Year 2 Review
Paris, November 8th and 9th, 2006

Scientific Highlights

Flexible scheduling framework

Michael González Harbour
Universidad de Cantabria, Spain

General assumptions of real-time theory

WCETs are measured and enforced

All timing requirements are hard

Industry is familiar with the details of real-time theory

Industry has real-time analysis tools integrated into their design processes

Operating systems provide adequate scheduling services

Is real-time theory useful?

What industry does in reality

- **no WCET estimation**
- **maximum use of the available resources**
- **no protection or fault detection due to added complexity**
- **no real-time analysis**
  - **timing requirements "proven" by testing**
  - **"develop and hope for the best" methodology**
  - **hard real-time analysis is too pessimistic**

Real-time scheduling theory regarded as "the solution to the wrong problem"

Yes: real-time theory is useful!

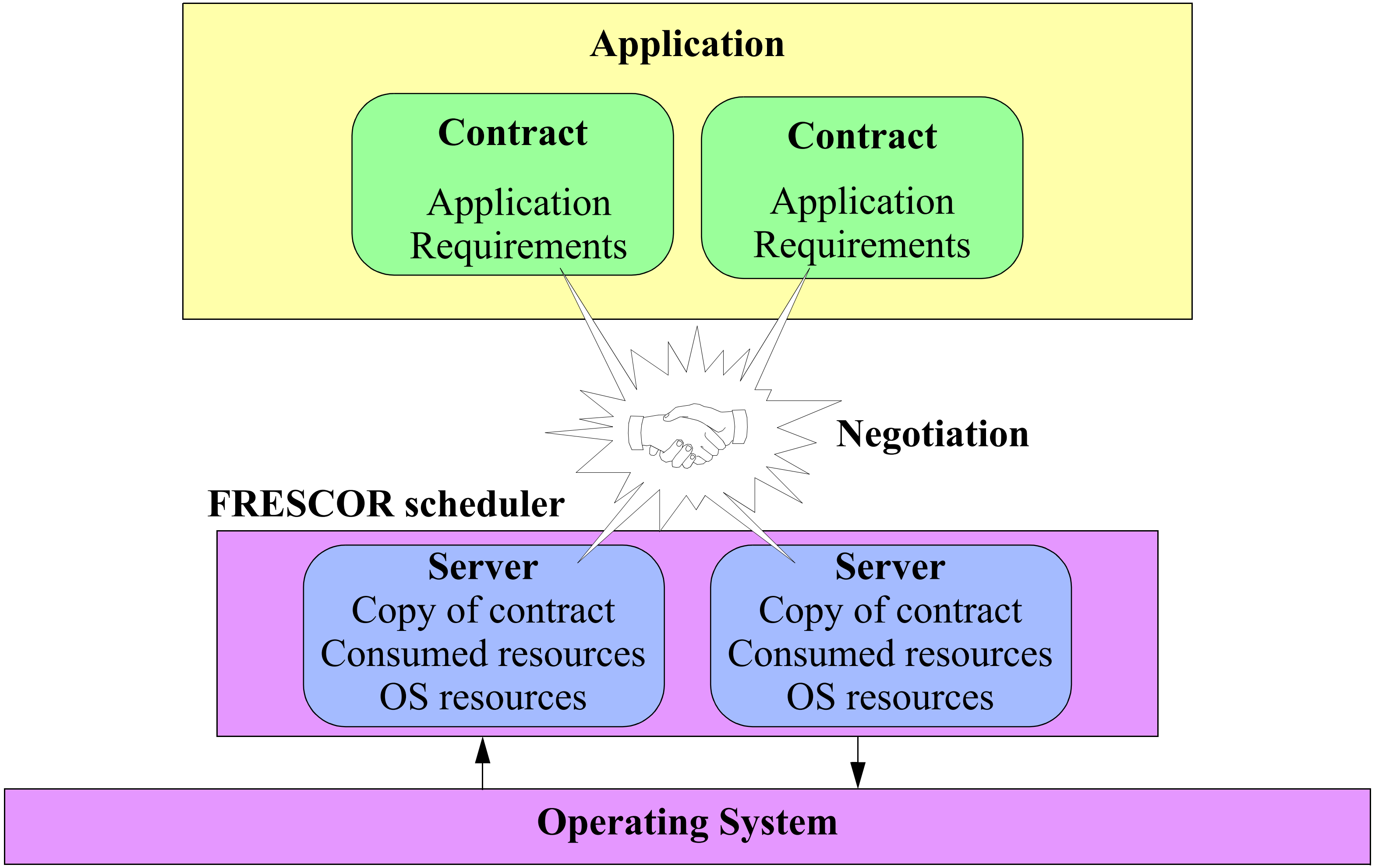
What we think:

- real-time scheduling theory is the right solution to the problem,
- but needs proper abstractions
- and needs to be integrated into the design process

What we propose

- application developer:
  - “tell me what you need”
- system:
  - tells you if the minimum requirements can be guaranteed
  - and distributes spare capacity to maximize quality

Solution: Service Contracts



Application

Contract

Application Requirements

Contract

Application Requirements

Negotiation

FRESCOR scheduler

Server

Copy of contract

Consumed resources

OS resources

Server

Copy of contract

Consumed resources

OS resources

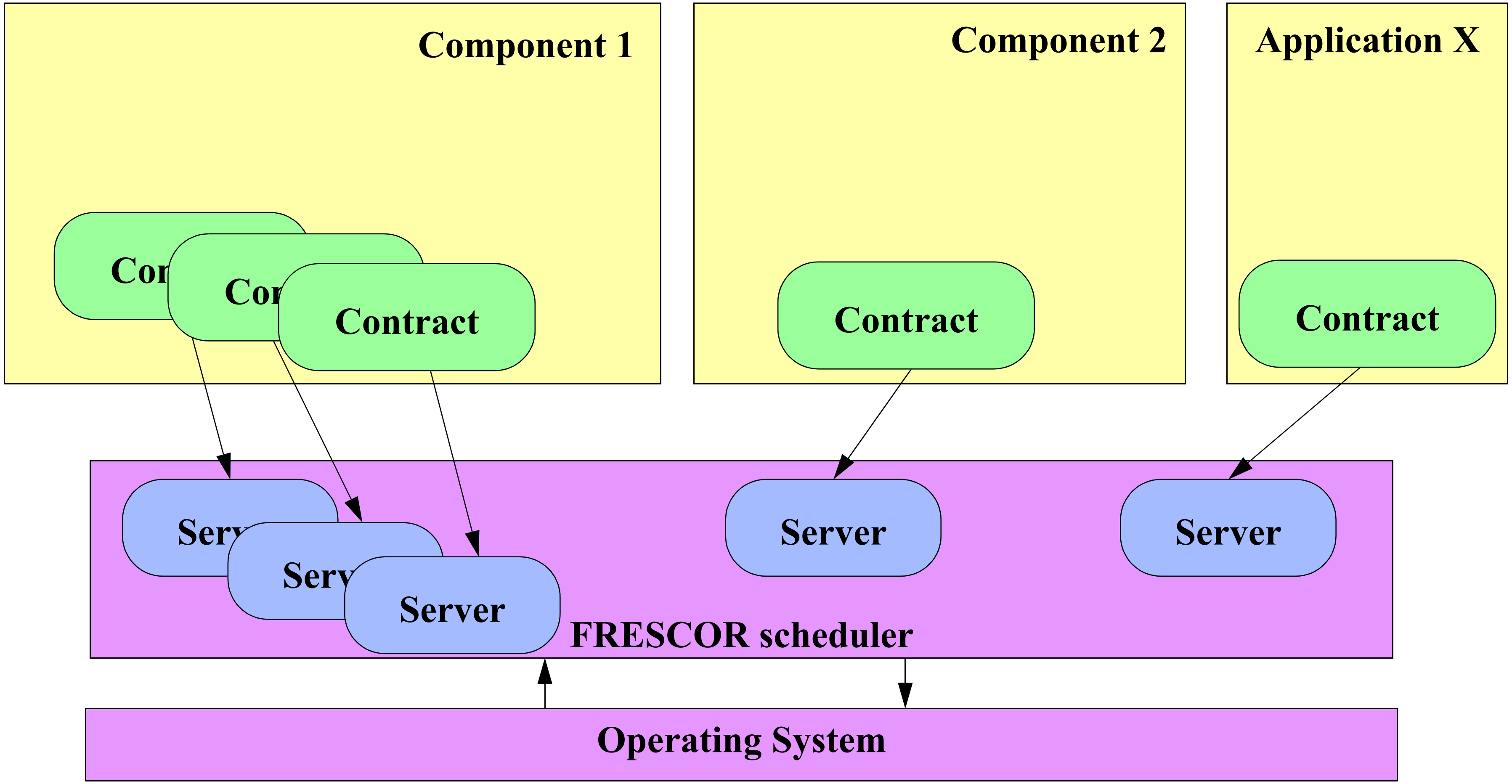
Operating System

Service Contracts (Cont'd)

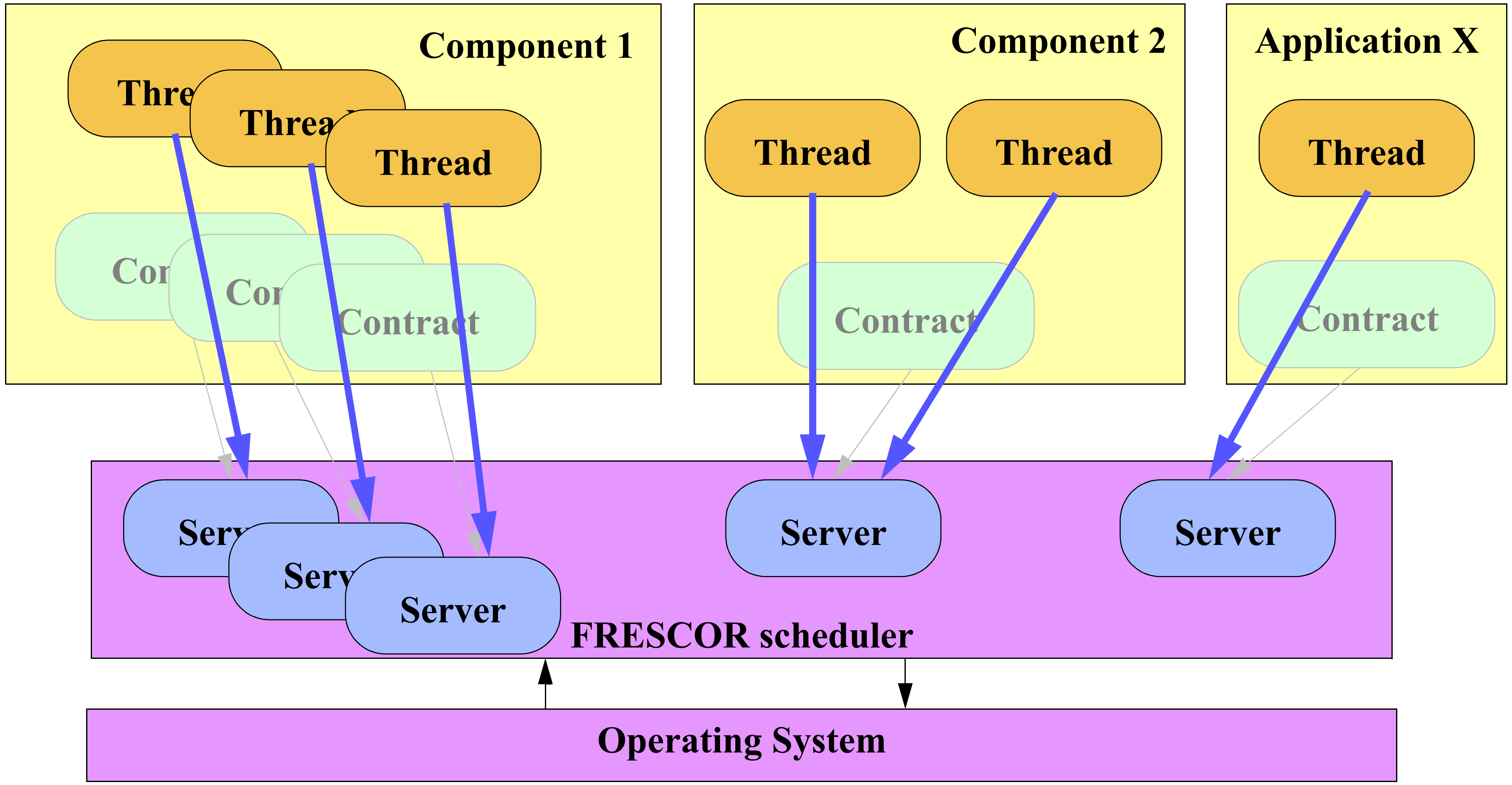
Contract-based server scheduling

- Contract specifies:
  - minimum requirements
  - how to make use of any spare capacity
- Online or offline acceptance test
- Spare resources are distributed according to *importance* and *weight*
  - global optimization policy (maximize available energy, maximize resource usage, ...)
- Renegotiation possible
- Independent of underlying scheduler

The application model: negotiation



The application model: binding of threads to servers



Objectives of the flexible-scheduling framework

- Contract model that specifies application requirements
  - required to be guaranteed
  - usable to increase quality of service
- Underlying implementation manages & enforces contracts
- Integrated resources
  - processors (single, multi, reconfigurable hardware, ...)
  - networks (general purpose, fieldbuses, wireless,...)
  - disk, buses
  - energy, memory, shared resources
- Adaptive QoS Manager
- Distributed transaction manager

Objectives of the flexible-scheduling framework (cont'd)

- Performance analysis via simulation
- Component-based framework bridges the gap with design methods
  - tools allow independent analysis
  - tools calculate contract parameters
  - tools obtain timing properties of the overall system
- Usable on different application domains
  - raise the level of abstraction
  - platform independent usage

Relation with FRESCOR IST project

ARTIST2 contributed to creating the critical mass to set up FRESCOR

- Shared objectives with flexible scheduling technologies activity

Role of ARTIST2 activity:

- Bring together a wide body of expertise to set the requirements for the flexible scheduling framework
- Evaluate the usefulness and applicability of a contract-based scheduling framework
- Contribute with new theoretical developments that can be used in the framework implementation
- Dissemination of results

Benefits from ability to influence FRESCOR and exploit its results