

Zwischenbericht

Diplomarbeit

Thema: Evaluation des Projekts
„Quality Objects“

Sven Harazim

sh17@inf.tu-dresden.de

Gliederung

- Aufgabenstellung
- Übersicht Dienstgüteparameter
- Framework „Quality Objects“
- Beispielanwendung
- Weiterer Verlauf der Arbeit

Aufgabenstellung

- Ausführung von Qualitätsanforderungen und deren Adaption bei Nichteinhaltung
- Umsetzung ausgewählter Anforderungen mittels des Frameworks „Quality Objects“
- Bewertung der Durchsetzung der Anforderungen und der weiteren Anwendungsmöglichkeiten der QDL
- Analyse des vom QDL-Compiler generierten Quelltextes

Dienstgüteparameter

- Allgemeine Dienstgüteparameter
 - Datendurchsatz
 - Übertragungsverzögerung
 - Verzögerungsschwankung
 - Datenfehlerrate
 - Verlustrate
 - Priorität

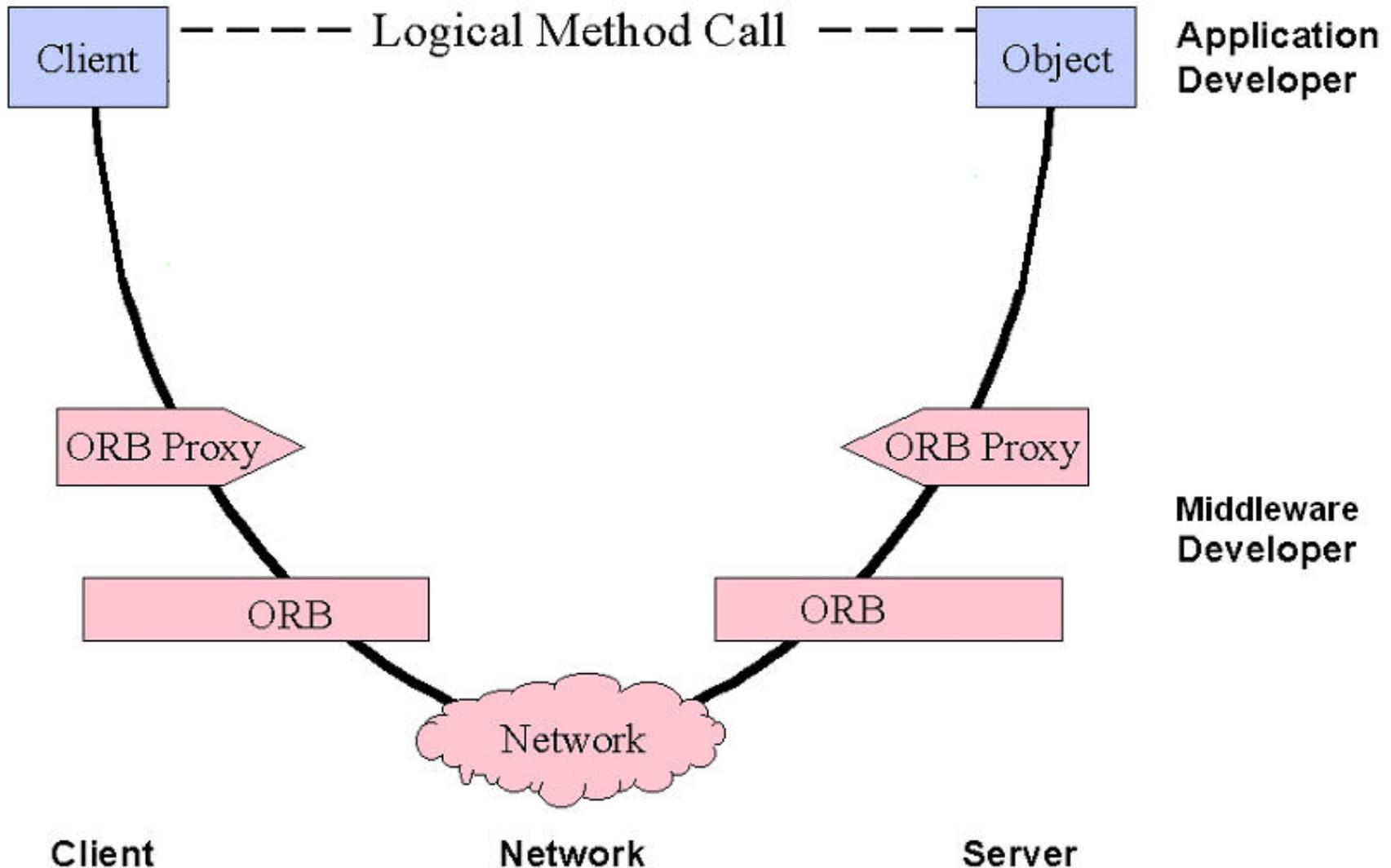
Dienstgüteparameter

- Dienstgüteparameter bei Videoanwendungen
 - Framerate
 - Bildauflösung
 - Kompressionstyp
 - Kompressionsstärke
 - Farbtiefe
- Weitere Parameter bei Audio- und Nachrichtendienstanwendungen

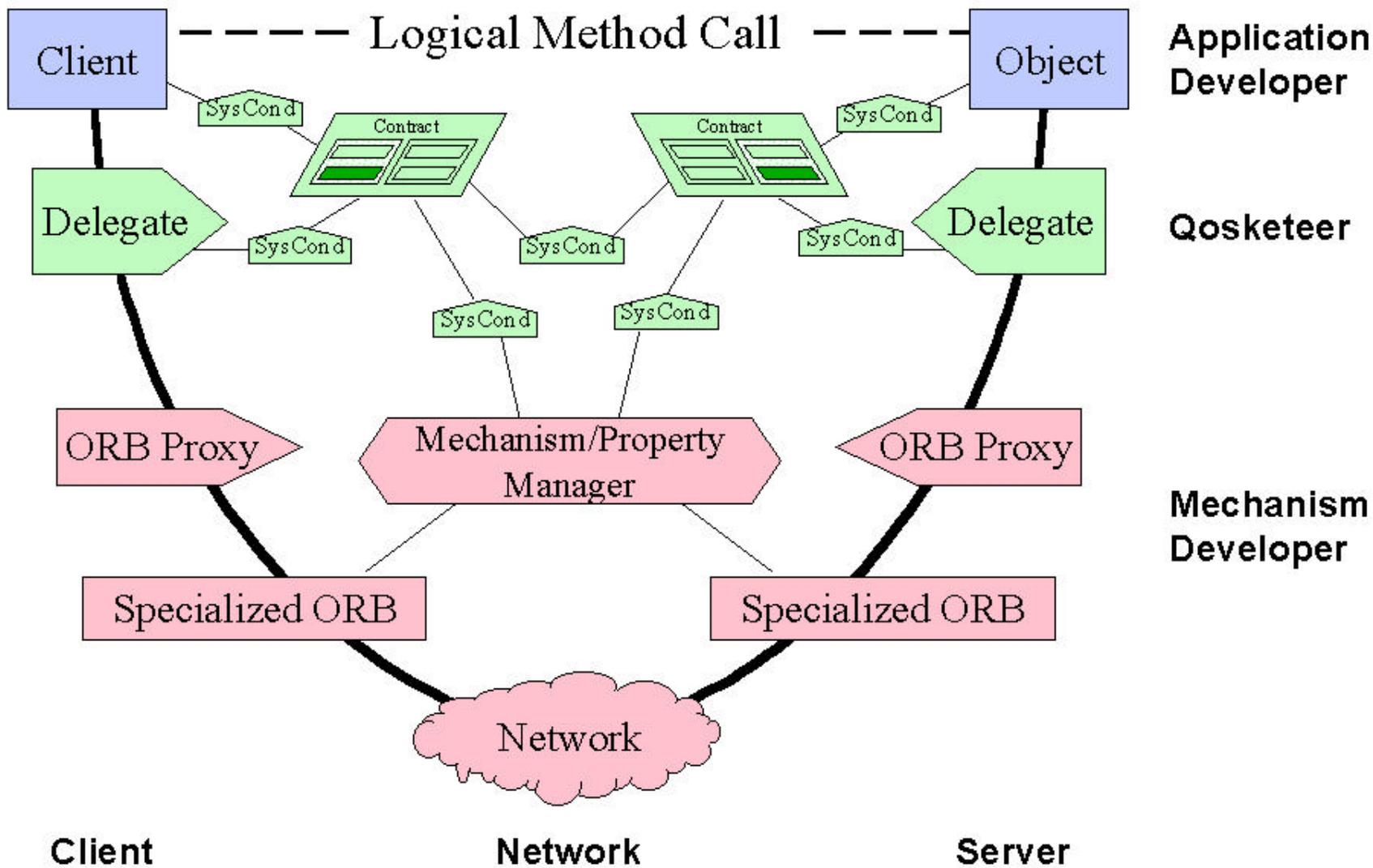
Framework und Toolkit QuO

- Infrastruktur zur Entwicklung von verteilten Anwendungen unter Einbeziehung von Dienstgüteparametern (CORBA, RMI)
- Erzeugung adaptiven wiederverwendbaren Codes (Qoskets) mittels Quality Description Languages (QDL)
 - Contract Description Language (CDL)
 - Aspect Structure Language (ASL)

Architektur von QuO



Architektur von QuO



Beispielanwendung

Börsenticker

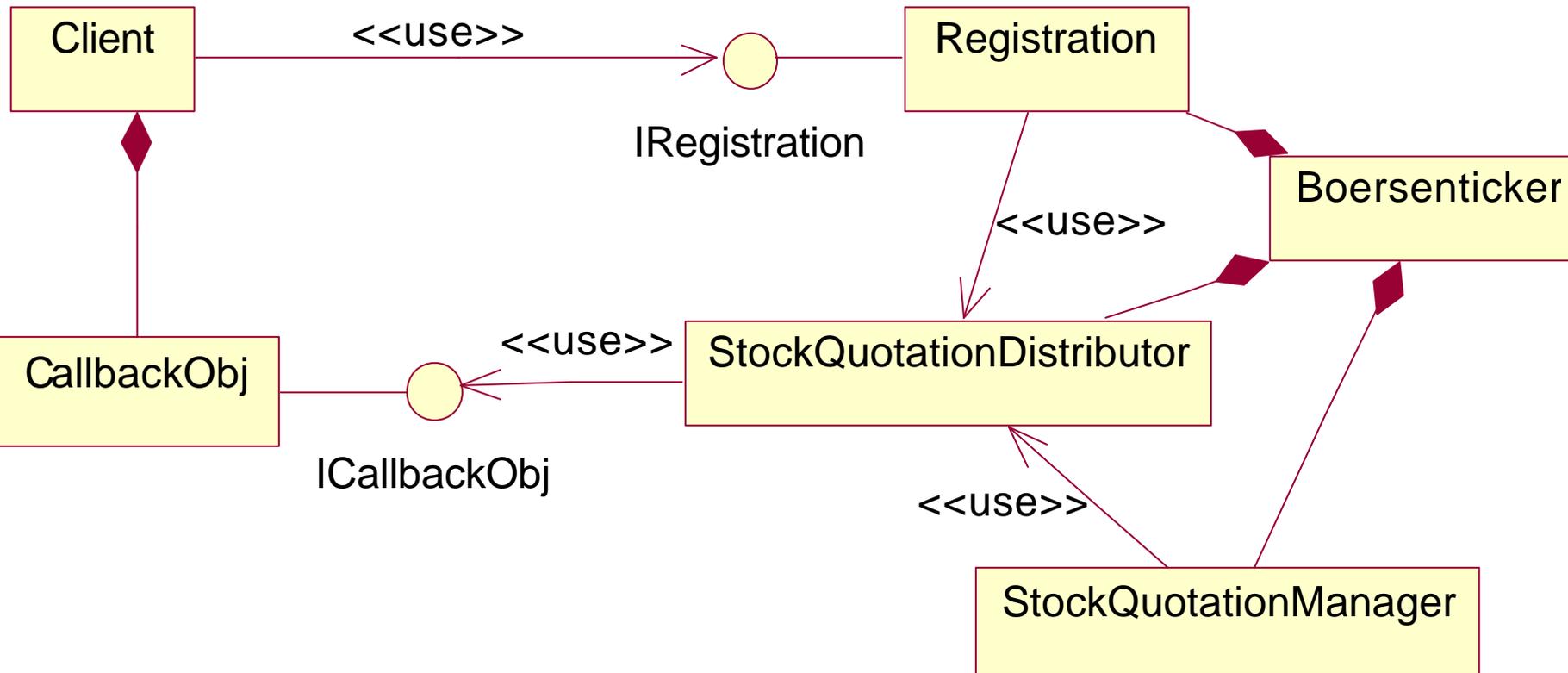
(Szenario aus dem COMQUAD-Projekt)

- Client abonniert bei einem Server einen Aktienkurs
 - Update-Rate
 - Priorität
 - Maximale Verzögerung
 - (Bandbreite)
- Die Authentizität wird durch den in den QuO-Kernel integrierten Security Service sichergestellt.

Entwicklungsreihenfolge

- Erstellung der geschäftlichen Anwendung ohne Dienstgüteparameter (DP)
- Erstellung bzw. Auswahl der Qoskets zur Überwachung und Kontrolle der DP
- Generierung des Adapters zw. Geschäfts-anwendung und Qoskets (ASL)
- Erstellung des Delegate-Objektes und Integration in Geschäftsanwendung

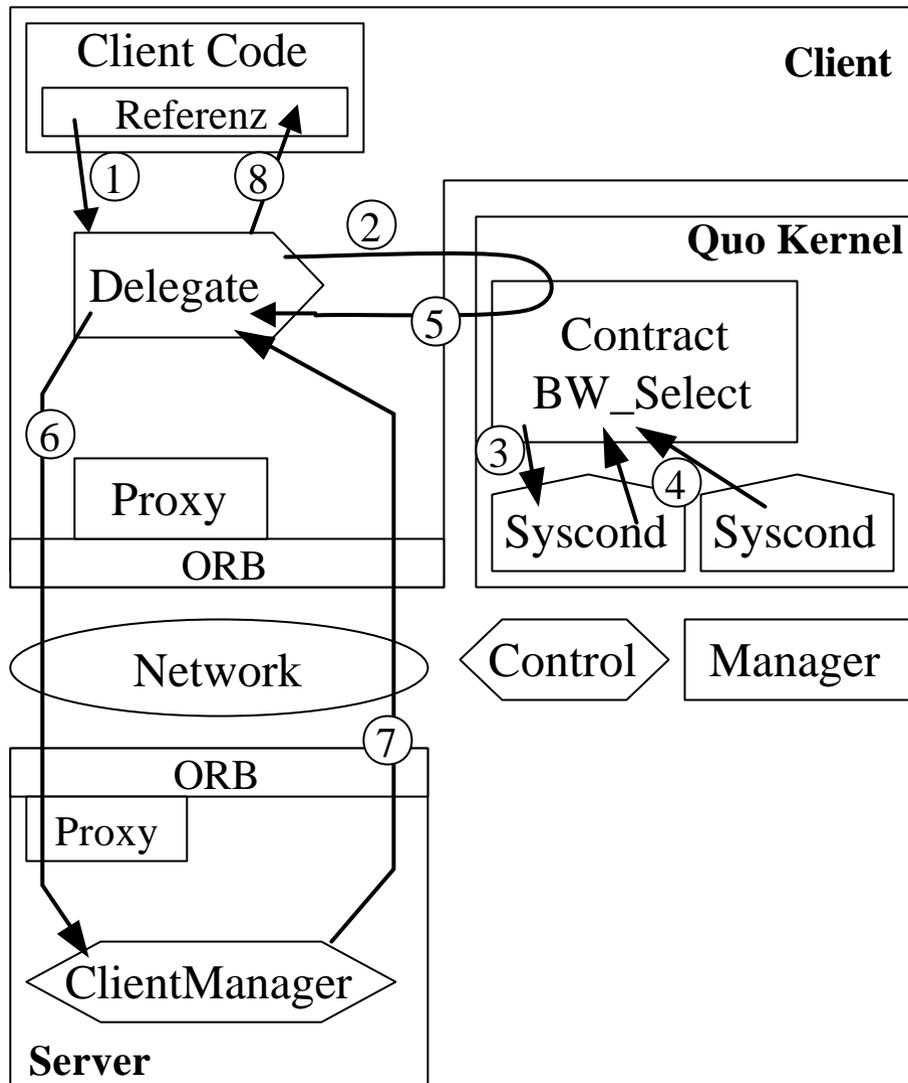
Klassendiagramm



Qoskets im Börsenticker

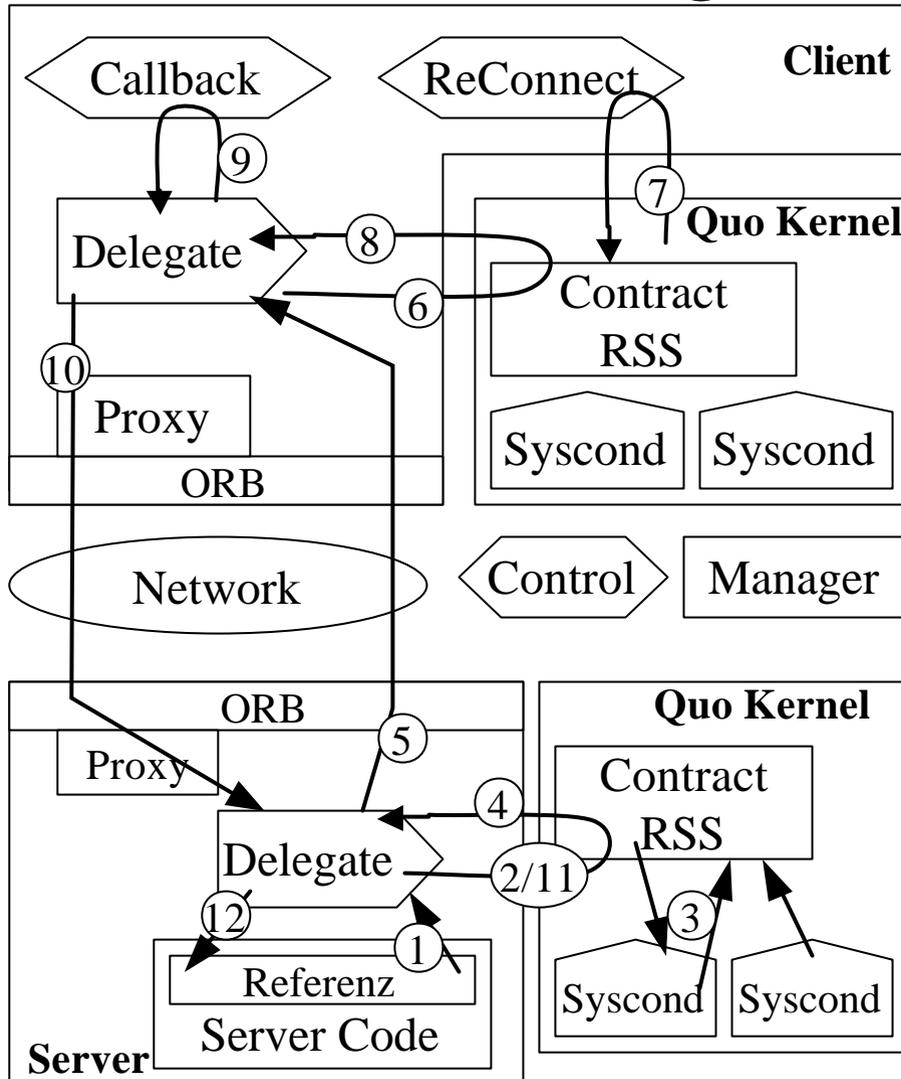
- BW_Select-Qosket
 - Wählt aus einer Liste von (Börsenticker-) Servern den mit der größten Bandbreite aus
- RSS-Qosket
 - Misst die verfügbare Bandbreite zw. Server und Abonnent und die benötigte Zeit für den Methodenaufruf.

Abonnement eines Aktienkurses



- 1) Client startet Registrierung
- 2) Delegate überprüft Contract
- 3) Bandbreiten zu einzelnen Servern werden bestimmt
- 4) Contract übernimmt die aktuellen Bandbreiten
- 5) Aktuelle Region (schnellste Server- verbindung) wird übergeben
- 6) Reg. wird an entsprechenden Server weitergeleitet
- 7) Server gibt Reg.-Kennung zurück
- 8) Weiterleitung des Rückgabewertes zum Client

Auslieferung eines Aktienkurses



- 1) Server liefert Aktienkurs
- 2) Delegate überprüft Contract
- 3) Bestimmung der Startzeit und der erwarteten Bandbreite zum Client
- 4) Aktuelle Region (Fast, Expected, Slow) und Zeit wird übergeben
- 5) Entsprechend Priorität und Bandbreite wird Aktienkurs weitergeleitet
- 6) Bestimmung der Ankunftszeit
- 7) Wenn Auslieferungszeit > Max. Verzögerung → Neuen Server suchen
- 8) Übergabe des Zeitwertes
- 9) Übergabe des Aktienkurses
- 10) Rückkehr des Aufrufes
- 11) Bestimmung der Rückkehrzeit
- 12) Rückkehr des Aufrufes

Weiterer Verlauf der Arbeit

- Fertigstellung der Beispielanwendung
- Untersuchung/Bewertung der Anwendungsmöglichkeiten der QDL
- Analyse des durch den QDL-Compiler erzeugten Quelltextes