorientierte Middleware.
- JMS definiert die **Syntax** und **Semantik** beim Zugriff auf eine nachrichtenorientierte Middleware und wurde erstmals im Jahre 1998 veröffentlicht.
- Die aktuelle Spezifikation stammt aus dem Jahr 2002 und trägt die Versionsnummer 1.1.

Einsatz von JMS

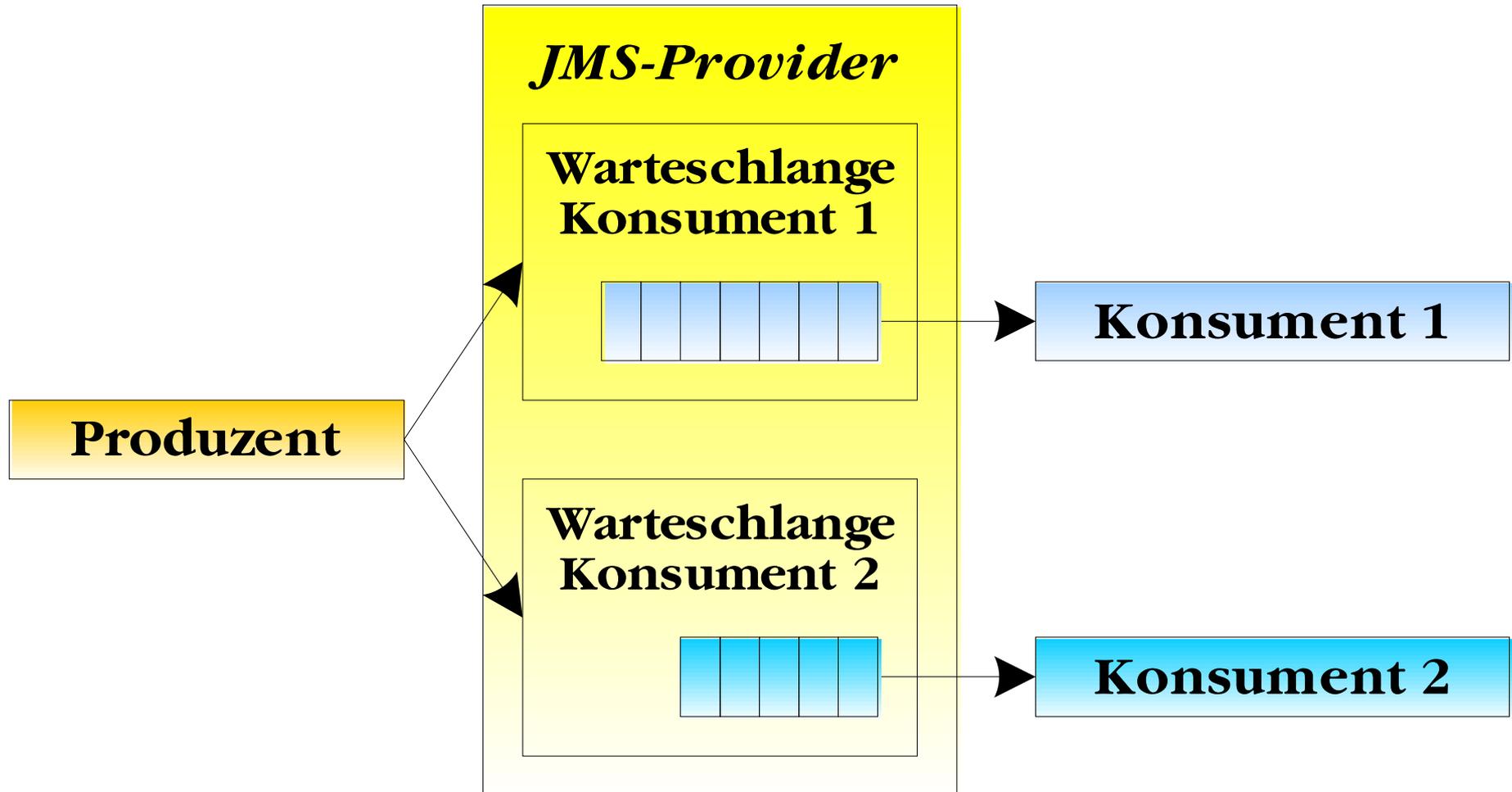
- MOM ohne JMS



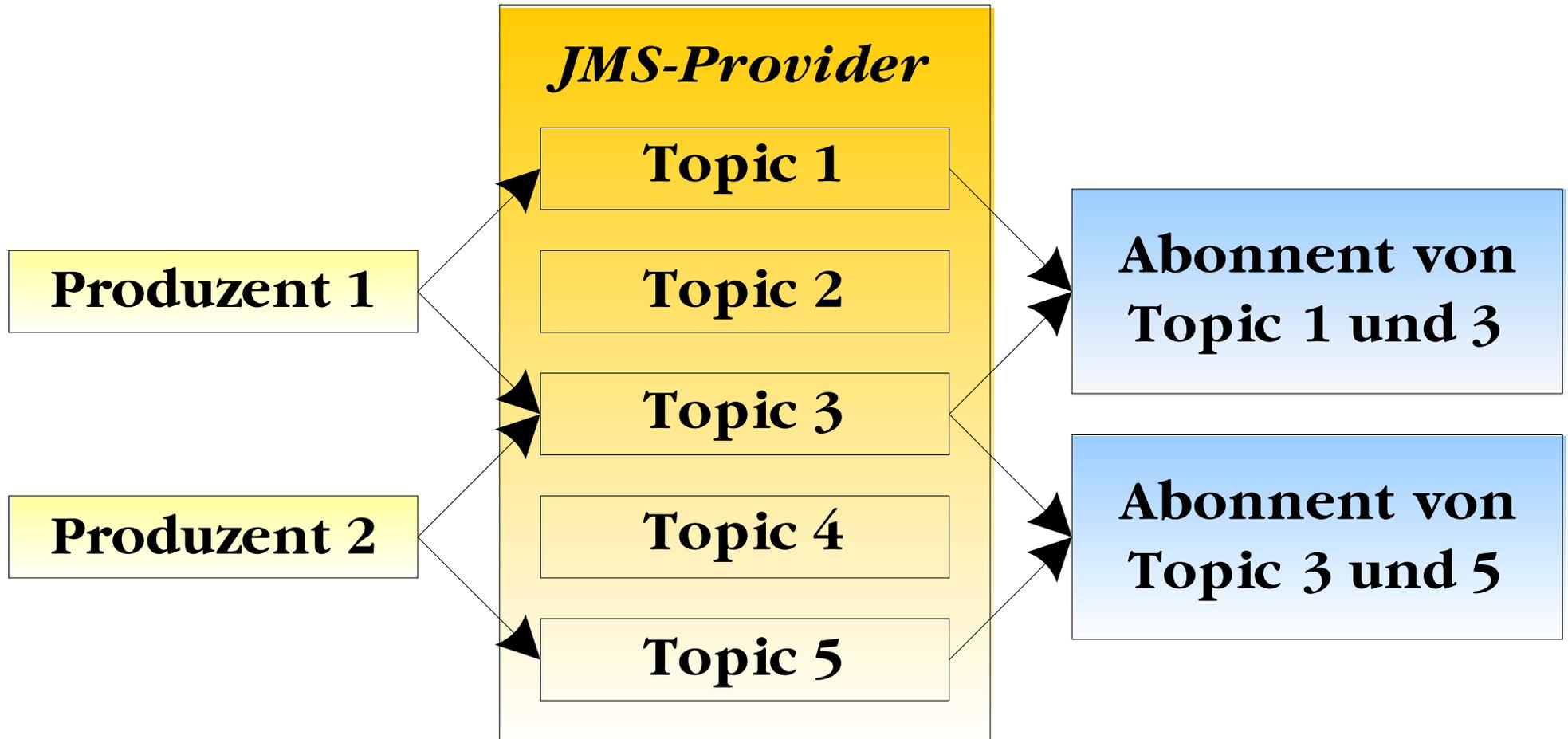
- MOM mit JMS



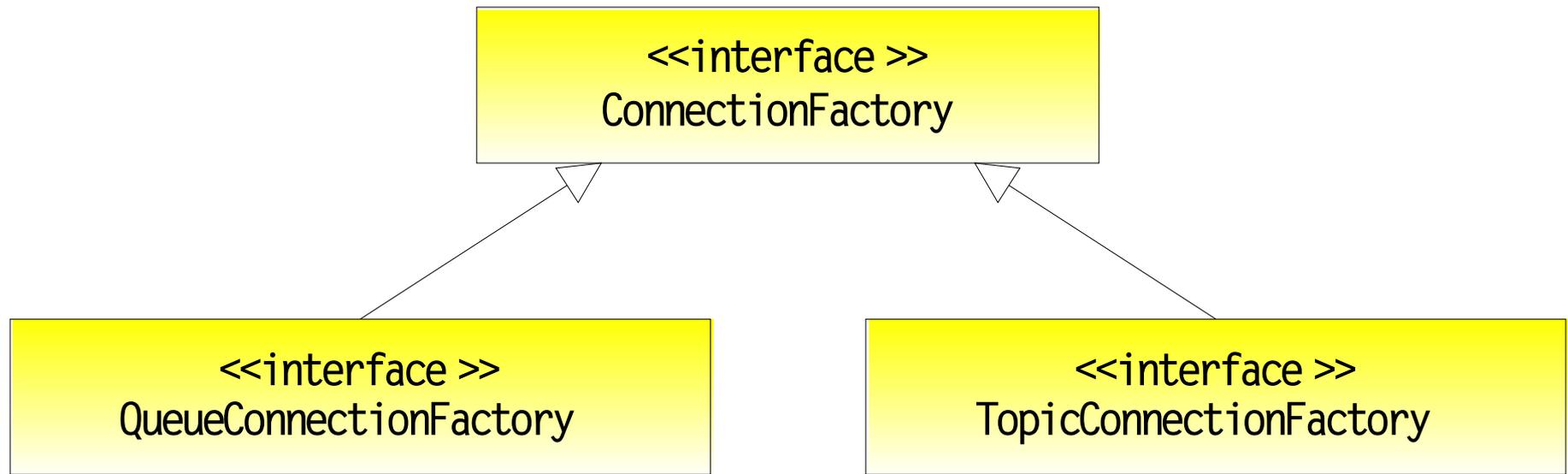
Point-to-Point (P2P)



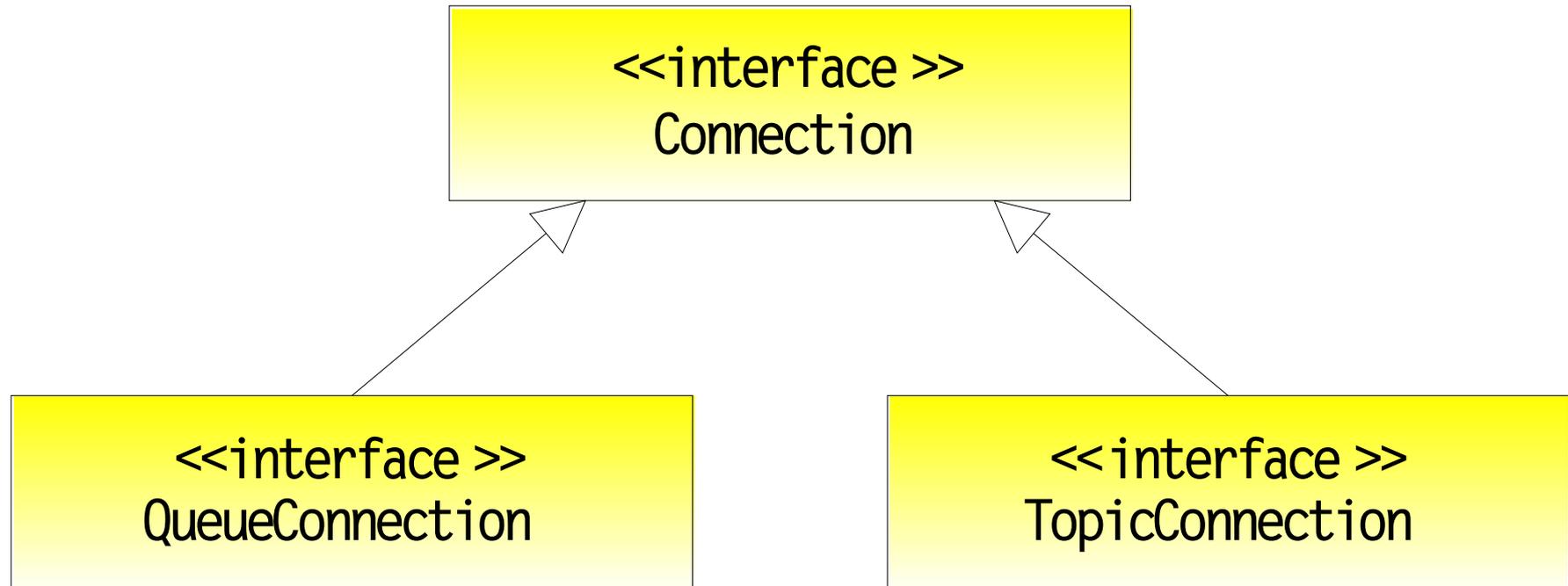
Publish/Subscribe (Pub/Sub)



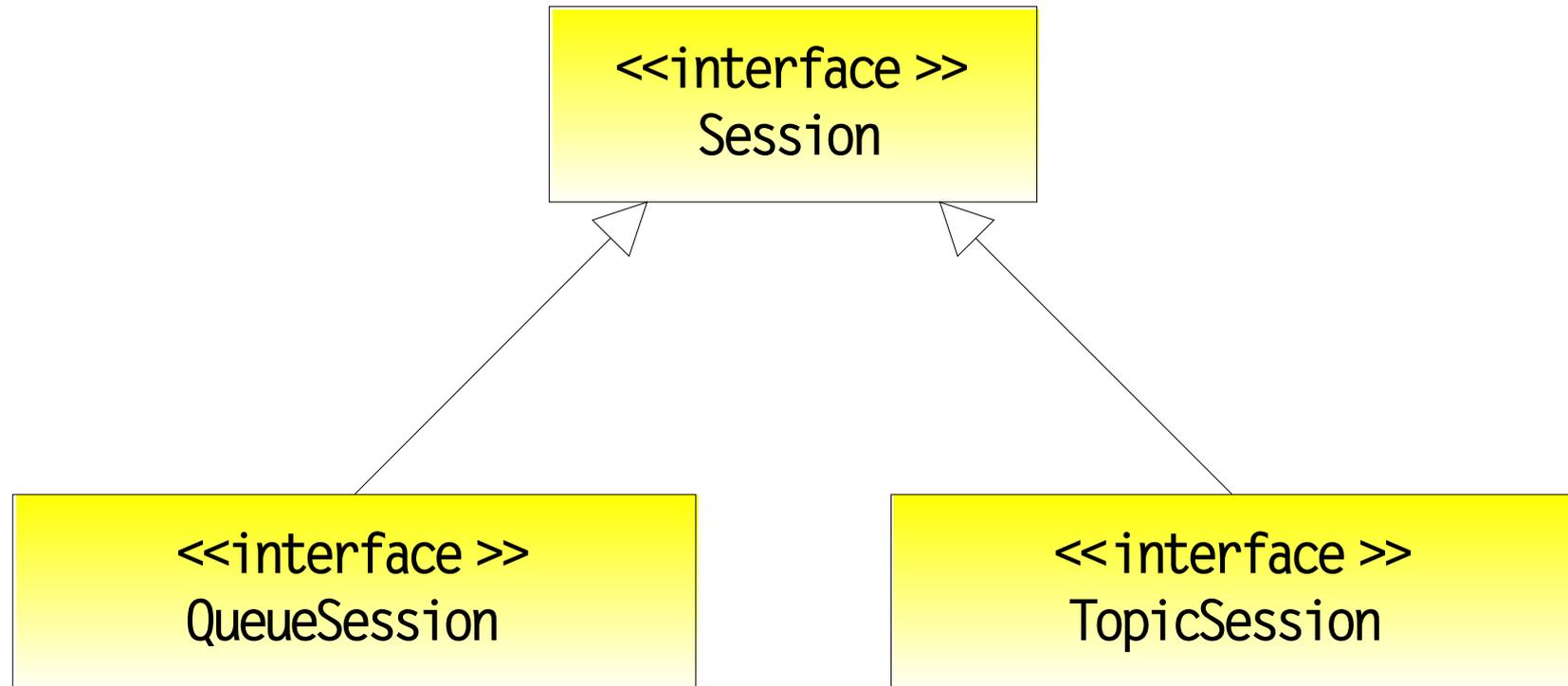
Verbindungsfabriken



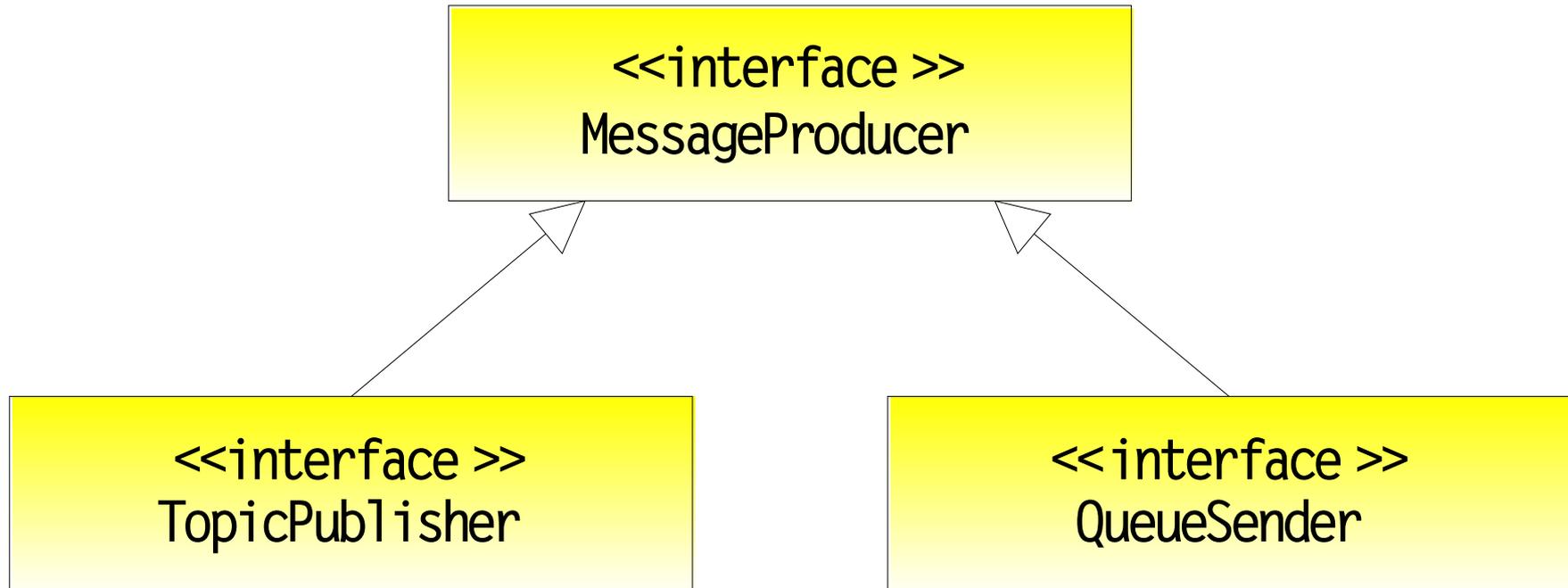
Verbindungen



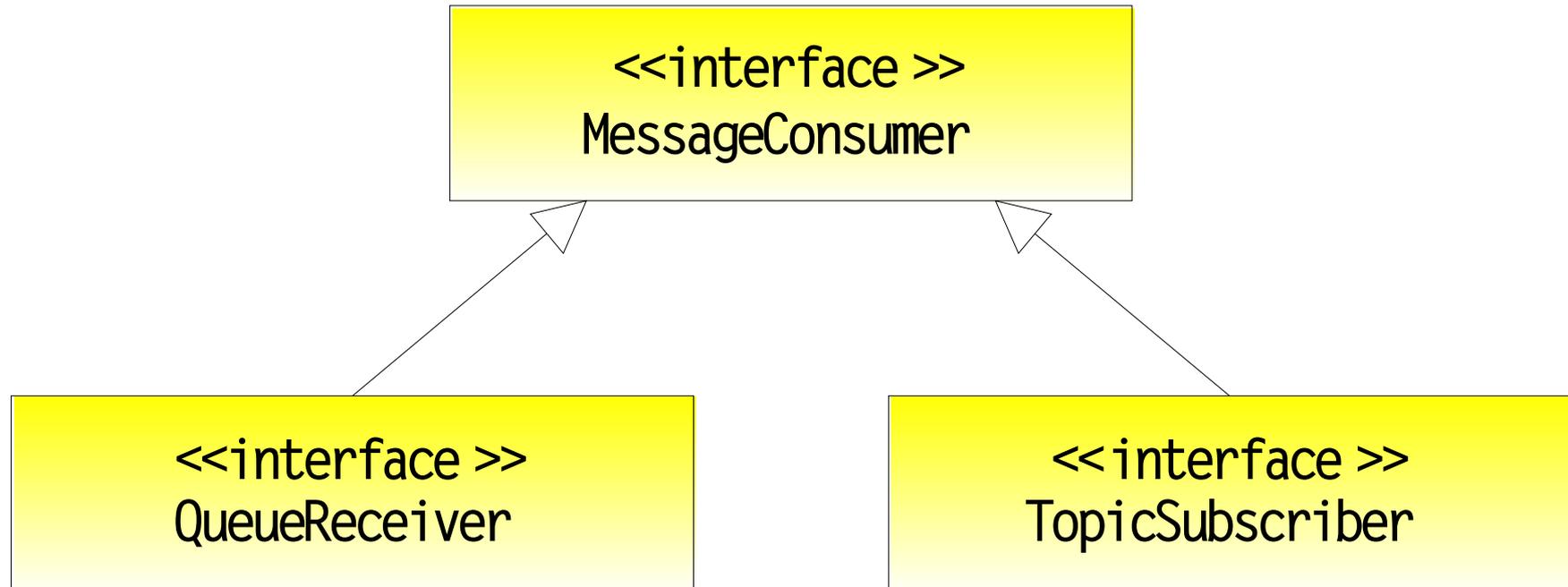
Sitzungen



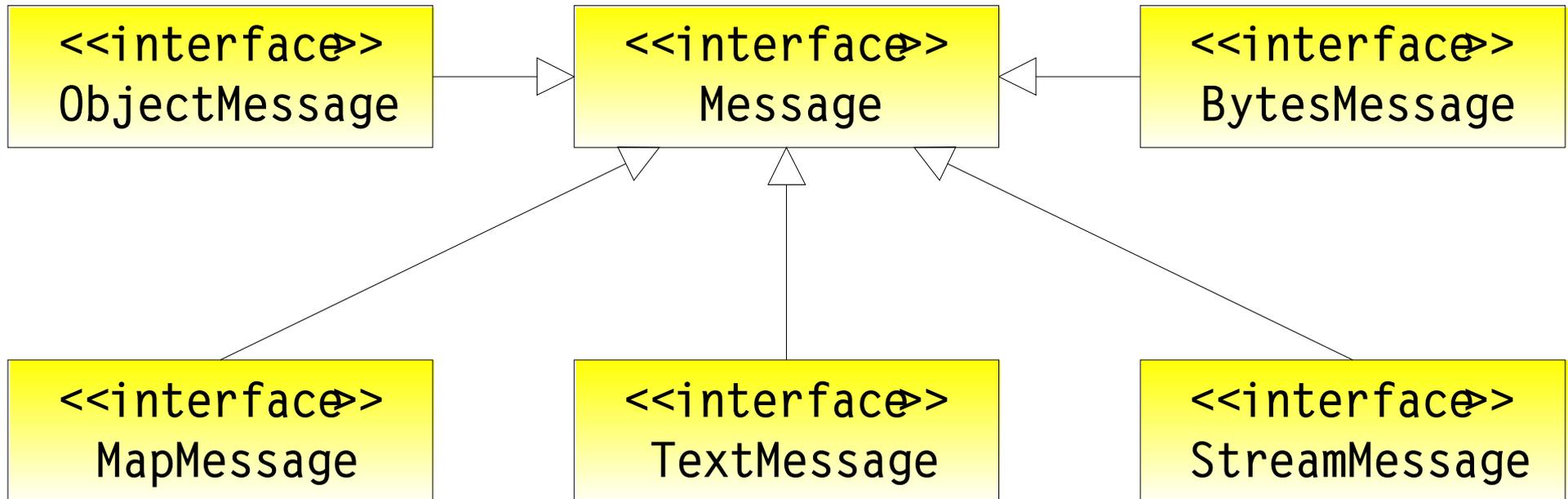
Nachrichtenproduzenten



Nachrichtenkonsumenten



Nachrichten



Aufbau einer JMS-Nachricht

- Nachrichtenkopf (***Header***)
 - Der Nachrichtenkopf wird zur Identifikation und zur Übertragung der Nachricht benötigt und umfasst mehrere Felder.
- Nachrichteneigenschaften (***Properties***)
 - Nachrichteneigenschaften erlauben das Speichern von Zusatzinformationen zur Nachricht.
- Nachrichtentrumpf (***Body***)
 - Der Nachrichtentrumpf enthält die Nutzdaten.

Nachrichtentypen

- `BytesMessage`
 - Übertragung einer Menge uninterpretierter Bytes
- `StreamMessage`
 - Übertragung sequentiell gelesen und geschriebener Daten
- `MapMessage`
 - Übertragung von Name/Wert-Paaren
- `TextMessage`
 - Übertragung von Textdaten
- `ObjectMessage`
 - Übertragung serialisierbarer Objekte

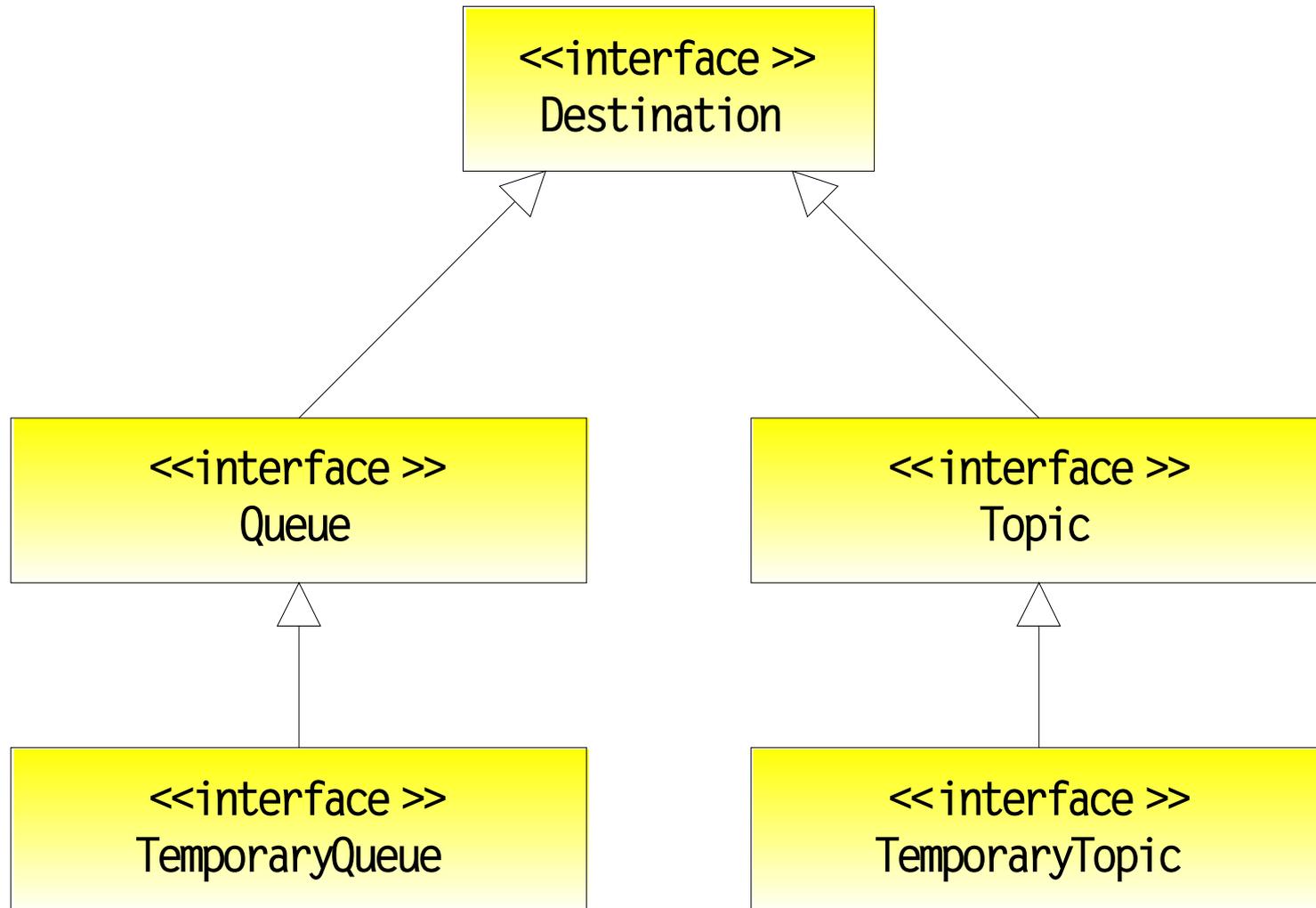
Header-Felder einer Nachricht (1)

- JMSCorrelationID
 - **Abhängigkeiten zwischen Nachrichten**
- JMSPersistenceMode
 - PERSISTENT
 - NON_PERSISTENT
- JMSDestination
 - **Nachrichtenziel**
- JMSExpiration
 - **Verfallsdatum der Nachricht**
- JMSMessageID
 - **eindeutiger Bezeichner**

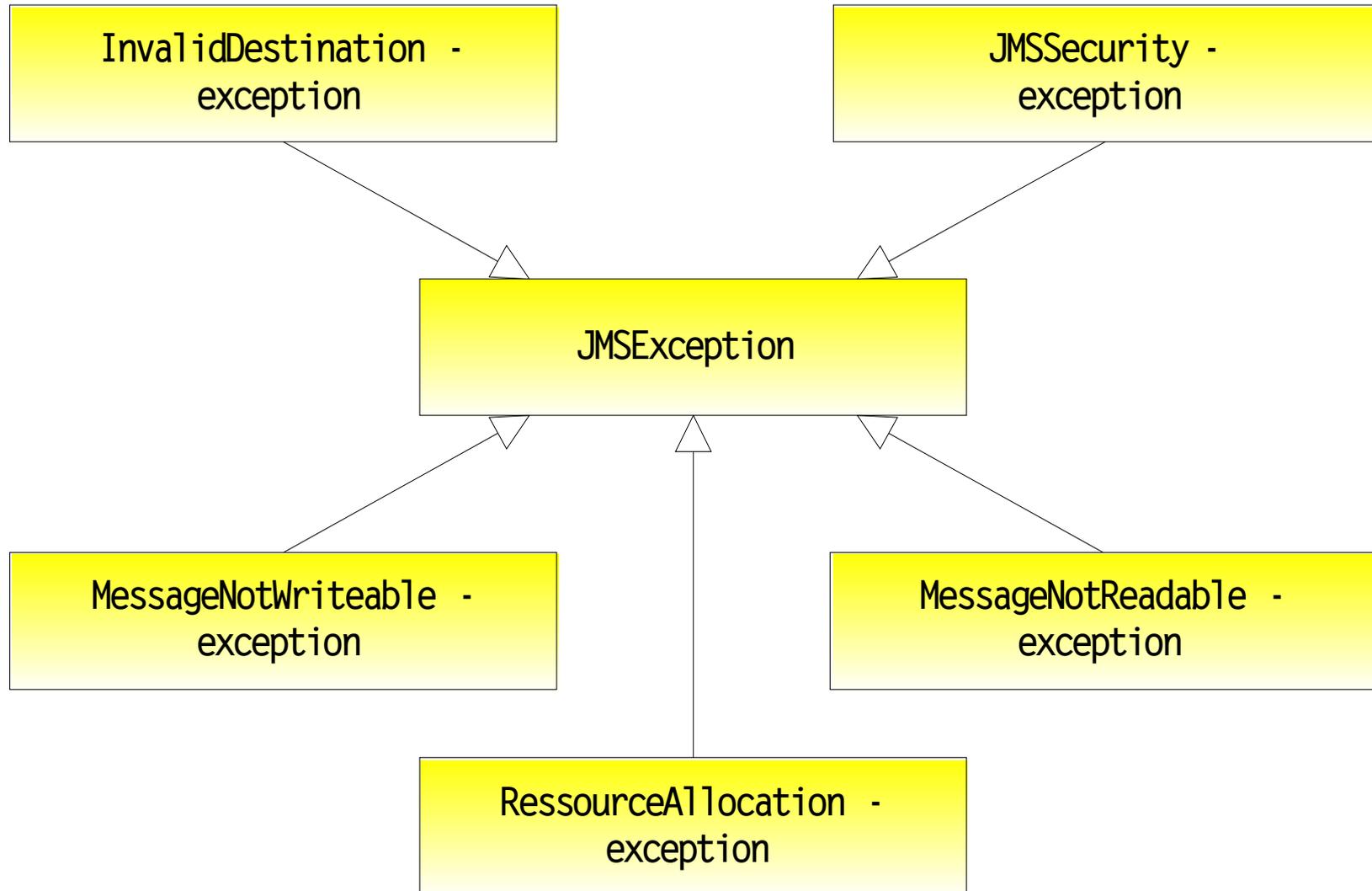
Header-Felder einer Nachricht (2)

- JMSPriority
 - **Priorität von 0 (niedrig) bis 10 (hoch)**
- JMSRedelivered
 - **Nachricht wurde mehrfach ausgeliefert**
- JMSReplyTo
 - **Ziel der Antwortnachricht festlegen**
- JMSTimestamp
 - **Übergabezeit der Nachricht an den JMS-Provider**
- JMSType
 - **Nachrichtentyp der Nachricht**

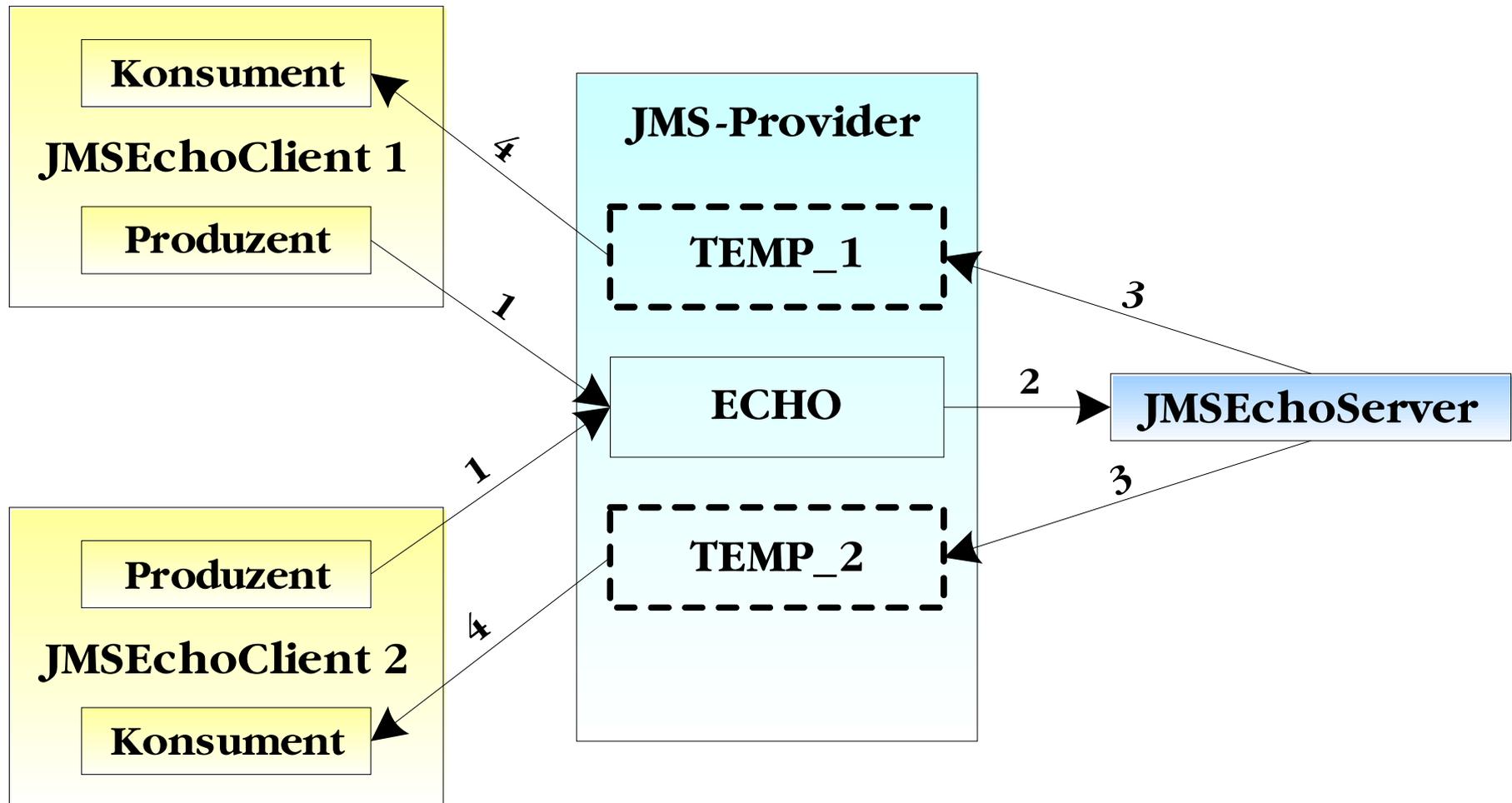
Nachrichtenziele



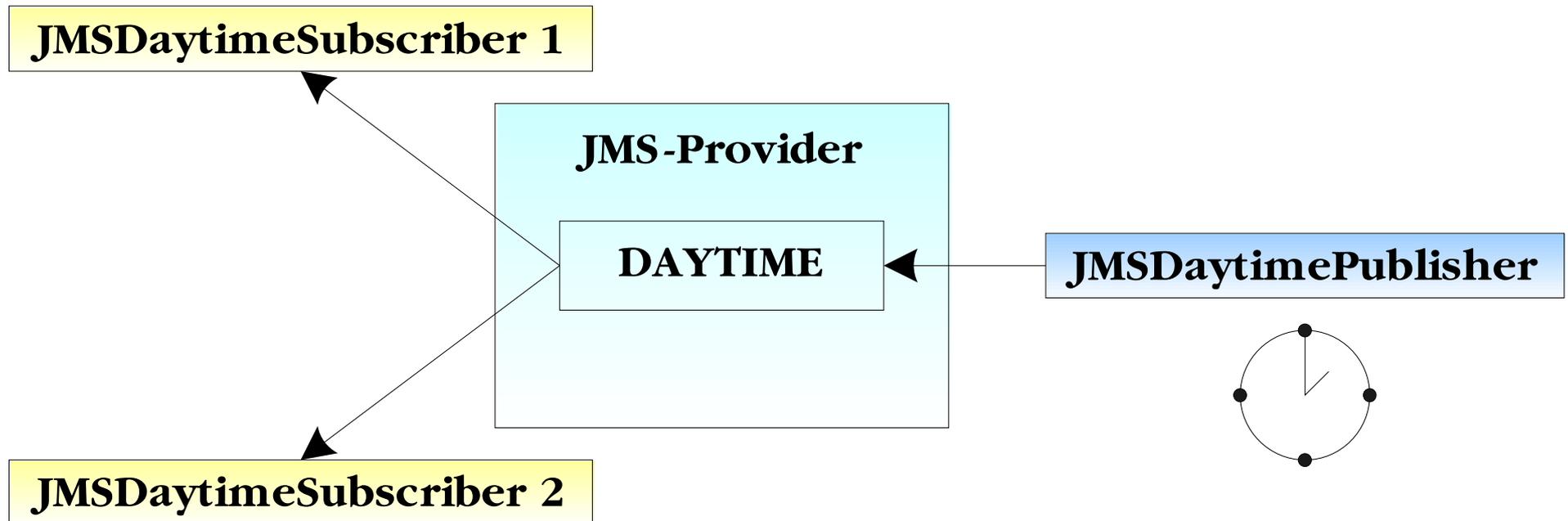
Ausnahmen in JMS



ECHO-Dienst mit P2P



DAYTIME-Dienst mit Pub/Sub



Literatur

- Ressourcen von Sun Microsystems Inc. zum Java Message Service (JMS); <http://java.sun.com/products/jms>
- Sun Microsystems Inc.: *Java Message Service Specification*; <http://java.sun.com/products/jms/docs.html>
- Sun Microsystems Inc.: *Java Message Service API Tutorial*; <http://java.sun.com/products/jms/tutorial/index.html>
- Richard Monson-Haefel, David A. Chappell: *Java Message Service – Creating Distributed Enterprise Applications*; O'Reilly 2001
- Levent Erdogan: *Java Message Service (JMS) for J2EE*; New Riders Publishing 2002
- Markus Mathes: *Zeitverhalten von nachrichtenorientierter Middleware und deren Einsatz in der Industrieautomation*; Diplomarbeit Fachhochschule Fulda 2004
- Florian Heidinger, Markus Mathes, Helmut Dohmann: *Java Messaging Service (JMS) – Einsatz in der Industrieautomation*; Automatisierungstechnische Praxis (atp) Ausgabe 05/2004

Aufgaben

In „*Middleware in Java*“
finden Sie

- Wiederholungs-,
- Vertiefungs-,
- Programmieraufgaben
zu den vorgestellten
Themen.

Zur Festigung und
Vertiefung des Erlernten
wird eine Bearbeitung der
Aufgaben empfohlen.

