

# Spontaneous Service Integration in Pervasive Environments

Noha Ibrahim

CITI - INSA Lyon  
INRIA AMAZONES

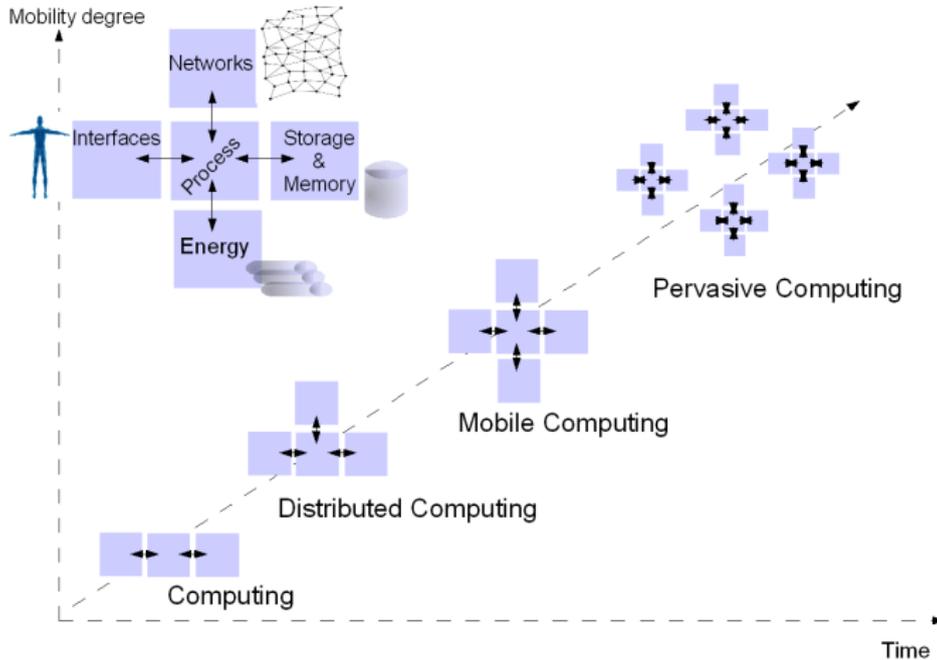
September 23, 2008

- Part I: Motivations & State of the Art
- Part II: Thesis Contributions
  - Unified Vision for the Service Integration: The SIM
  - Functional & Non Functional Service Integration Relations
  - Spontaneous Service Integration
- Part III: Conclusions & Perspectives

## Part I

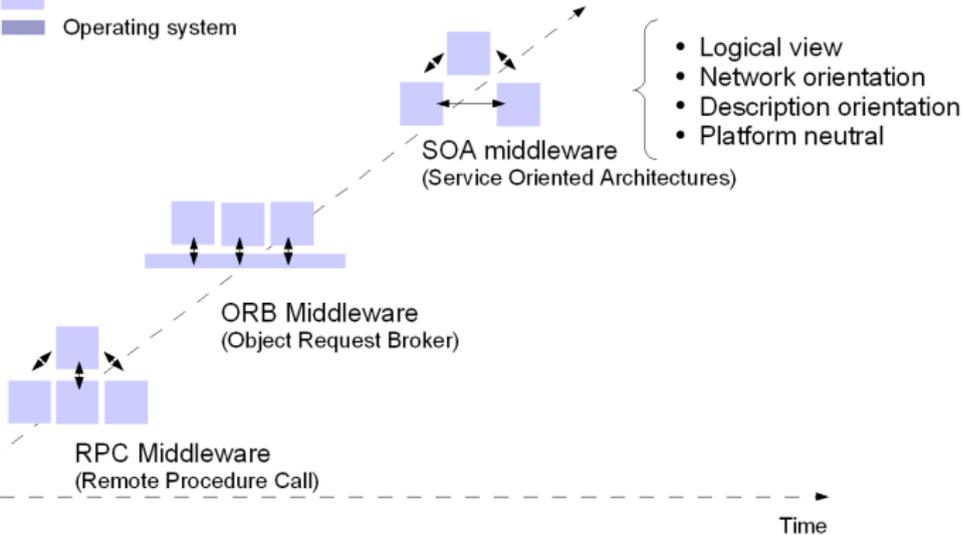
# Motivations and State of the Art

# From Mobile to Pervasive..



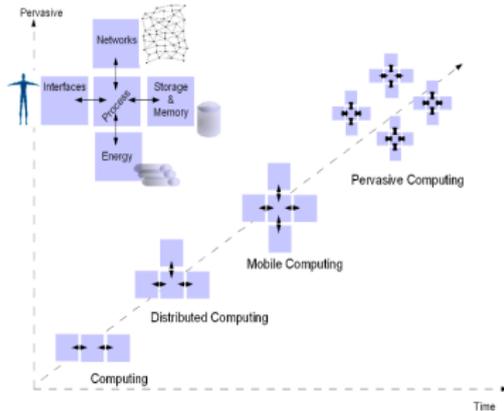
# From RPC to SOA..

Middleware evolution

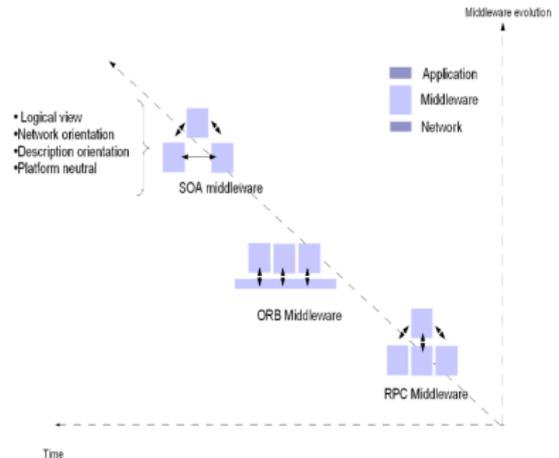


# Pervasive Meeting SOA

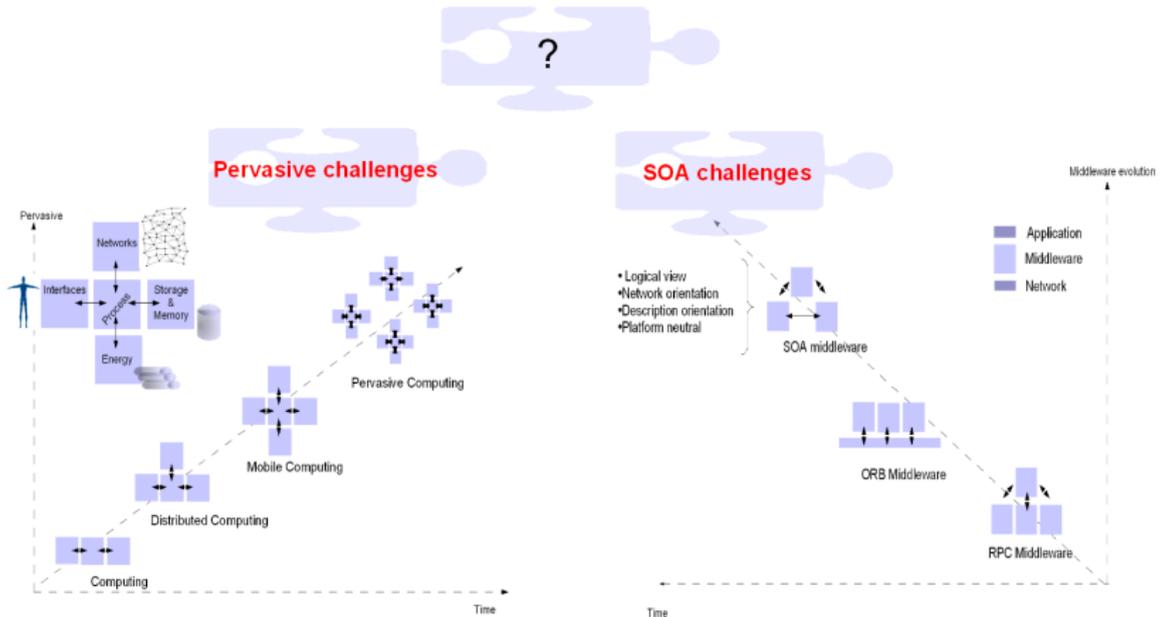
## Pervasive challenges



## SOA challenges

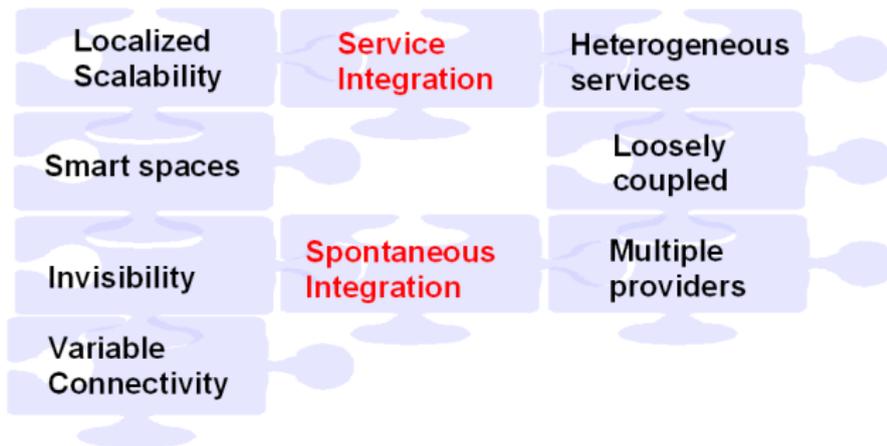


# Pervasive Meeting SOA



# Problem Statement: Spontaneous Service Integration

Pervasive challenges



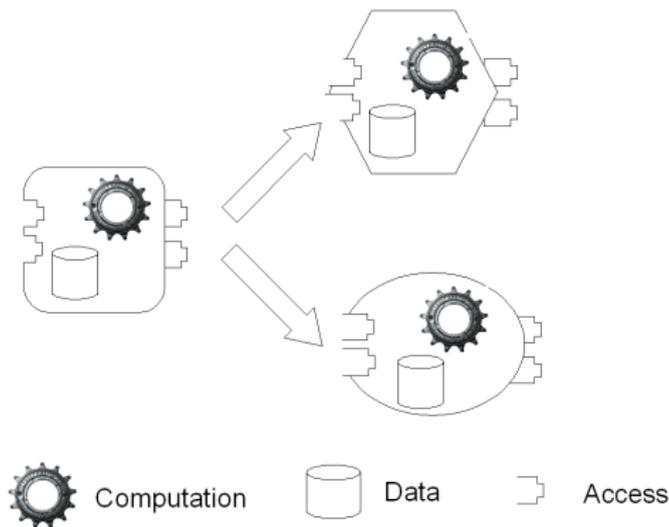
SOA challenges

# Service Integration

- Service transformation
- Service composition
- Service adaptation

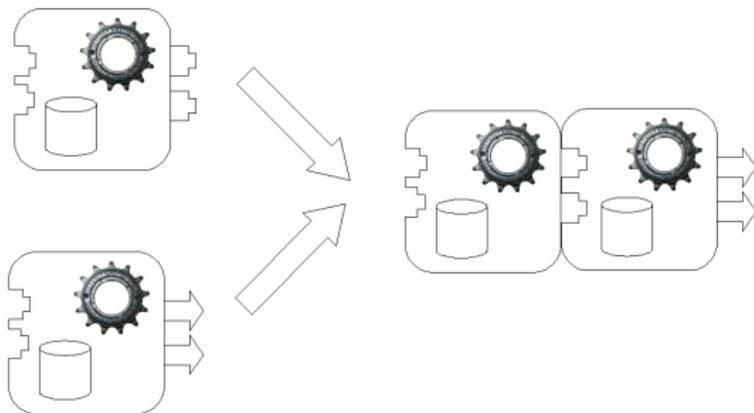
# Service Integration

Service **transformation** is the process or result of changing from one appearance, state, or phase to another



# Service Integration

Service **composition** is the process of binding two or more entities into a new one



Computation



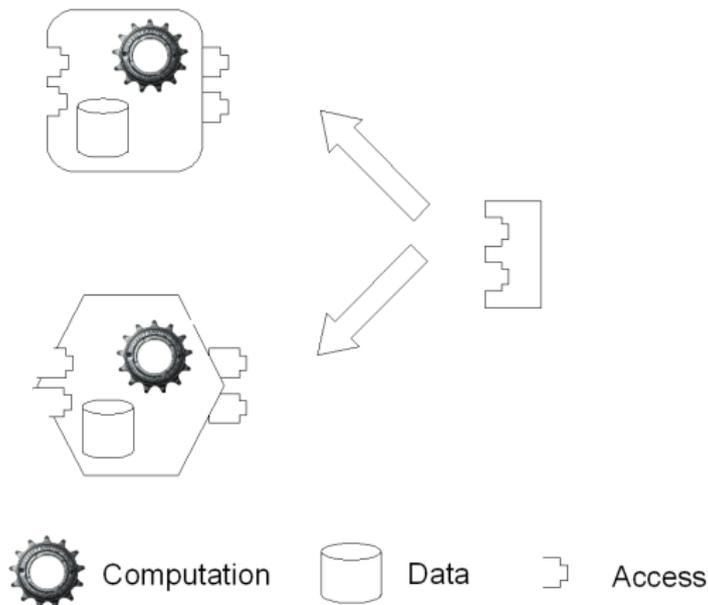
Data



Access

# Service Integration

Service **adaptation** is the process of making adjustments to suit the environment and to adjust to different conditions



## Goal-Oriented versus...

Goal-Oriented Integration: decided by applications and/or users



The middleware is reactive to the application and/or users needs

## ...Spontaneous

Spontaneous Integration: decided by the middleware for the applications and/or users



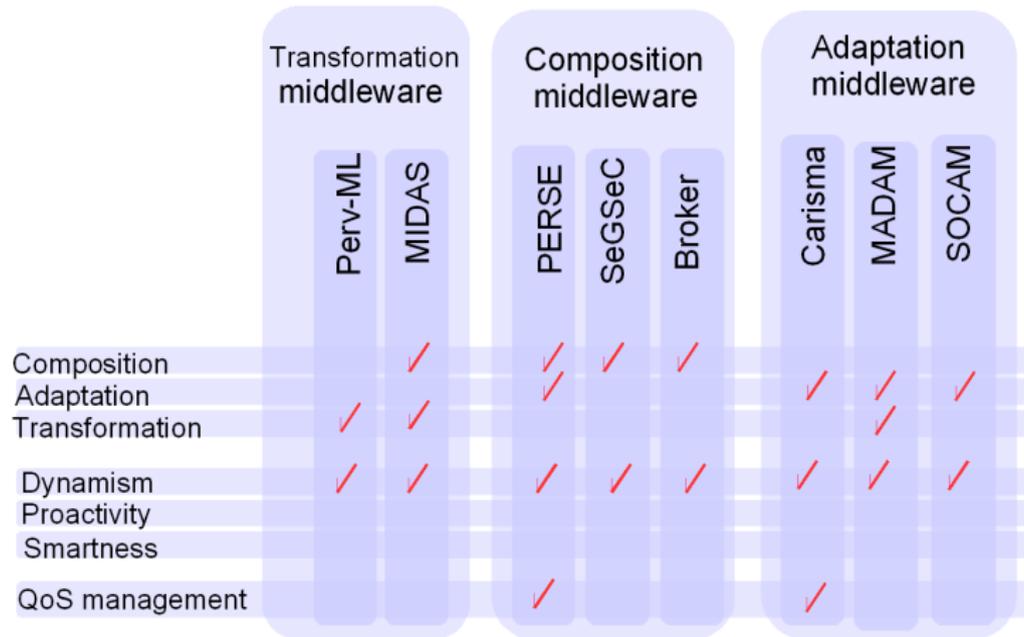
Transparent for applications and users, the middleware is proactive

# State of the Art

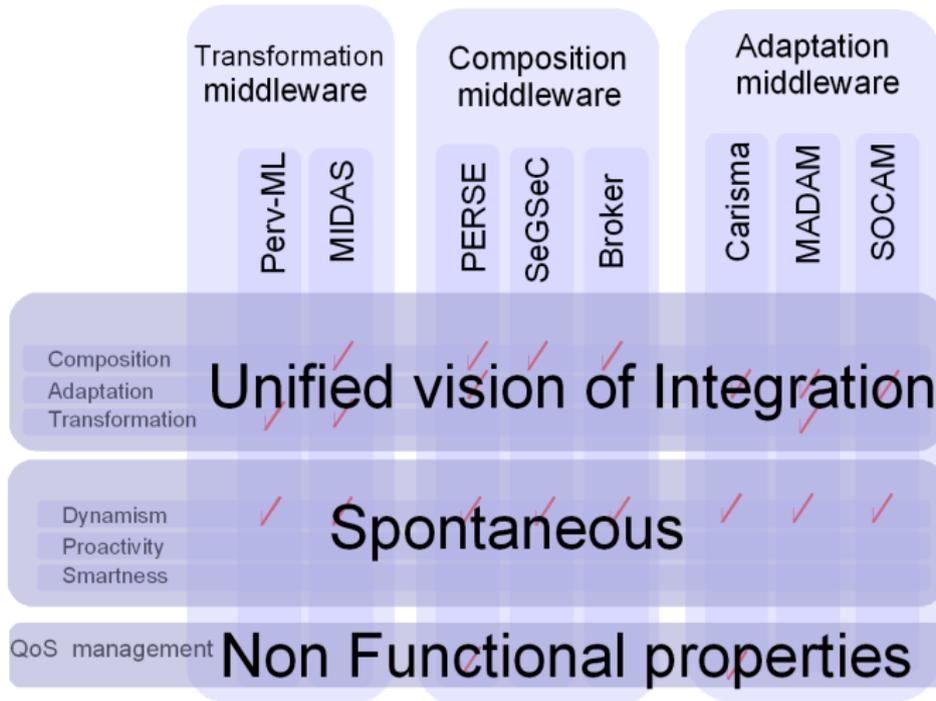
# Service Integration Middleware

- Service Transformation Middleware are MDA based approach: MIDAS[2006], Perv-ML[2004]
- Service Composition Middleware are goal-oriented based: Perse[2007], SeGSeC[2005], Broker[2005]
- Service Adaptation Middleware are based on the disappearance of services: MADAM[2006], Carisma[2003], Socam[2004]

# Classification & Issues



# Classification & Issues



## Part II

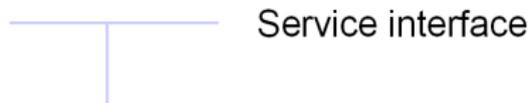
# Thesis Contributions

# Threefold Thesis Contribution

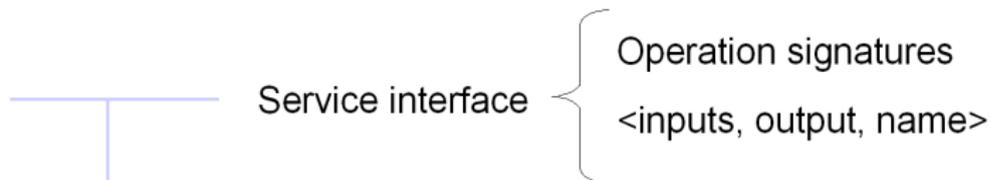
- Unified vision for the service integration. Proposing a middleware model that provides the necessary modules for service transformation, composition and adaptation: The SIM Middleware Model
- **Functional & Non Functional Service Integration Relations**
- **Spontaneous Service Integration for Pervasive Environments**

# Service Model

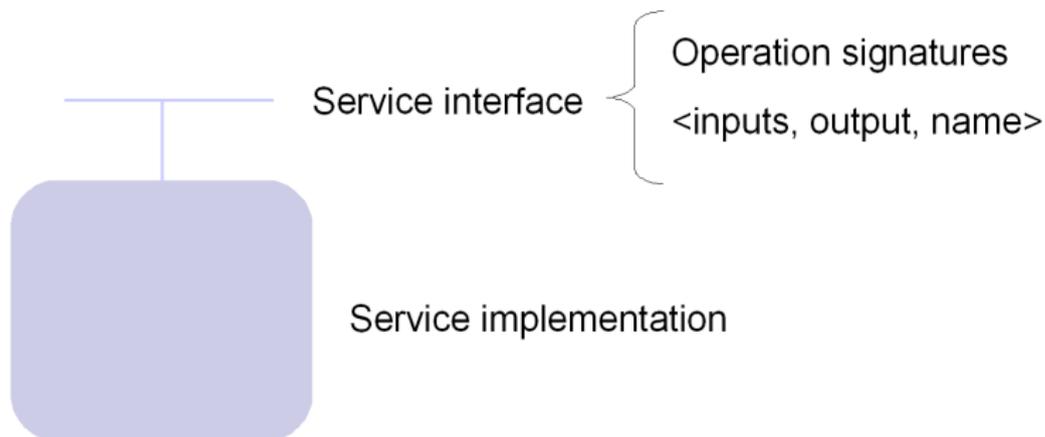
# Service Model



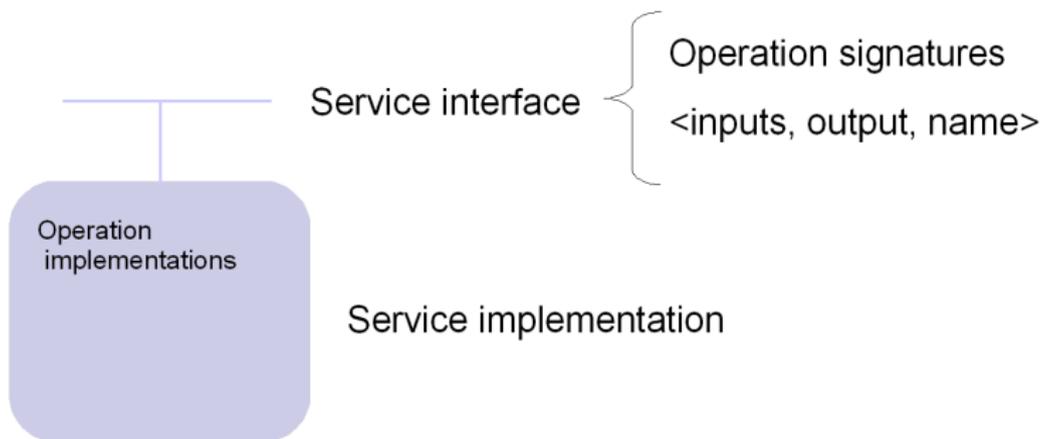
# Service Model



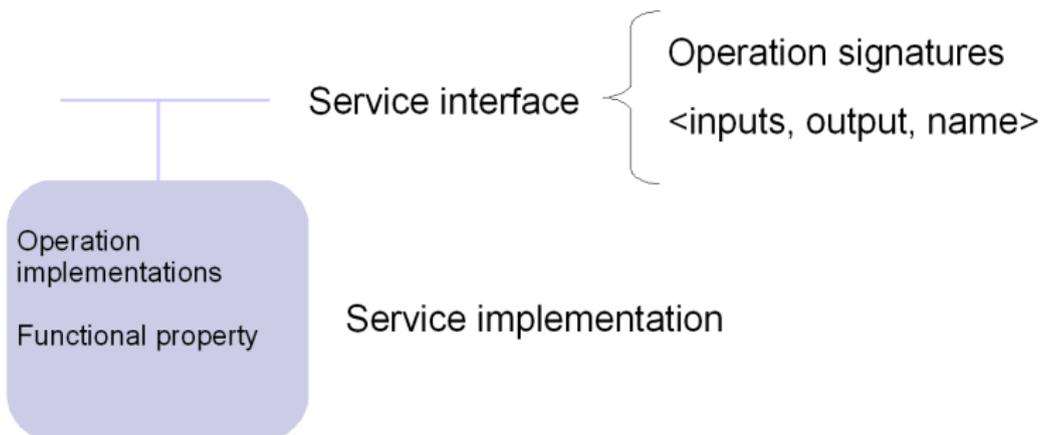
# Service Model



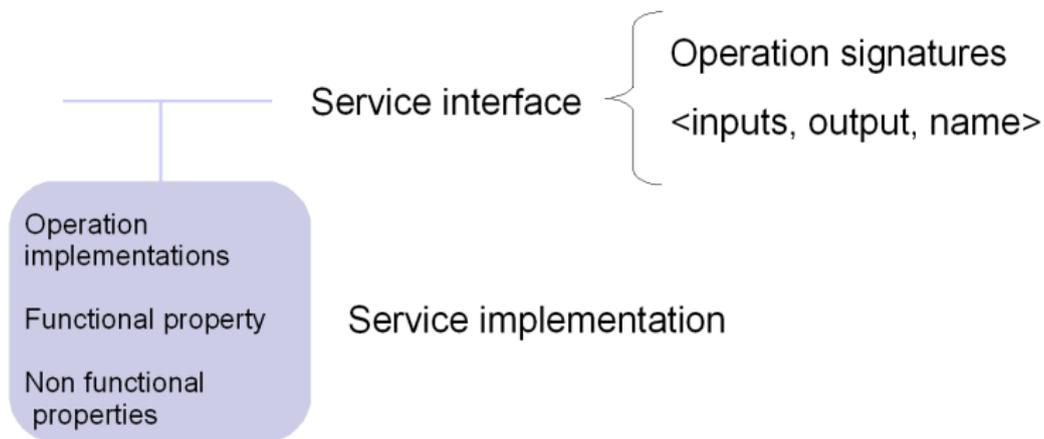
# Service Model



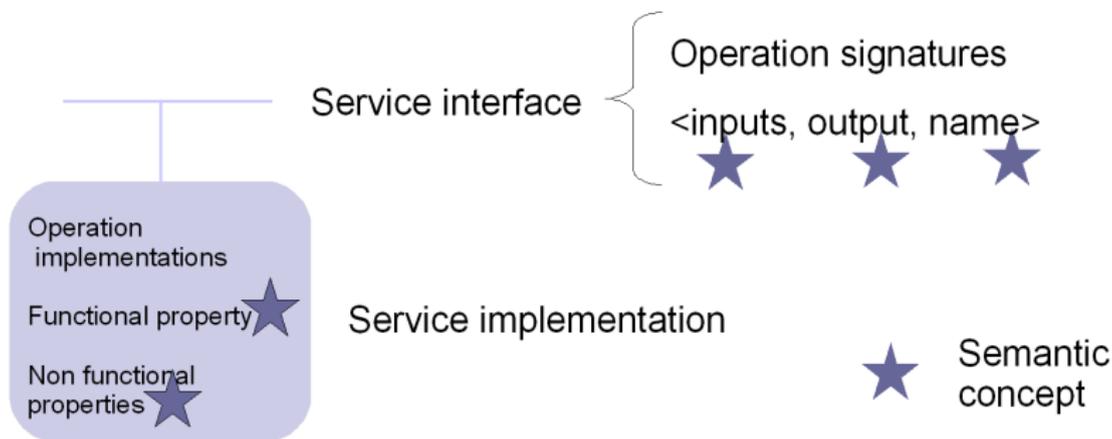
# Service Model



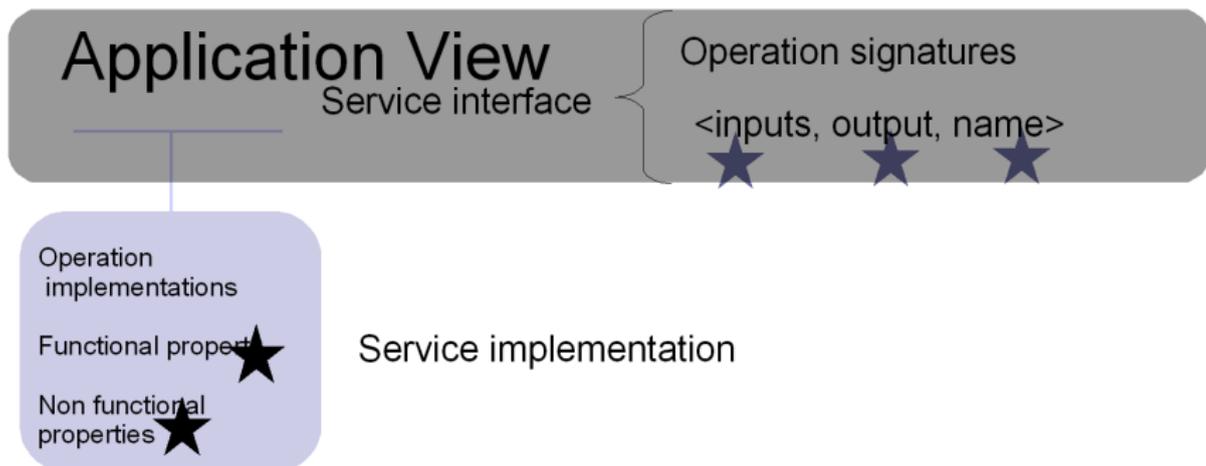
# Service Model



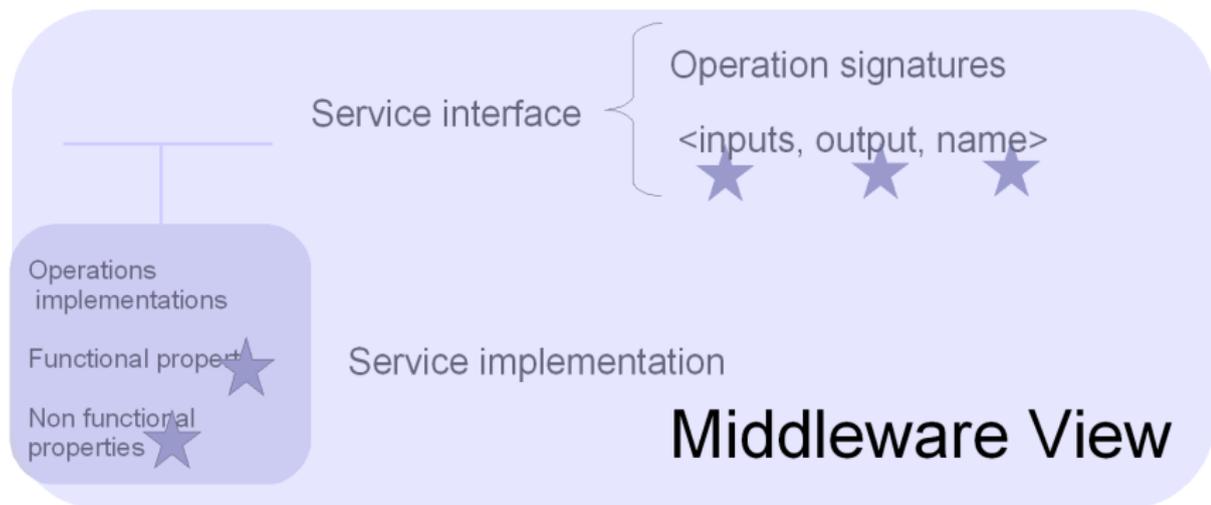
# Service Model



# Service Model

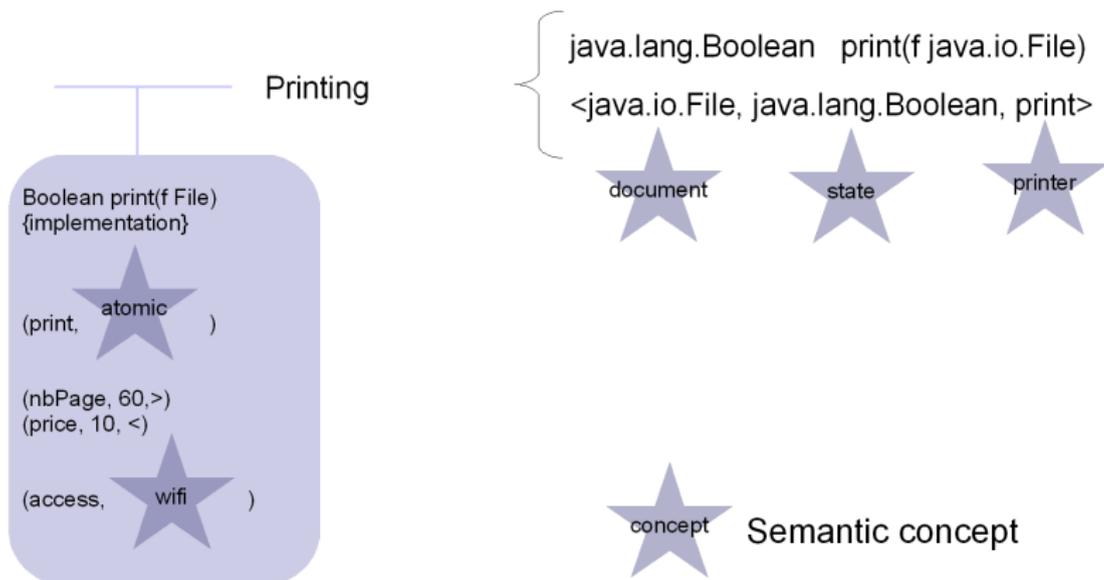


# Service Model



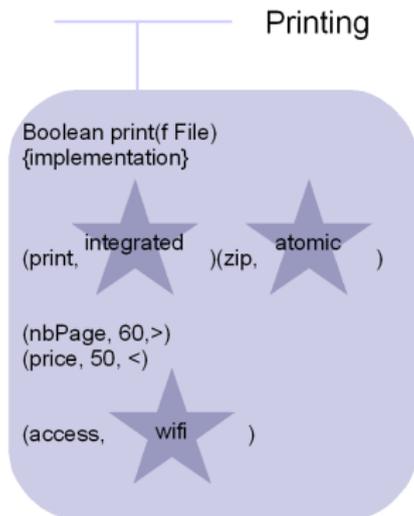
# Service Model

## An atomic printing service example



# Service Model

## An integrated printing service example



```
java.lang.Boolean print(f java.io.File)
<java.io.File, java.lang.Boolean, print>
```



document



state



printer



concept

Semantic concept

# Service Relation

- Service Equivalence Relation: Two services are equivalent and can substitute one another
- Service Composition Relation: Two services are composable and offer new functionalities resulting from the composition



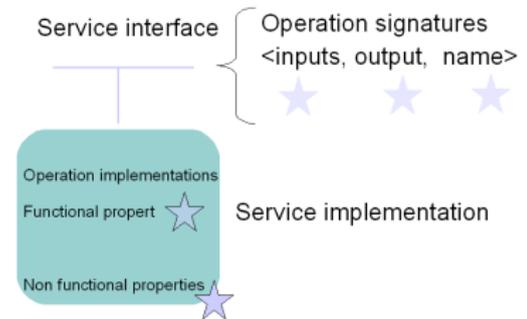
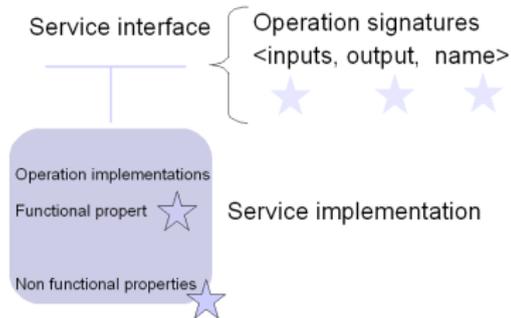
Services equivalence



Services composition

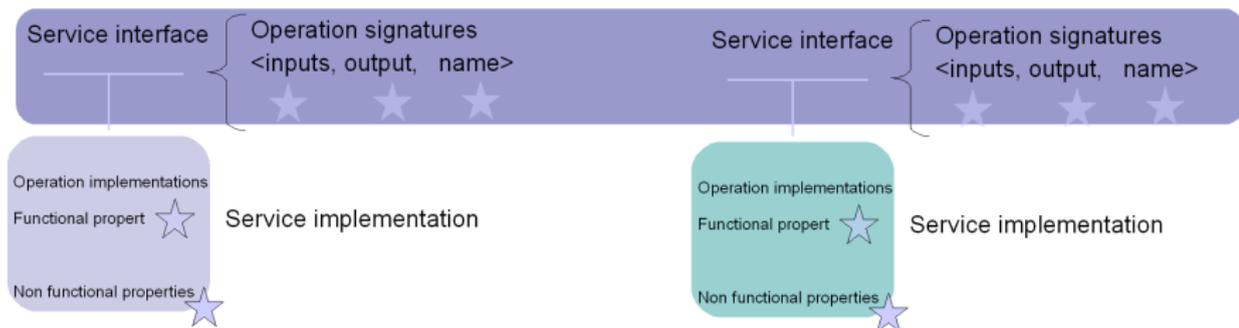
# Service Equivalence

# Service Equivalence

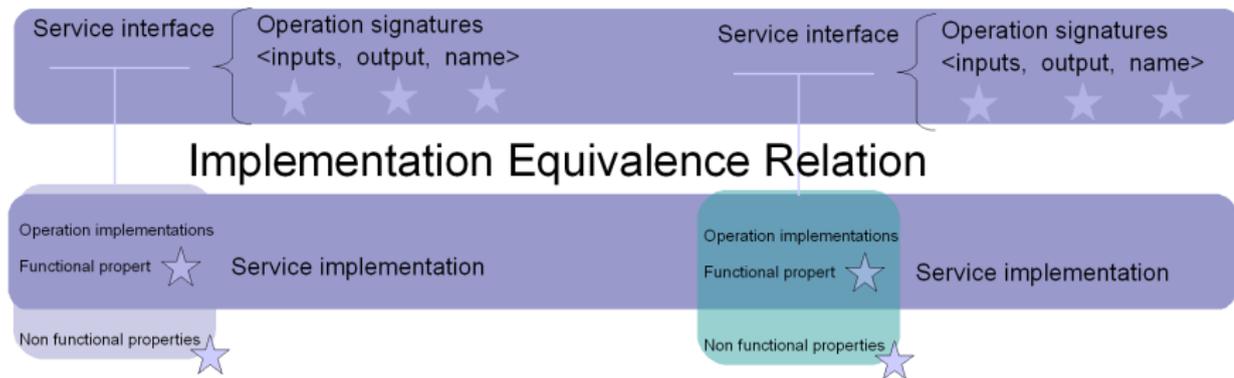


# Service Equivalence

## Interface Equivalence Relation

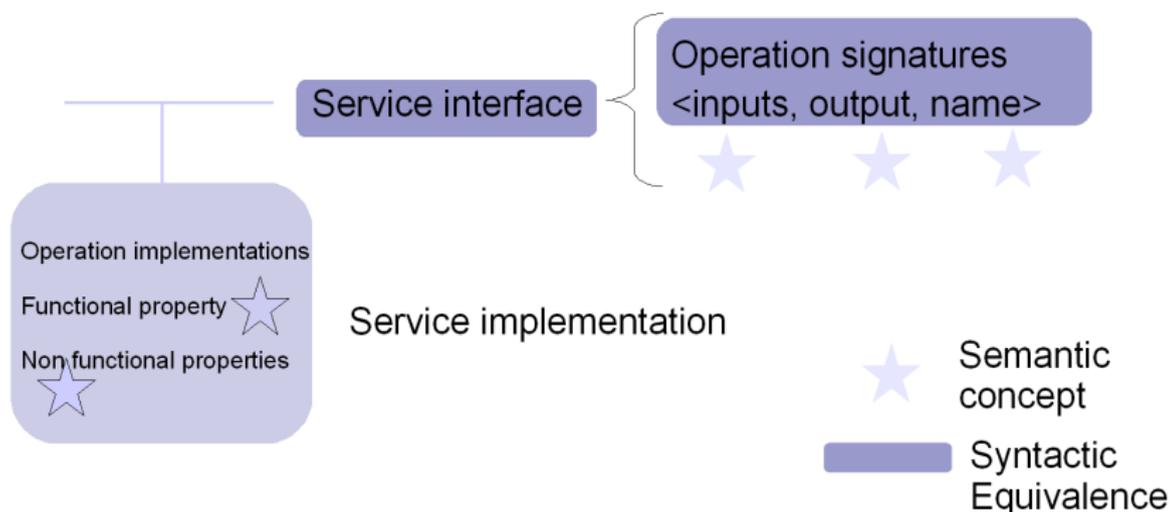


# Service Equivalence



# Interface Equivalence

## Interface Syntactic Equivalence



# Interface Equivalence

## Interface Syntactic Equivalence Example

<(java.awt.Image, java.lang.String), java.lang.Boolean, size>

<(java.awt.Image, java.lang.String), java.lang.Boolean, size>

<(java.awt.image.BufferedImage, java.lang.String), java.lang.Boolean, size>

# Interface Equivalence

## Interface Syntactic Equivalence Example

<(java.awt.Image, java.lang.String), java.lang.Boolean, size>

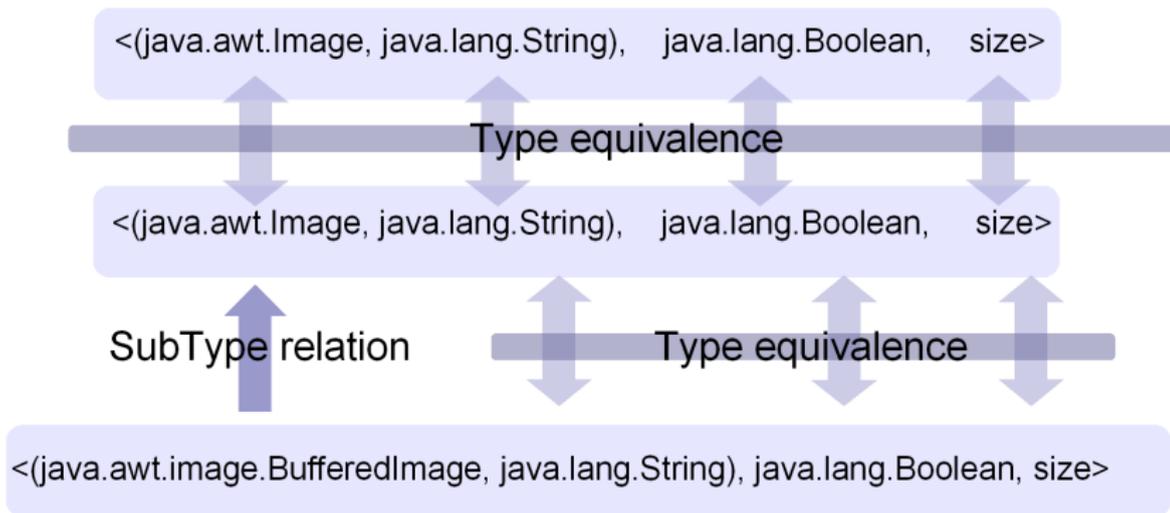
Type equivalence

<(java.awt.Image, java.lang.String), java.lang.Boolean, size>

<(java.awt.image.BufferedImage, java.lang.String), java.lang.Boolean, size>

# Interface Equivalence

## Interface Syntactic Equivalence Example



# Interface Equivalence

## Interface Syntactic Equivalence Example

<(java.awt.Image, java.lang.String), java.lang.Boolean, size>

Operation Syntactic Equivalence

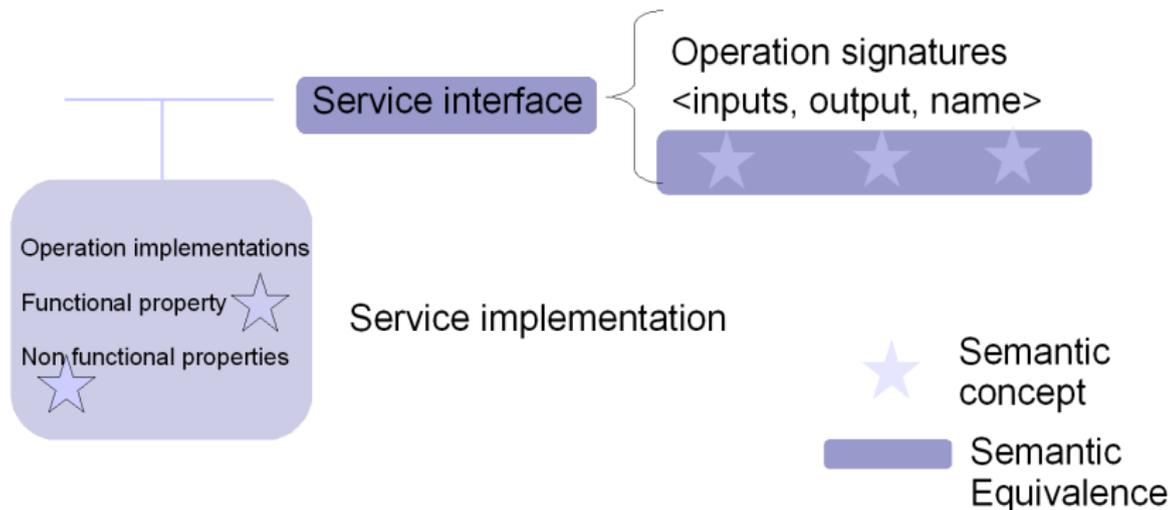
<(java.awt.Image, java.lang.String), java.lang.Boolean, size>

Operation Syntactic almost Equivalence

<(java.awt.image.BufferedImage, java.lang.String), java.lang.Boolean, size>

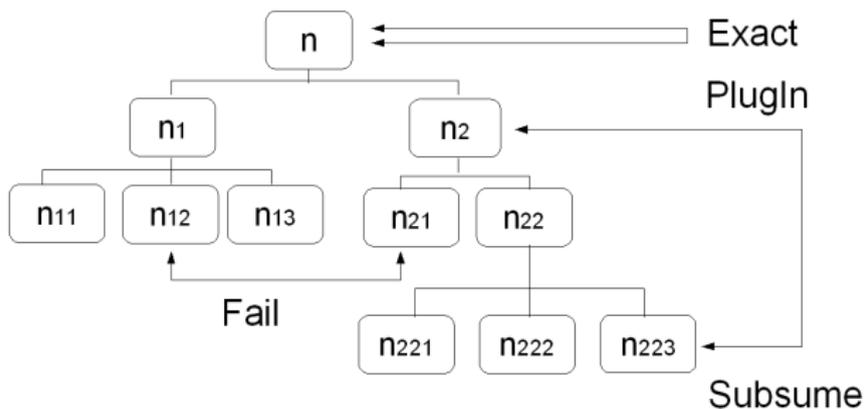
# Interface Equivalence

## Interface Semantic Equivalence



# Interface Equivalence

## Interface Semantic Equivalence Example (Paolucci[2002])



# Interface Equivalence

## Interface Semantic Equivalence Example

$\langle (n11, \quad n112), \quad n, \quad n22 \rangle$

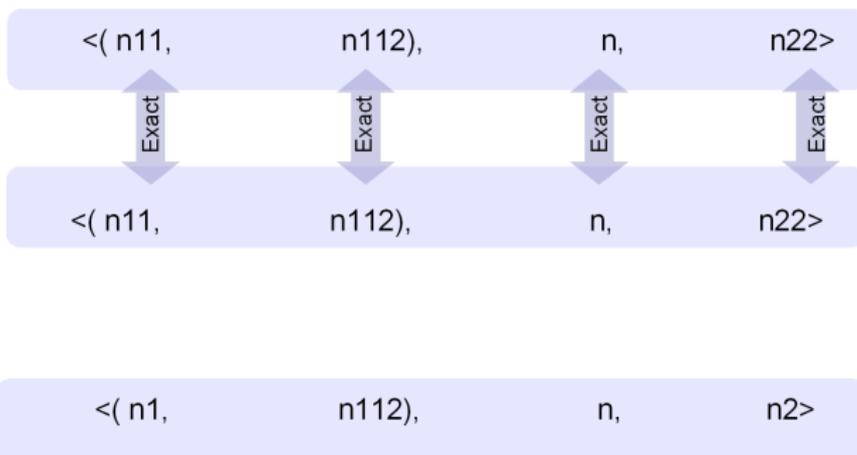
$\langle (n11, \quad n112), \quad n, \quad n22 \rangle$

$\langle (n1, \quad n112), \quad n, \quad n2 \rangle$

# Interface Equivalence

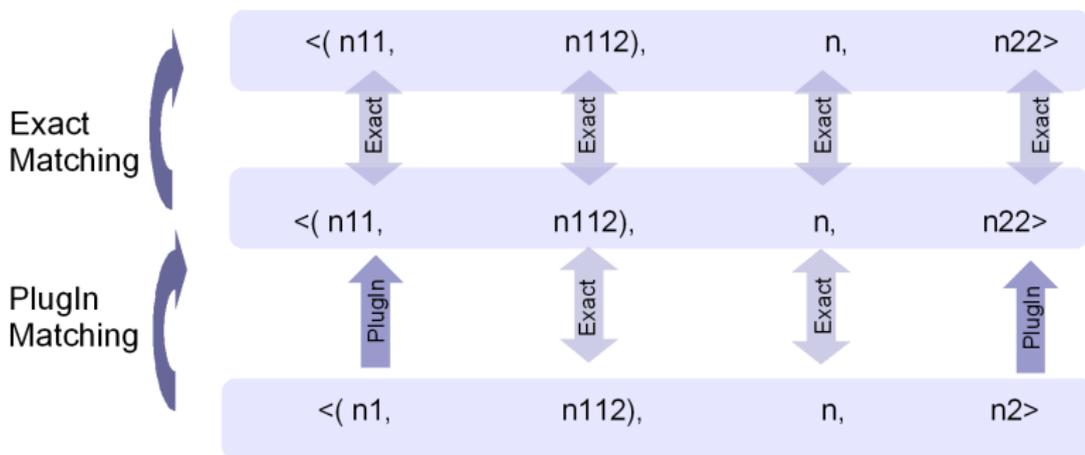
## Interface Semantic Equivalence Example

Exact  
Matching



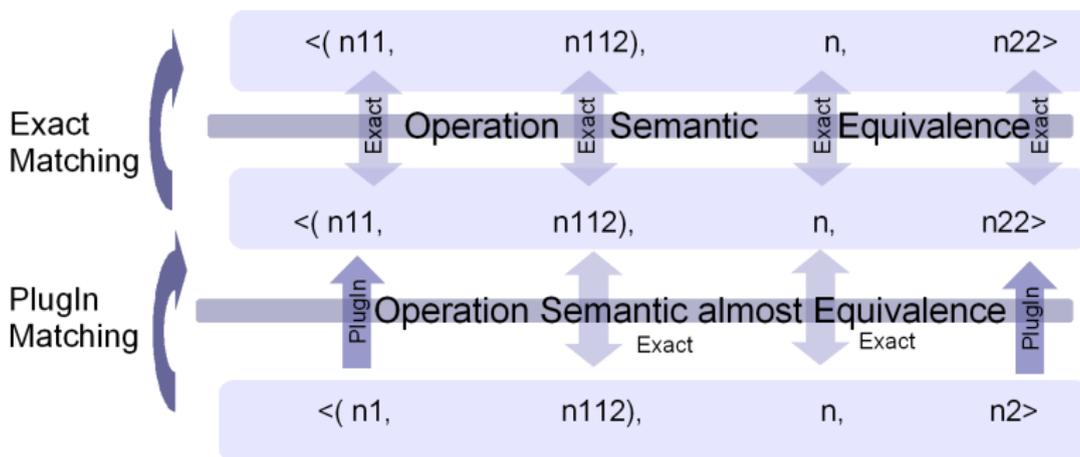
# Interface Equivalence

## Interface Semantic Equivalence Example



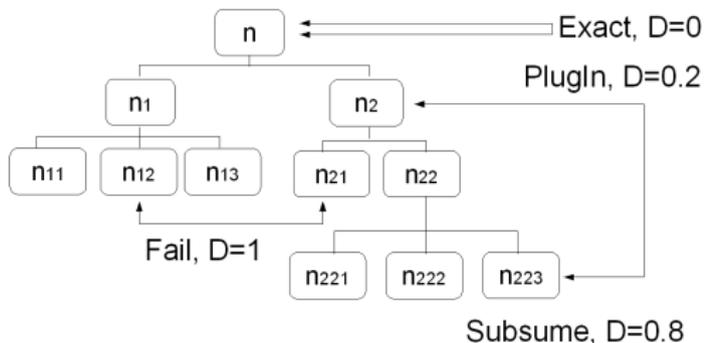
# Interface Equivalence

## Interface Semantic Equivalence Example



# Interface Equivalence

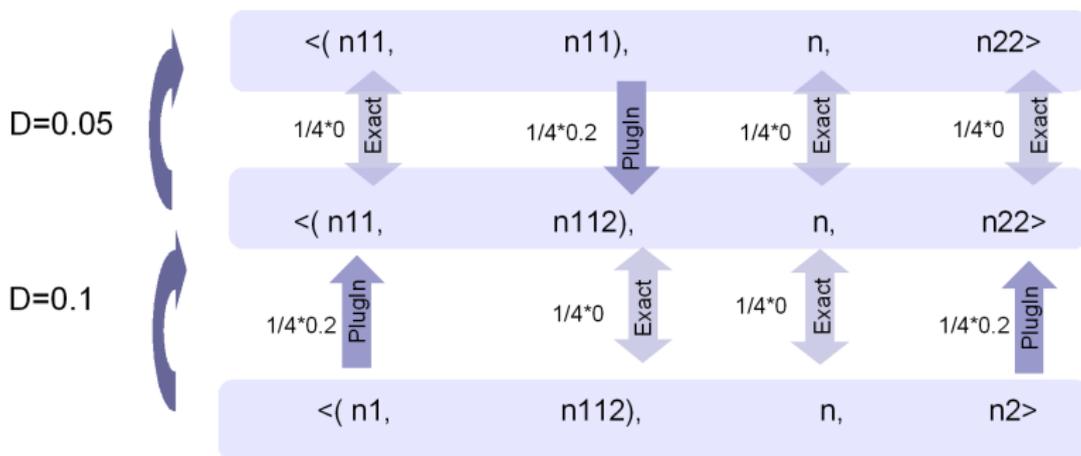
- Several operations may be PlugIn semantic matching
- Semantic distance between concepts to evaluate semantic distance between operations



Predefined values of semantic distance  $D$   
 Without considering concept levels

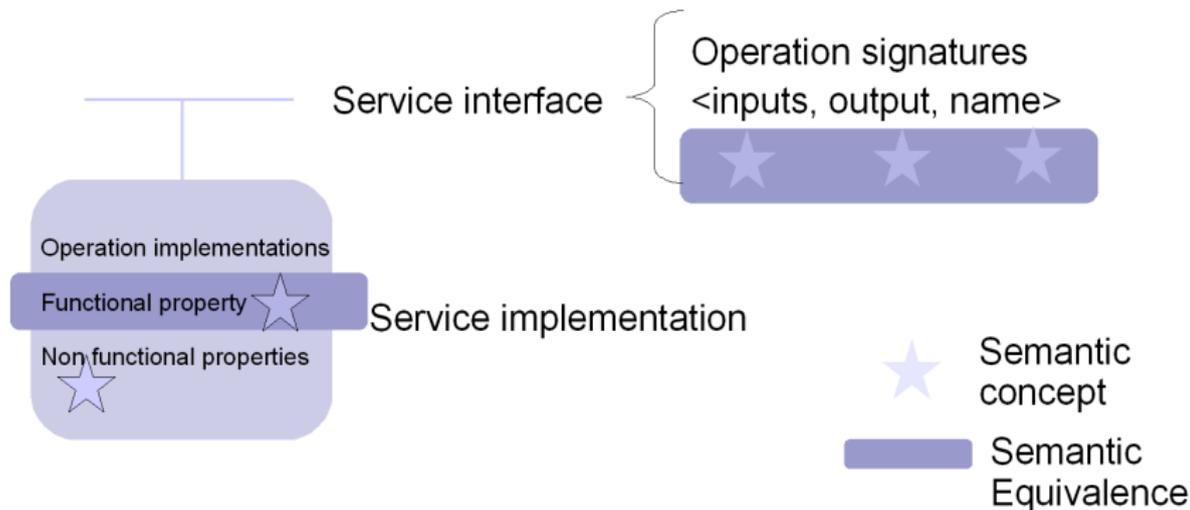
# Interface Equivalence

## Three operations plugin semantic matching

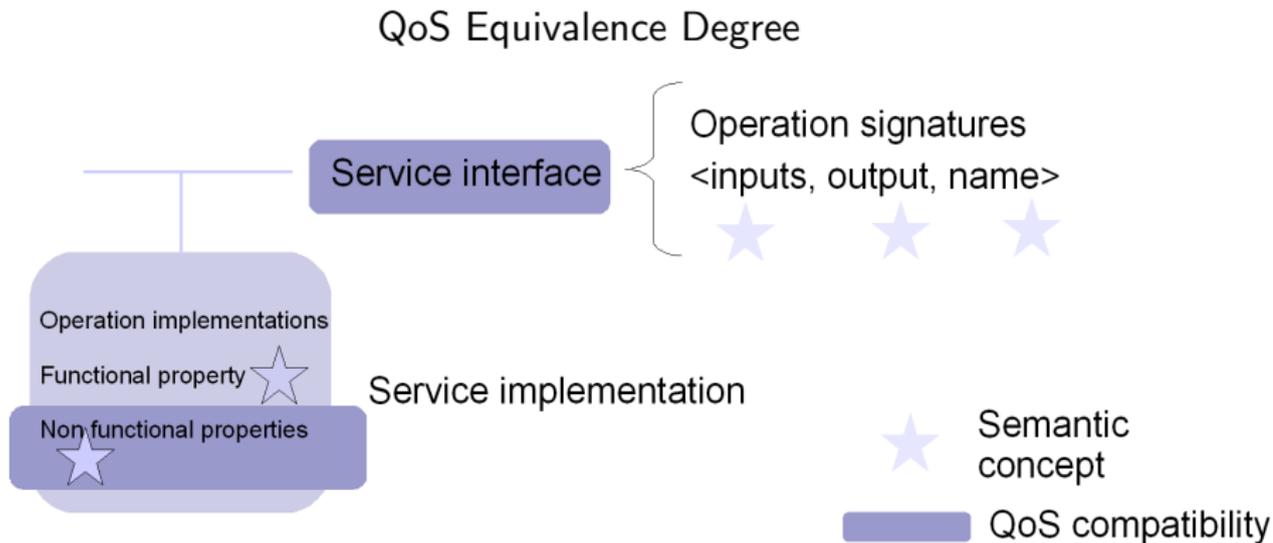


# Implementation Equivalence

## Implementation Equivalence



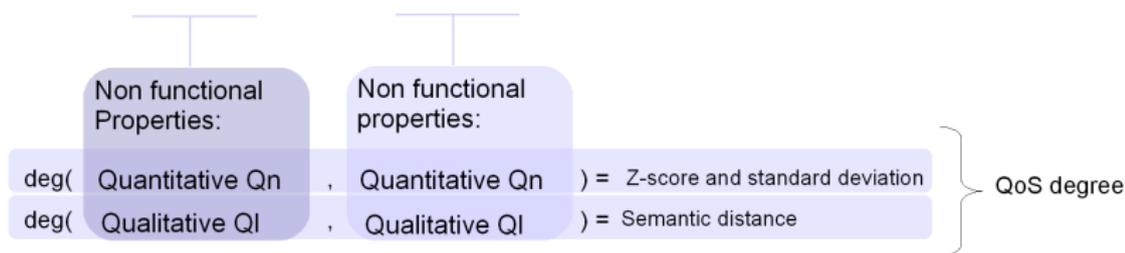
# QoS Equivalence Degree



# QoS Equivalence Degree

- Two operations can be equivalent (interface and implementation) with different non-functional properties.

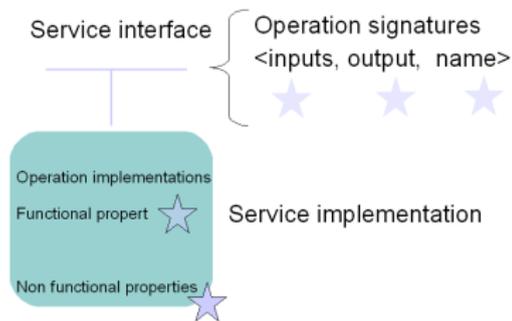
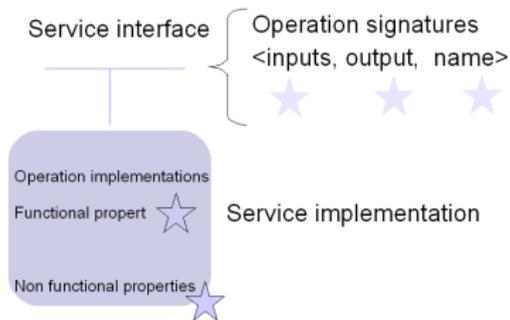
- $$QoS_{Degree}(opi, opj) = \sum_{k=1}^{|Np_{opi}|} w_k * deg(npk_{opi}, npk_{opj})$$



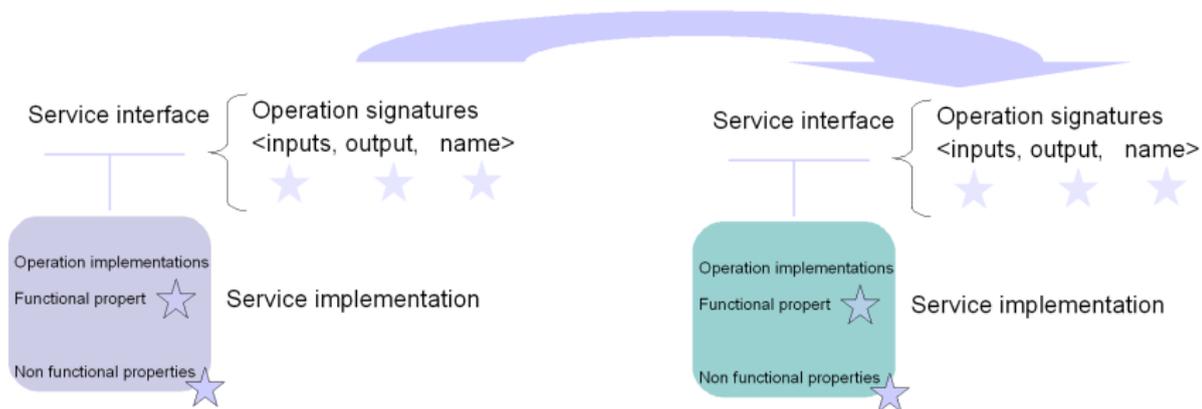
- For three equivalent operations  $opi, opj, opk$   
 $QoS_{Degree}(opi, opj) \neq QoS_{Degree}(opi, opk) \neq QoS_{Degree}(opj, opk)$

# Service Composition

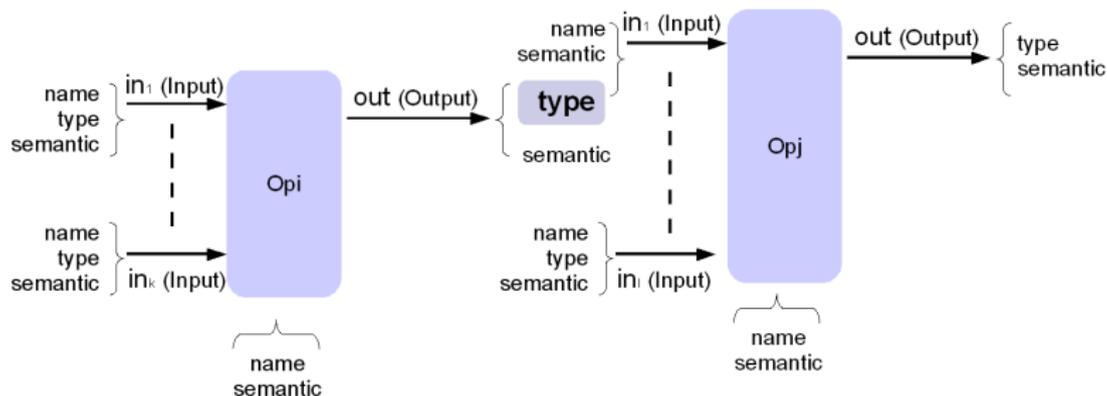
# Service Composition Relation



# Service Composition Relation

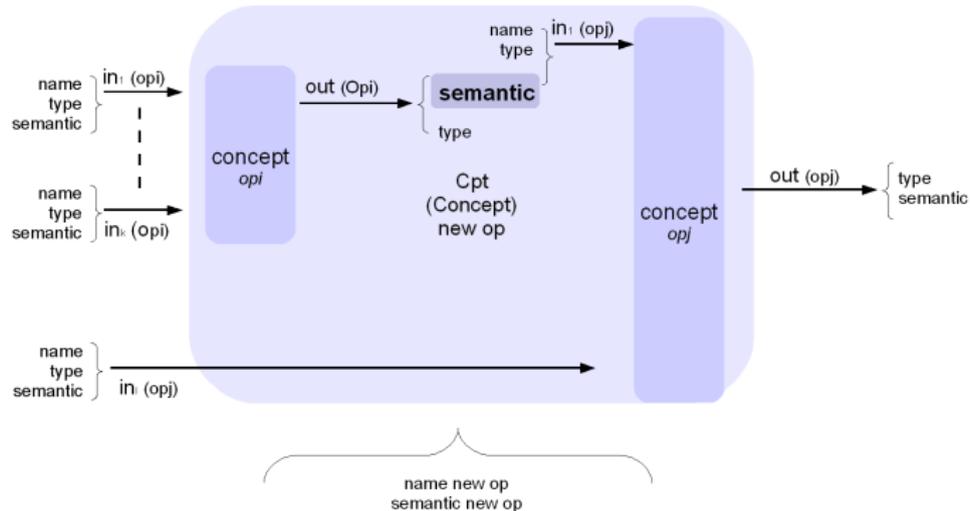


# Service Composition Relation

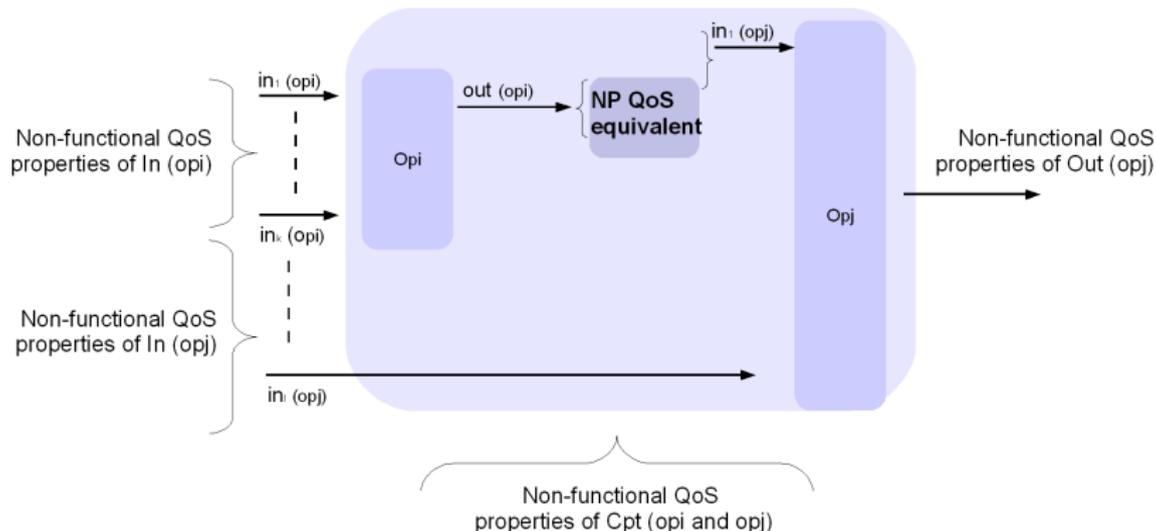




# Service Composition Relation

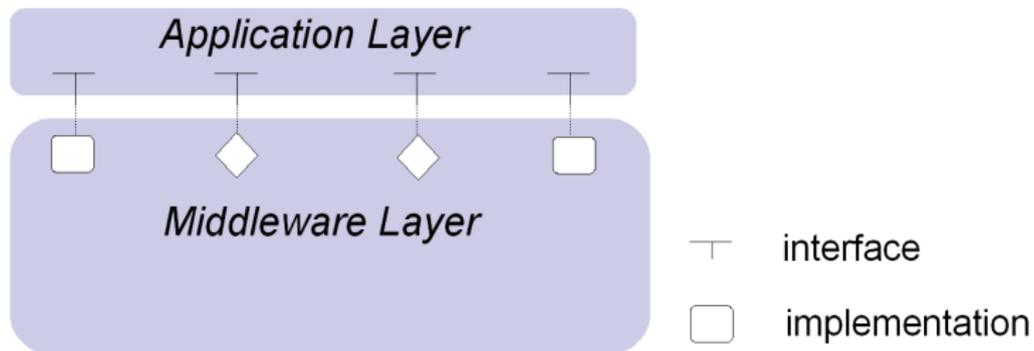


# Service Composition Relation



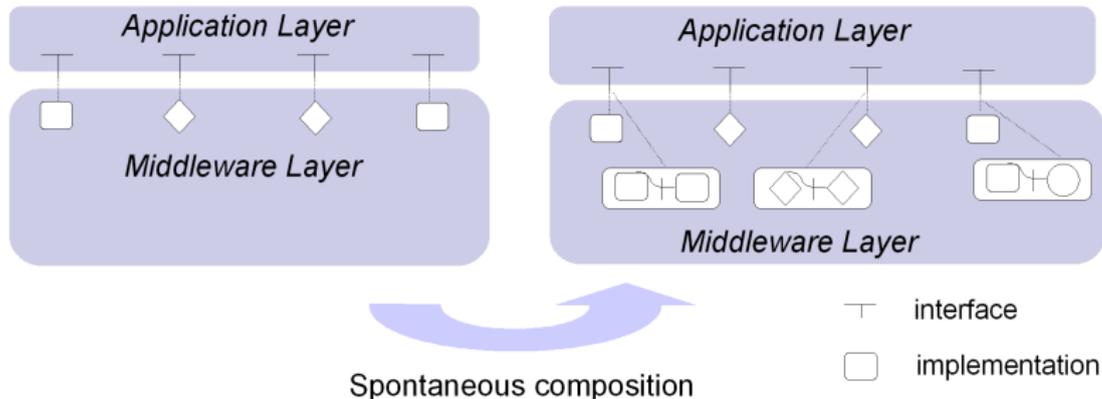
# Spontaneous Service Composition

# Spontaneous Service Composition



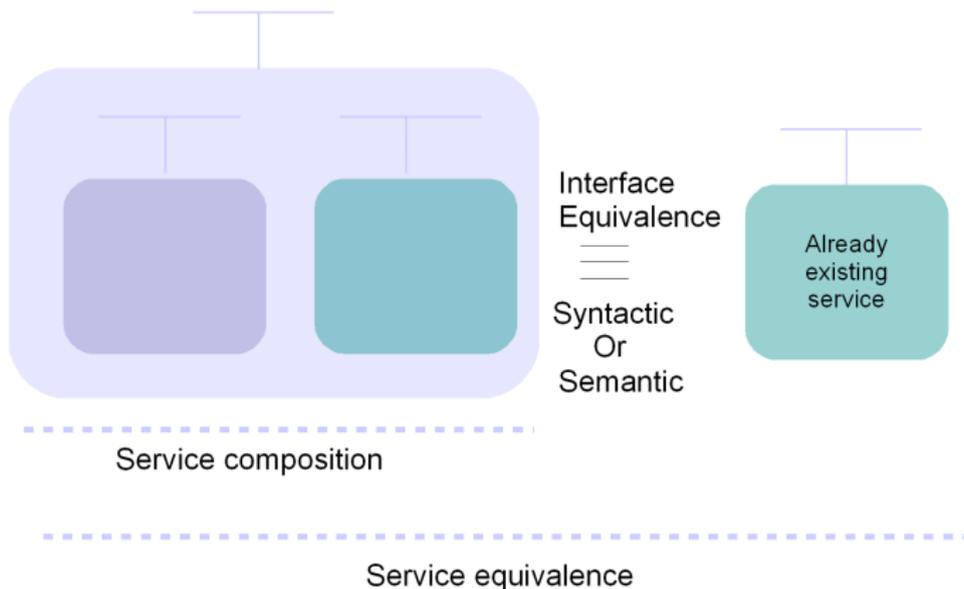
# Spontaneous Service Composition

How: Application Transparent Service Composition



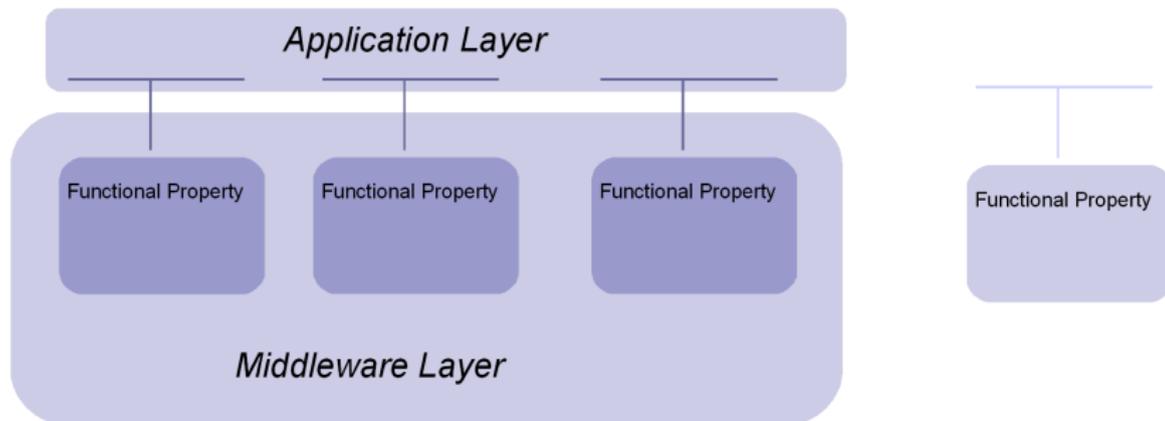
# Spontaneous Service Composition

How: combination of service equivalence and service composition relations

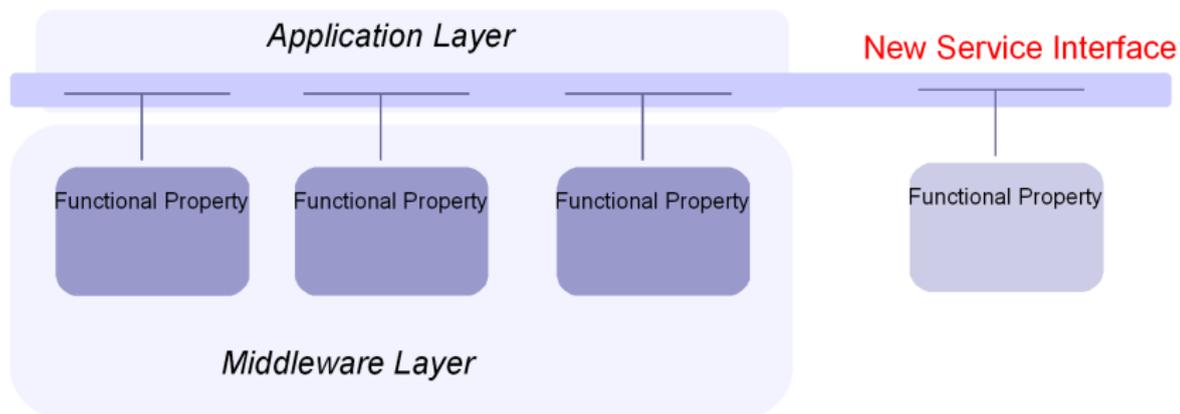


# Spontaneous Service Composition

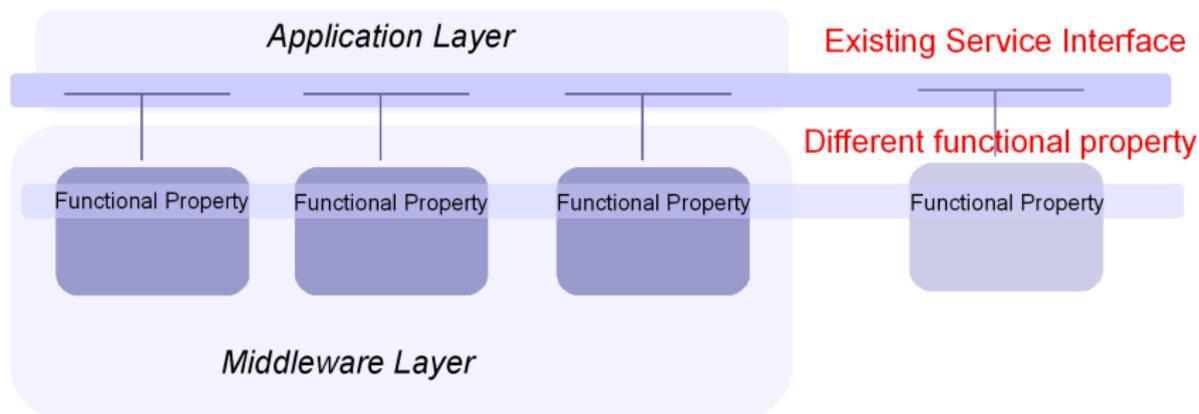
When: Appearance of a new service



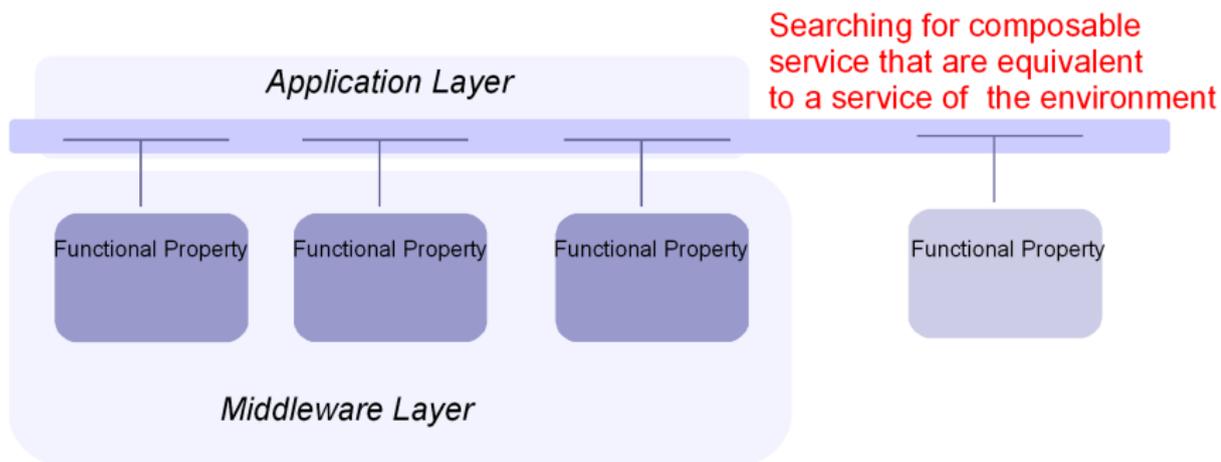
# Spontaneous Service Composition



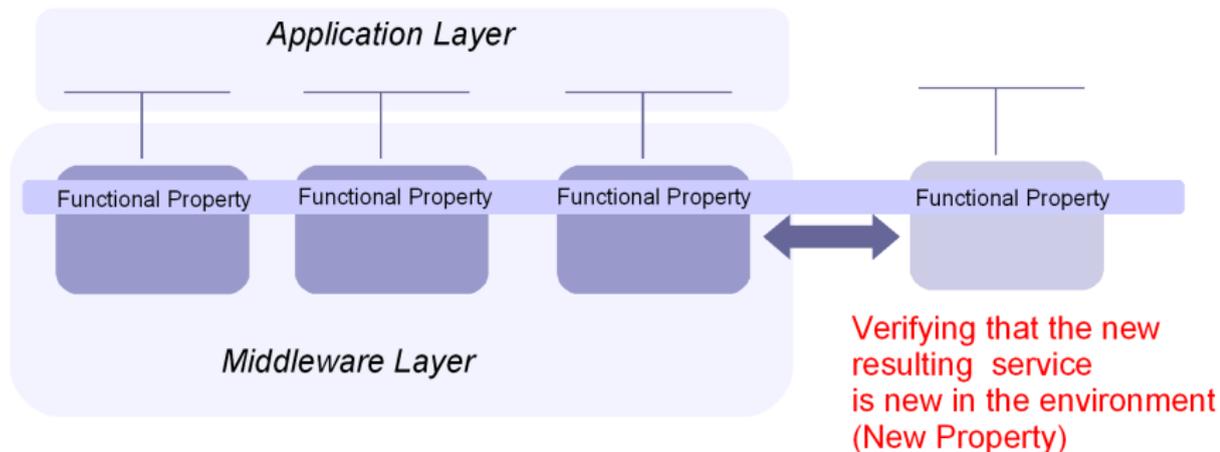
# Spontaneous Service Composition



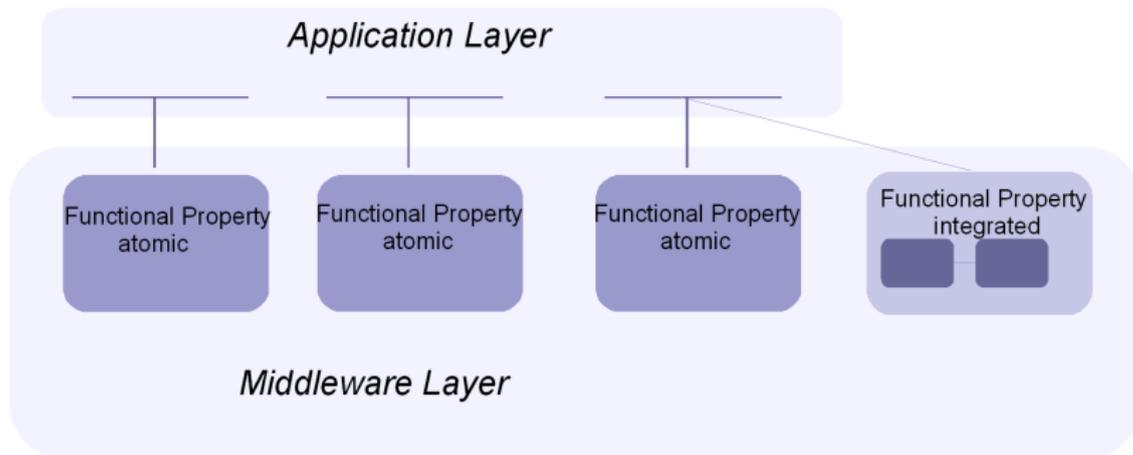
# Spontaneous Service Composition



# Spontaneous Service Composition

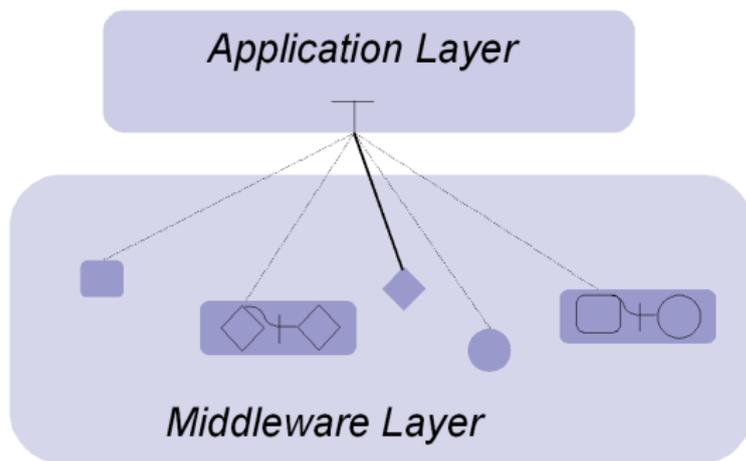


# Spontaneous Service Composition



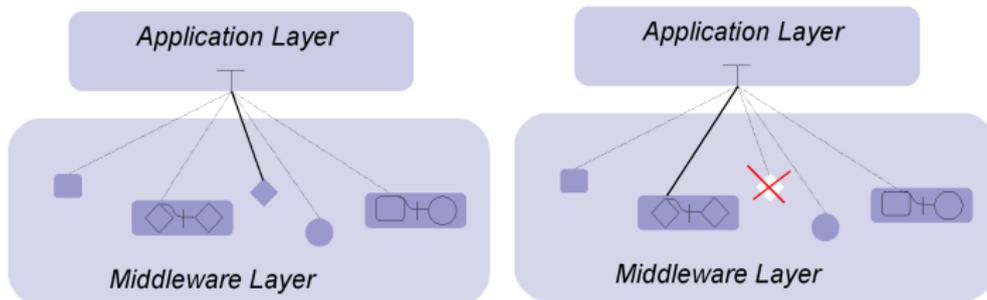
# Spontaneous Service Adaptation

# Spontaneous Service Adaptation



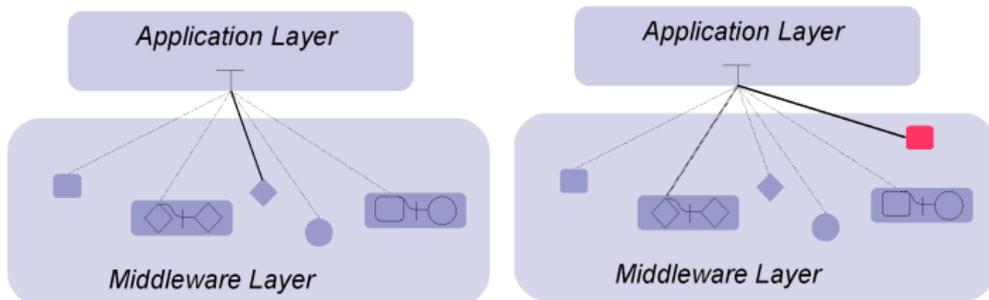
# Spontaneous Service Adaptation

Service disappears: Search for equivalent or almost equivalent services that best fit the non functional properties of the disappearing service



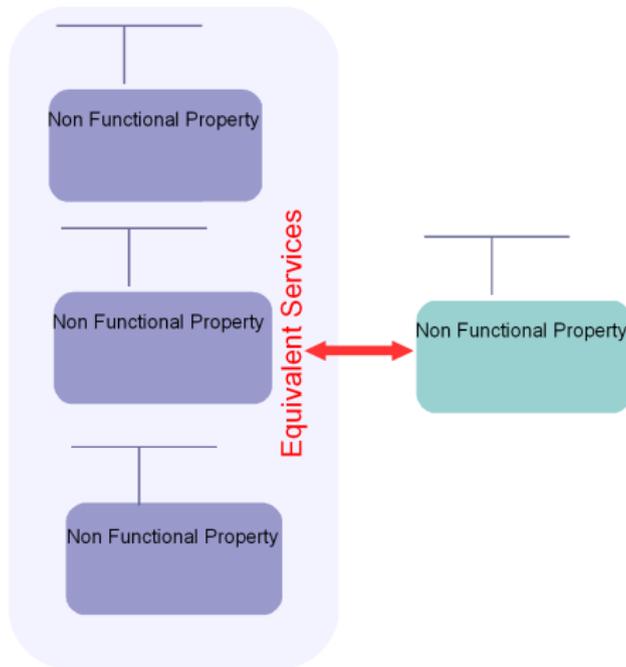
# Spontaneous Service Adaptation

Equivalent or almost equivalent service appears: Possible substitution if the new service fits best the non functional properties of the applications



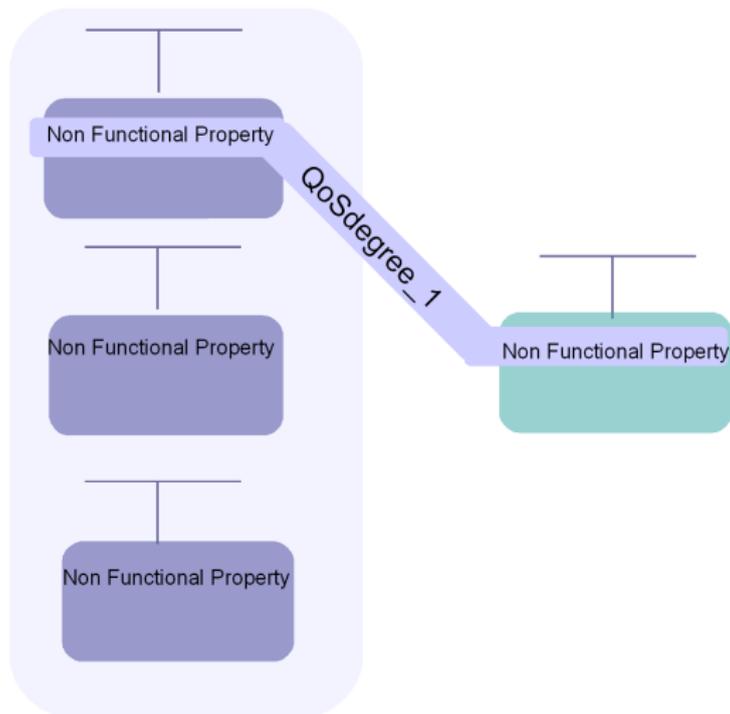
# Spontaneous Service Adaptation

Considering a set of services equivalent or almost equivalent to the service to substitute



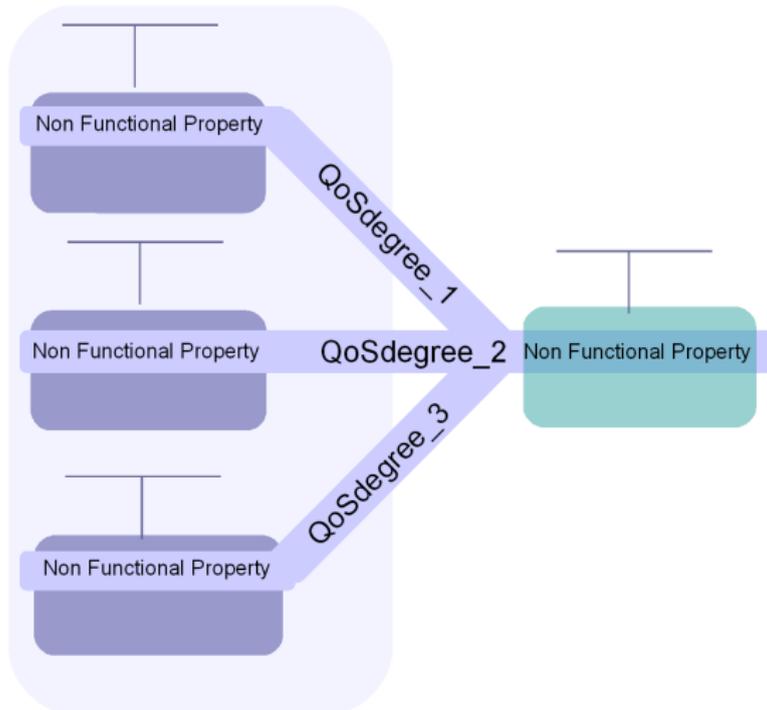
# Spontaneous Service Adaptation

## QoSdegree computing



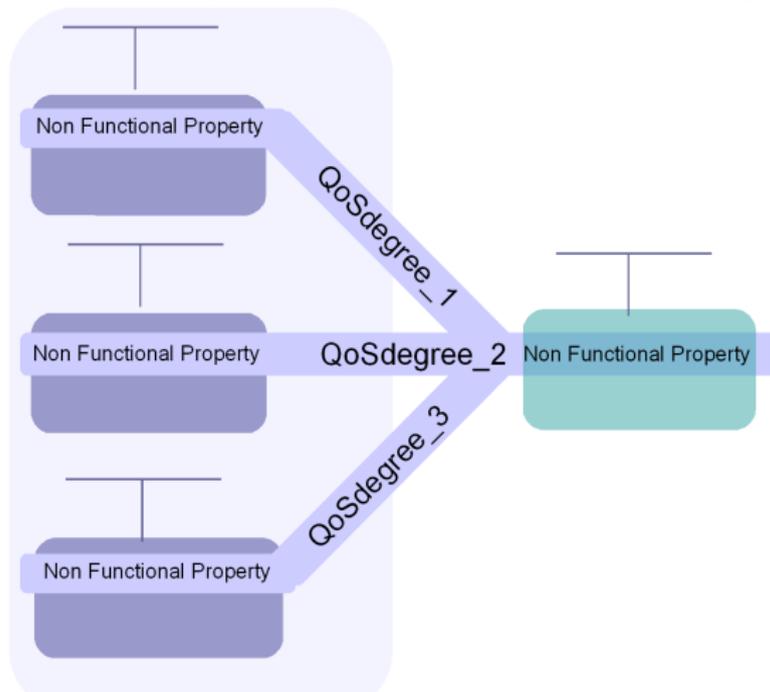
# Spontaneous Service Adaptation

## QoSdegree computing



# Spontaneous Service Adaptation

Best service corresponding to the minimal QoSdegree value



$$\text{Min}(\text{QoSdegree}_1, \text{QoSdegree}_2, \text{QoSdegree}_3)$$

# Use Case Implementation

# MyStudio

## Before the spontaneous service integration



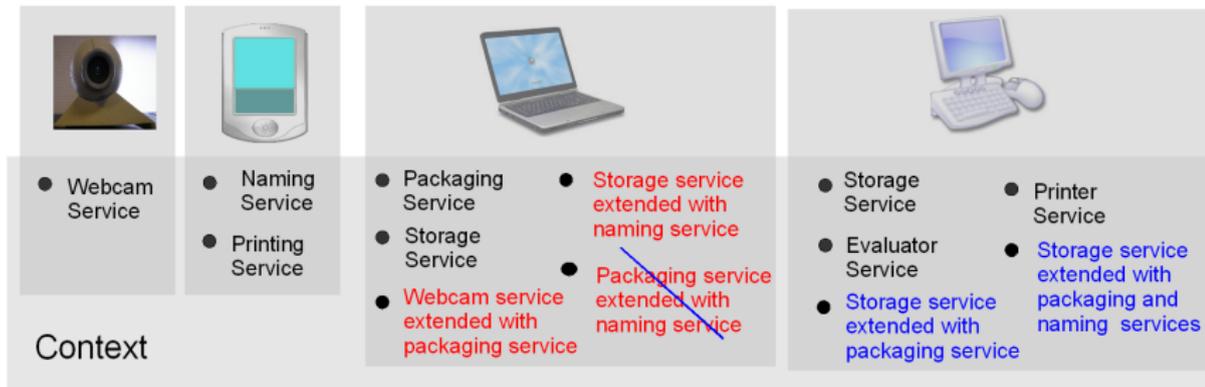
# MyStudio

After the spontaneous service syntactic composition



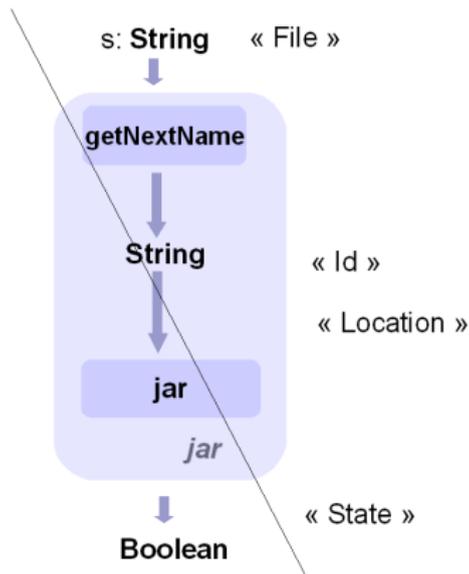
# MyStudio

## After the spontaneous service semantic composition



# MyStudio

## Semantic control

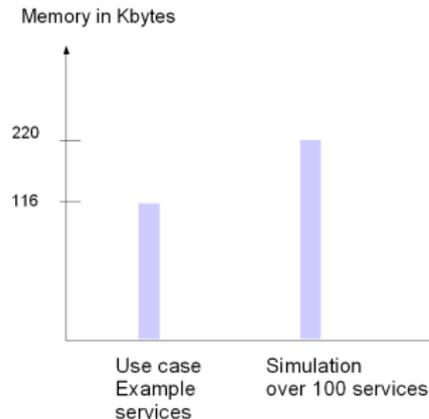
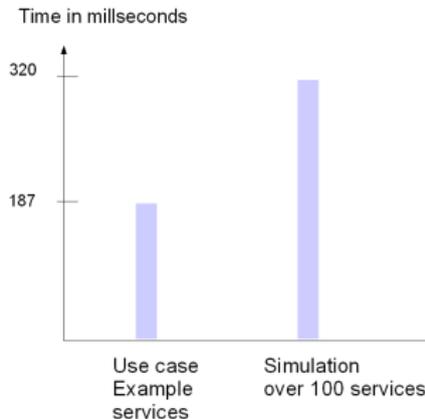


Jar & getNextName are Syntactic operation compatible

Jar & getNextName are Semantic operation incompatible

# Evaluation

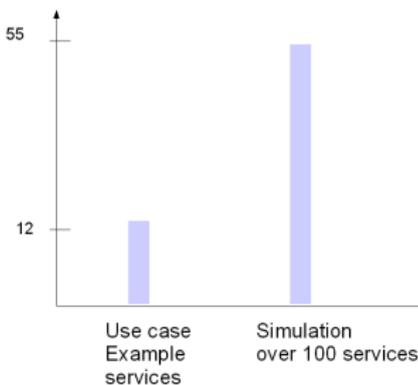
## Syntactic Interface Matching



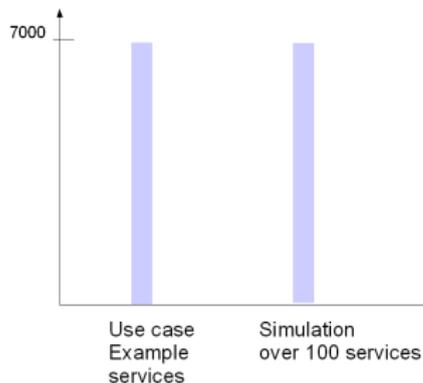
# Evaluation

## Semantic Interface Matching

Time in seconds



Memory in ko



## Implementation issues

- Project: ANIS, Inria gforge
- Code: JVM, OSGi/Felix, 47 classes, 6407 lines
- Hardware: 2 laptops (Windows XP, Kubuntu 7.04), webcam logitech
- Technical points:
  - Service generation via bundle creation (`sun.tools`)
  - Remote call vi RMI
  - OWL-S Mindswap ontology reasoner (<http://www.mindswap.org/>)

## Part III

# Conclusions & Perspectives

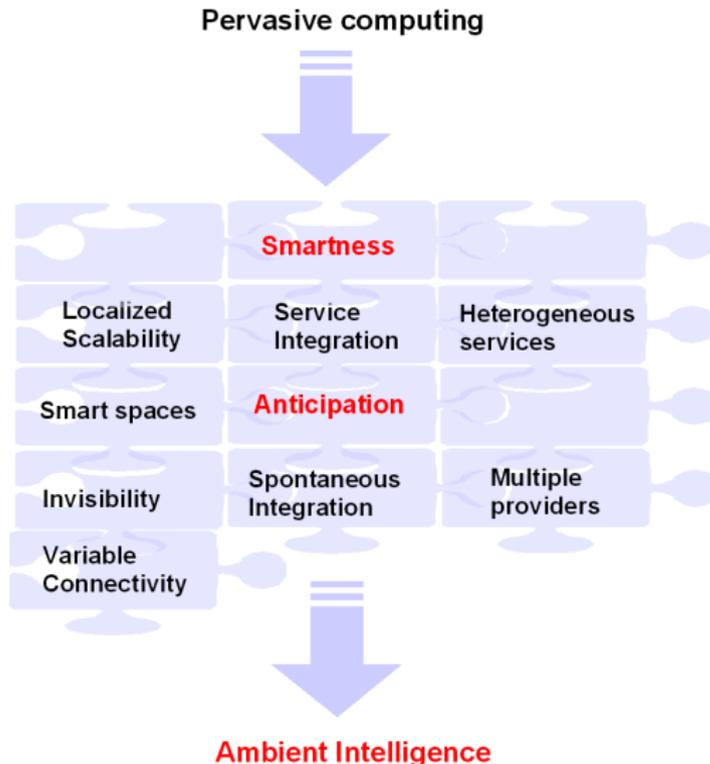
# Conclusion

- Unified vision for the service integration
  - A generic service integration middleware: SIM
  - A spontaneous instantiation of SIM: MySIM middleware
- Functional and Non-Functional service integration relations: the equivalence and composition relations
  - Equivalence, almost equivalence and composition relations over service interfaces and implementations
  - Introducing new metrics: semantic distance, QoS degree function
- Spontaneous service integration for pervasive environments
  - Application transparent service composition with semantic control
  - Service adaptation upon appearance and disappearance of services based on functional and non functional properties

# Improving MySIM middleware

- Service model
  - Functional property enabling to compose two operations twice upon different inputs.
  - Semantic distance and concept level considerations
- Service transformation
  - Automating the transformation model
  - Extending the prototype to other platforms
- Service composition
  - Combining n services
- Service adaptation
  - Taking the state of a service into account

# From Pervasive to Ambient..



# Publications

## Chapter Book:

N. Ibrahim, F. Le Mouël, and S. Frénot *Middleware technologies for ubiquitous computing*. Handbook of Research on Next Generation Networks and Ubiquitous Computing. IGI Global Publication. To appear 2009.

## International Conferences:

N. Ibrahim, F. Le Mouël, and S. Frénot *C-ANIS: a Contextual, Automatic and Dynamic Service-Oriented Integration Framework*. In International Symposium on Ubiquitous Computing Systems (UCS'2007). LNCS, November 2007.

N. Ibrahim, F. Le Mouël, and S. Frénot *Automatic service-integration framework for ubiquitous environments*. In Proceedings of the International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM'2007). Papeete, French Polynesia (Tahiti), France, November 2007.

N. Ibrahim and F. Le Mouël *Anis: A negotiated integration of services in distributed environments*. In Proceedings of the 8th International Symposium on Distributed Objects and Applications (DOA'2006), Montpellier, France, October 2006.

W. Jouve, N. Ibrahim, L. Réveillère, F. Le Mouël, and C. Consel. *Building home monitoring applications: From design to implementation into the amigo middleware*. In the proceedings of the 2nd International Conference on Pervasive Computing and Applications (ICPCA'2007), Birmingham, UK, July 2007.

## International Workshops:

N. Ibrahim, F. Le Mouël *Context-aware Specialization of Semantic Rules for choosing Services in Pervasive Environments*. 2nd IEEE International Workshop on Services Integration in Pervasive Environments (SIPE'2007), Istanbul, Turkey, 2007.

P. Parrend, Y. Royon and N. Ibrahim *Service-Oriented Distributed Communities in Residential Environments*. 1st IEEE International Workshop on Services Integration in Pervasive Environments (SIPE'2006), Lyon, France, 2006.

N. Ibrahim, F. Le Mouël, Y. Royon and S. Frénot *Semantic Deployment of Services in Pervasive Environments*. RSPSI workshop at Pervasive 2006, Dublin, Ireland, May, 2006.

N. Ibrahim, F. Le Mouël, and S. Frénot *Automatic negotiated integration of services in pervasive environments*. In Proceedings of the Middleware for Web Services Workshop (MWS'2005), Enschede, Netherland, September 2005. "Most Promising Research" Award, NICTA.

Thank You..

