An Approach for QoS Contract Negotiation in Distributed Component-Based Software

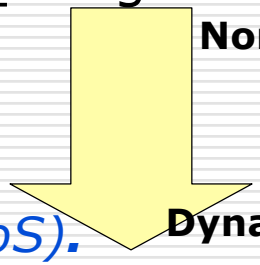
Mesfin Mulugeta and Alexander Schill
Dresden University of Technology,
Germany

CBSE 2007 Medford, MA, USA
July 9 – 11, 2007

Motivation

- COMQUAD research project (2002 – 2004), TU Dresden

- Component Contracts [Beugnard et al, 1999]:
  - *Syntactic*,
  - *Behavioral*,
  - *Synchronization*, and
  - *Quality of Service (QoS)*.

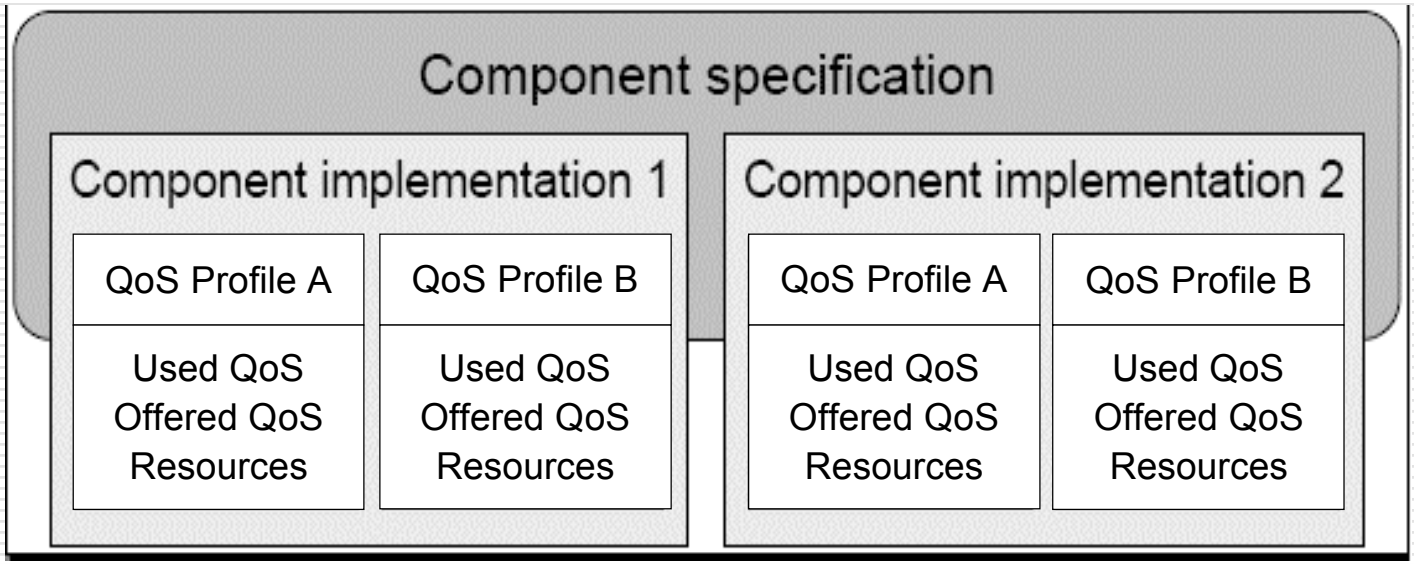


- *A QoS contract* specifies constraints on the non-functional properties like response time, throughput, etc.

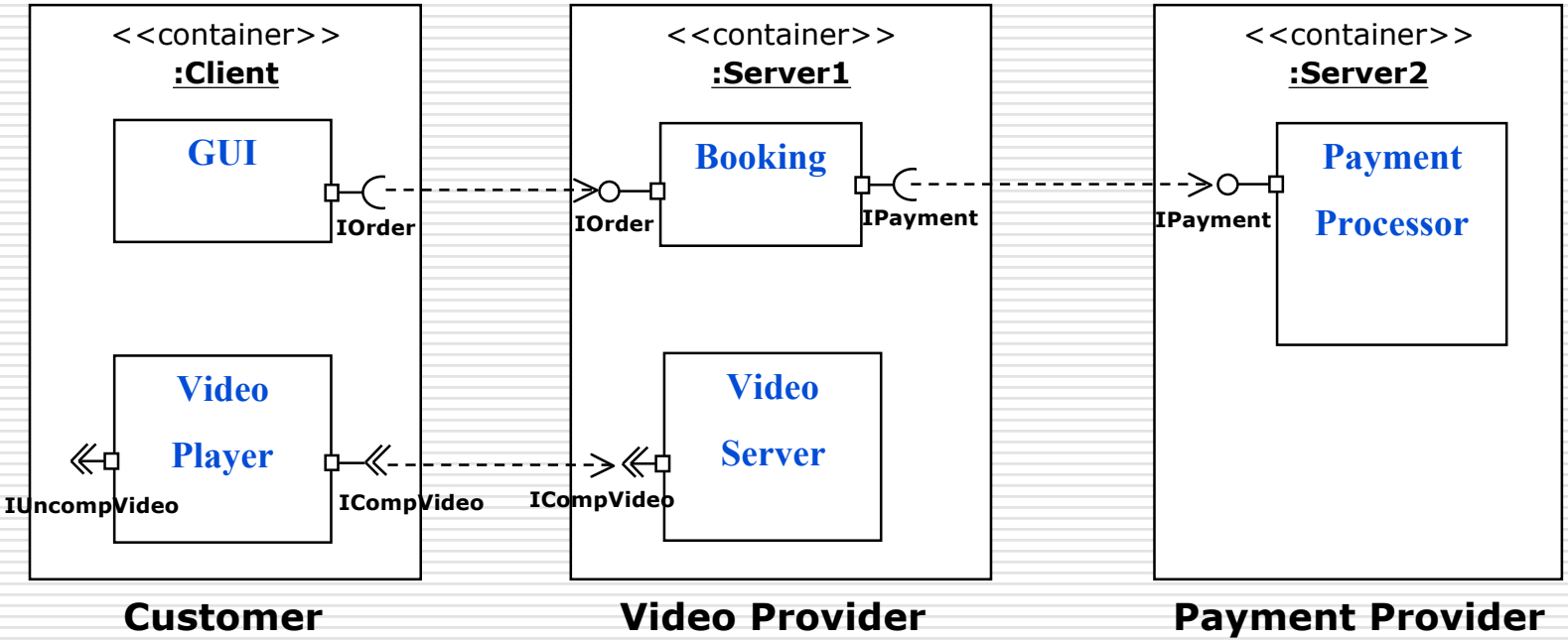
→ The consideration of QoS contracts requires the support of *QoS Contract Negotiation*

QoS Contract & Its Specification

- Component Specification (CQML+)



Example: Componentized Video Streaming



- "order service" – `GUI`, `Booking`, `PaymentProcessor`
- "stream video" – `VideoPlayer`, `VideoServer`

Example: QoS Contract Specification
➔ QoS-Profiles

a QoS-Profile

VideoPlayer				VideoServer	
	uses ICompVideo (resolution, frame rate in s^{-1})	provides IUnCompVideo (resolution, frame rate in s^{-1})	Resource (CPU in %, bandwidth in Kbps, memory KB)	Provides ICompVideo (resolution, frame rate in s^{-1})	Resource (bandwidth in Kbps)
1	352x288, 30	352x288, 30	13.23, 2165, 31.6	352x288, 30	2165
2	352x288, 15	352x288, 15	8.90, 2146, 30	352x288, 15	2146
3	352x288, 5	352x288, 5	5.90, 1852, 29.5	352x288, 5	1852
4	352x288, 1	352x288, 1	2.31, 1644, 29.2	352x288, 1	1644
5	176x144, 30	176x144, 30	0.97, 321, 25.6	176x144, 30	321
6	176x144, 15	176x144, 15	0.90, 252, 18.8	176x144, 15	252
7	176x144, 10	176x144, 10	0.62, 208, 24.4	176x144, 10	208
8	176x144, 5	176x144, 5	0.39, 135, 24.0	176x144, 5	135

Problem & Challenges

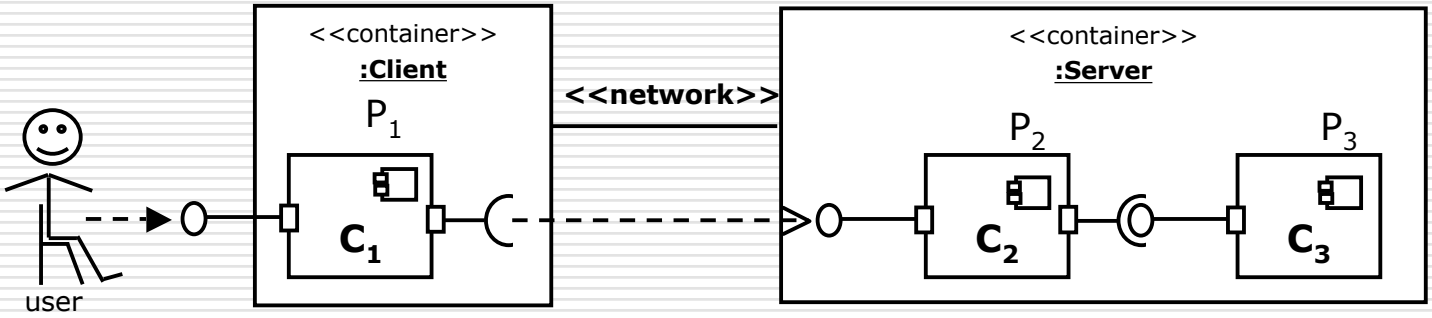
- *Problem:*
  - ★ How to select concrete QoS contracts at the ports of interacting components that are deployed in distributed nodes?

- *Challenges:*
  - Find a solution that satisfies a number of different types of constraints.
  - Find a "better" solution. (*A* is a "*better" solution* than another solution *B* if *A's* utility is higher than that of *B)*
  - Find the solution efficiently.

Proposed Approach

- QoS Contract Negotiation as a Constraint Satisfaction Problem (CSP)
- Multi-Phase Negotiation: *Coarse-grained* & *Fine-grained*
- Heuristics
  - Single Client – Single Server Scenario
  - Multiple-Clients Scenario

QoS Contract Negotiation as a CSP



Variables: **P_1, P_2, P_3 (QoS-Profiles to be used)**

Domain: **a set of QoS-Profiles are specified for each component**

QoS properties: **d_1, d_2, …, d_k**

Constraints: **user's, conformance, and resource**

User's QoS Requirement on d_1 > P_1.Offered.d_1 **(User's constraint)**

P_2.Required.d_1 => P_3.Offered.d_1 **(conformance constraint)**

P_1.Required.responseTime => P_2.Offered.responseTime + delayInNetworkContainers

P_2.Resources.memory + P_3.Resources.memory ≤ Server.Resources.memory **(resource constraint)**

“Coarse-grained” and “fine-grained” Negotiation

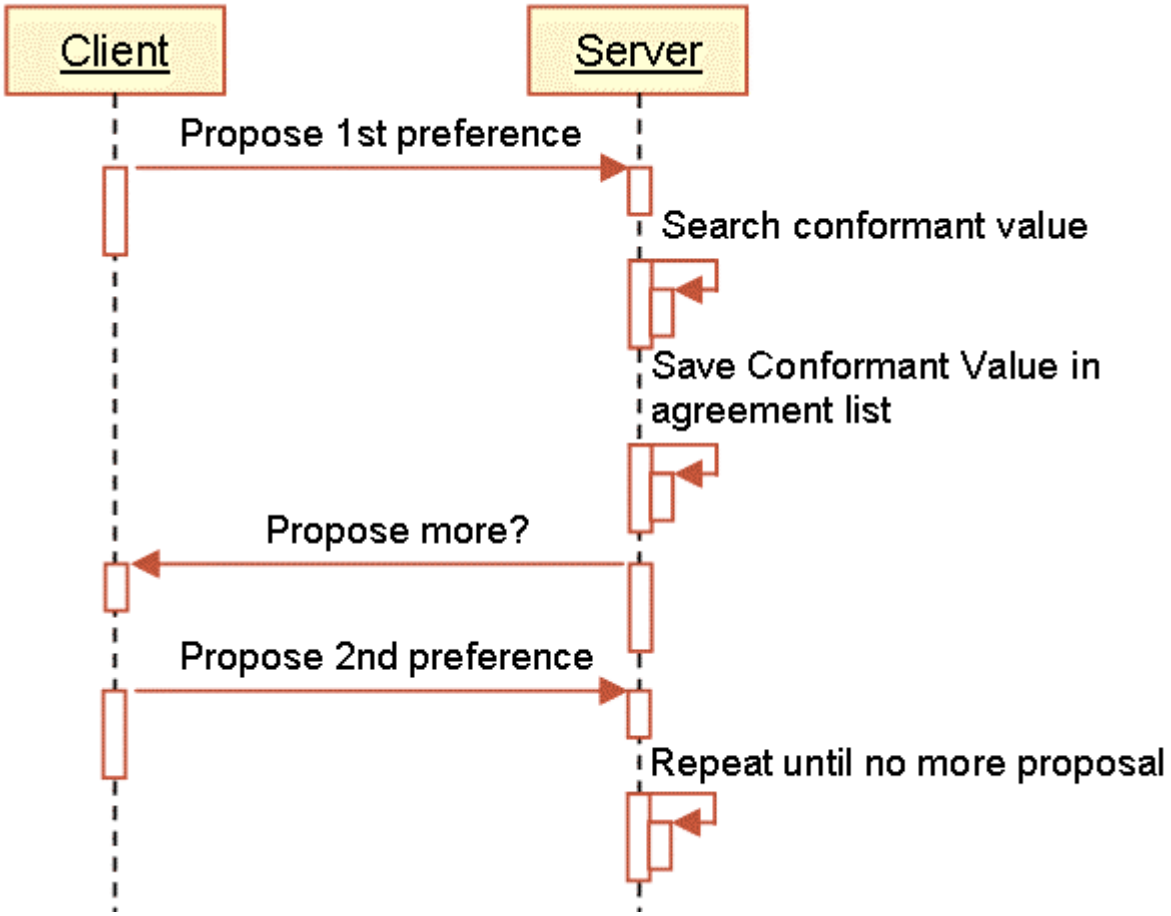
- We classify properties as “coarse-grained” and “fine-grained”
  - → Simplifies negotiation
  - → Speeds up negotiation

- A coarse-grained property is associated with one or multiple fine-grained properties. For a certain value of the coarse-grained property, the fine-grained properties can possibly take different values depending on the allocated resource.

- Examples:

“Coarse-grained”	“fine-grained”
video coding	frame rate, resolution
security goals	security mechanisms

Coarse-grained Negotiation

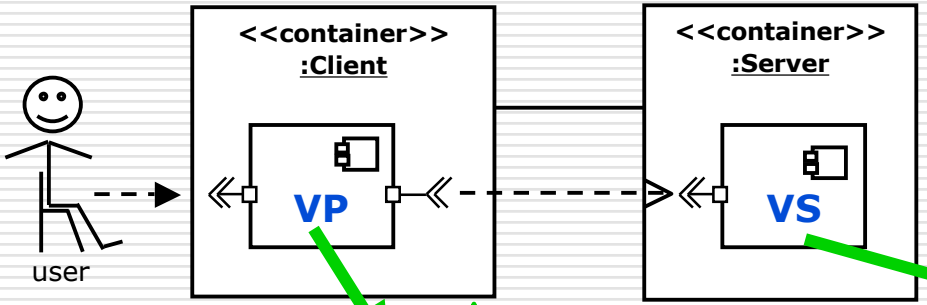


- e.g. Video Coding Property
- Client's preferences
  - *mp42*
  - *mpg4*
  - *h263*
- Server's preferences
  - *mp43*
  - *mp42*
  - *mpg4*
  - *h264*
- Result:
  - No agreement
  - Agree on one value
  - Agree on multiple values

Fine-grained Negotiation

- Constraint Satisfaction Optimization Problem (CSOP) is used to model this phase.
- We use the standard branch and bound technique (B&B).
- For the B&B, we need to define:
  - Variable & value selection policies
  - Objective function, *f*
  - Heuristic function, *h*

Example: Fine-grained Negotiation Algorithm (Single-Client – Single-Server)



User's requirement: resolution ≥ 176x144 and frame rate > 12fps

Available resources: Client's CPU 80%, Server's CPU 80%, and end-to-end bandwidth 1Mbps

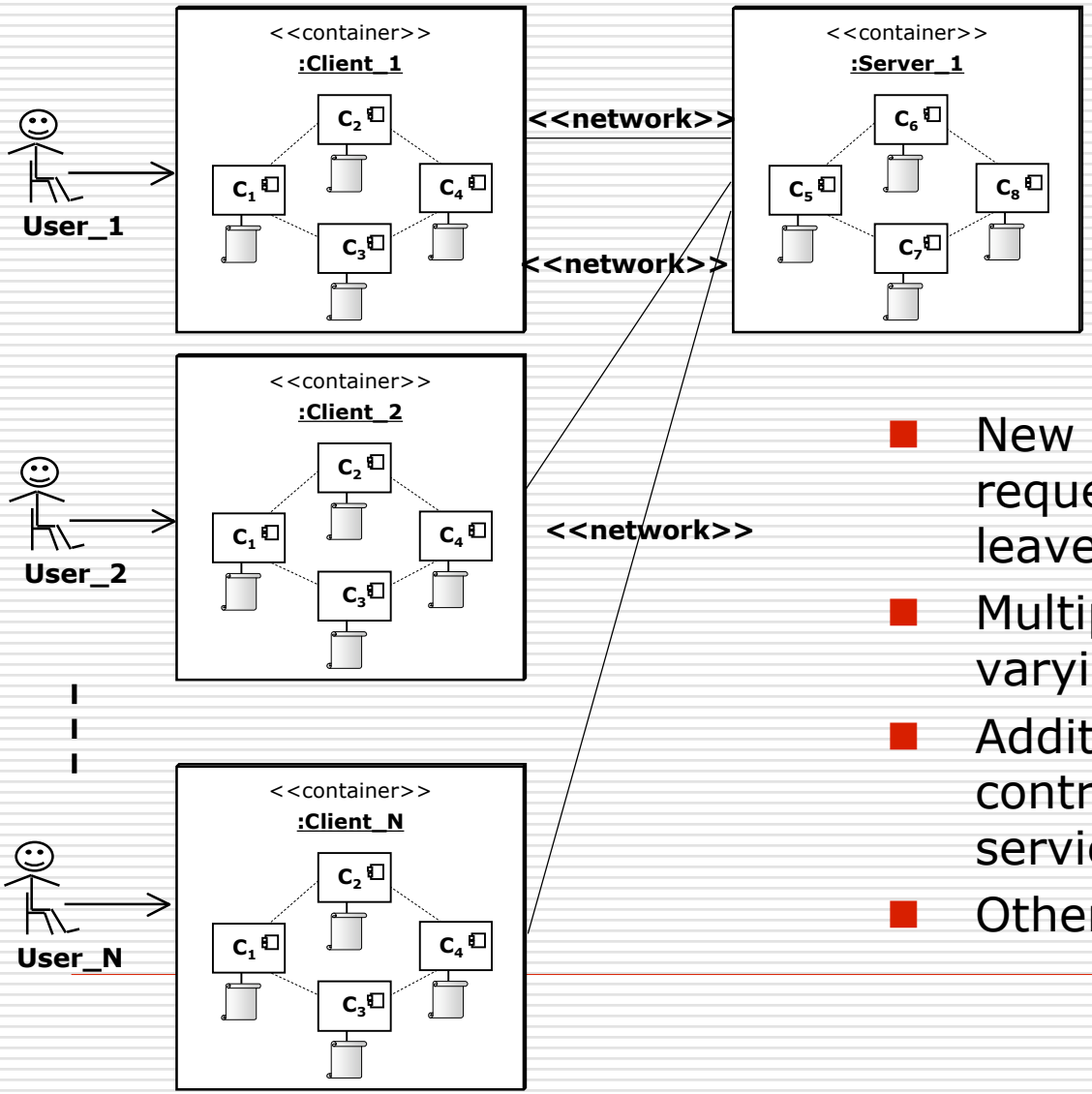
Selected QoS-Profiles

A solution found!

	VideoPlayer			VideoServer	
	uses ICompVideo (resolution, frame rate in s^{-1})	provides IUnCompVideo (resolution, frame rate in s^{-1})	Resource (CPU in %, bandwidth in Kbps, KB)	Provides	Resource (bandwidth in Kbps)
1	352x288, 30	352x288, 30	13.23, 2165, 31.6	352x288, 30	2165
2	352x288, 15	352x288, 15	8.90, 2146, 30	352x288, 15	2146
3	352x288, 5	352x288, 5	5.90, 1852, 29.5	352x288, 5	1852
4	352x288, 1	352x288, 1	2.31, 1644, 29.2	352x288, 1	1644
5	176x144, 30	176x144, 30	0.97, 321, 25.6	176x144, 30	321
6	176x144, 15	176x144, 15	0.90, 252, 18.8	176x144, 15	252
7	176x144, 10	176x144, 10	0.62, 208, 24.4	176x144, 10	208
8	176x144, 5	176x144, 5	0.39, 135, 24.0	176x144, 5	135

***f, h* - utility**

Multiple-Clients Scenario - challenges



- New clients constantly send requests while existing clients leave
- Multiple clients may have varying requirements
- Additional parameters (e.g. contract duration, time of service delivery)
- Other negotiation goals

Multiple-Clients Scenario – Addressing the challenges

- Resource allocation strategy
  - Light-load
  - Over-load
  - Clients' request rate known

- Classes of users & service class

	premium service class (resolution, frame rate in s^{-1})	*normal* service class (resolution, frame rate in s^{-1})
1.	352x288, 30	176x144, 30
2.	352x288, 25	176x144, 15
3.	352x288, 20	176x144, 5

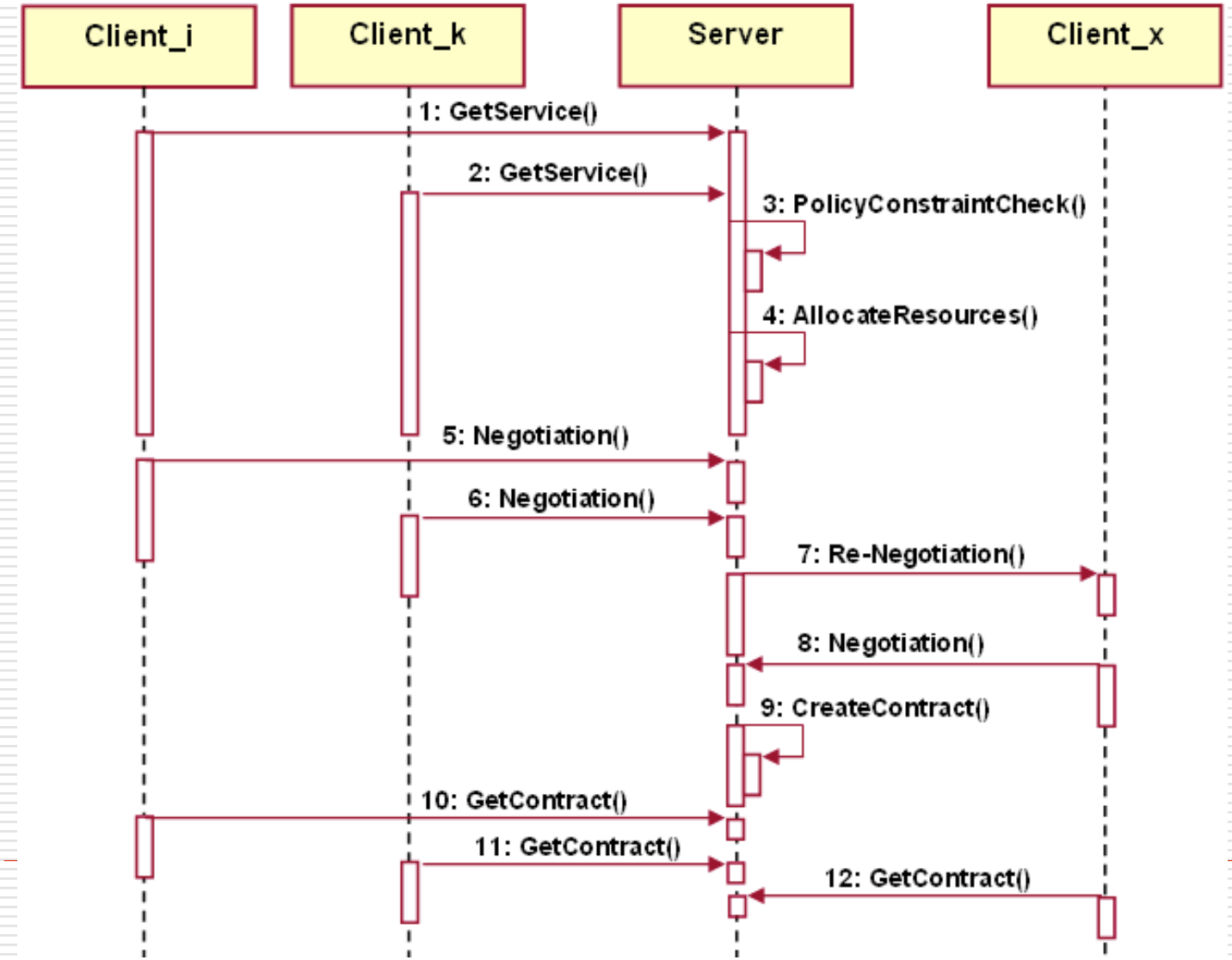
- Utility functions

$$U = \sum_{clients} \alpha_i U_{s_i}$$

- Policy Constraints

e.g. How to favor clients of the same class when re-negotiating contracts?

Possible interaction between multiple clients and a server



Conclusions & Outlook

- Conclusions:
  - We have modeled QoS contract negotiation as a CSOP and have performed negotiation in multiple-phases to get a good solution.
  - The algorithm in a single-client – single-server scenario is $O(nd^2)$ (n=#components, d=#QoS-Profiles)
  - We generalized our approach to a multiple-clients scenario.

- Outlook:
  - Globally optimal solutions
    - Defining utility functions
    - Considering more parameters in a utility function

Thank You!