Sistemas Distribuídos

Visão Geral Expandida

Visão Geral

- Infra-estrutura
- Ambientes de execução e de programação
- Projeto
- Configuração
- Simulação
- Testes

Visão Geral

- Infra-estrutura
- Ambientes de execuç
- Projeto
- Configuração / *deployment*
- Simulação
- Testes

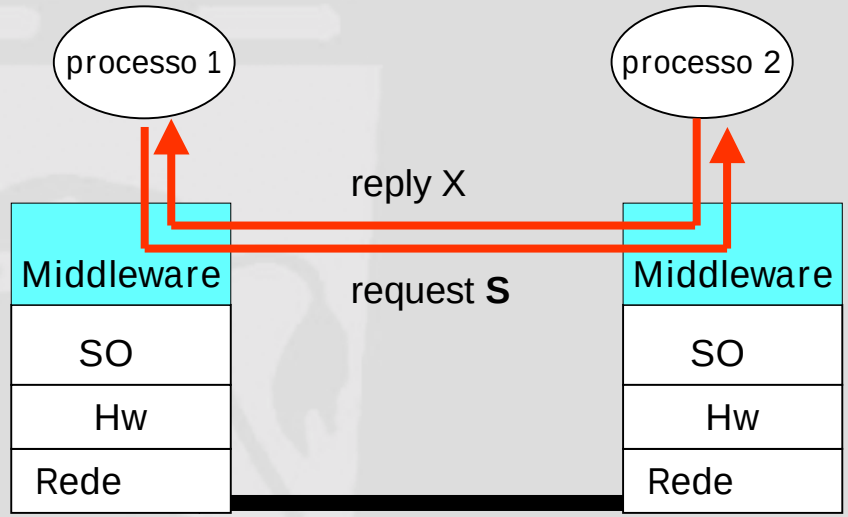
Heterogeneidade

- Redes
- Sistemas Operacionais
- Middleware

O que é um *Middleware* ?

- Camada de software que permite a comunicação entre aplicações (distribuídas)
- Um conjunto de serviços que fornece comunicação e distribuição de forma transparente à aplicação
- Componentes
  - ambiente de programação
  - ambiente de execução

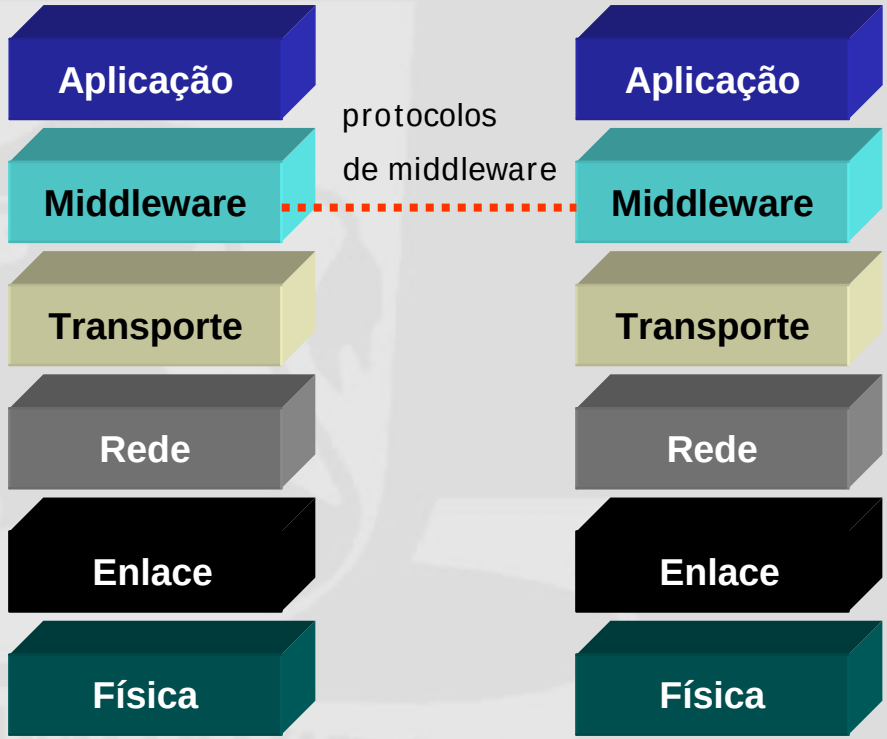
Modelo de Interação



O processo precisa saber o nome do serviço desejado, mas não onde ele se localiza!

Contexto do *Middleware*

- O *middleware* está localizado entre as camadas de aplicação e transporte
- O *middleware* implementa seus próprios protocolos
  - protocolos que suportam os serviços fornecidos pelo *middleware*
    - ex., protocolos de autenticação



Contexto do *Middleware*

- O *middleware* está localizado entre as camadas de aplicação e transporte
- O *middleware* implementa seus próp
  - protocol
    serviços fornecidos pelo *middleware*
    - ex., protocolos de autenticação

O middleware faz o papel da camada de apresentação no RM-OSI.

Serviços Fornecidos pelo Middleware

- Comunicação
  - esconde os detalhes da rede
  - ex., chamada de procedimento remoto, invocação de objeto
- Serviço de nomes
  - ex., páginas brancas
- Transações
  - ex., atomicidade
- Segurança
  - a camada de middleware deve implementar mecanismos de segurança

Serviços Fornecidos pelo Middleware

- Comunicação
  - esconde os detalhes da rede
  - ex., chamada de procedimento remoto, invocação
- Serviço de n
  - ex., págin
- Transações
  - ex., atomicidade
- Segurança
  - a camada de middleware deve implementar de segurança

O serviços fornecidos pelos *middleware* variam de ambiente para ambiente.

Modelos de Middleware

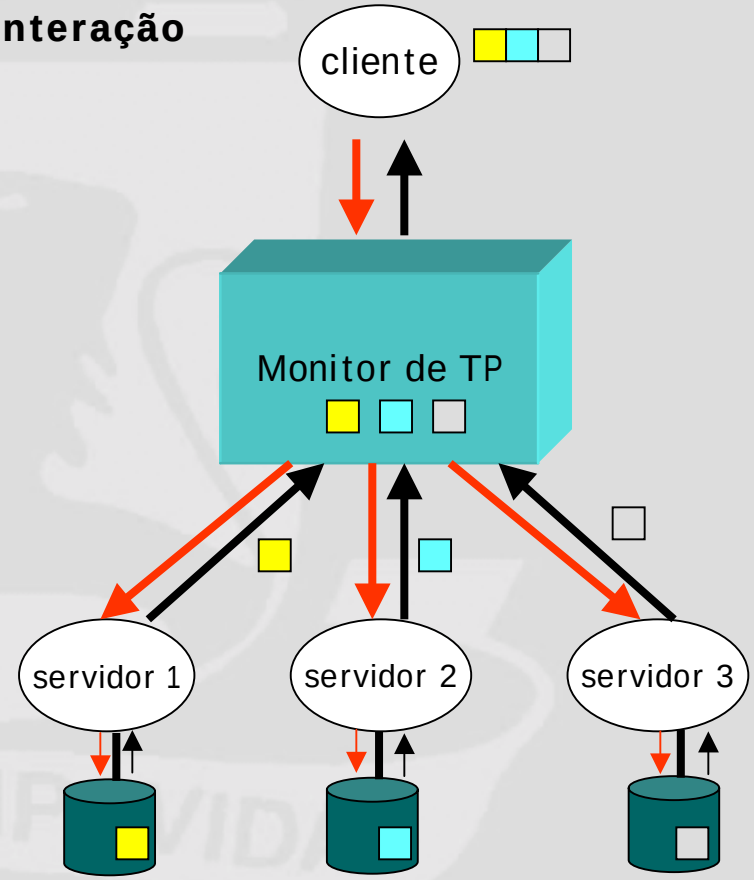
- Parâmetros para classificação dos middleware
  - ambiente de programação
    - ex., tipo de abstração
  - ambiente de execução
    - ex., tipo de comunicação

- Modelos
  - middleware orientado à transação
  - Remote Procedure Call
  - middleware reflexivo
  - middleware baseado em eventos
  - middleware orientado a objetos
  - middleware orientado a mensagem
  - middleware multimídia
  - middleware para sistema móveis

Middleware Orientado a Transação

- Chamada de procedimento remoto integrada + controle de transações
- Serviços
  - comunicação síncrona/assíncrona
  - transação
- Usado em aplicações que demandam rapidez na execução de transações remotas
- Heterogeneidade de linguagens de programação
- ex., BEA Tuxedo, Linda TS, DCE

Modelo de Interação



Middleware Orientado a Mensagem (MOM)

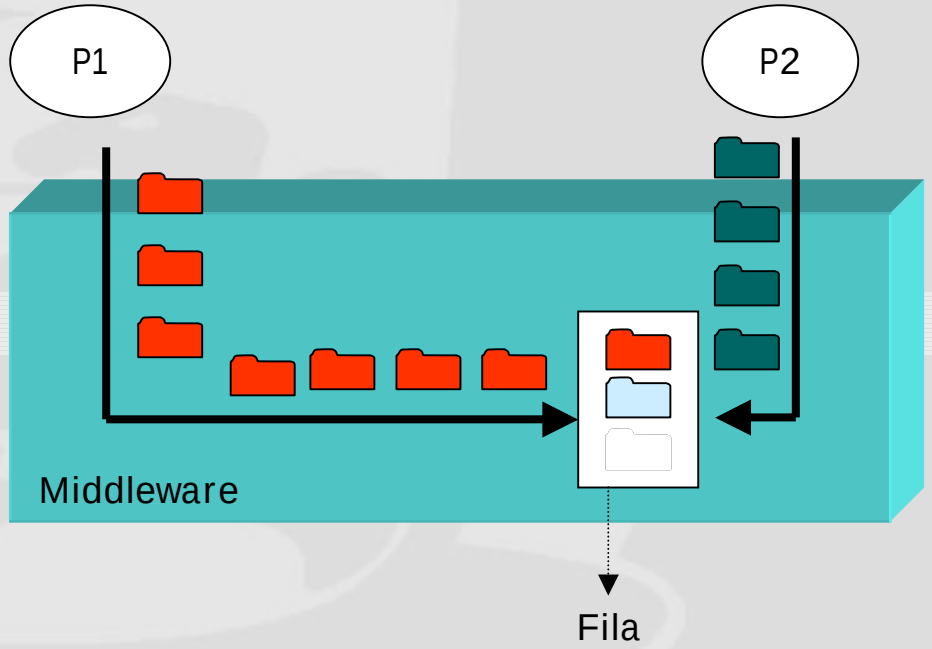
- Comunicação através de mensagens
- Serviços
  - segurança
  - comunicação assíncrona
  - priorização de mensagens
  - replicação
- Não assume um transporte confiável
- ex., MQSeries (IBM), JMS (Sun)

- Variações
  - baseado em fila
  - passagem de mensagem

MOM – Message Queuing

- Fila de mensagens
- Ambientes orientados a transação
- Não há conexão direta entre as partes

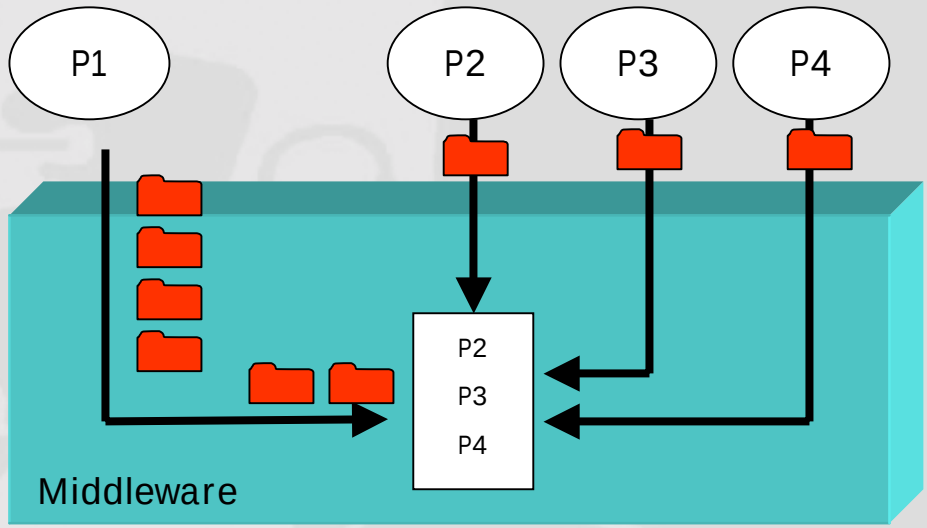
Modelo de Interação



MOM - Publish/Subscribe

- Clientes se registram para receber mensagens (*subscribe*)
- O *middleware* envia mensagens para vários clientes (*publish*)

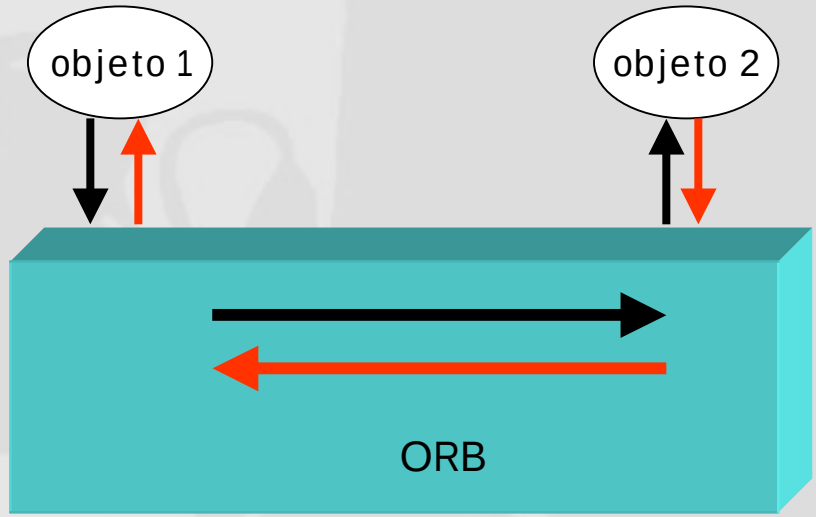
Modelo de Interação



Middleware baseado em Objetos

- Ambiente de programação
  - distribuição + OO = Objetos distribuídos
  - IDLs
- "Barramento" de integração/comunicação
  - ORB (Object Request Broker)
- Assume um transporte confiável
- ex., CORBA (OMG), DCOM (Microsoft), RMI e EJB (Sun)

Modelo de Interação



Middleware para Sistemas Móveis

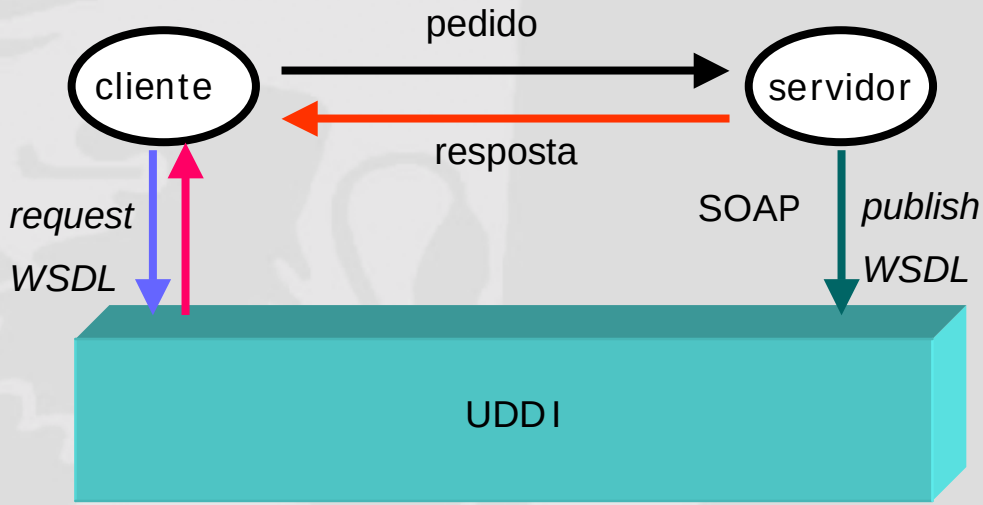
- Carga computacional leve
  - os dispositivos móveis tem limitações de processamento, armazenamento
- Comunicação assíncrona
  - clientes e servidores nem sempre estão conectados
- Reconfiguração dinâmica
  - ex., largura de banda, serviços disponíveis
- Desenvolvedor ciente das mudanças (contexto)

- Tipos de middleware móveis
  - reflexivos
    - reconfiguráveis
    - light-weight
  - context-aware
    - infomações contextuais são fornecidas a aplicação

Middleware para WEB

- *WebServices*
  - HTTP (padrão)
    - ubigüidade
  - XML (padrão)
    - codificação dos dados
  - SOAP – *Simple Object Acess Protocol* (padrão)
    - comunicação (RPC)
    - executa sobre o HTTP
  - WSDL – *Web Service Definition Language* (padrão)
    - o que, onde, como
  - UDDI – *Universal Description Discovery and Integration* (padrão)
    - trader

Modelo de Interação



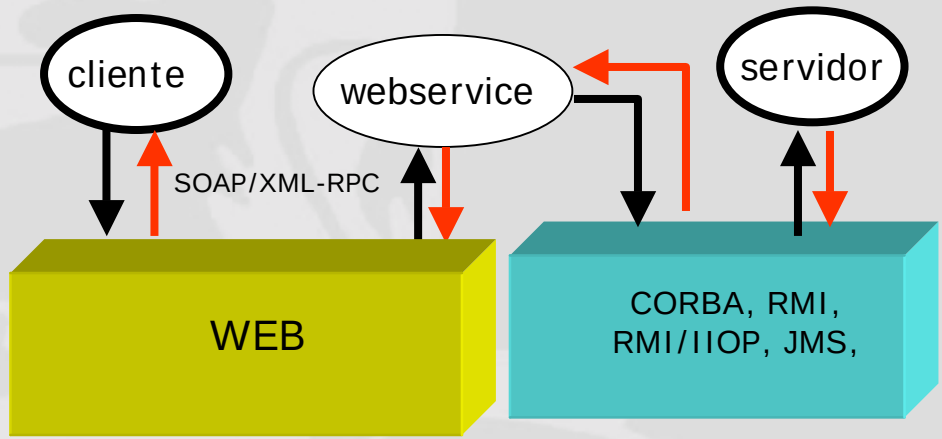
Web Services

- Papéis
  - Service provider
  - Service broker
  - Service requester
- Operações
  - Publish
  - Find
  - Bind

WebServices e Outros *Middleware*

- *WebServices* agem como uma “interface” para acessar os serviços providos por outros *middleware*

Modelo de Interação



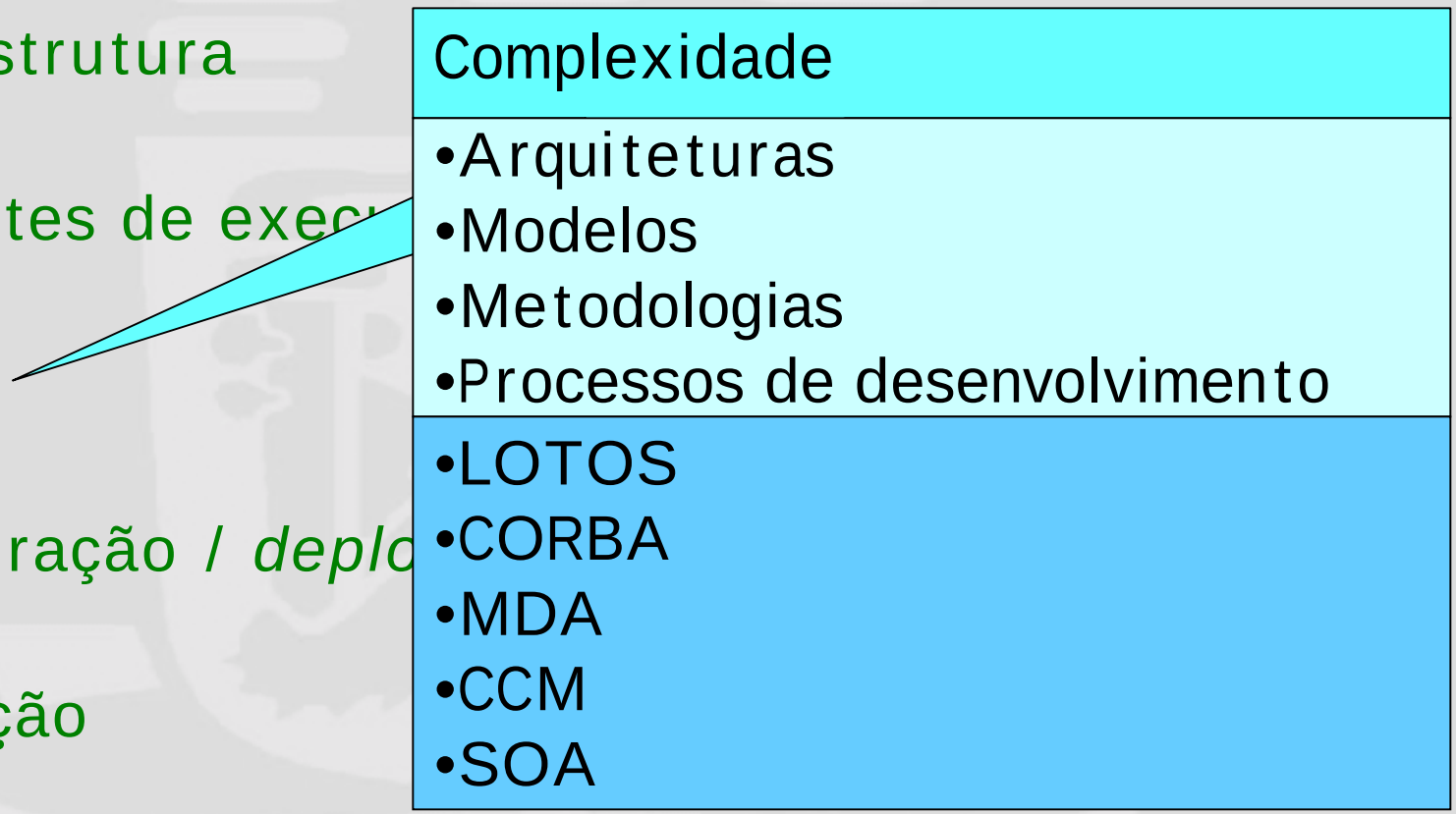
Visão Geral

- Infra-estrutura
- Ambientes de execução e de programação
- Projeto
- Configuração / *deplo*
- Simulação
- Testes

- Produtos CORBA: Orbix, Visibroker, ...
- JavaRMI
- Linguagens de programação, sockets, RPC
- WebServices
- IDL

Visão Geral

- Infra-estrutura
- Ambientes de exec
- Projeto
- Configuração / *deplo*
- Simulação
- Testes



MDA

- Model Driven Architecture
  - PIM: Platforma-Independent Model
    - É mais fácil validar o modelo abstraindo-se de semânticas específicas da plataforma
    - É mais fácil produzir implementações em diferentes plataformas partindo da estrutura essencial e do comportamento preciso do sistema.
    - Integração e interoperabilidade podem ser definidos plataforma
  - PSM: Platform-Specific Model

CORBA

- Common Object Request Broker Architecture

- CORBA tem sido utilizado por um número crescente de aplicações distribuídas (ver “success stories” em http://www.omg.org), devido a características como:
  - Independência de linguagem, sistema operacional, hardware e localização relativa entre cliente e servidor
  - Serviços como *naming*, *trading* e *event notification*, que fornecem reusabilidade
  - Serviços essenciais como os de transações e de segurança

CORBA

- Entretanto, a especificação de CORBA 2.3 não atende bem a pontos tais como:
  - Falta de padrão para implantação (*deployment*) de objetos
  - Falta de suporte para o uso de subconjuntos de funcionalidades “comuns”
    - Por exemplo, um grande número de aplicações utiliza apenas um subconjunto das possíveis configurações do POA e, apesar disso, o desenvolvedor deve conhecer muitas políticas do POA para conseguir o efeito desejado
  - Dificuldade para estender funcionalidades de objetos CORBA – possível apenas através de herança
  - Falta de definição de serviços obrigatórios: a aplicação deve possuir código que trate a indisponibilidade de serviços, por ex.
  - Falta de padrão para gerenciamento de ciclo-de-vida de objetos: apesar da existência do “Lifecycle Service”, seu uso não é obrigatório, o que dificulta o trabalho no cliente

CCM

- Problemas como os citados levam à produção de aplicações com objetos fortemente acoplados (*tightly coupled*)
  - difíceis de modelar, reutilizar, implantar, manter e estender
- Com o objetivo de solucionar esses problemas, a OMG adotou o **CORBA Component Model** (**CCM**) como parte da especificação de CORBA 3.0
- CCM estende o modelo de objetos de CORBA, definindo funcionalidades e serviços que permitem que o desenvolvedor implemente, gerencie, configure e implante componentes que integrem os serviços de CORBA, como segurança, transações, persistência e eventos, em um ambiente padronizado
- CCM permite maior reutilização de servidores e mais flexibilidade para configuração dinâmica de aplicações CORBA

SOA

- "The **Service Oriented Architecture** (**SOA**) is a distributed software model. The primary characteristics of a SOA are:
  1. Services are used to divide larger applications into smaller discrete modules
  2. Services are integrated via service composition mechanisms to create larger applications
- It is said that a SOA is usually comprised of three primary parties:
  - A Producer (of services)
  - A Consumer (of services)
  - A Directory (of services)
- WebServices are considered an example of a SOA. Service Networks take on the properties of a SOA."

SODA

- "**Service-Oriented Development of Applications** (**SODA**) is a new style of developing software to work specifically within Service Oriented Architectures (SOA)
- SOAs represent a collection of loosely coupled heterogeneous components that can be easily snapped together using WebServices
  - The result is
    - enhanced developer productivity,
    - code reuse and
    - business agility."

SOA - referências

- http://www.serviceoriented.org/service_oriented_architecture.html
- http://www.gartner.com > Application Development > Application Architecture > Service-Oriented Architecture > Introduction to Service-Oriented Architecture > $$$
- The Benefits of a Service-Oriented Architecture
- http://www.2gamma.com

Projeto

- Proposta: 15/01/2004
- Quantas equipes?
- Próxima aula: discussão sobre os projetos em minha sala