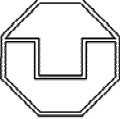


Types of Quality of Service Contracts for Component-Based Software

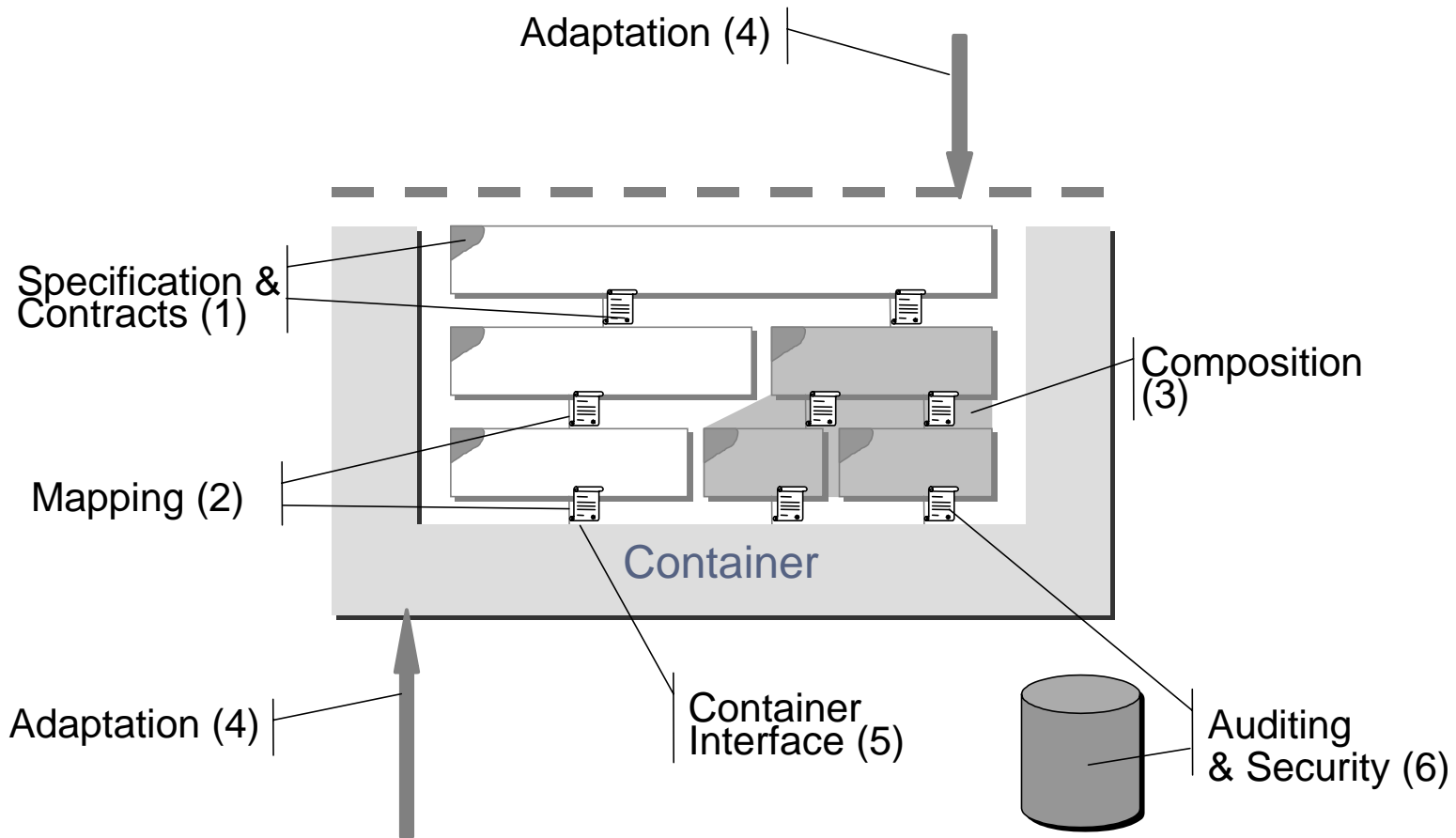
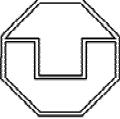
Steffen Zschaler, Simone Röttger
COMQUAD project
Dresden University of Technology

Outline

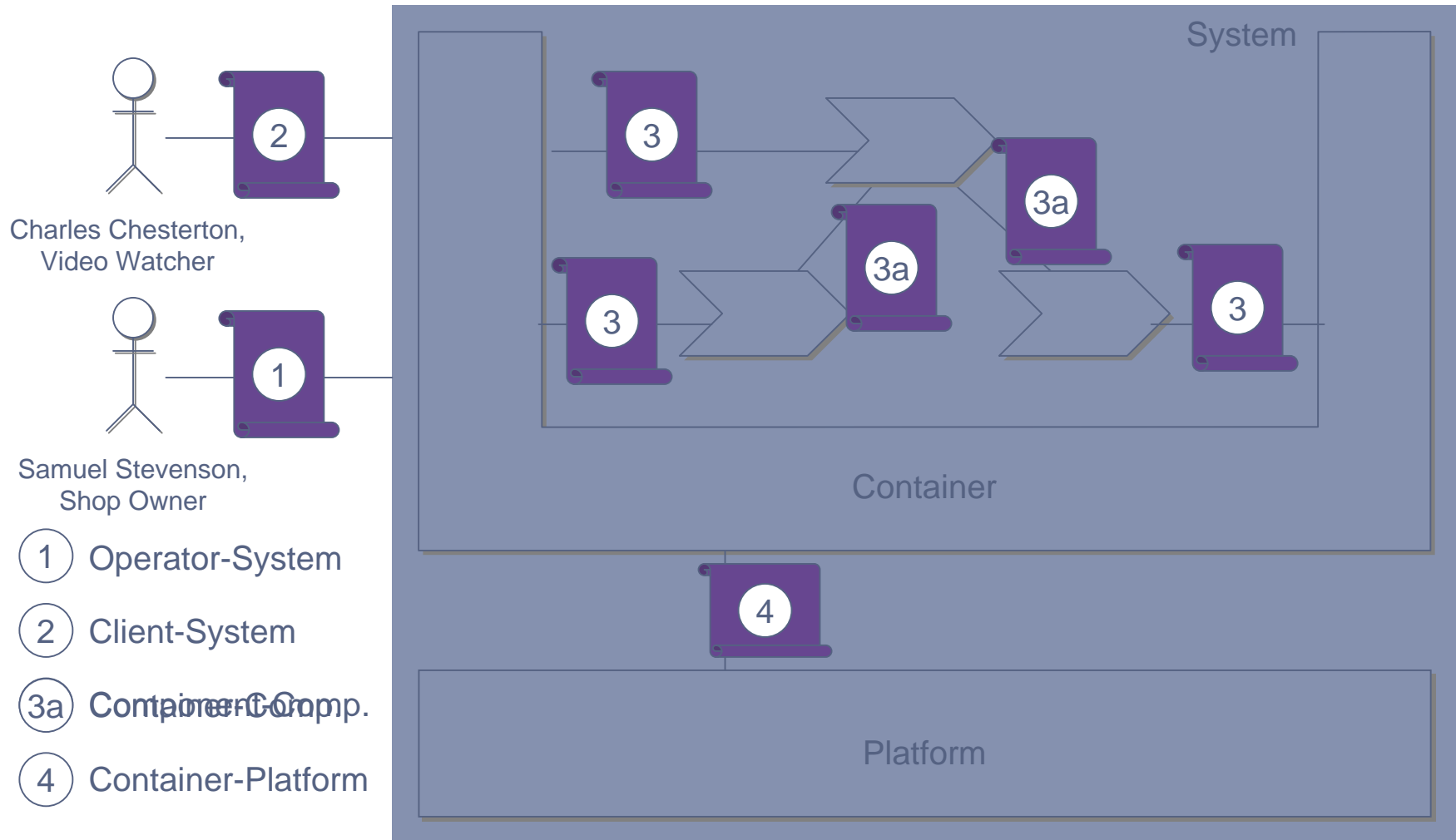
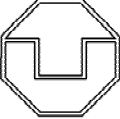
- **Motivation**
- **Contracts and Roles**
- **Putting it all together**
- **Conclusion**



Motivation



Roles and Contracts



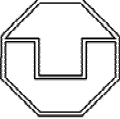
Operator-System Contract

- **Global view over all clients**
- **Operator guarantees load profile**
 - Distribution of request interarrival times
 - Average number of clients per second
 - ...
- **System guarantees QoS-level**
 - For each request or for a percentage
- **Enables scheduling of all requests in advance**

Example:

Video-on-Demand System:

- Load profile: Number of Clients ≤ 15 Clients/min
- QoS-level:
 - For 90% of requests: framerate ≥ 15 frms/sec
 - Remaining 10%: framerate ≥ 12 frms/sec



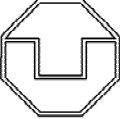
Client-System Contract

- **For each request**
 - *In accordance to operator-system-contract*
 - Otherwise system may reject
- **System guarantees QoS-level for request**
- **Client guarantees time of usage**
 - Start of request
 - For streams: Will there be breaks? How often? For how long?

Example

Client Charles wants to watch „The Matrix“ (15 frm/s, 800x600) in 5 minutes from now.

- QoS-level:
 - 15 frm/s, 800x600 Pixel → only if Charles is among the “90%”
- Client guarantee:
 - Movie starts in 5 minutes
 - Maximum break length: 15 minutes



Container-Component Contract

- **Selects component's working region**
- **Container guarantees resources and services from other components**
- **Component guarantees QoS-level**

- **Extends component-component contracts**
- **Container determines required number of instances based on load profile from operator-system contract**

Example

Videoplayer component requires 30ms of CPU every 60ms to provide 15 frms/sec.

- QoS-level:
 - 15 frm/s, 800x600 Pixel
- System guarantee:
 - 30ms of CPU every 60ms



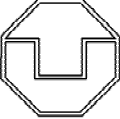
Container-Platform Contract

- Backs resource demand of the complete application
- Platform guarantees required resources
- Container guarantees upper usage limit

- Required resources are computed from:
 - Load profile,
 - Component network, and
 - Resource demand of individual components
- Scheduling of component usage allows detailed planning of resource demand

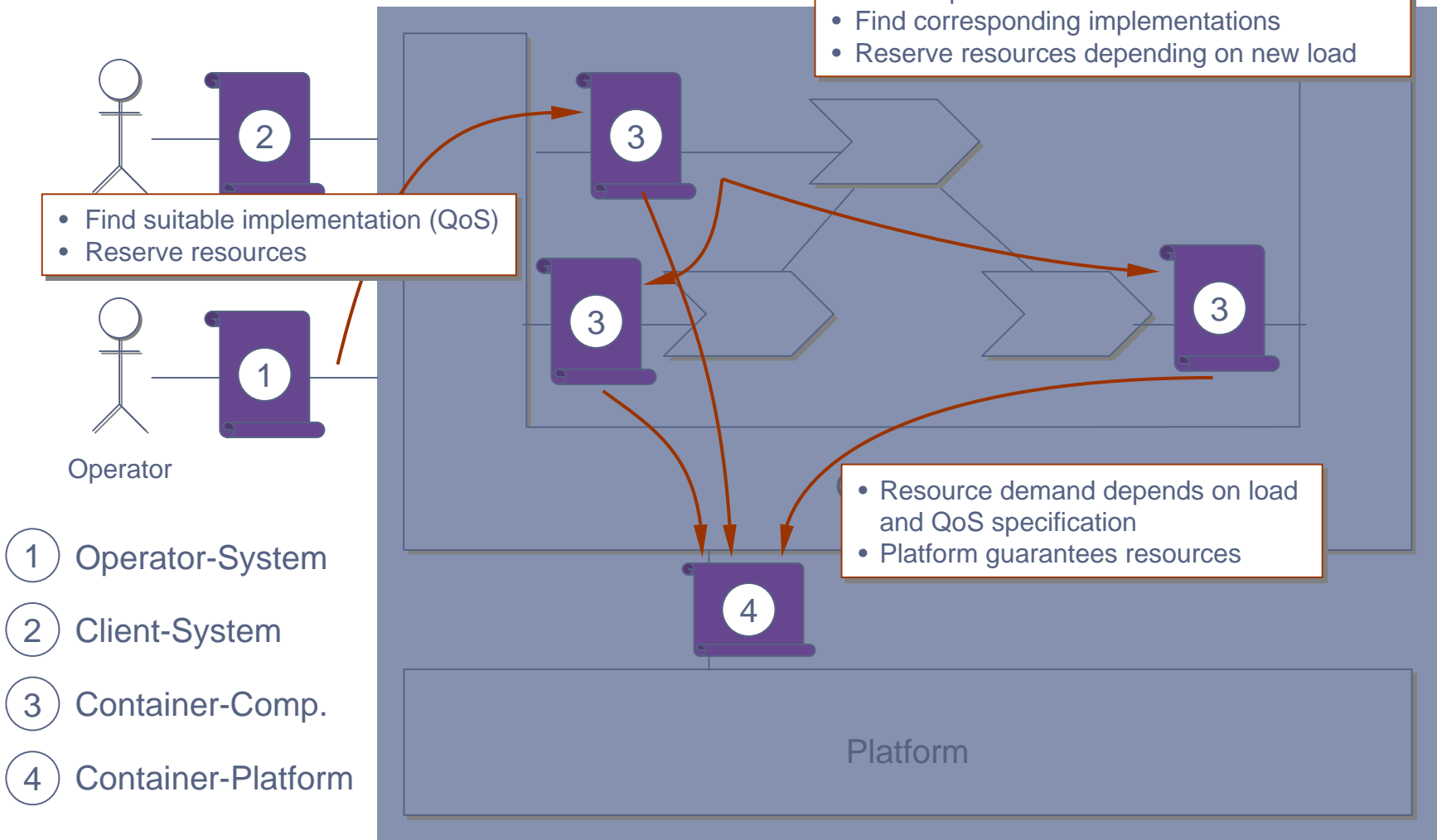
Example

- Operator-System-Contract specifies Load: *5 requests/min* and QoS: *avg(response) < 500ms*
- There are two implementations: **A**: *response < 300ms, cpu = 40%*
B: *response < 600ms, cpu = 20%*
- Possible Component Schedule:
 - 40% of requests: **A**; 60% of requests: **B**
 - *avg(response) <= 480ms, cpu = 52%*
 - Schedule fulfils requirements



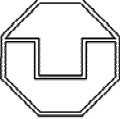
Negotiation of Contracts

At Deployment

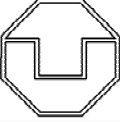


- **Identified roles and contracts based on what is needed to guarantee non-functional properties.**

- **Work in progress:**
 - How is this related to development time roles?
 - How can the contracts be formalized?
 - What do concrete component scheduling techniques look like? How can they be extended to non-time-based properties?



The End



Dresden
University of
Technology

