BEEP

The Blocks Extensible Exchange Protocol Core

KM-/VS-Seminar

Wintersemester 2002/2003

Betreuer: Frank Strauß

TECHNISCHE UNIVERSITÄT
CAROLO-WILHELMINA
ZU BRAUNSCHWEIG
Institut für Betriebssysteme und Rechnerverbund
http://www.tu-bs.de http://www.ibr.cs.tu-bs.de

Übersicht

- Einleitung
- Konzepte von BEEP
  - Nachrichtaustausch
  - Frames
  - Sitzungsverwaltung
  - Abbildung auf TCP
- Implementierungsbeispiel: Java BEEP
- Zusammenfassung

Einleitung

- BEEP – Ein Framework für Anwendungsprotokolle
  - Probleme bei Entwürfen neuer Internetanwendungen
  - RFC 3080 und RFC 3081, IETF

- Zielsetzung von BEEP
  - Verbindungsorientierung,
  - Nachrichtenorientierung und
  - Asynchronität.

Konzepte von BEEP

- BEEP implementiert Peer-to-Peer-Prinzip.
  - Peer-to-Peer: bilaterale Kommunikation zwischen zwei oder mehr gleichberechtigten Endpunkten.

- Funktionalitäten von BEEP:
  - Trennen einer Nachricht von der Nächsten (Framing),
  - Kodierung der Nachrichten,
  - Unterstützung mehrfacher logischer Kommunikationskanäle,
  - Fehlerbehandlung,
  - Aushandlung von Verschlüsselungsparametern,
  - Aushandlung von Authentisierungsparametern.

Nachrichtaustausch

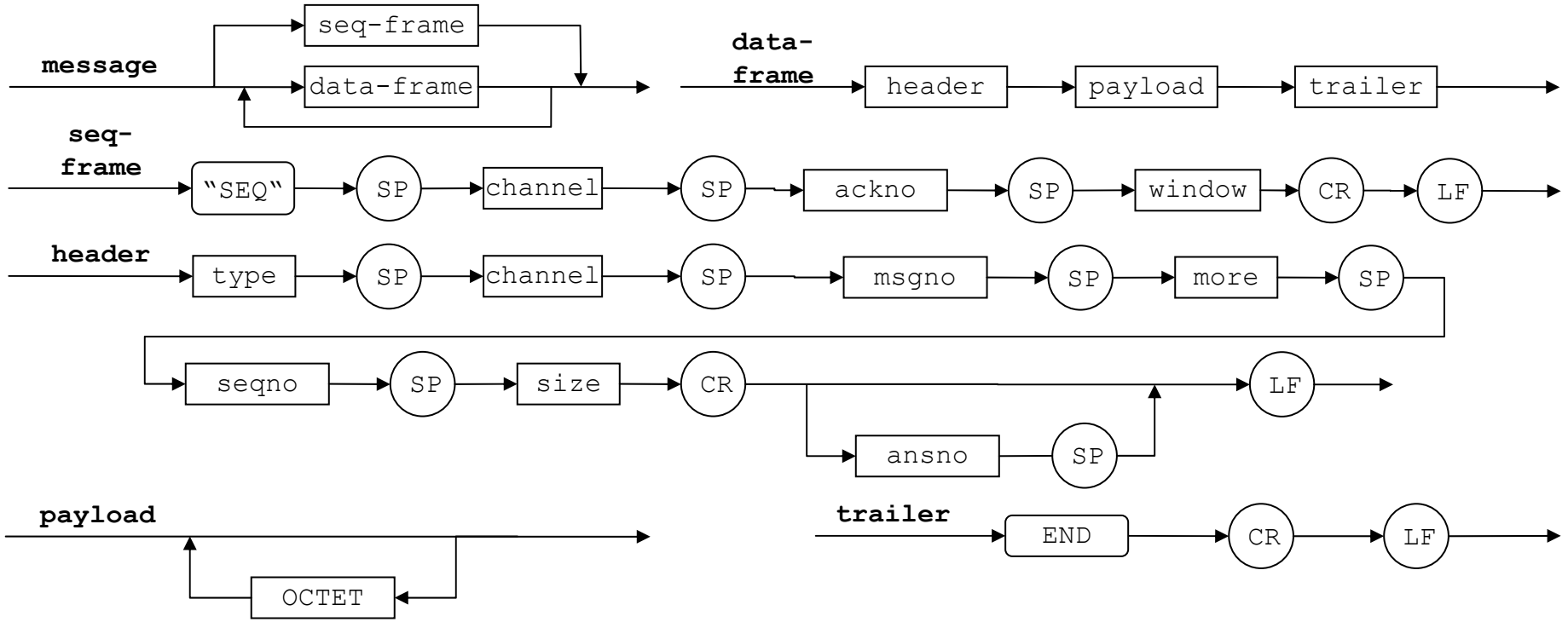
- Rollen
  - Listener – wartet auf eine eingehende Verbindung und
  - Initiator – will eine Verbindung zum Listener erstellen,
  - Client – löst Datenaustausch aus (Request) und
  - Server – reagiert auf Request (Response).

- Exchange Styles
  - message/reply,
  - message/error und
  - message/answer

Frames

- Beschreibung der Inhaltstypen in der Regel durch MIME-Kopf - „entity-headers“. Defaultwerte:
  - `Content-Type: application/octet-stream`
  - `Content-Transfer-Encoding: binary`
- Beschreibung der Nachrichten in der Regel durch XML.
  - Nachrichteninhalt: XML-Element,
  - *Profile*: DTD.

Frame Syntax




```
type      = "MSG" / "RPY" / "ANS" / "ERR" / "NUL"
channel   = 0..2^31-1
msgno     = 0..2^31-1
more                = "*" / "."
seqno     = 0..2^32-1
size                = 0..2^31-1
ansno     = 0..2^31-1
ackno     = seqno
window    = size
```

Frame Semantics

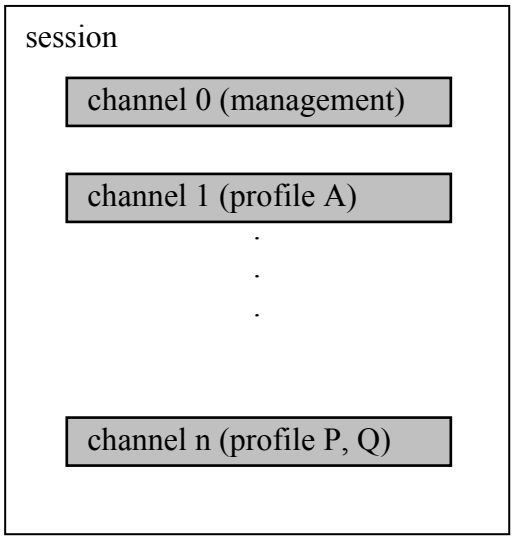
- Kanalspezifische Semantik – In der Channel-Profile werden definiet:
  - Initialisierungsnachricht,
  - Nachricht in der Nutzlast und
  - Nachrichtensemantik.

- Ein Beispiel:

```
S:  MSG 0 1 . 52  0
S:  Content-Type: application/beep+xml
S:
S:  ...
S:  END
```

Sitzungsverwaltung

- Eine Sitzung besteht normalerweise aus mehreren Kanälen.
- Kanal 0: Verwaltungskanal.
- Weitere Kanäle spezifiziert durch *Profile.*



Sitzungsverwaltung

- Profile
  - Ein Profile stellt dar, welche Nachrichten über den Kanal übertragen werden sollen/können.
  - Ein Profile ist durch einen URI identifiziert und regelt die Syntax und Semantik zugelassener Nachrichten.
- Vordefinierte Profile für:
  - Channel Management (Kanal 0)
  - TLS – „Transport Layer Security“
    - `uri = http://iana.org/beep/TLS`
  - User Authentification: SASL Family
    - `uri = http://iana.org/beep/SASL/mechanism`
- Weitere anwendungsspezifische Profile
  - SOAP, RFC 3288
  - …

Sitzungsverwaltung

- Nachrichten
  - Nachrichten werden repräsentiert durch XML-Elemente.
  - Fünf Grundtypen von Nachrichtenelementen für Sitzungsverwaltung.
  - Beziehungen zwischen Rollen, Nachrichtenarten und Nachrichtenelementen:

Rolle	MEG	RPY	ERR
I und L		greeting	error
I oder L	start	profile	error
I oder L	close	ok	error

Sitzungsverwaltung

Greeting-Nachrichten

```
L:      RPY 0 0 . 0 98
L:      Content-Type: application/beep+xml
L:      <greeting>
L:              <profile uri='http://iana.org/beep/TLS' />
L:      </greeting>
L:      END
I:      RPY 0 0 . 0 47
I:      Content-Type: application/beep+xml
I:      <greeting />
I:      END
```

Start-Nachrichten

```
C:      RPY 0 1 . 30 169
C:      Content-Type: application/beep+xml
C:
C:      <start number='1'>
C:              <profile uri='http://iana.org/beep/SASL/OTP' />
C:              <profile uri='http://ibr.cs.tu-bs.de/fangming/MYCONTACT' />
C:      </start>
C:      END
```

Sitzungsverwaltung

Close-, Error-/OK-Nachrichten

```
C:      RPY 0 2 . 390 66
C:      Content-Type: application/beep+xml
C:      <close number='1' code='200' />
C:      END
S:
S:      RPY 0 2 . 211 42
S:      Content-Type: application/beep+xml
S:
S:      <ok />
S:      END
```

Abbildung auf TCP

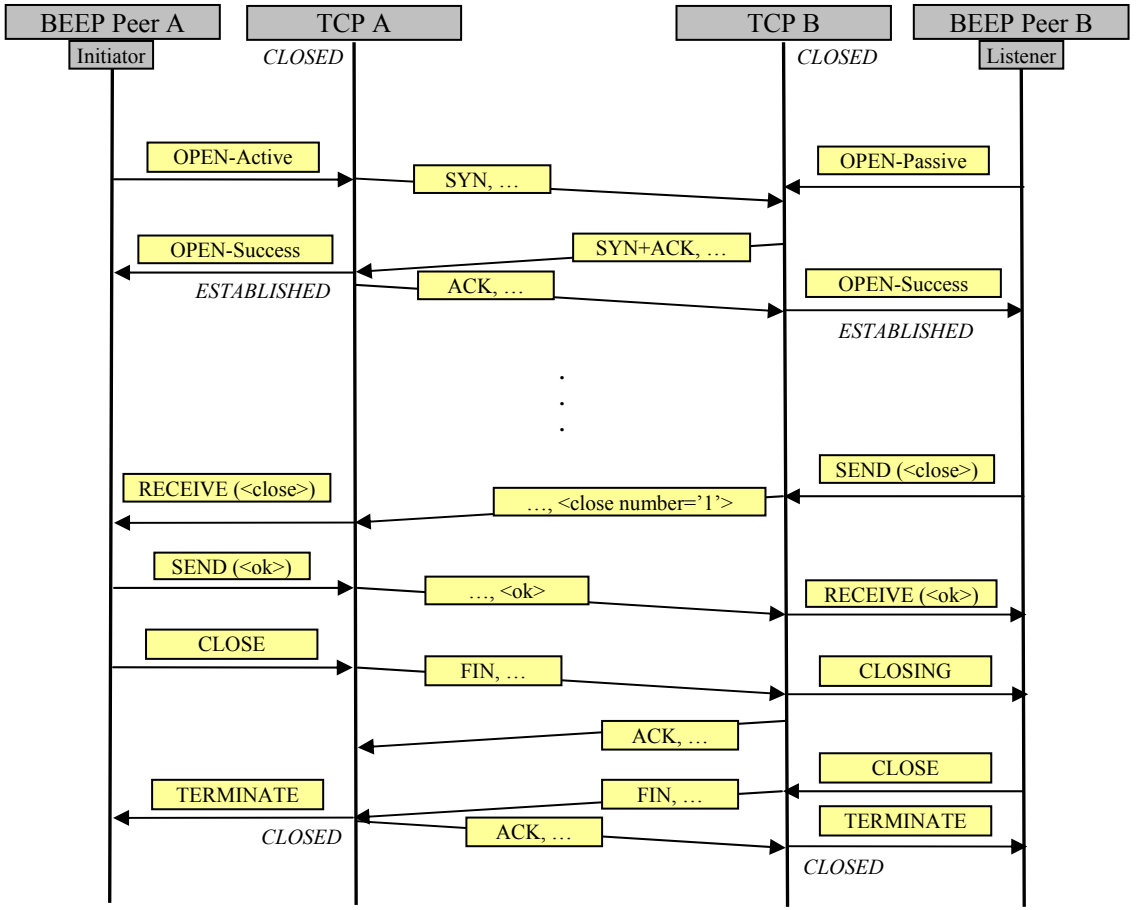
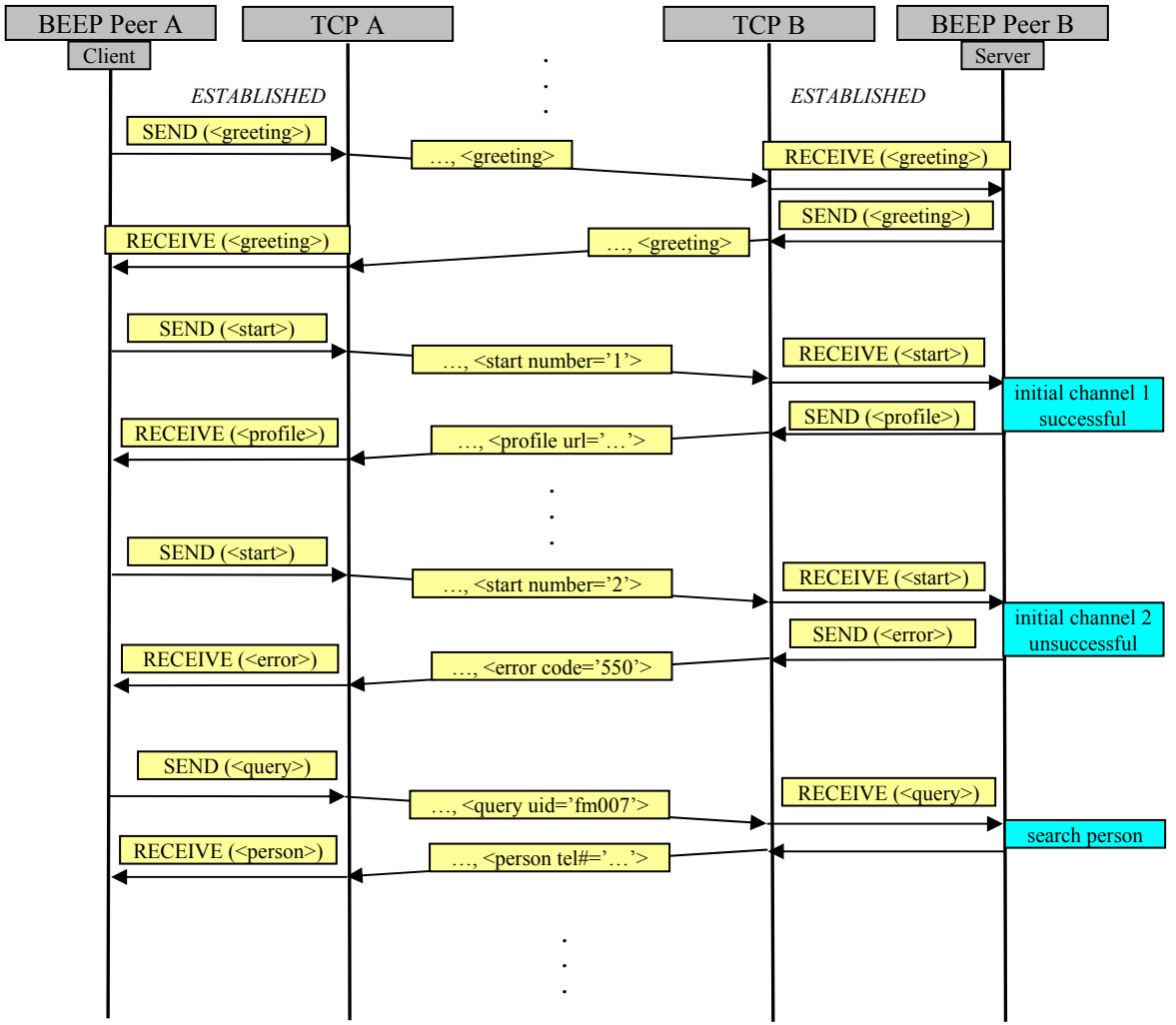
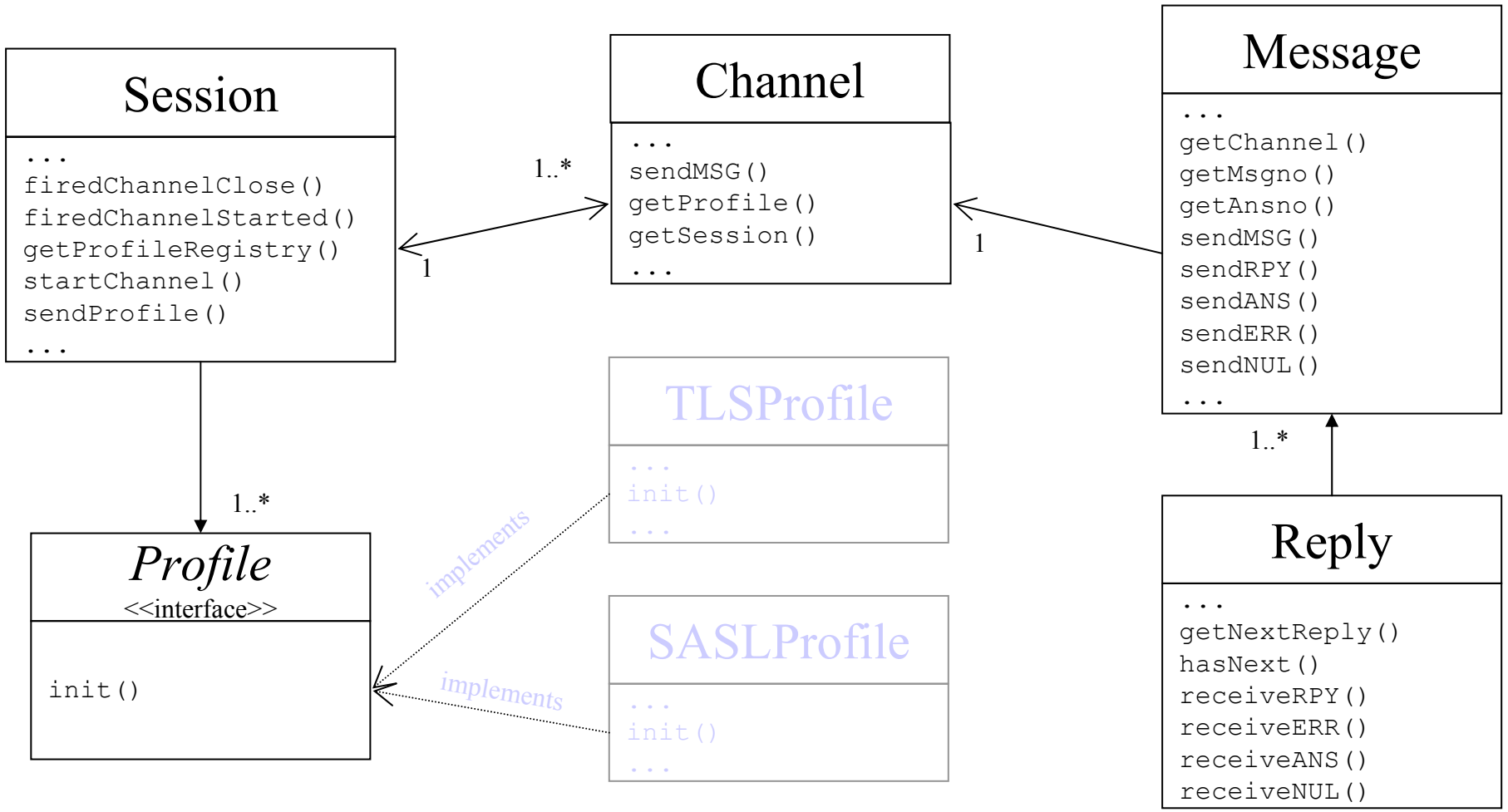


Abbildung auf TCP

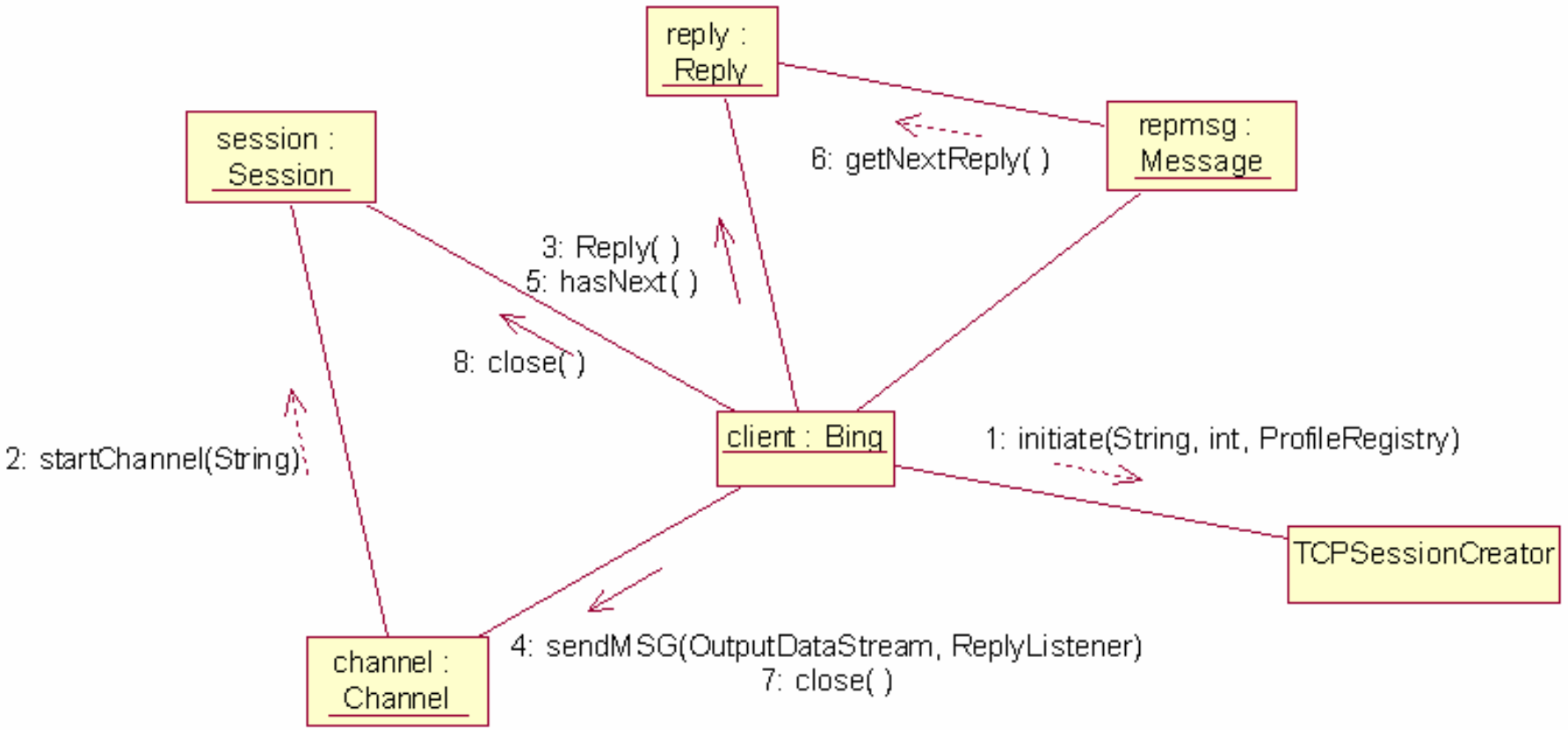


Implementierungsbeispiel: Java BEEP



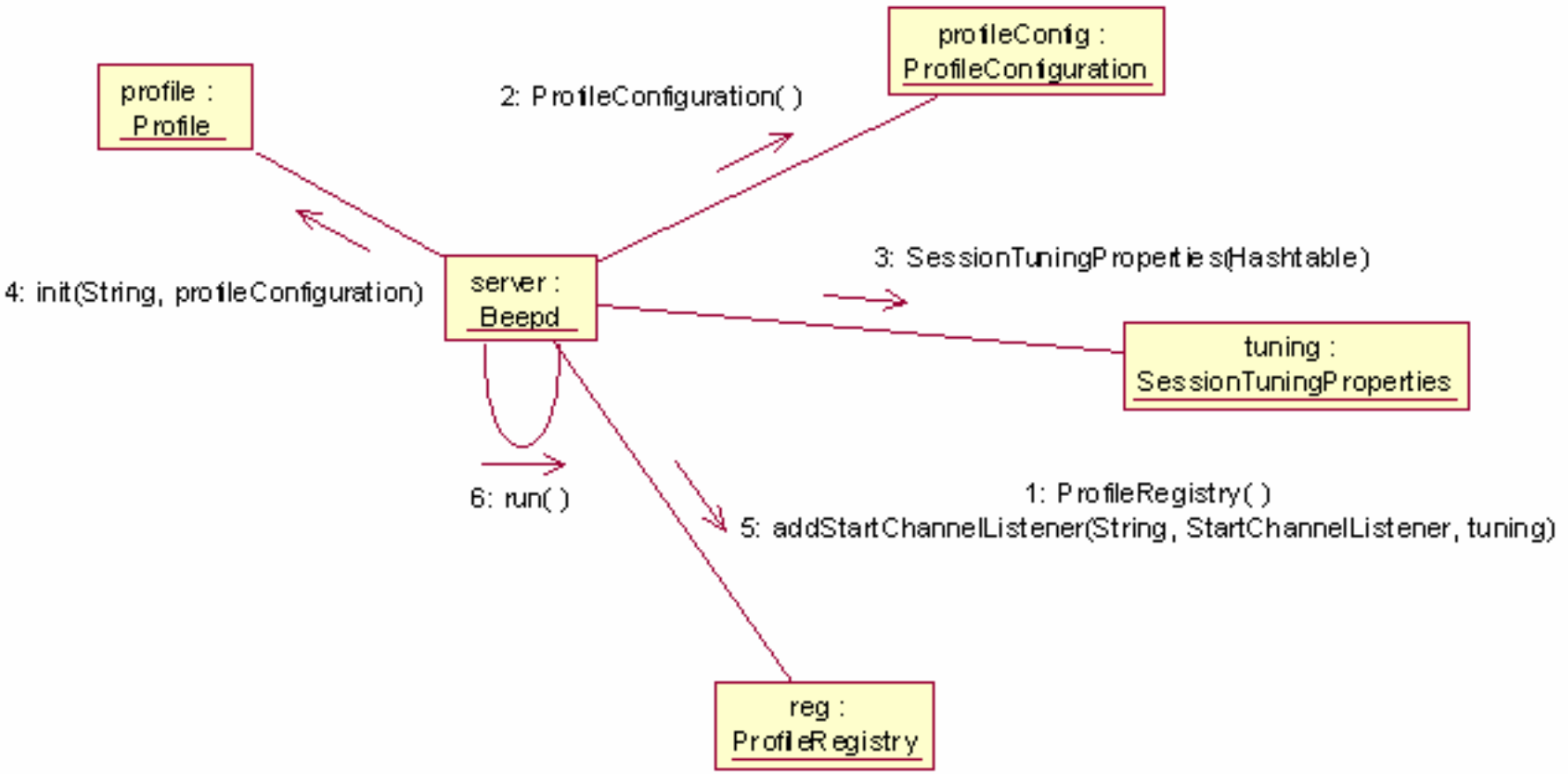
Implementierungsbeispiel: Java BEEP

Client



Implementierungsbeispiel: Java BEEP

Server



Zusammenfassung

- BEEP ist ein Framework für Anwendungsprotokoll.
  - Asynchronität,
  - XML Nachrichtenformate,
  - Unterstützt Peer-toPeer-, Client/Server- und Server-to-Server-Kommunikation,
  - Profile für standardisierte und spezifische Anwendungen.
- Der „Missbrauch“ von HTTP für Anwendungsprotokolle kann durch BEEP elegant abgelöst werden.
- BEEP Home - http://www.beepcore.org