



# Tailor Made Containers: Modeling Non-Functional Middleware Service

**Ronald Aigner**, Christoph Pohl,  
Martin Pohlack, Steffen Zschaler

Technische Universität Dresden, Germany

12. October 2004, *Lisbon, Portugal*



- Components may have non-functional requirements, such as response time, reliability, service quality, ... (you name it)
- Requirements on other components and on middleware
- Middleware has to provide support for these components with non-functional requirements
- COMQUAD project ([www.comquad.org](http://www.comquad.org))  
Components with Quantitative Properties and Adaptivity



- Example Non-functional Property (Experiences with Realtime)
- Influence on Middleware Design
- Modeling Non-functional Properties

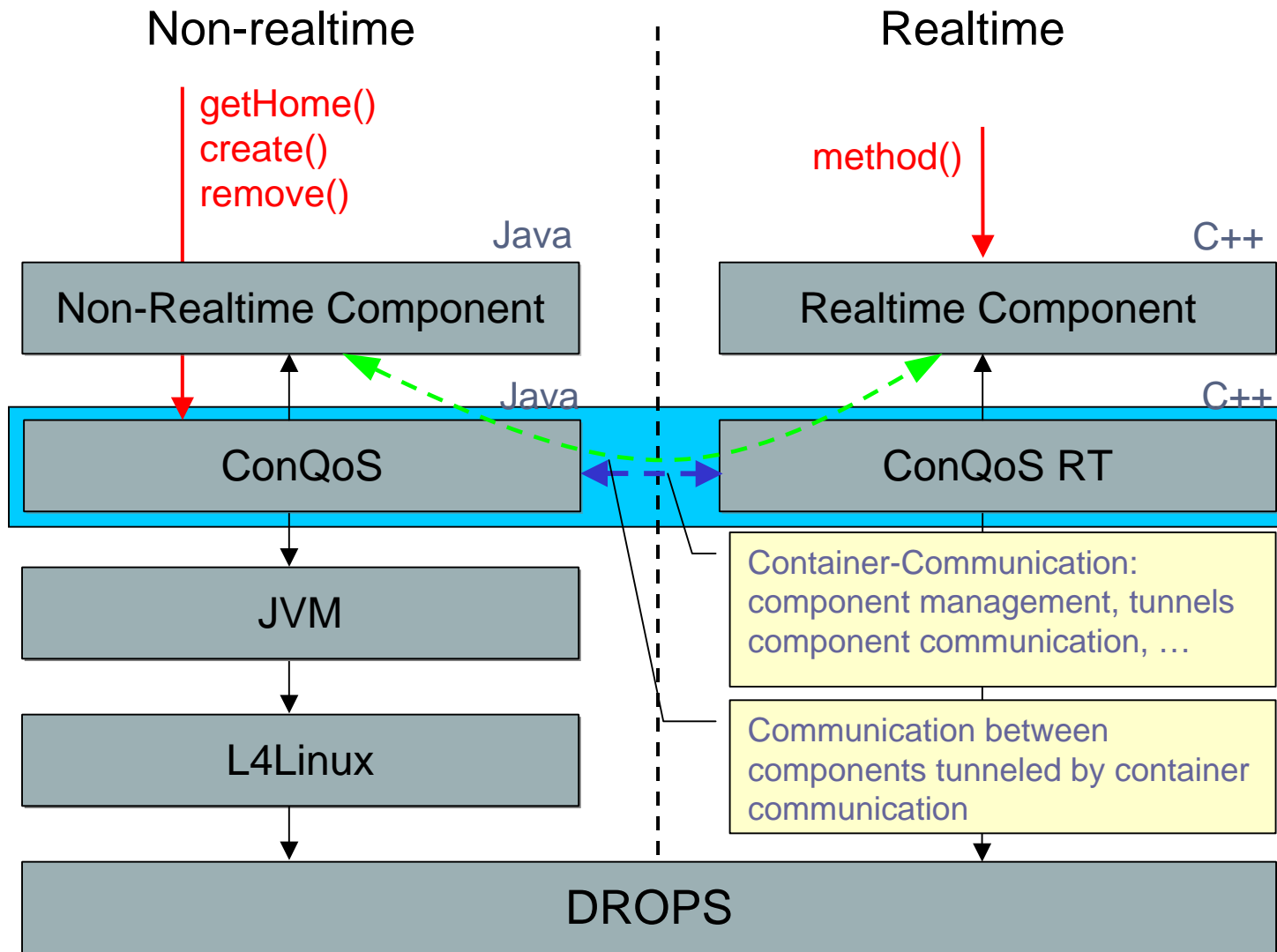


# The Realtime Property

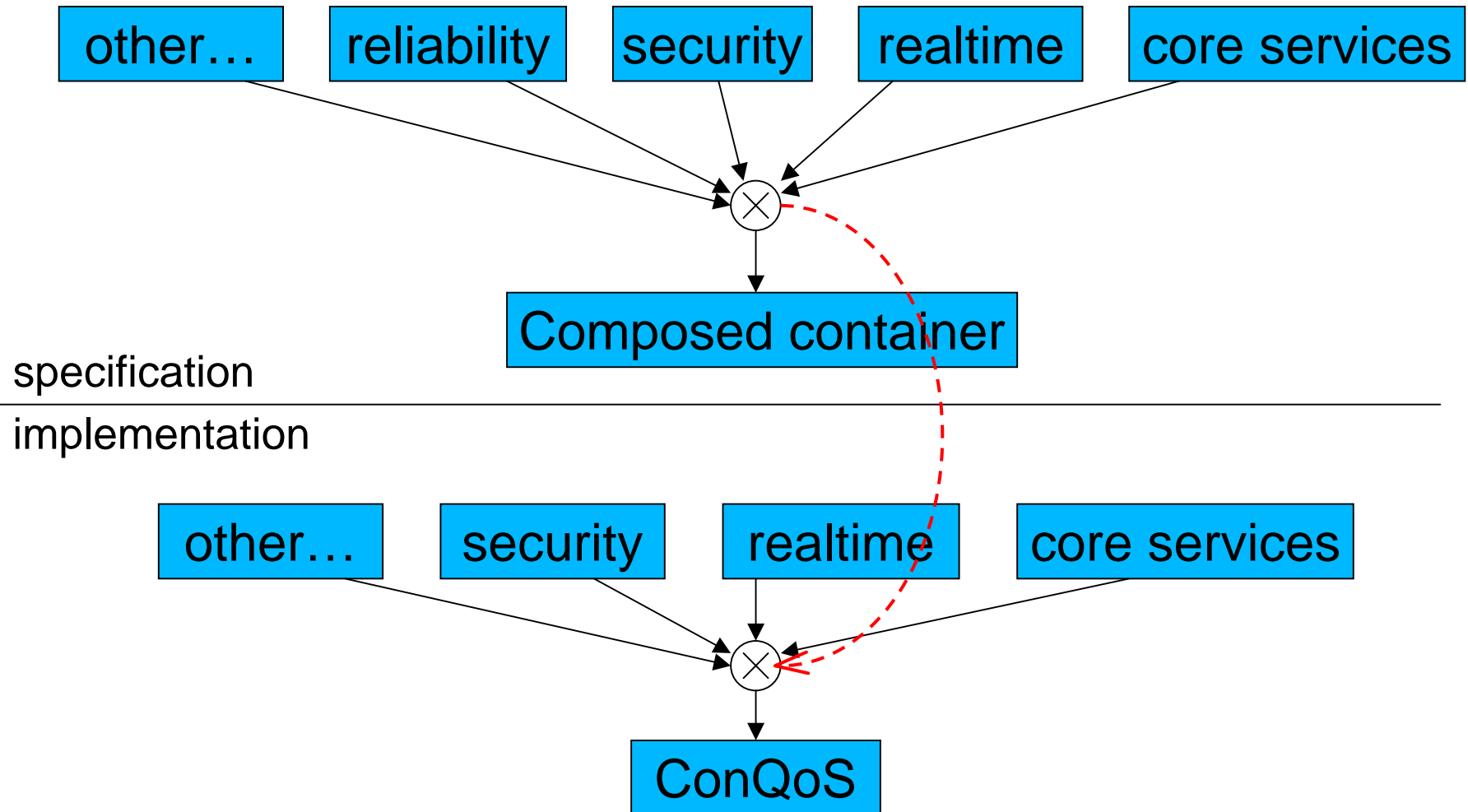
- Realtime is timeliness of execution
- Requires “timeliness” middleware – the container
- Different approaches:
  - Extend container with realtime support or
  - Split of container into
    - small realtime capable part and
    - big non-realtime capable part
  - Split looks most promising

- Features of a realtime capable container
  - Contract negotiation
  - Resource reservation:  
to guarantee timeliness access to resources
  - Timeliness execution of components:  
no unpredictable interference of other software
  - Timeliness communication:  
non-realtime components do not delay realtime components

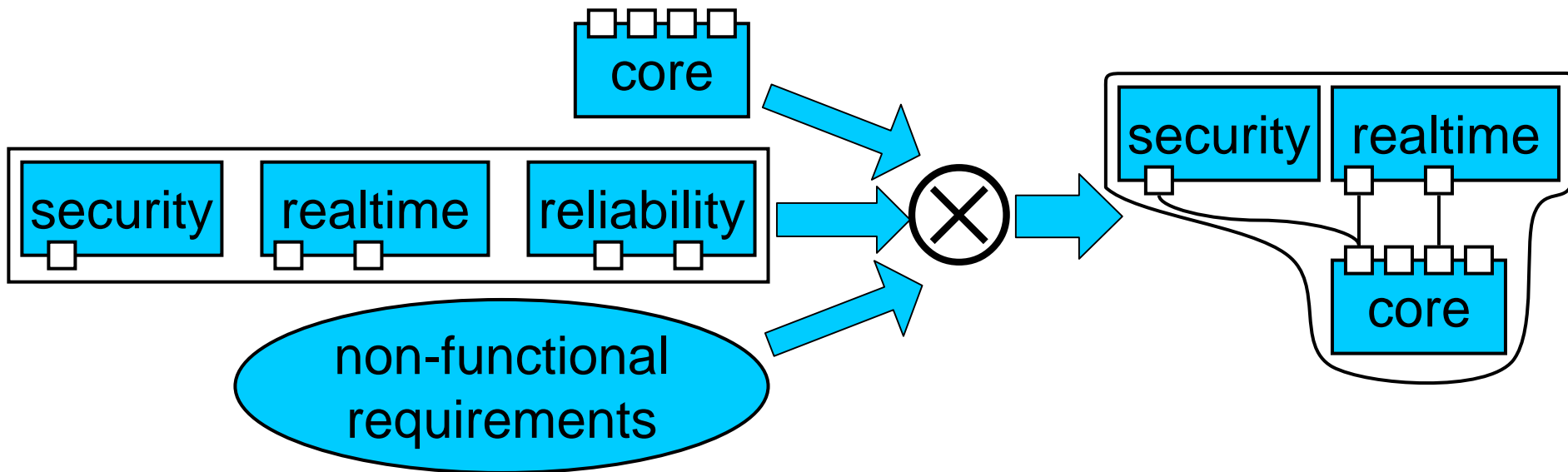
# Split Container



# From Split Container to Model



- Apply split for other non-functional properties as well





- Security: covert timing channels
    - Information transfer can only be avoided with **imprecise** clocks
  - Realtime:
    - Requires **precise** clocks to meet deadlines on time
- Negotiate clock precision: tradeoff between bandwidth of covert channel and precision of met deadlines

- **Pro:**
  - Container composed from modules
  - Time-Sources can be modeled as modules
  - To make time imprecise
    - wrap time-modules
- **Con:**
  - Too many time sources (CPU, sound card, network, ...)
  - Requires invasive manipulation of container and components
  - Modules are non-invasive
- **It's not working.**



# Modeling with Aspects

- Places to manipulate described by join points  
(time sources)
- Aspect denotes action  
(clock imprecision)
- Conditions when to apply aspect  
(when security is required)

- **Join points:**
  - At run-time realized with interceptors
  - At meta-level required by state changes  
→ special reflective support required
- **Pre-Conditions:**
  - Other required aspects
  - Required methods (realtime only works with periodic components → periodic work method)
- **Parameterized aspects:**
  - One aspect used at different join points with different parameters → consider at weaving time
  - Example: "response time" with different values



# Conclusion

- Model non-functional properties using componentized container
- Build container for a component from:
  - A set of services and
  - The component's (non-functional) requirements
  - Using operator.
- Model non-functional properties of components using aspects