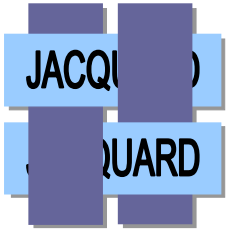
OpenCCM : une infrastructure à composants pour le déploiement d'applications à base de composants CORBA

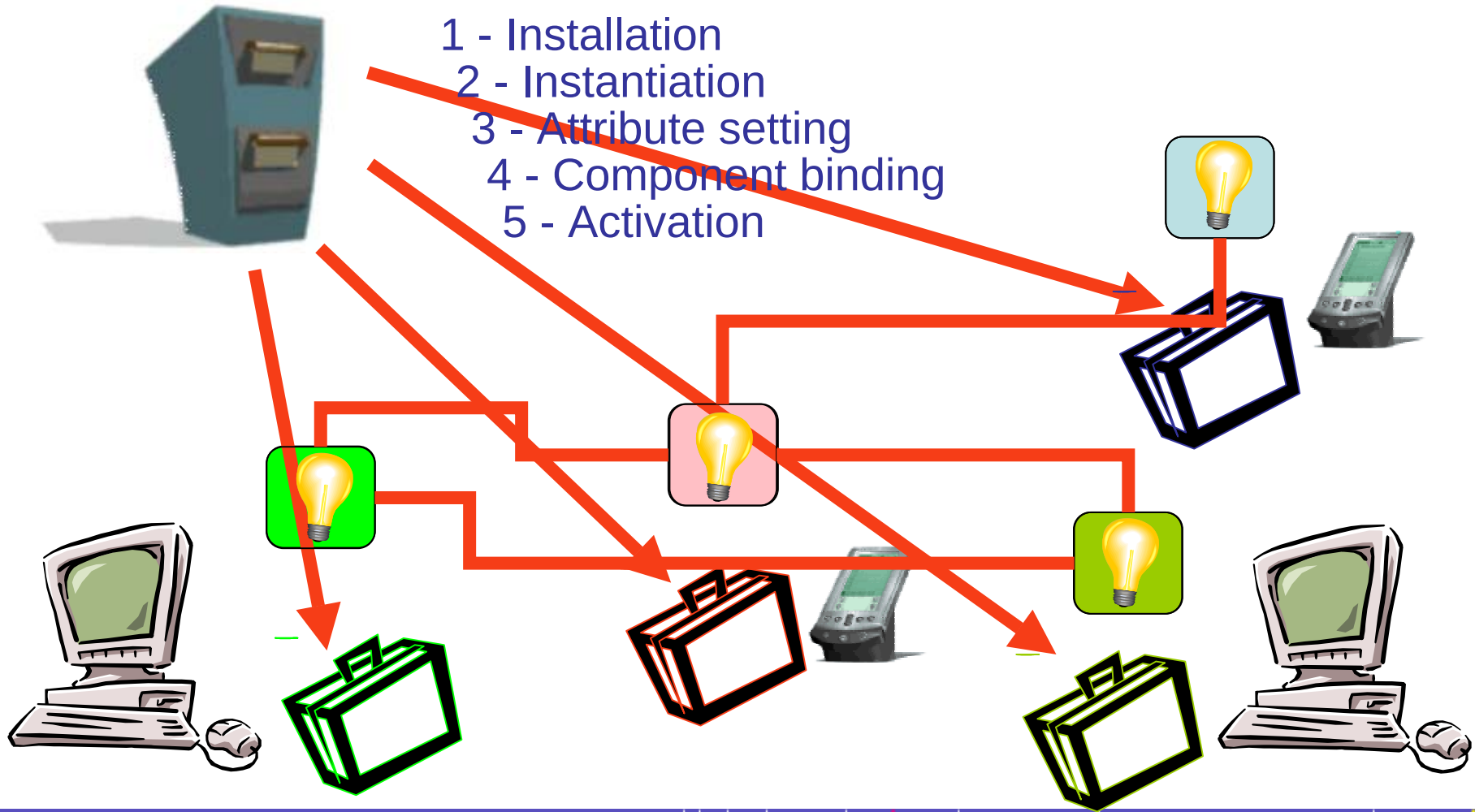
Frédéric **BRICLET**
Christophe **CONTRERAS**
Philippe **MERLE**
openccm@objectweb.org
Project INRIA Jacquard



Outline

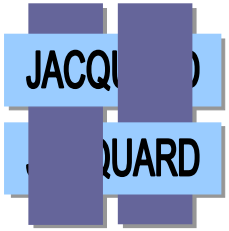
- Component-based software deployment
- The CORBA component deployment
  - Architecture & scenario
  - Identified issues
- OpenCCM's Distributed Computing Infrastructure (DCI)
  - Global architecture
  - Benefits from the CCM design
  - The domain layer's features
  - The distributed engine layer's features
  - The target nodes layer's features
- Experimentations
- Conclusion and perspectives

Distributed Component-based Software Deployment (1/2)



Distributed Component-based Software Deployment (2/2)

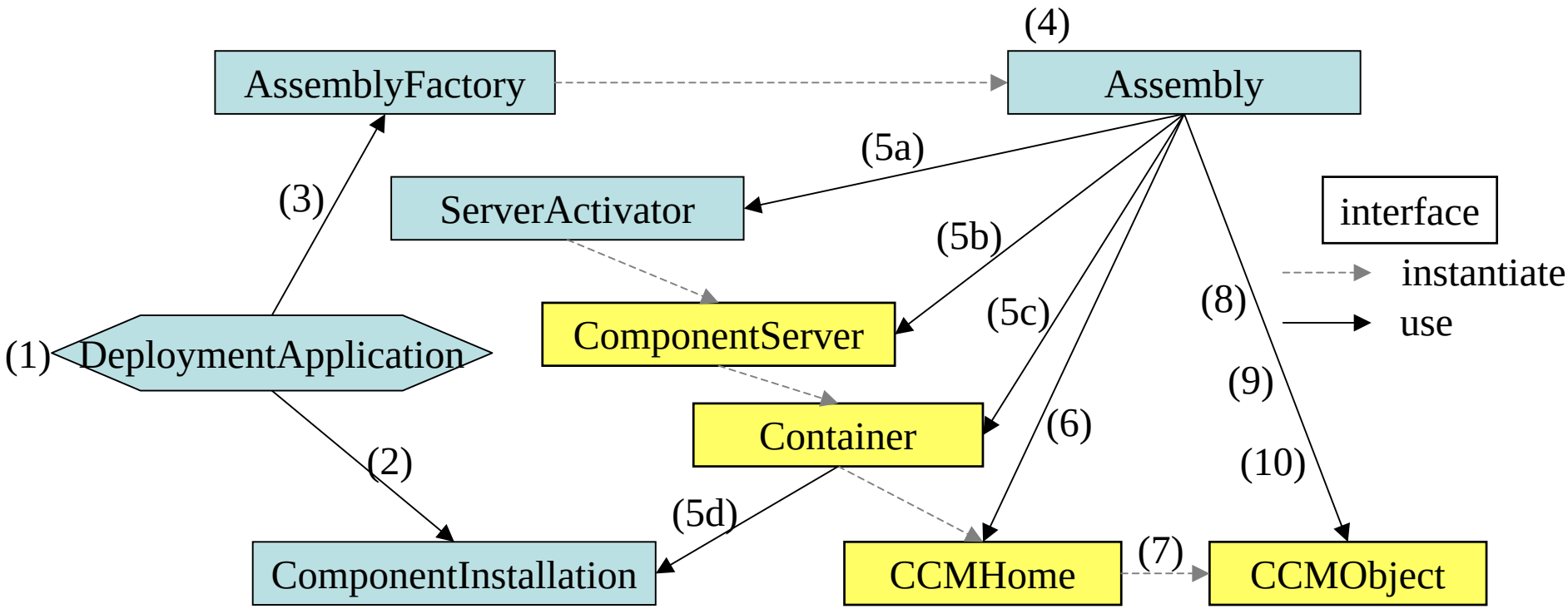
- Requires an automated process instead of a manual one
- Implies a distributed infrastructure
  - Deployment process controller
  - Target hosts controllers
- Our context: The CORBA Component Model (CCM)
- Our contribution: The OpenCCM deployment infrastructure



Outline

- Component-based software deployment
- The CORBA component deployment
  - Architecture & scenario
  - Identified issues
- OpenCCM's Distributed Computing Infrastructure (DCI)
  - Global architecture
  - Benefits from the CCM design
  - The domain layer's features
  - The distributed engine layer's features
  - The target nodes layer's features
- Experimentations
- Conclusion and perspectives

The CORBA Components Deployment - Architecture & Scenario



(2) Installation

(3-4) CCM Assembly creation

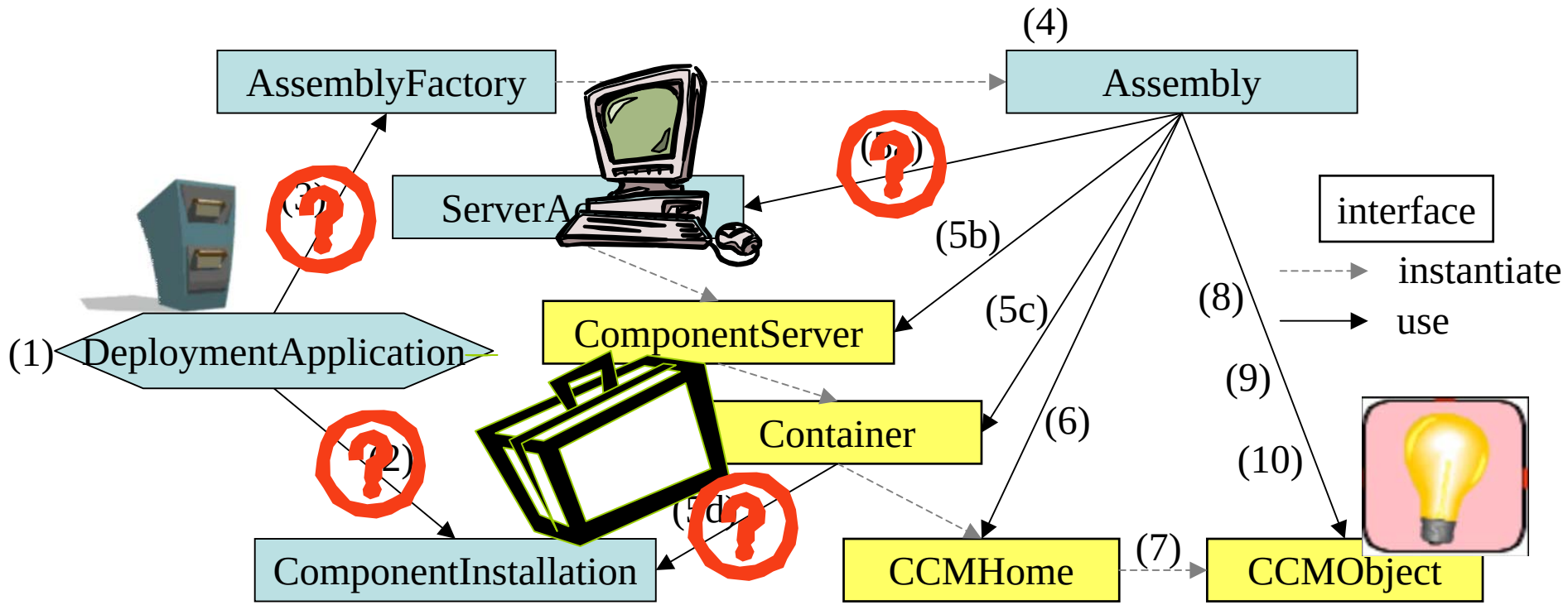
(5-6-7) Instantiation (factory pattern)

(8) Attribute setting

(9) Component binding

(10) Activation

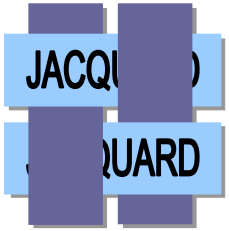
The CORBA Components Deployment - Identified Issues



√ Some features are well addressed…

X How to retrieve references of the AssemblyFactory, the ServerActivators and the ComponentInstallations?

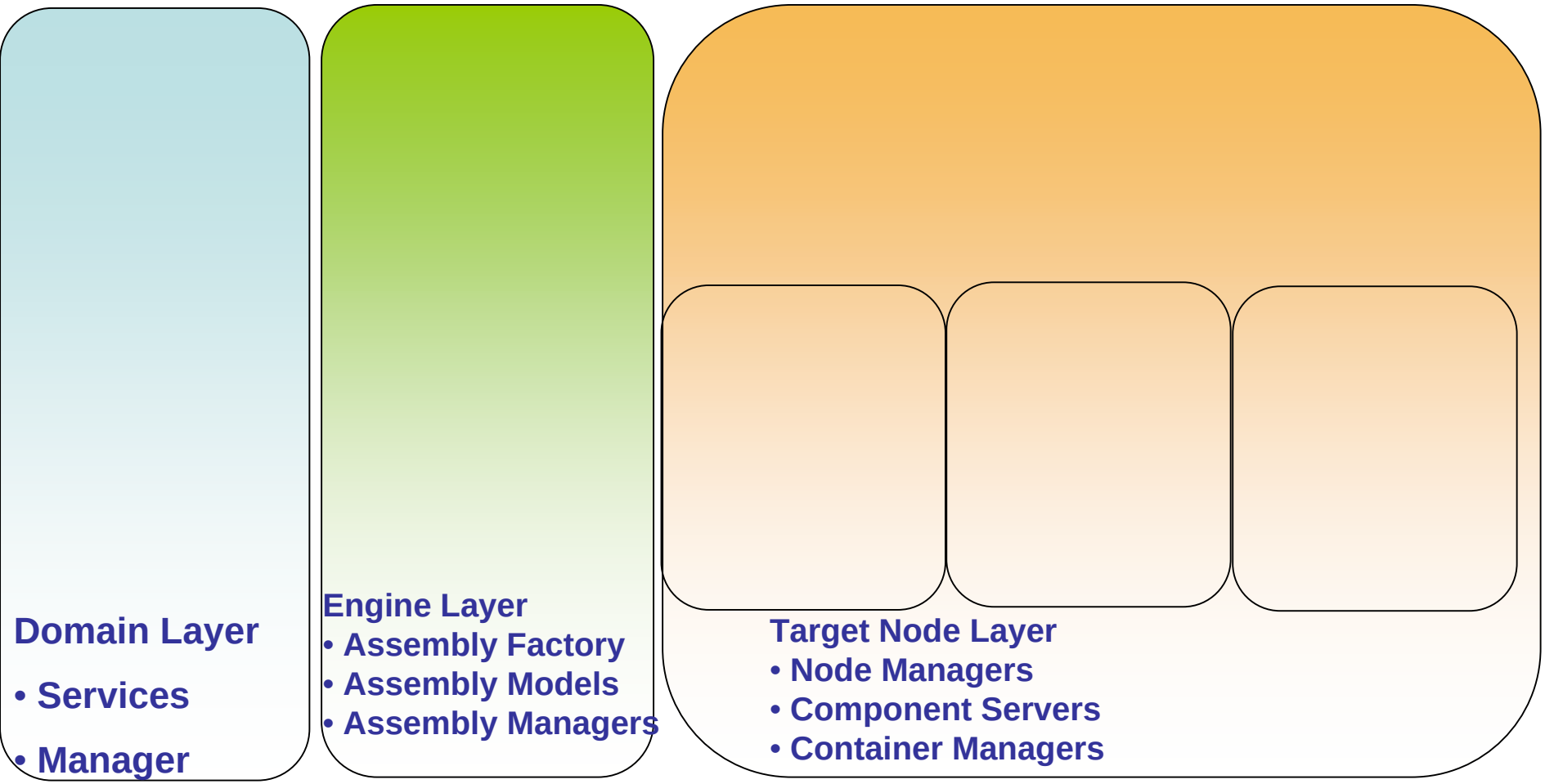
⇒ Specific and implicit implementation choices



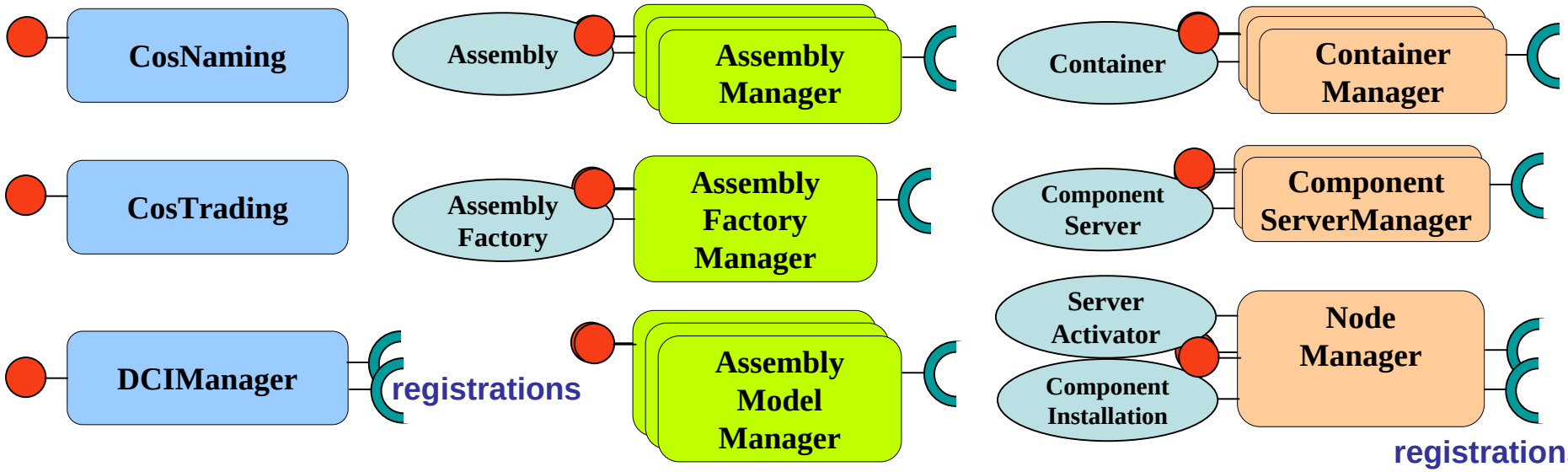
Outline

- Component-based software deployment
- The CORBA component deployment
  - Architecture & scenario
  - Identified issues
- OpenCCM's Distributed Computing Infrastructure (DCI)
  - Global architecture
  - Benefits from the CCM design
  - The domain layer's features
  - The distributed engine layer's features
  - The target nodes layer's features
- Experimentations
- Conclusion and perspectives

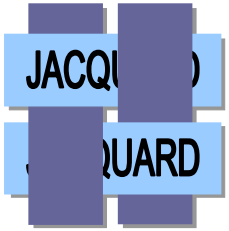
Distributed Computing Infrastructure (DCI) - A Global View Of Architecture



CORBA Component Model Benefits For The Distributed Computing Infrastructure

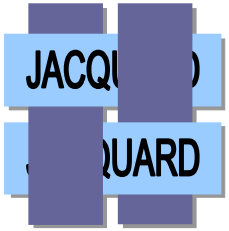


- √ Each component offers its services via CCM facets
- √ Each requirement is expressed with CCM receptacles
- √ OMG's CCM Deployment interfaces are implemented
- √ "Internal" CCM ports maintain consistency by registering components with each others, which allows introspection & control
- √ No hidden communication pathes between components



Outline

- Component-based software deployment
- The CORBA component deployment
  - Architecture & scenario
  - Identified issues
- OpenCCM's Distributed Computing Infrastructure (DCI)
  - Global architecture
  - Benefits from the CCM design
  - The domain layer's features
  - The distributed engine layer's features
  - The target nodes layer's features
- Experimentations
- Conclusion and perspectives



The Domain Layer Features

(1) Global consistency

- CosNaming & CosTrading references are kept atomic for the whole distributed infrastructure
- Nodes can register themselves to be known as ready to be used

(2) Access to meta information from each DCI component

- Nodes characteristics, installed and running Assemblies, …

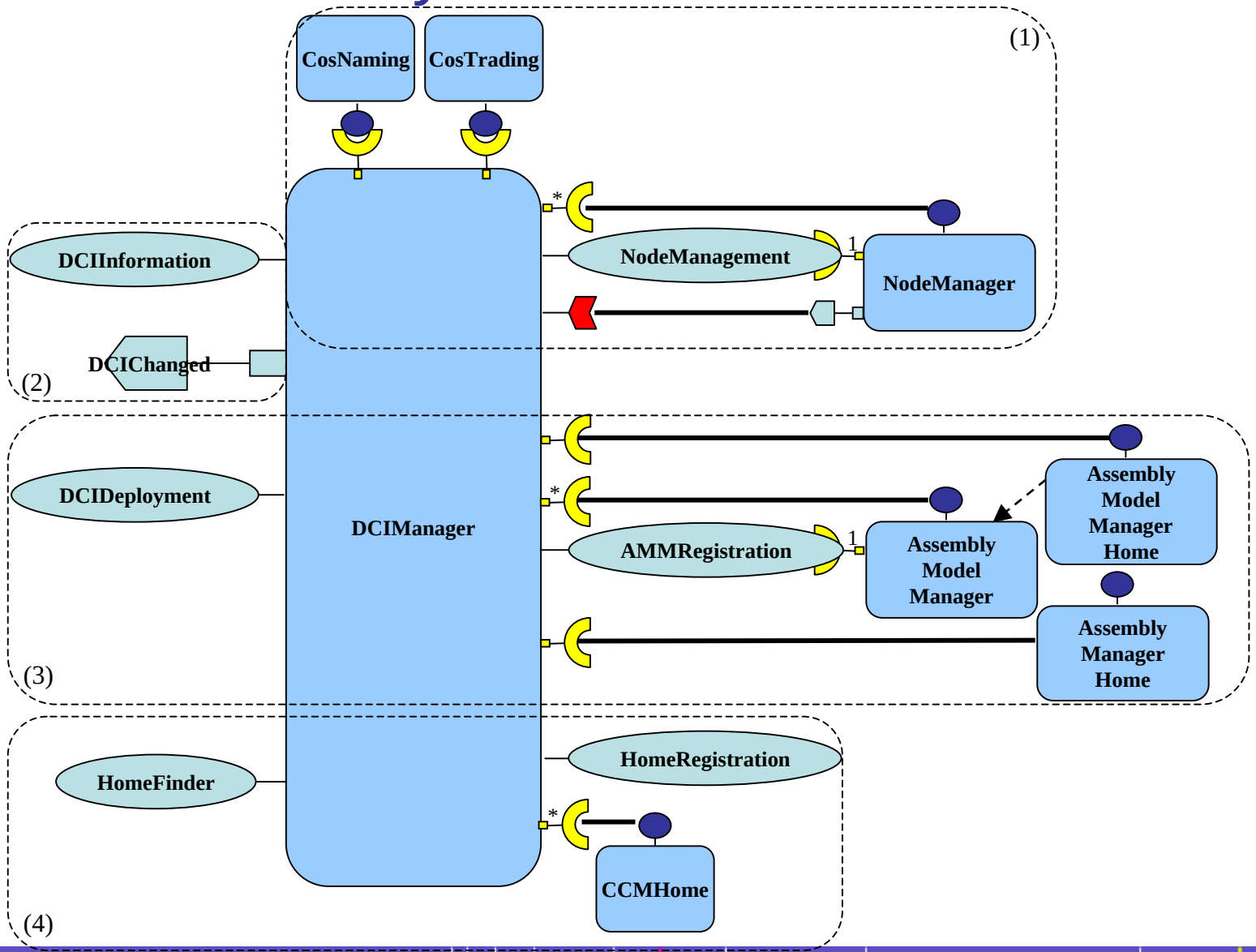
(3) Operating the deployment

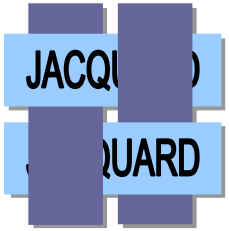
- Installation, update, listing and searching archives
- Creation and destruction of component assemblies

(4) Home Finder service

- Used by an assembly to retrieve pre-deployed CCM Homes

The Domain Layer - CCM View





The Distributed Engine Layer Features

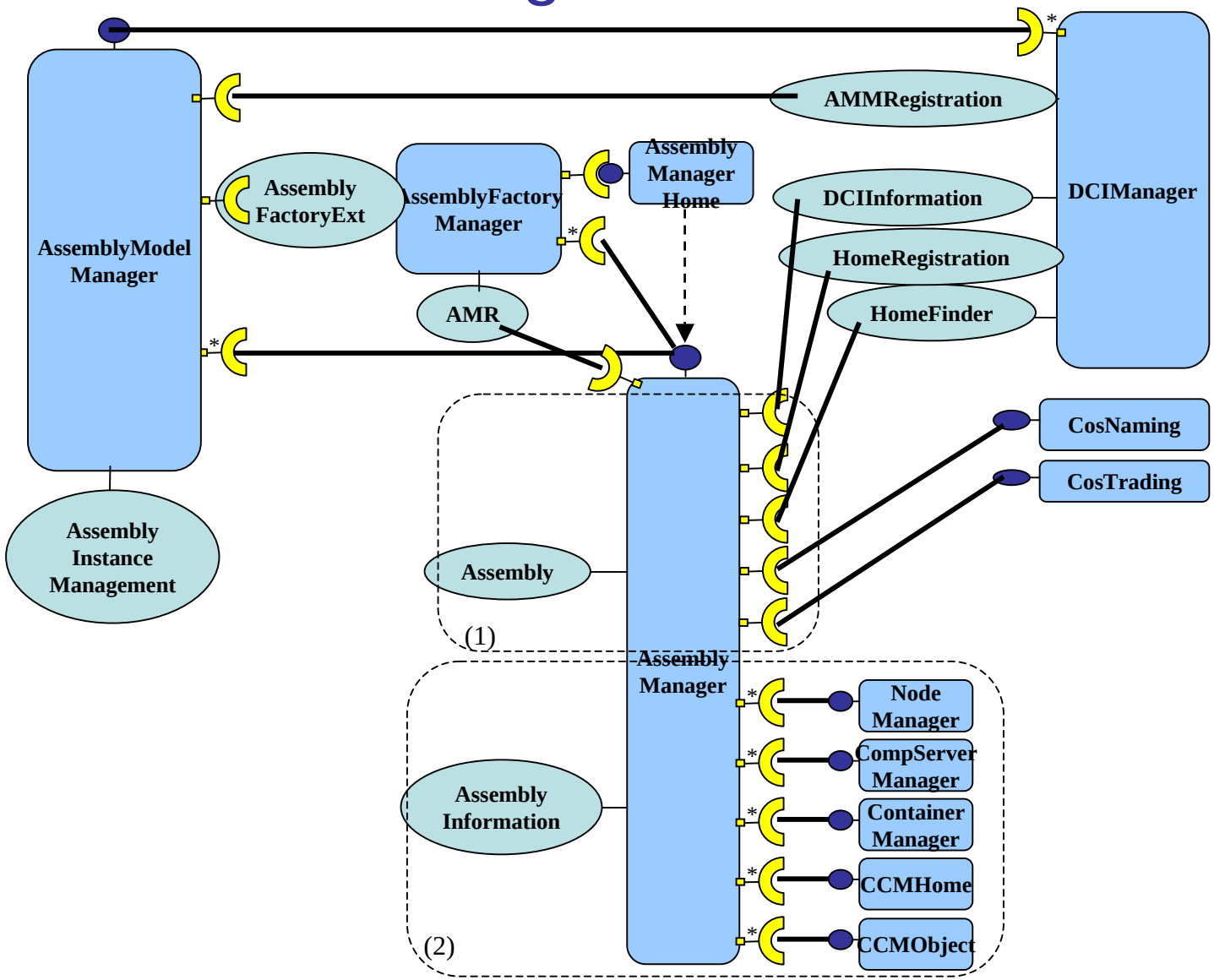
(1) Deployment control via the Assembly interface

- Assembly Manager discovers the Node Layer using the Domain Layer

(2) Introspection of the deployed Assemblies

- Multiple CCM receptacles register each CCM Objects, CCM Homes, Containers, Component Servers and Nodes
- This is *very* important to build a management tool

The Distributed Engine - CCM View



The Node Layer Features

(1) Providing of meta information about the Node's host
- Content includes processor, memory, operating system, available software, system load and network bandwidth ; the XML format allows an exploitation by automatic assignment or load balancing tools

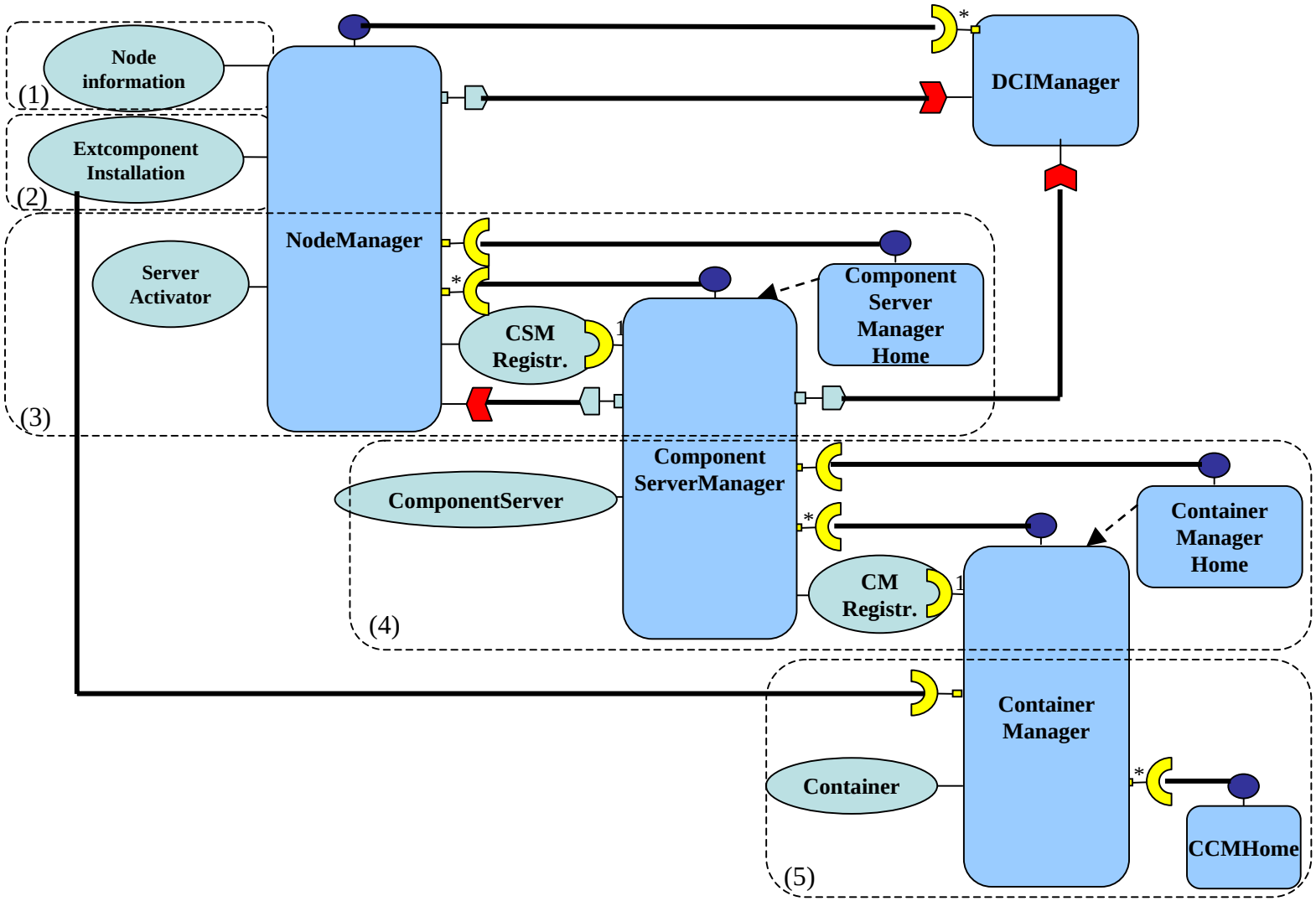
(2) Archives management
- Installation, listing, searching and removal of component archives
- The CCM *pull* operation can be used, or a new *push* (sending) operation

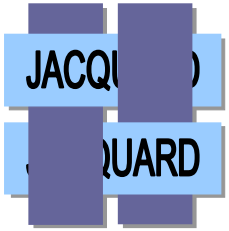
(3) Component Servers management
- Creation, listing, searching and removal of component servers
- The Component Servers register themselves to their parent Node

(4) Container management
- Creation, listing, searching and removal of containers
- The Container register themselves to their parent Component Server

(5) Component Homes management
- Creation, listing, searching and removal of component homes
- The Container gets archives via the ExtComponentInstallation facet of the Node

The Node Layer - CCM View





Outline

- Component-based software deployment
- The CORBA component deployment
  - Architecture & scenario
  - Identified issues
- OpenCCM's Distributed Computing Infrastructure (DCI)
  - Global architecture
  - Benefits from the CCM design
  - The domain layer's features
  - The distributed engine layer's features
  - The target nodes layer's features
- Experimentations
- Conclusion and perspectives

Experimentations

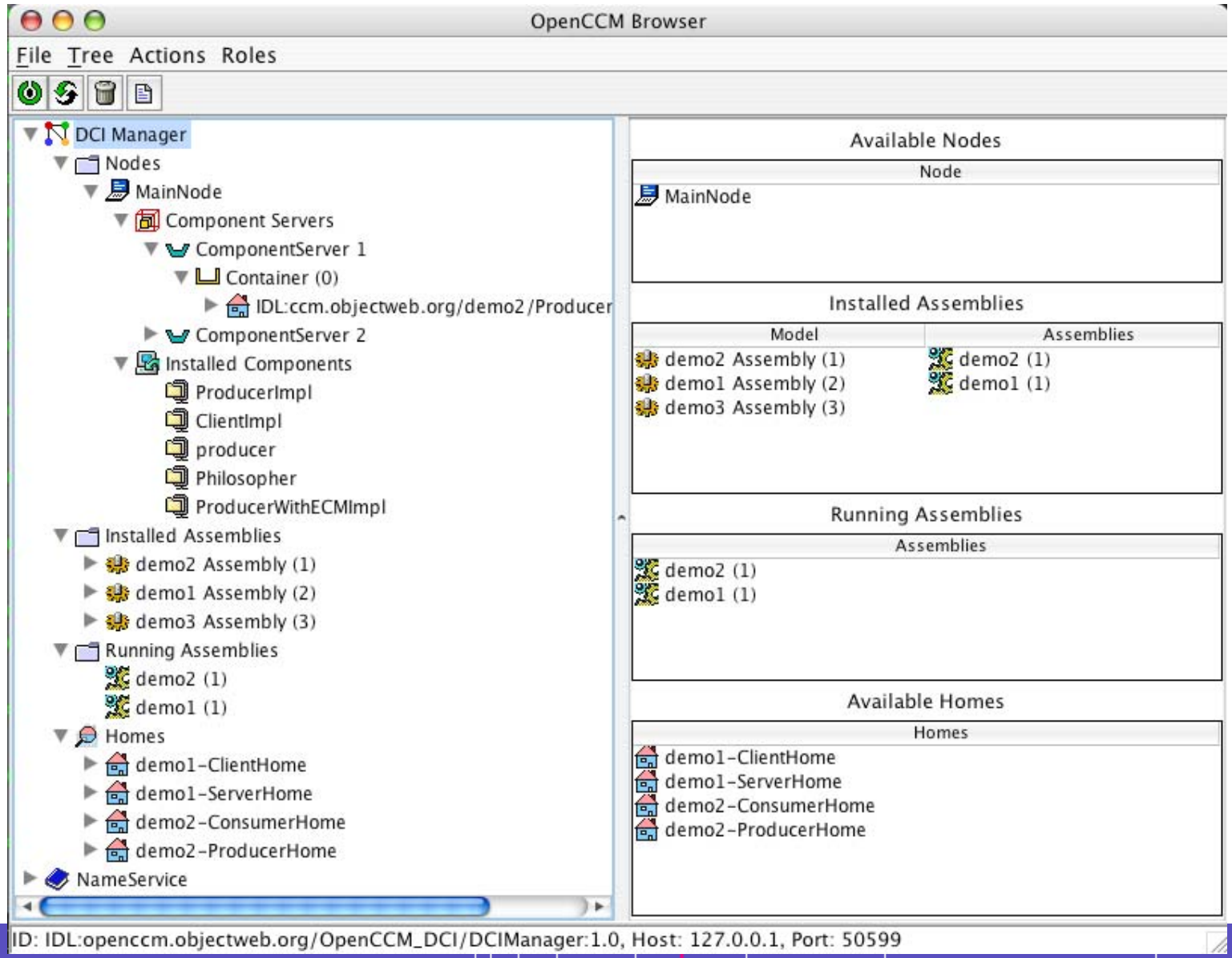
The IST funded COACH project

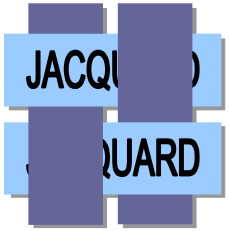
- Definition of the DCI involving the whole consortium
- Implementation in both Java (OpenCCM) and C++ (Qedo)
- Demonstration through a Network Management Framework (NMF)

OpenCCM's DCI supports

- Many ORBs: BES, JacORB, ORBacus, Orbix, OpenORB, ZEN
- Usual operating systems: Linux, Windows, MacOS
- Various devices, including PDAs

Exploration of The DCI

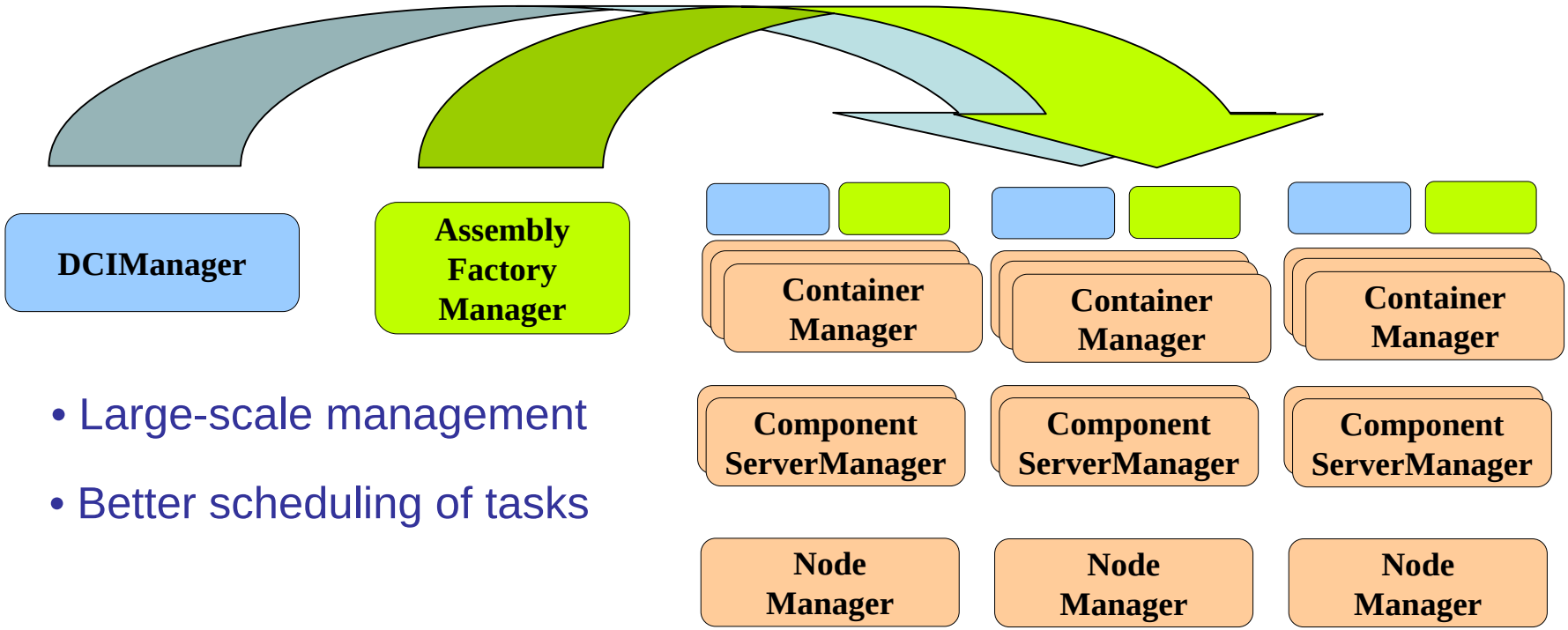




Perspectives

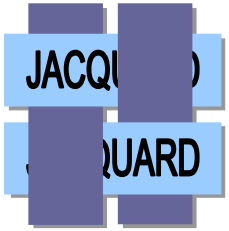
- Identified the need of an optimization of archives downloading
  - Intracom's Network Management Framework (large-scale application made of CCM Assemblies)
- The DCI Manager should be distributed
  - Addressing large scale deployments
  - Switching from centralized management to a peer-to-peer system
- The Assembly Manager should be distributed to
  - Better schedule deployment operations
  - Reduce the number of network operations

Domain & Engine Layers Distribution



- Large-scale management
- Better scheduling of tasks

Idea: Avoid network operations, leaving the nodes in charge of local ops.

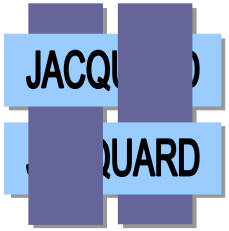


Conclusions

The OpenCCM DCI is an operational CCM deployment platform

Open perspectives will drive our future research works:

- Distributing the layers (modification of the DCI architecture)
- Optimization of network accesses and local threads
- Gain benefits from the CCM Container to inject non functional properties (security, transaction, persistence and so on)
- Make DCI components re-usable to build deployment systems adapted to any other component technology
- Towards an implementation of the OMG's Deployment & Configuration specification



Questions

OpenCCM - The Open CORBA Component Model Platform

http://openccm.objectweb.org

openccm@objectweb.org