

Message Oriented Middleware am Beispiel von XMLBlaster

Vortrag im Seminar
XML und intelligente Systeme
an der Universität Bielefeld
WS 2005/2006

Vortragender:
Frederic Siepmann
fsiepman@techfak.uni-bielefeld.de

Outline:

Middleware

- *Weg zur Middleware*
- *Funktionsweise*
- *Kommunikation*

Message Oriented Middleware (MOM)

- *Funktionsweise & Kommunikation*
- *Elementare Funktionen*
- *Vor- und Nachteile*

XMLBlaster

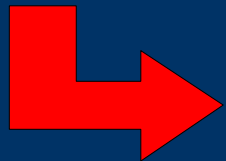
- *Was ist das ?*
- *Aufbau der Nachrichten*
- *Funktionsweise*
- *Anwendungsbereiche*

Diskussion

Long walk to Middleware...

Weiterentwicklung der Mikroprozessoren/Netzwerktechnik führt zu “verteilten Systemen”

“Ein verteiltes System ist eine Menge voneinander unabhängiger Computer, die dem Benutzer wie ein einzelnes kohärentes System erscheinen.”



für den Anwender ist die Verteilung des Systems nicht mehr sichtbar

Wesentliche Untergruppen:

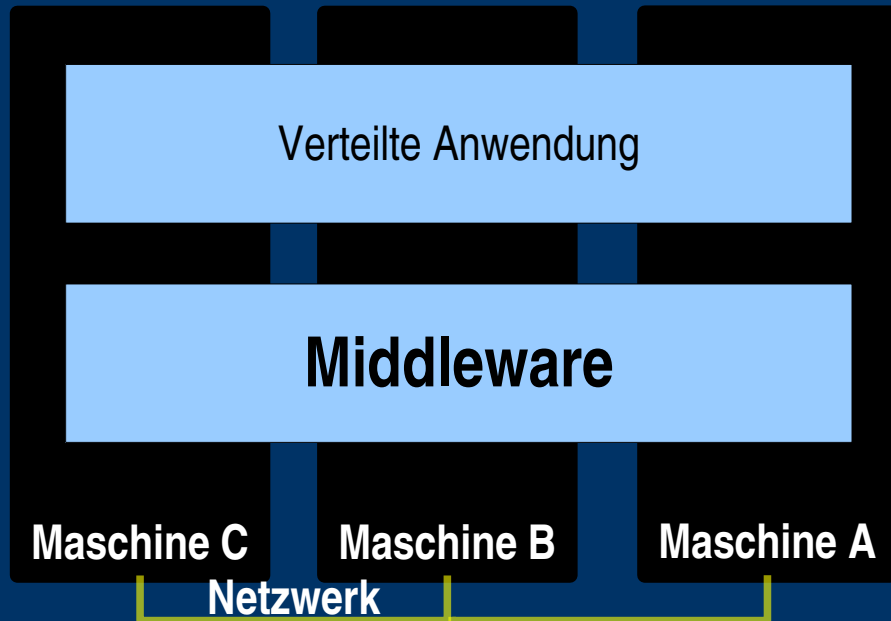
kommunikationsorientierte Middleware:

CORBA → Abstraktion von der Netzwerkprogrammierung

anwendungsorientierte Middleware:

XMLBlaster → Kommunikation + Unterstützung verteilter Anwendungen

Middleware einsortiert



und seine Ziele

Einfache Verbindung zwischen
Benutzer/Ressourcen

Transparenz

Offenheit

Skalierbarkeit

Kommunikation & Middleware

- Anforderungen und Anwendungsbereiche bzw.
- “Art” der Kommunikation

bestimmen den Typ der Middleware:

Transaction Processing ... Database Middleware ... RPC Middleware ... Objekt Middleware ... **Message Oriented Middleware (MOM)**

Middleware setzt auf der “Transportschicht” auf, da die grundlegenden Funktionen hier oftmals nicht ausreichen.

Anwendung
Middleware
Transport
Netzwerk
Sicherheit
Physisch

Anwendungsprotokoll (FTP, HTTP)
Middleware-Produkt
Transportprotokoll (TCP)
Netzwerkprotokoll (IP)
Sicherungsprotokoll
Bitübertragungsprotokoll (RS-232-C)

Was ist Message Oriented Middleware (MOM)?

- Synchronität von “Sender” und “Empfänger” wird aufgehoben durch Nachrichten (Parameter für Funktionsaufruf)
VS. Sender und Empfänger gleichzeitig “online”
- Die Kommunikation ist dabei persistent (asynchron)
VS. Daten werden direkt übertragen und nicht vom Server gespeichert
- Schnittstellenstandardisierung erlaubt Zugriff über einfache Menge von Elementarfunktionen
äquivalent
- Garantie der Zustellung der Nachricht, nicht über Zeitpunkt der Zustellung bzw. Lesen der Nachricht
VS. Direkte Zustellung und Zeitpunkt garantiert

MOM's elementare Funktionen

Elementare Funktion

Bedeutung

Put	Eine Nachricht an angegebene Warteschlange anhängen
Get	Blockieren, bis die angegebene Warteschlange nicht leer ist, und die erste Nachricht daraus entfernt
Poll	Eine angegebene Warteschlange auf Nachrichten überprüfen und die erste Nachricht daraus entfernen; kein Blockieren
Notify	Eine Verarbeitungsroutine installieren, die aufgerufen wird, wenn eine Nachricht in die angegebene Warteschlange gestellt wird

MOM's Vor- und Nachteile

- speichern/"routing"/transformieren von Nachrichten
- parallele Verarbeitung von Nachrichten möglich
- höhere Flexibilität bei Änderungen am System (Funktionen o.ä.)
- Es werden zusätzlich Komponenten bzw. Infrastruktur benötigt
- Kommunikation zwischen Applikationen ist oft SYNCHRON
- Viele unterschiedliche Standards
- Ausfall des MOM = "Totalausfall"

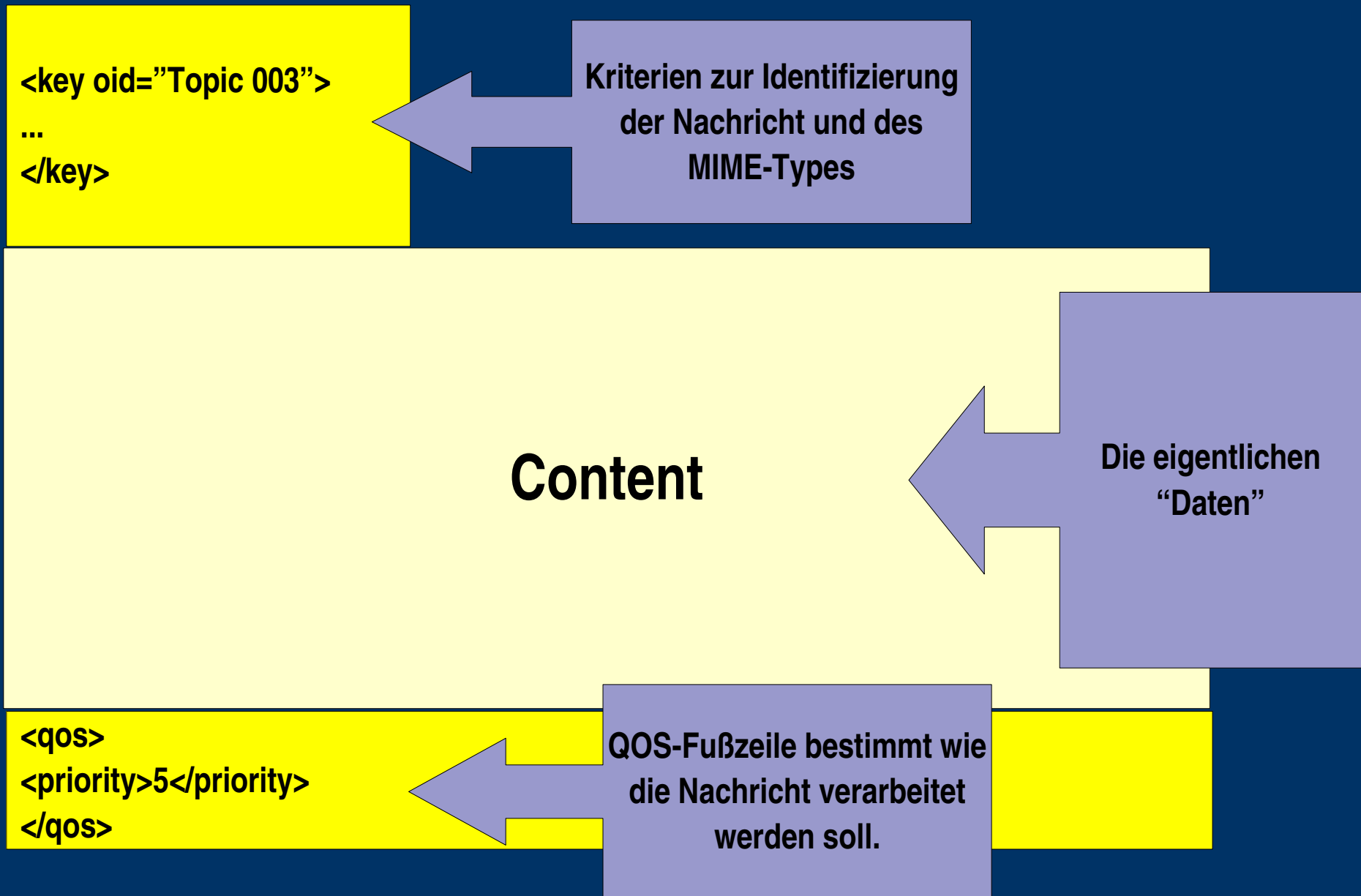
Was ist XMLBlaster ?



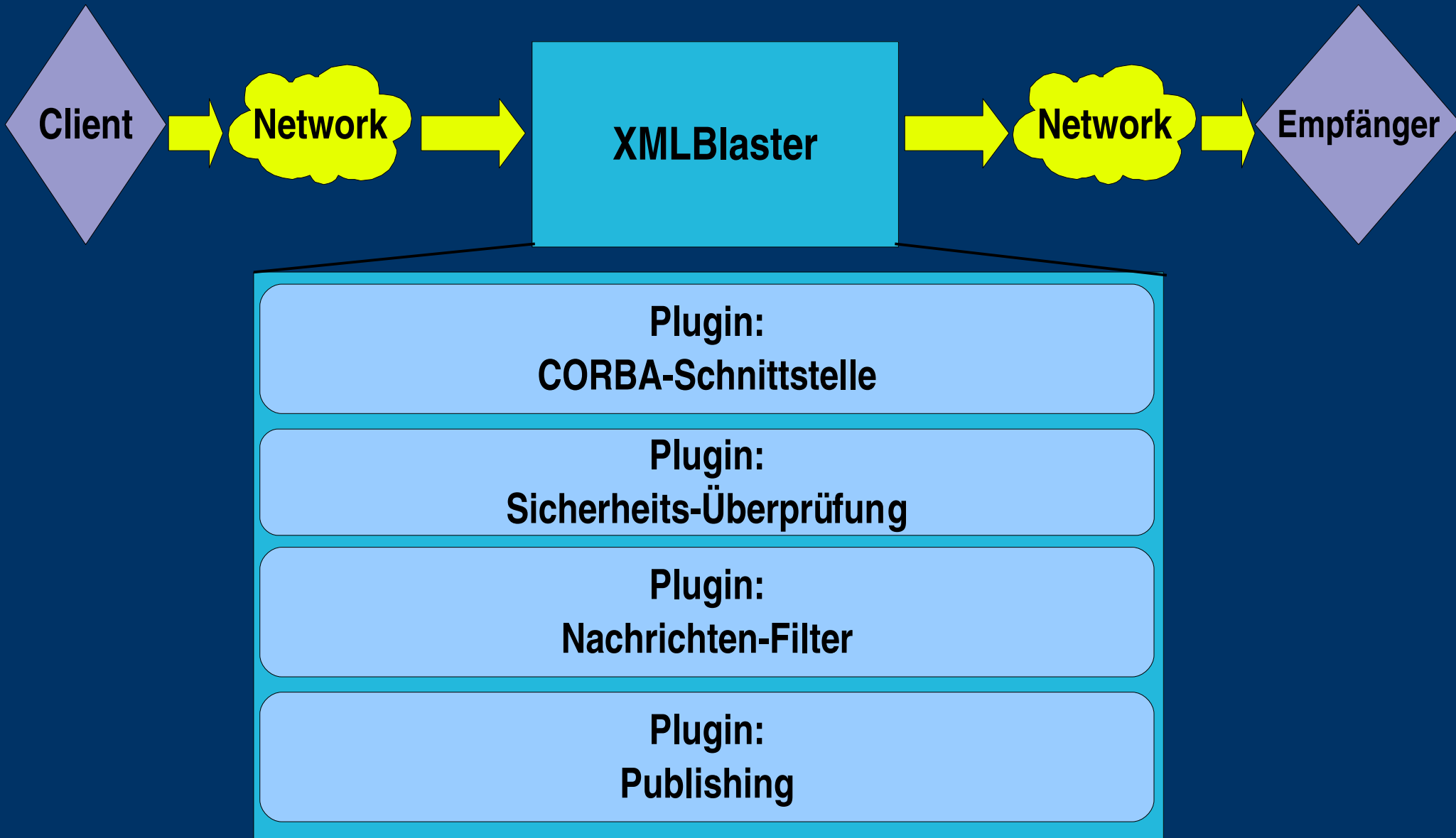
Auf Java[®] basierender MOM-Server

- Nachrichten sind XML-Dokumente
- benötigte Warteschlangen werden bei Bedarf “on the fly” erstellt
- Clientseitige Warteschlangen möglich
- asynchrone / synchrone Kommunikation
- Plugins für Funktionalität/Schnittstellen

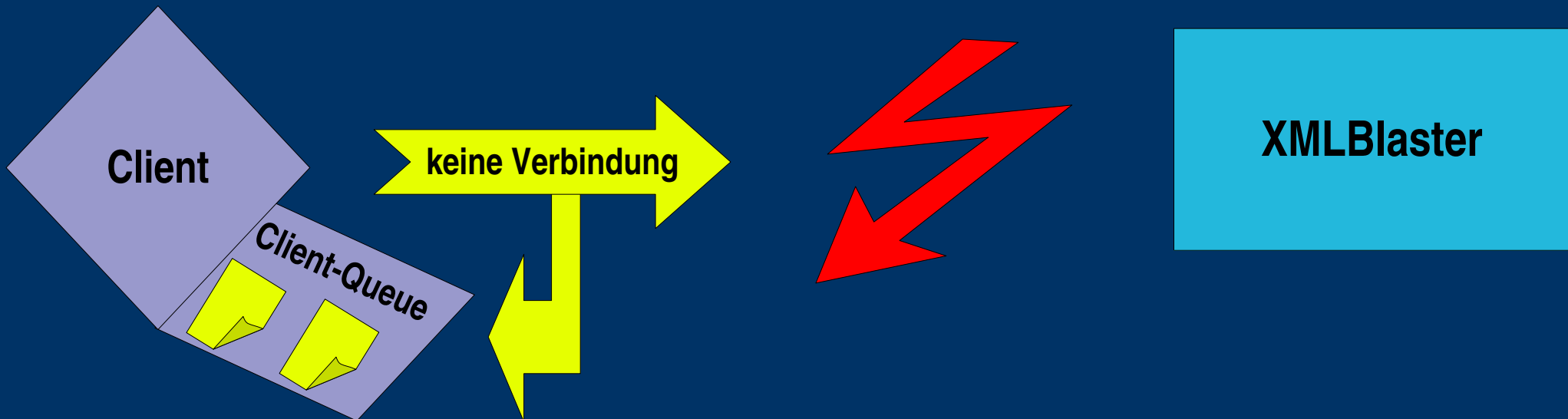
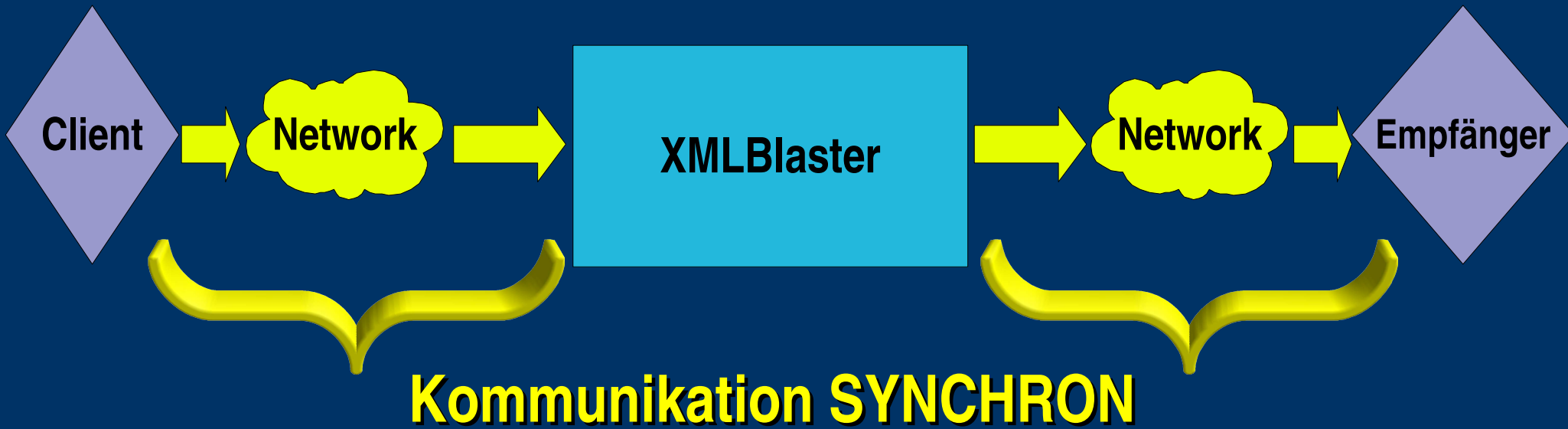
Wie sind Messages im XMLBlaster aufgebaut ?



Funktionsweise des XMLBlaster



Client-Queues beim XMLBlaster



Wie funktioniert der XMLBlaster ?

Funktionen:

publish

get

subscribe

callback

Entspricht etwa

PUT

GET

POLL

NOTIFY

Nachrichten durchlaufen verschiedene “Plugins”, die verschiedene Funktionalität anbieten:

- Anbieten von Schnittstellen
- Sicherheit / Zugangsberechtigung
- Filtern

Anwendungsgebiete und Anwendungsfälle für den XMLBlaster

- **Forstware – GeoMail**

XMLBlaster wird zur Kommunikation zwischen Applikation/Business benutzt.

- **doubleSlash – integrative measure platform for T-Com:**

XMLBlaster versorgt den HelpDesk mit Informationen verschiedener subsystems

- **SES ASTRA – Digital Automatic Analysing System (DAAS):**

6 Teilsysteme zur Kontrolle der Signalqualität (ASTRA 1,2 und 3), die via XMLBlaster kommunizieren.

Vielen Dank für die Aufmerksamkeit !

Quellen:

- www.xmlblaster.org
- www.wikipedia.org
- A. Tannenbaum, M. van Steen,
“Verteilte Systeme - Grundlagen
und Paradigmen”,
Pearson Studium, München 2003



Kontakt:

fsiepman@techfak.uni-bielefeld.de

Diskussion:

?!/?!?!?

- ▶ In welchen Situationen ist Middleware/MOM notwendig bzw. sinnvoll ?
- ▶ Warum XML ?
- ▶ Wie kann man die Performance solcher Systeme verbessern ?