

# An Aspect-Oriented Implementation Method with support to Progressive Implementation

Sérgio Soares and Paulo Borba

Software Productivity Group

Informatics Center - Federal University of Pernambuco - Brazil

{scbs,phmb}@cin.ufpe.br

http://www.cin.ufpe.br/spg

## Characteristics:

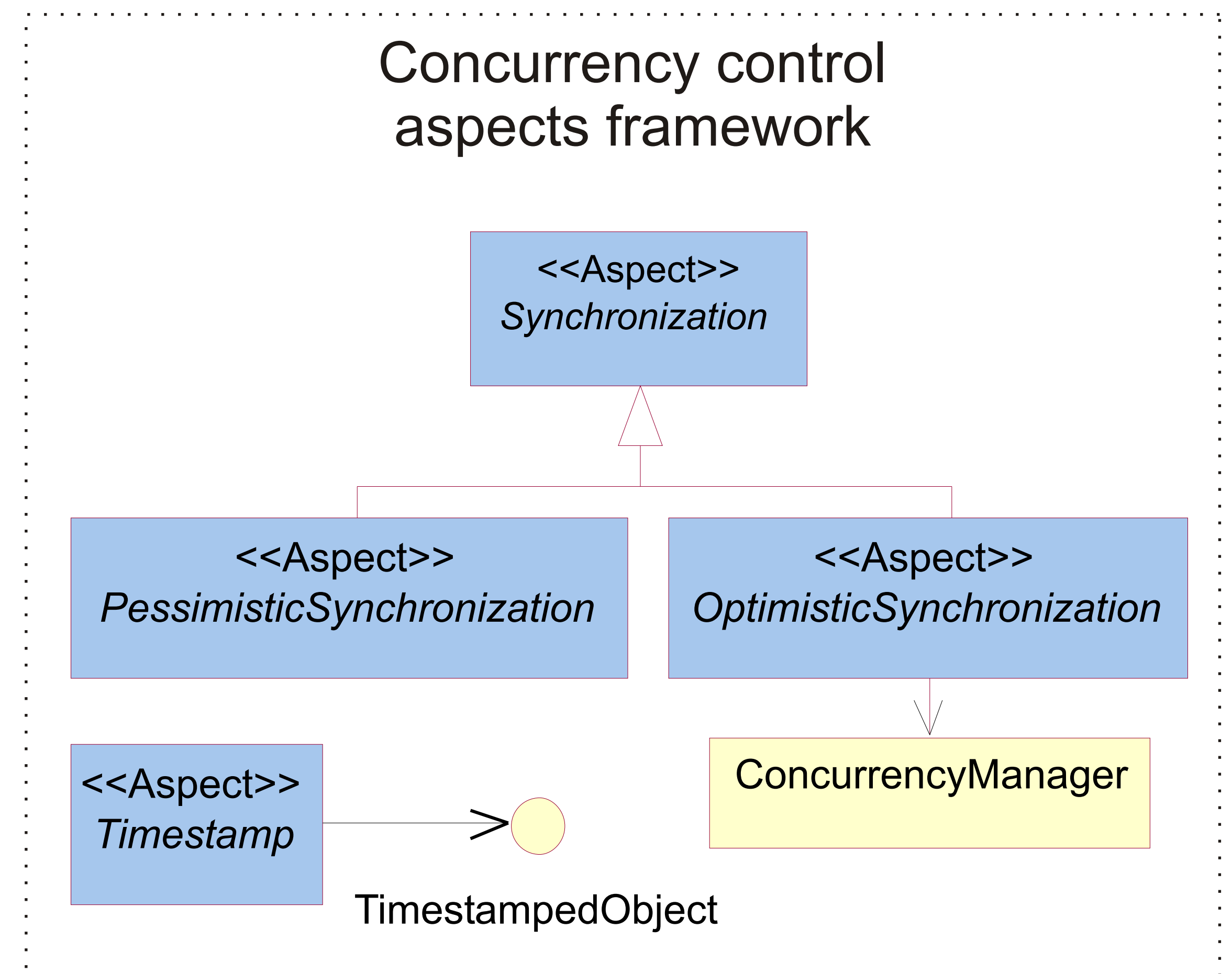
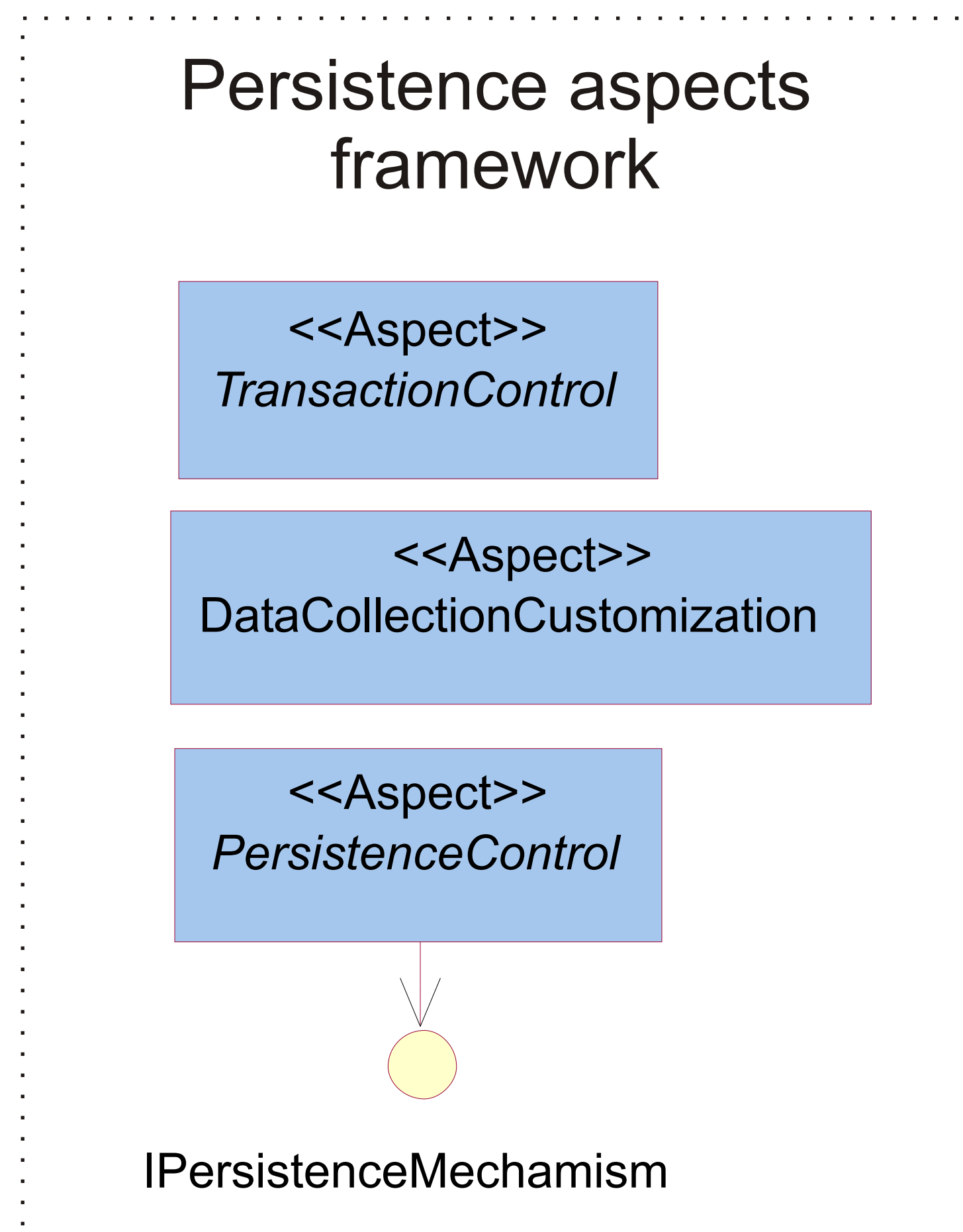
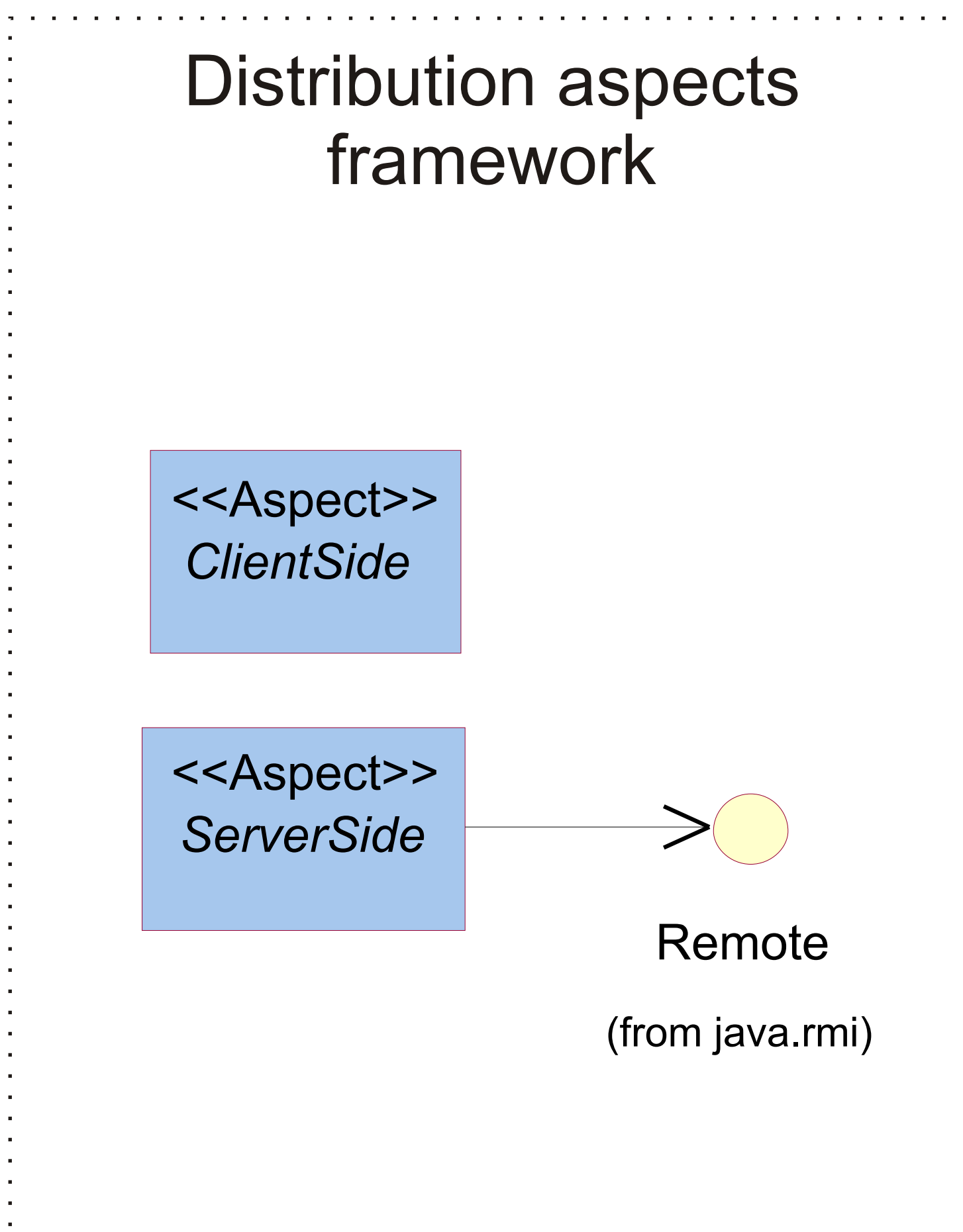
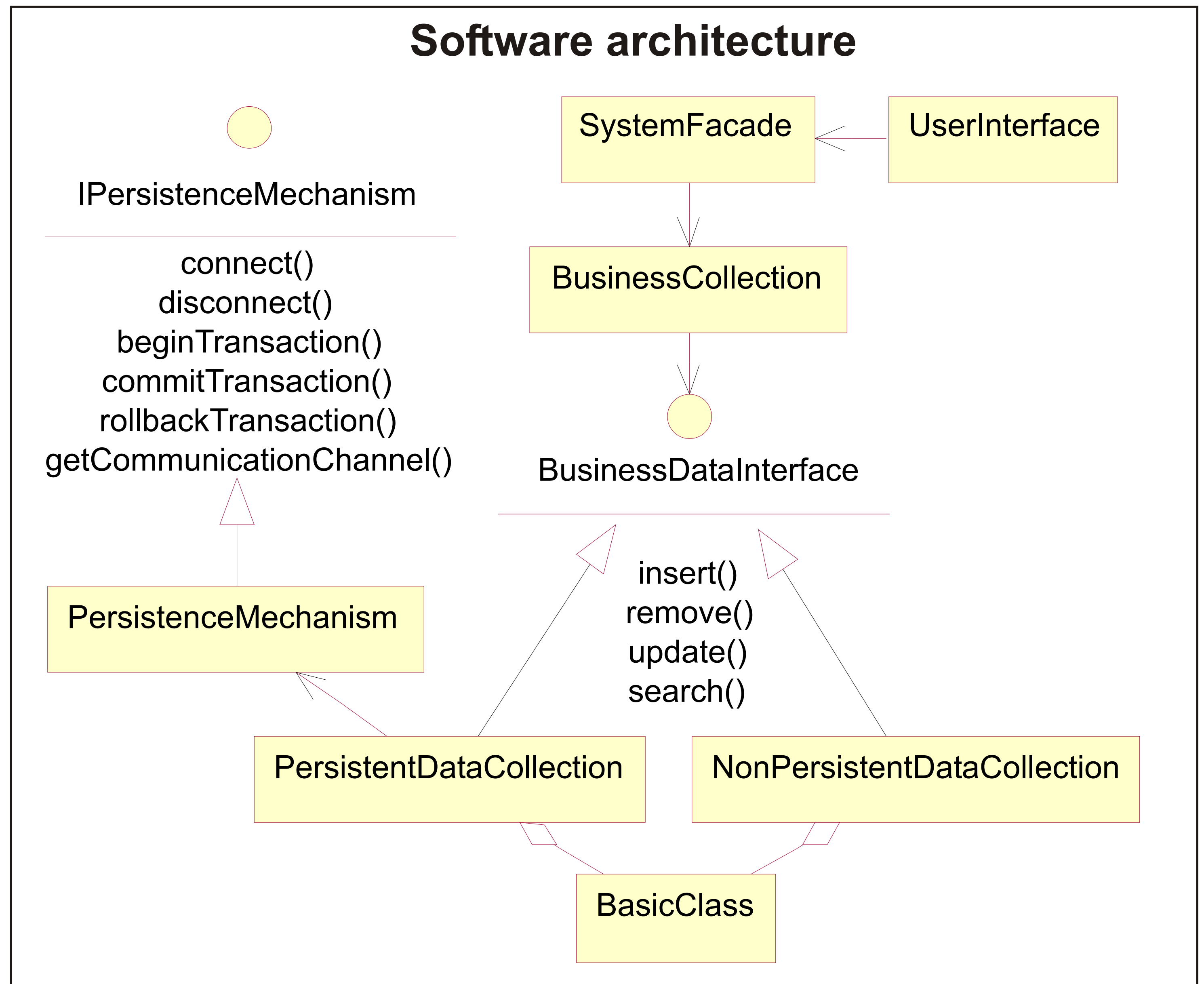
- Implementation activities
- Modifications in analysis, design, and tests activities
- Patterns for structuring the system architecture
- Aspects frameworks to implement crosscutting concerns:
  - Persistence, distribution, and concurrency control
- Integration with Use Case Driven Development

## Benefits:

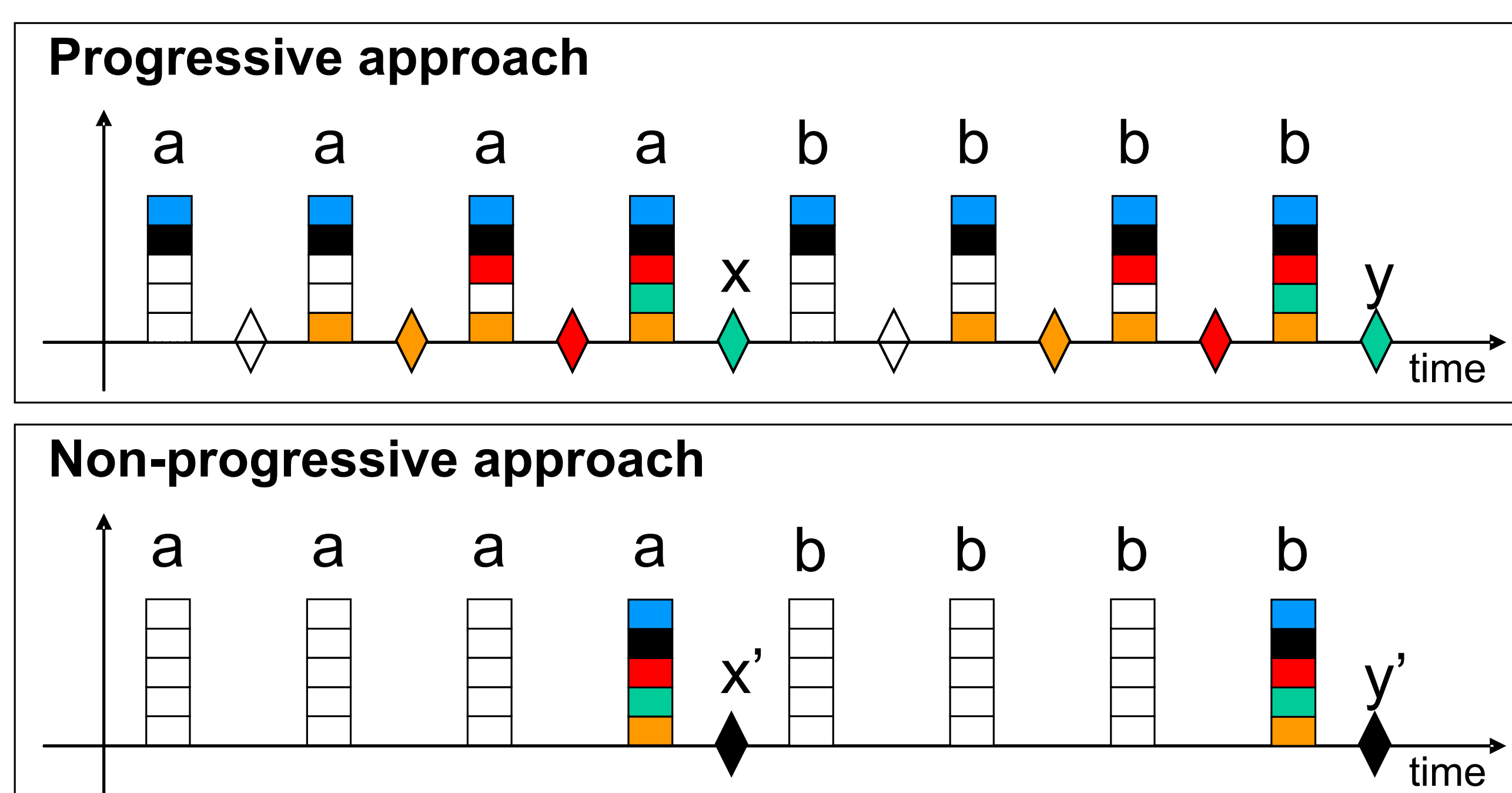
- Guarantees the effort given to requirements and design
- Guidelines to aspect-oriented development
- Guidelines to aspect-oriented refactoring
- Two implementation approaches
  - Experimentation for analysis
- Tool support
  - Aspect and classes generation
  - Increased productivity

## Drawbacks:

- Interferences between aspects
- Tailored to a specific software architecture
  - Allows implementing several kinds of systems
  - Used in several real systems
  - More precise guidelines

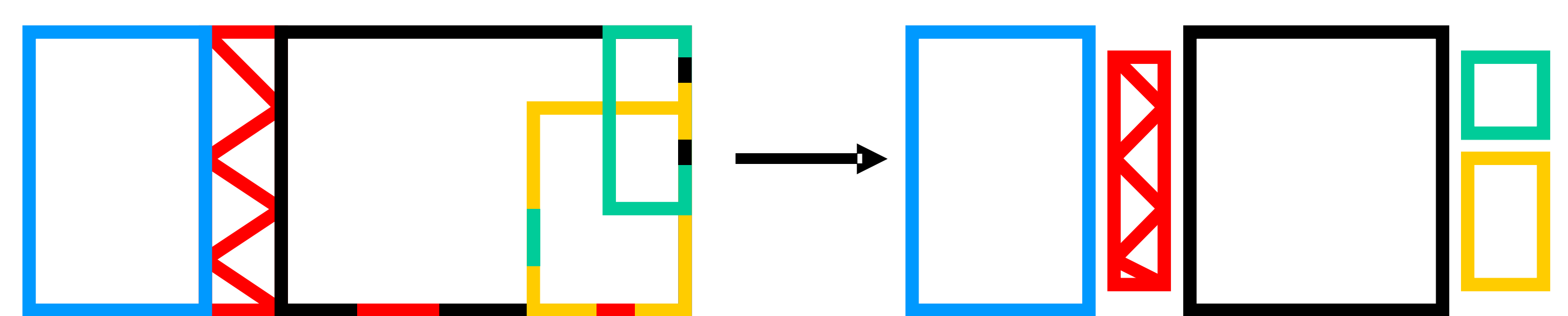


## Aspect-oriented development



The implementation method is integrated to Use-case Driven Development in two different approaches

## Aspect-oriented refactoring



- Functional requirements
  - Non-persistent data management
  - Persistent data management
  - ◆ Milestone (end of iteration)
  - Concurrency control
  - User interface
  - Distribution
- a and b are use-cases or sets of use-cases

The implementation method can be used to refactor an object-oriented application to an aspect-oriented one