



**João Américo and Walter Rudametkin**  
Bull S.A.S./LIG Grenoble

**Predictability vs. Dynamism:**  
managing dynamic real-time  
applications

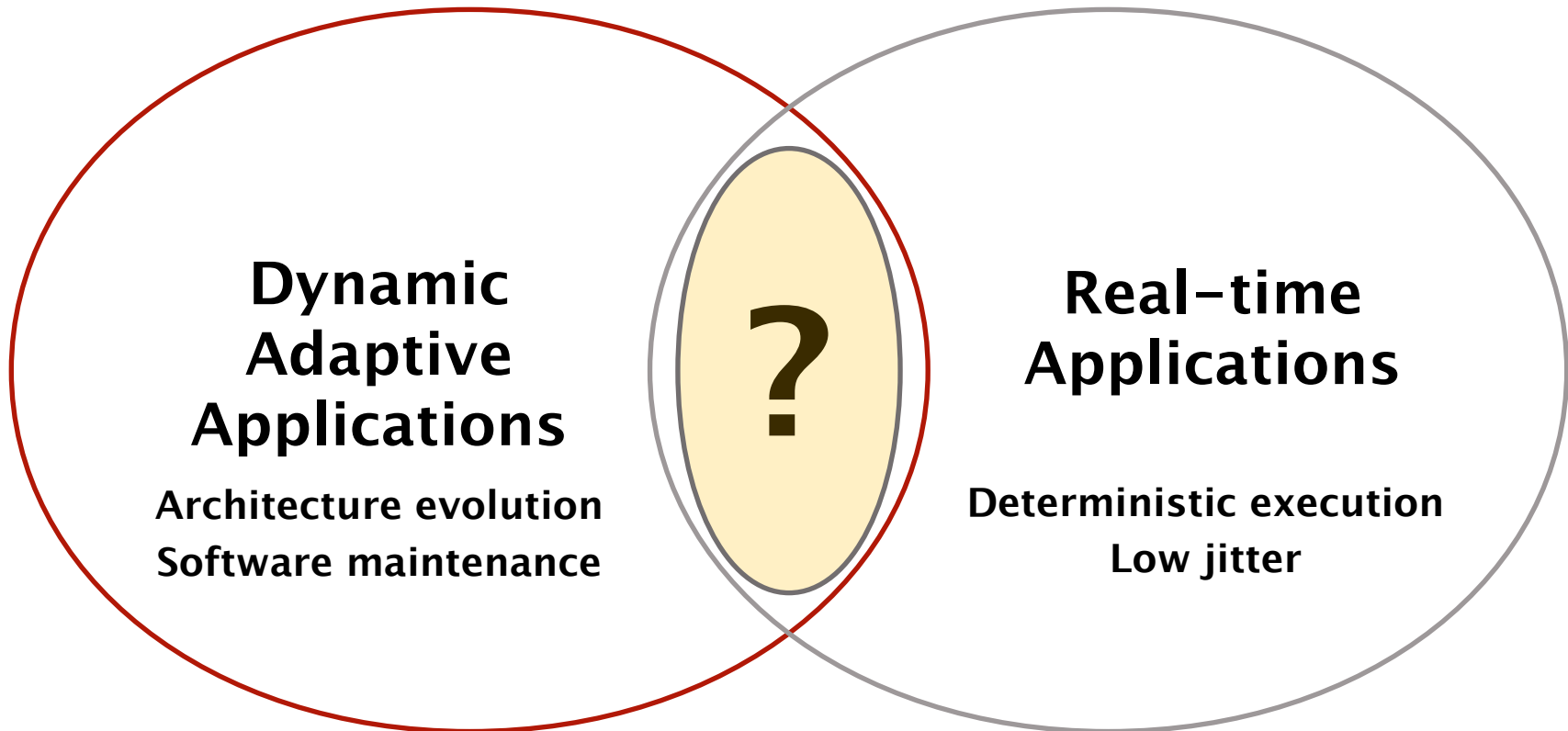
# Outline

- Context
- State-of-the-art
- Problem Identification
- Suggested Approach
- Limitations
- Conclusions and perspectives

# About

- **Walter RUDAMETKIN**
  - PhD student at Université de Grenoble
- **João AMÉRICO**
  - PhD student at Université de Grenoble
  - BSc at UFRGS (2010), MSc at Université Joseph Fourier (2010), Engineer Degree at ENSIMAG (2009)

# Context



# State-of-the-art

## RTSJ: Real-time Specification for Java

- Issues: garbage collection, dynamic class loading, thread scheduling, etc.

## Dynamic Evolution/Adaptation

- Architecture modification at runtime

## Real-time dynamic adaptive software

- Based on QoS objects (QoSKeys), modes (SOFA-HI/Blue-ArX), and real-time adaptations for CCM (CIAO, Cardamom).

# State-of-the-art

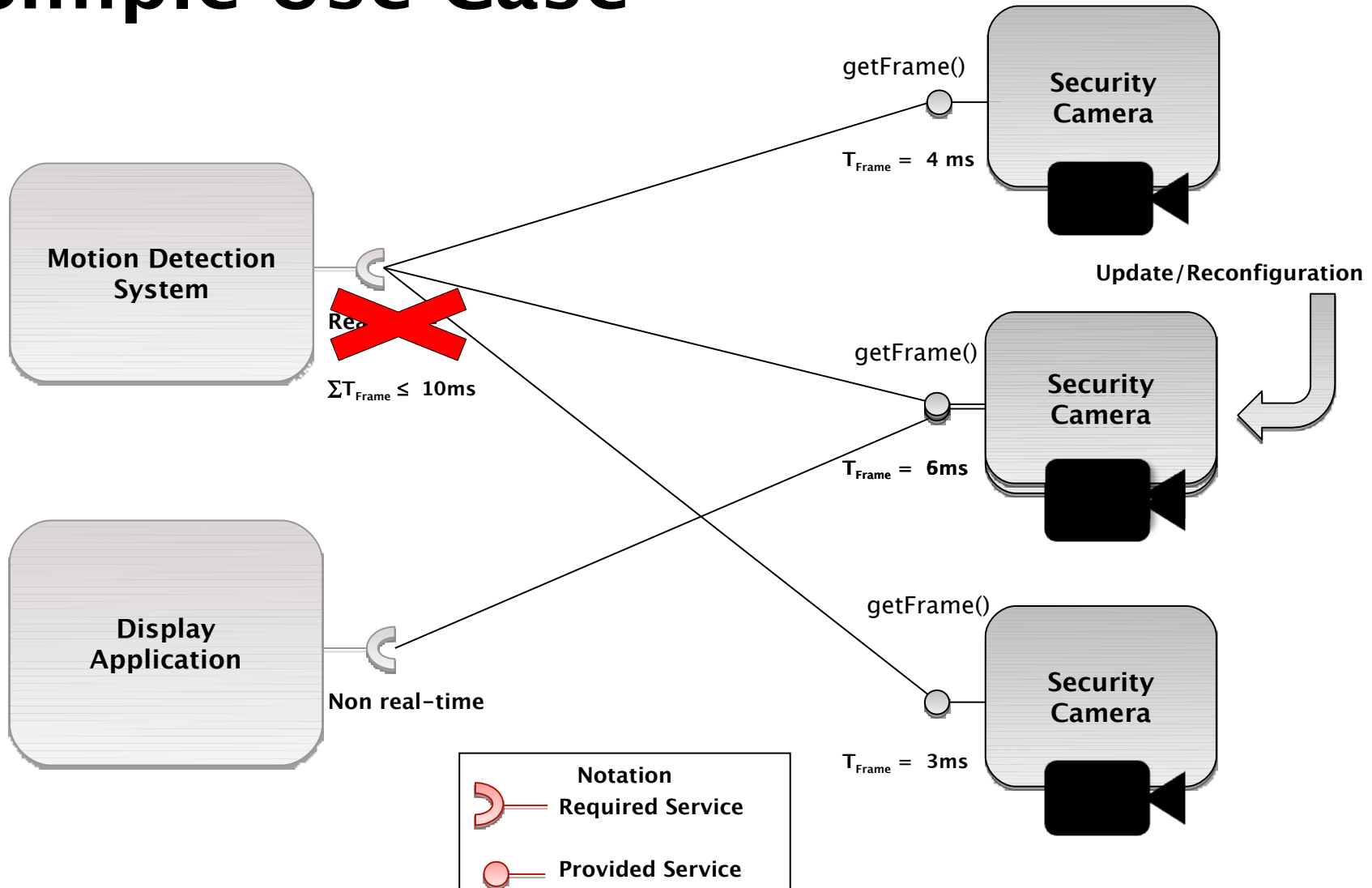
## Real-time OSGi

- Works focused mainly on isolation issues: ARFLEX Project, [Richardson, 2009], AONIX's Real-time OSGi model
- Industry initiatives: Oracle/BEA's WebLogic Real-time, Integration between Perc and mBS

# Problem Identification

- OSGi platform is inappropriate for real-time applications
  - Memory issues
  - Scheduling issues
  - Isolation issues
  - **Runtime software evolution**

# Simple Use Case

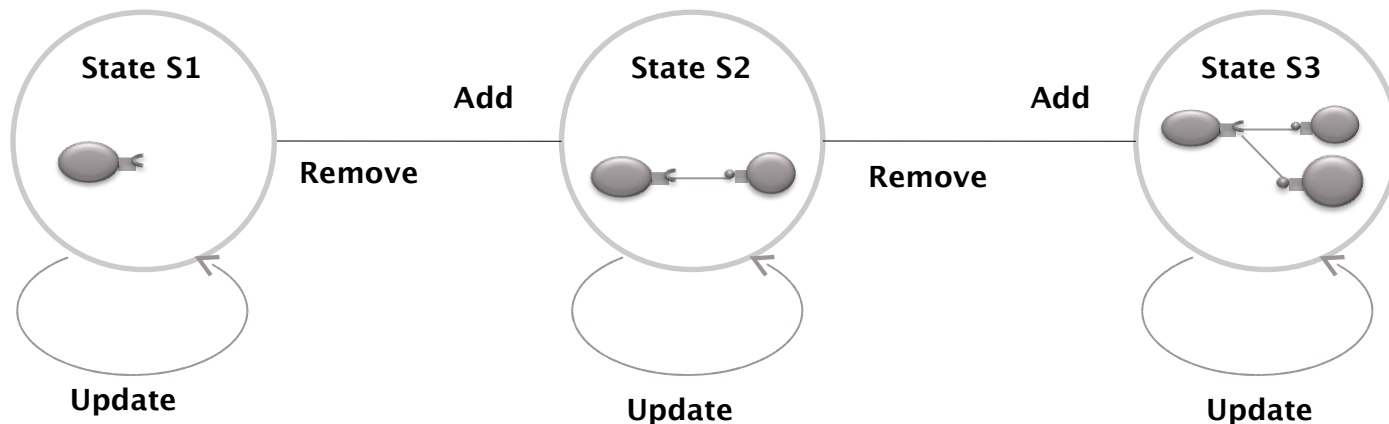


# Suggested Approach

- Distinction between critical and non-critical code
  - Architecture freezing policy
  - Dynamic Real-time SLA

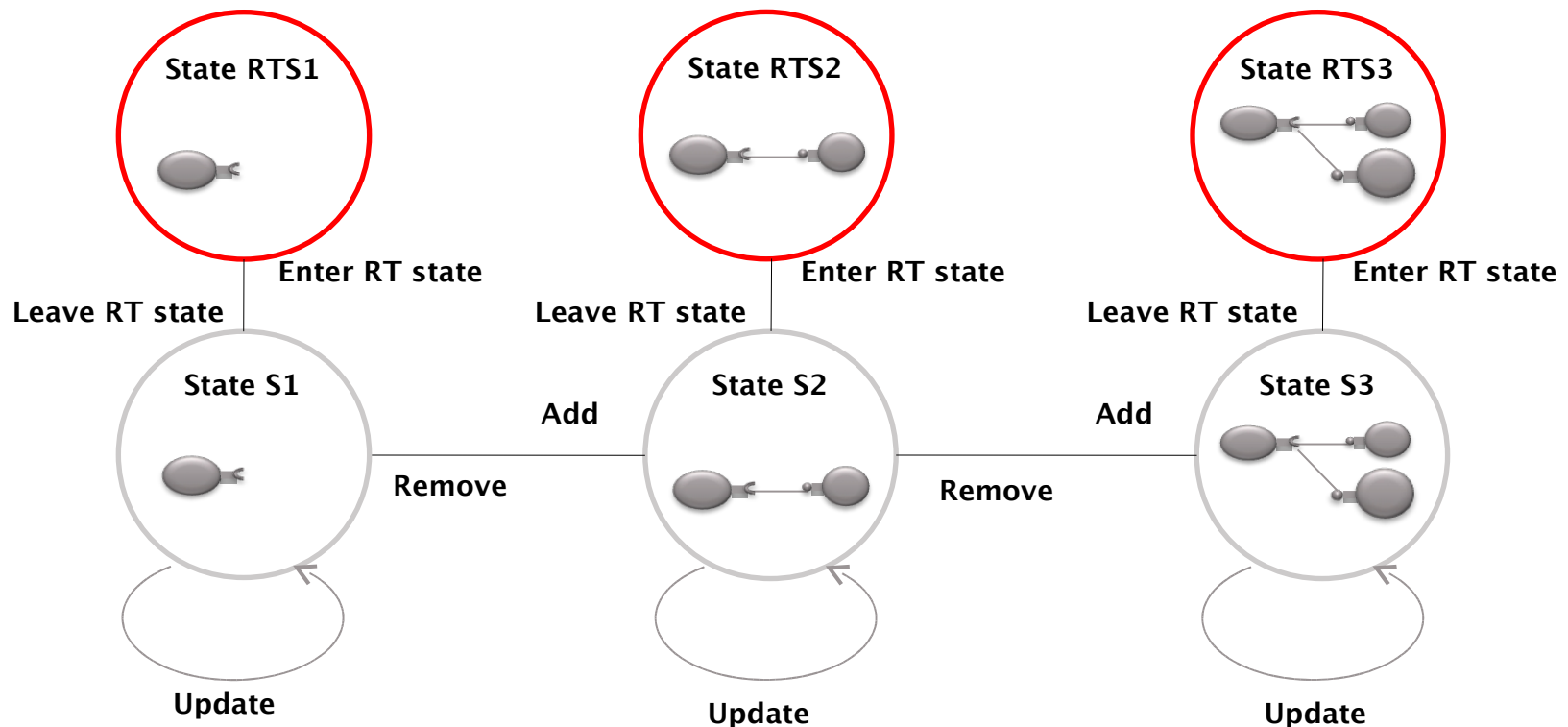
# Architecture Freezing

- Application = set of states
  - Each state corresponds to an architecture (service bindings)

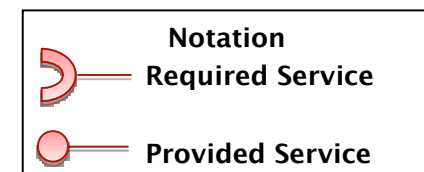
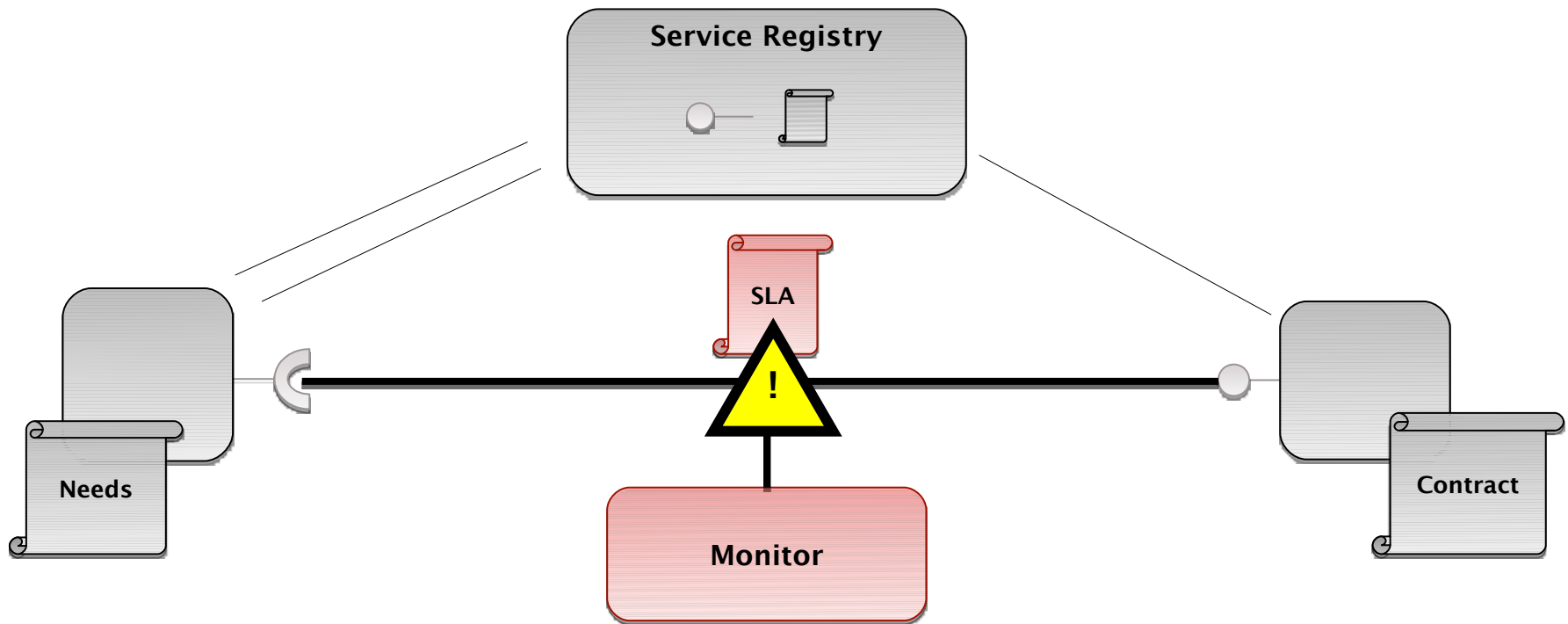


# Architecture Freezing

- Real-time processing states
  - Architecture modifications forbidden



# Service Level Agreement

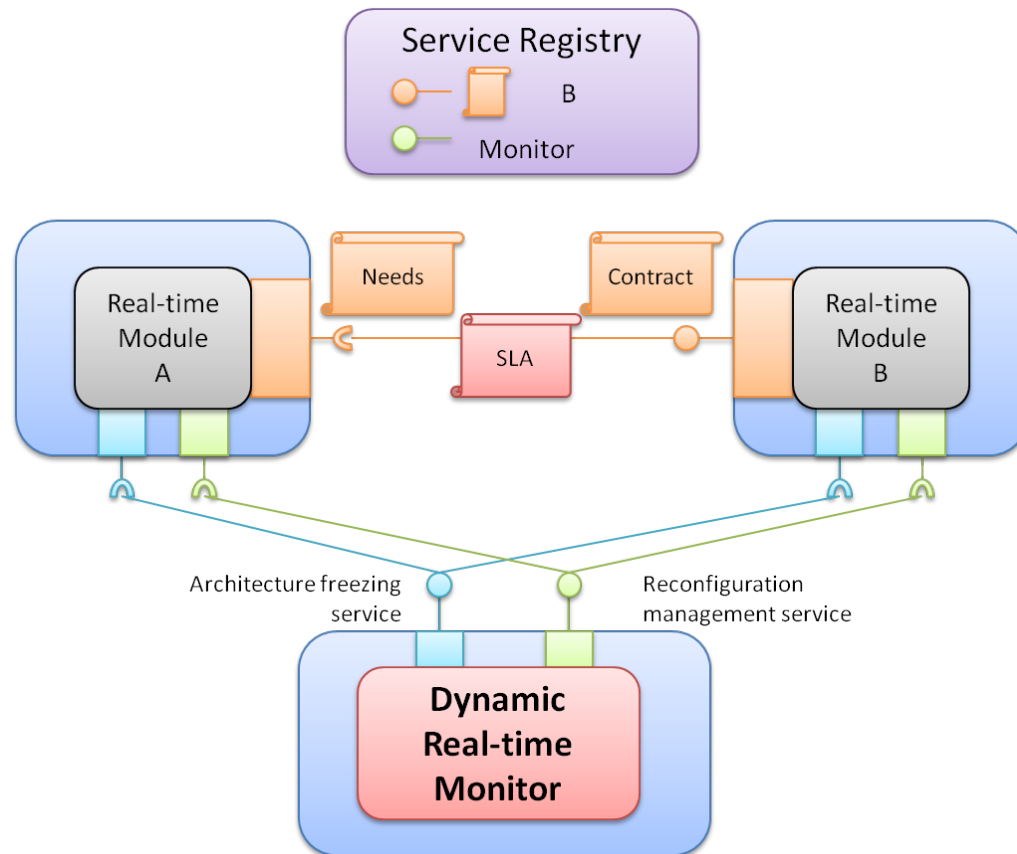


# Real-Time Dynamic SLA

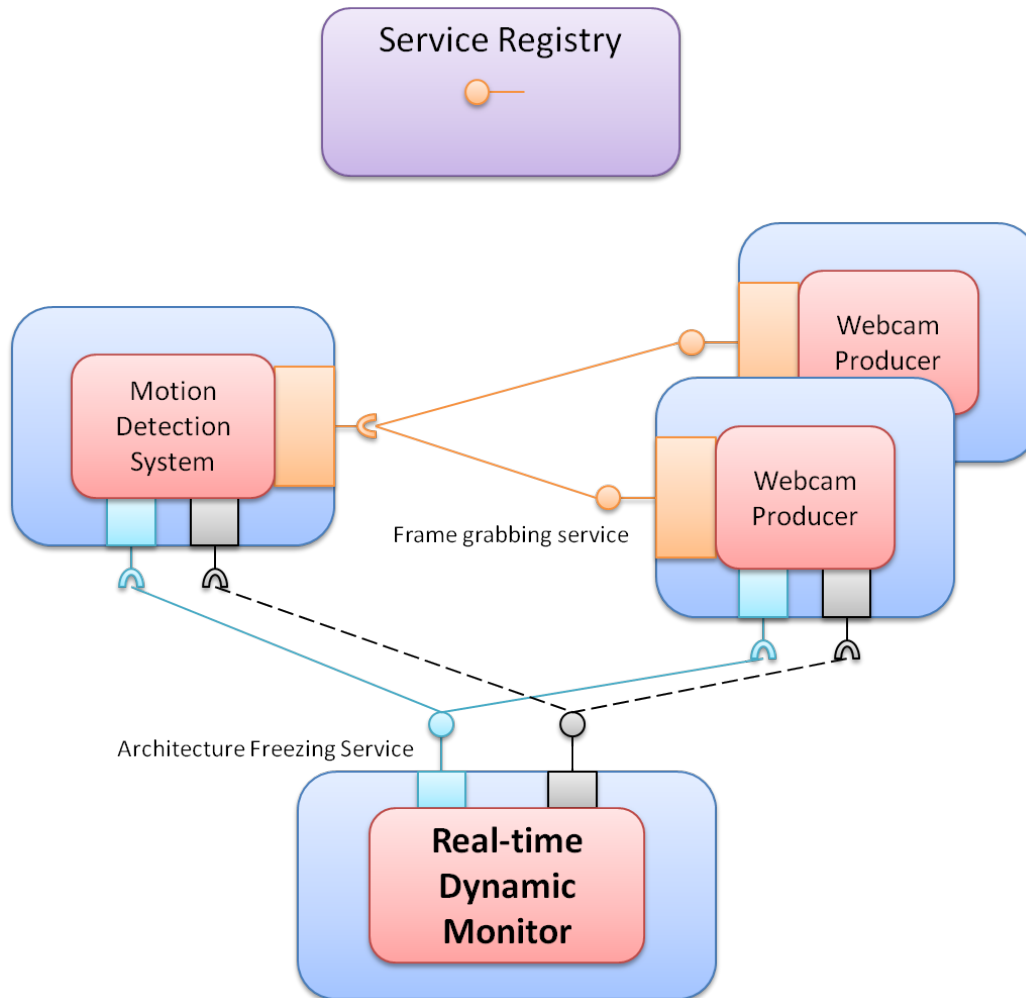
- Extension to the D-SLA model [Touseau, 2010]
  - Task type
  - Period
  - Worst case execution time (WCET)
  - Resource Utilization
  - Priority

# Implementation

- iPOJO component model extension



# Validation



- Architectures frozen during real-time processing states
- SLM not implemented

# Limitations

- One real-time application at a time
- Unknown update times
- Component characterization
  - Resource utilization measures
- Overhead

# Results

- Architectural Freezing solves:
  - Dynamic update
  - Service interruptions
    - but not disappearance of physical devices
- Dynamic RT–SLA solves:
  - Service admission
    - based on resource consumption, deadlines, ...
- Both require modifying apps (explicit notifications)

Contact: {Joao.Americo, Walter.Rudametkin}@imag.fr

**THANK YOU FOR YOUR  
ATTENTION!**