

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

Sistemas Distribuídos

Introdução

Definição de Sistemas Distribuídos (1)

- Sistema no qual **componentes** localizados em uma rede de computadores **comunicam e coordenam suas ações** somente **através de passagem de mensagens**.

Questões relacionadas à definição de Sistemas Distribuídos

- Concorrência
- Não há relógio global
- Falhas são independentes

Motivação

- Compartilhamento de recursos
- Elevada disponibilidade de serviço
- Extensibilidade
- Desempenho
- Suporte a organizações distribuídas
- Interação e coordenação entre *smart devices*
- Comunicação

Figure 1.1 (see book for the full text)

Selected application domains and associated networked applications

<i>Finance and commerce</i>	eCommerce e.g. Amazon and eBay, PayPal, online banking and trading
<i>The information society</i>	Web information and search engines, ebooks, Wikipedia; social networking: Facebook and MySpace.
<i>Creative industries and entertainment</i>	online gaming, music and film in the home, user-generated content, e.g. YouTube, Flickr
<i>Healthcare</i>	health informatics, on online patient records, monitoring patients
<i>Education</i>	e-learning, virtual learning environments; distance learning
<i>Transport and logistics</i>	GPS in route finding systems, map services: Google Maps, Google Earth
<i>Science</i>	The Grid as an enabling technology for collaboration between scientists
<i>Environmental management</i>	sensor technology to monitor earthquakes, floods or tsunamis

Tipos de Sistemas Distribuídos

- Computação Distribuída de Alto Desempenho
- Sistemas de Informação Distribuída
- Sistemas Pervasivos

Tipos de Sistemas Distribuídos

- Computação Distribuída de Alto Desempenho
 - Cluster computing
 - Grid computing
 - Cloud computing
- Sistemas de Informação Distribuída
 - Processamento de transações distribuídas (DTP)
 - Integração de aplicações empresariais (EAI)
- Sistemas Pervasivos
 - Computação ubíqua e móvel
 - Sistemas sensíveis ao contexto
 - Redes de sensores

Cluster computing

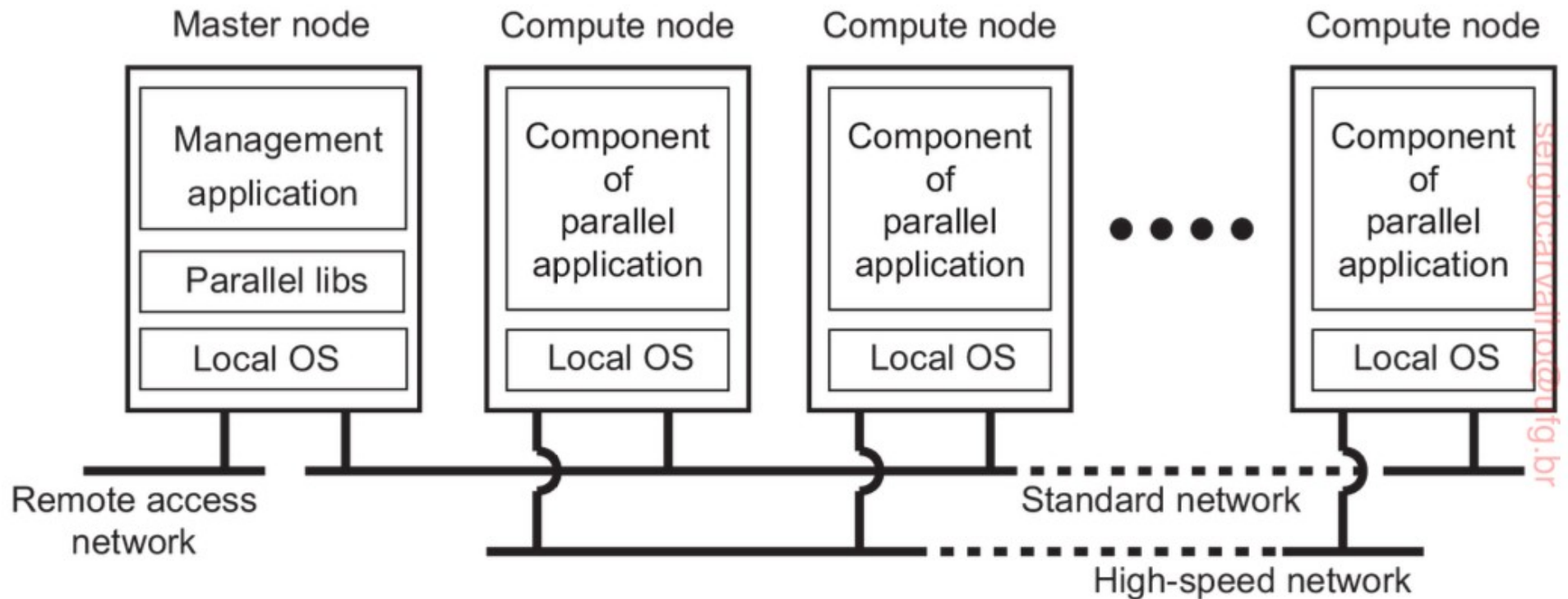


Figure 1.7: An example of a cluster computing system.

Cloud computing

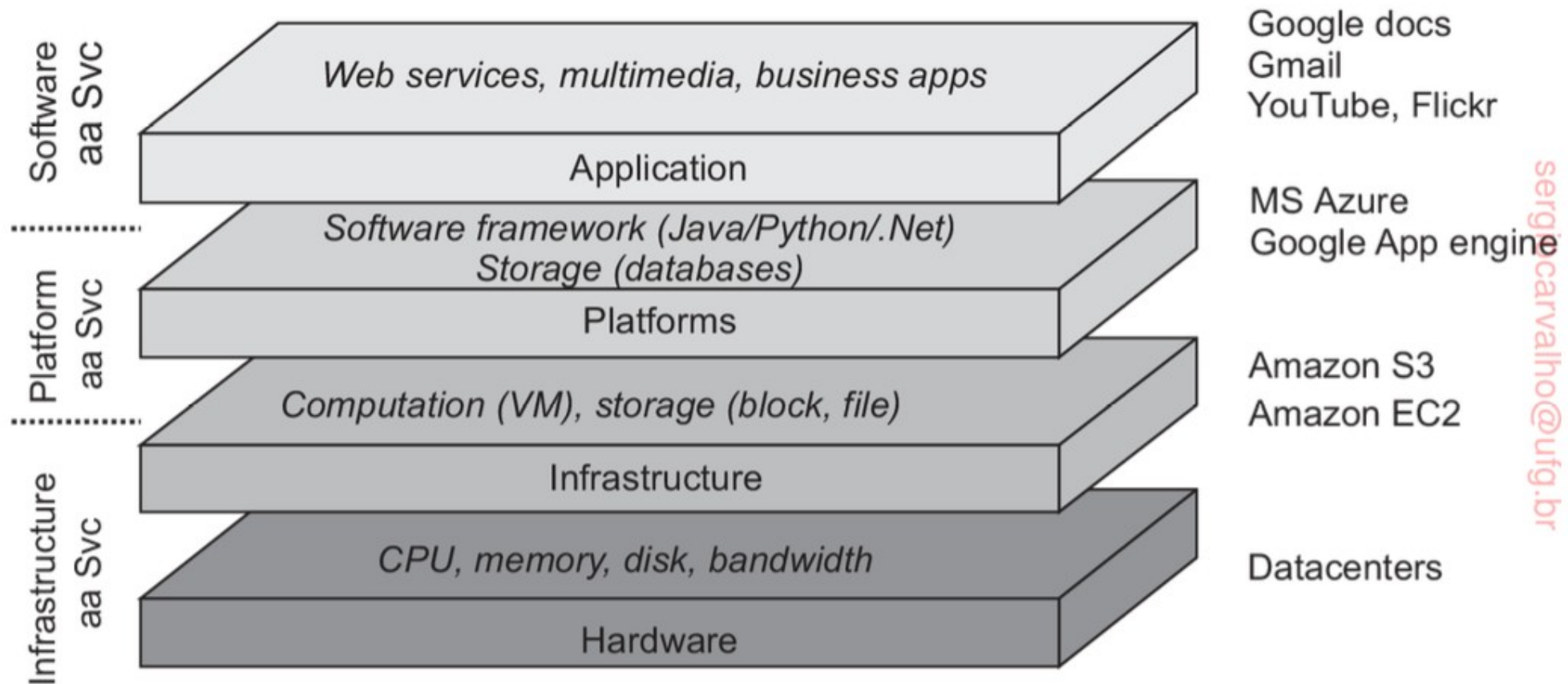


Figure 1.9: The organization of clouds (adapted from Zhang et al. [2010]).

Transações Distribuídas

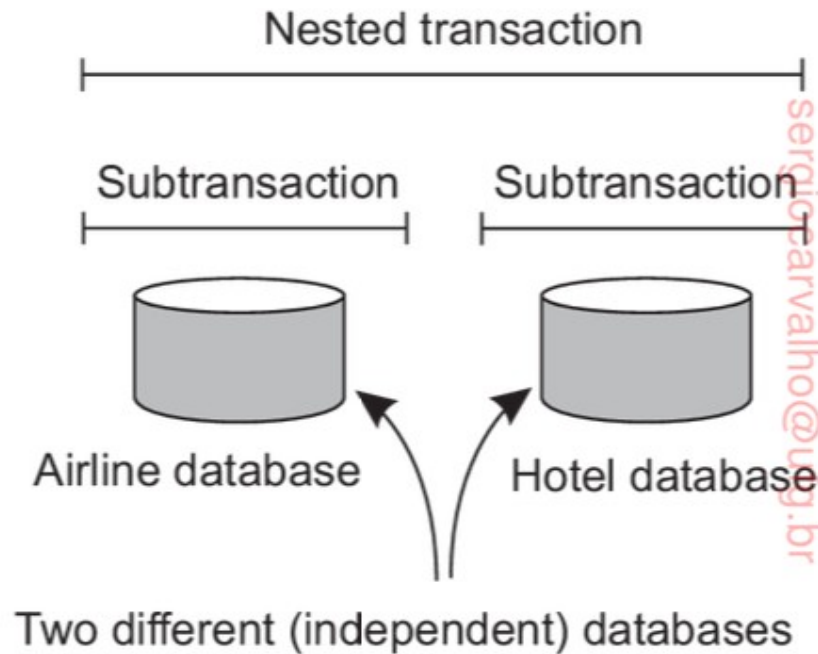


Figure 1.11: A nested transaction.

Transações Distribuídas

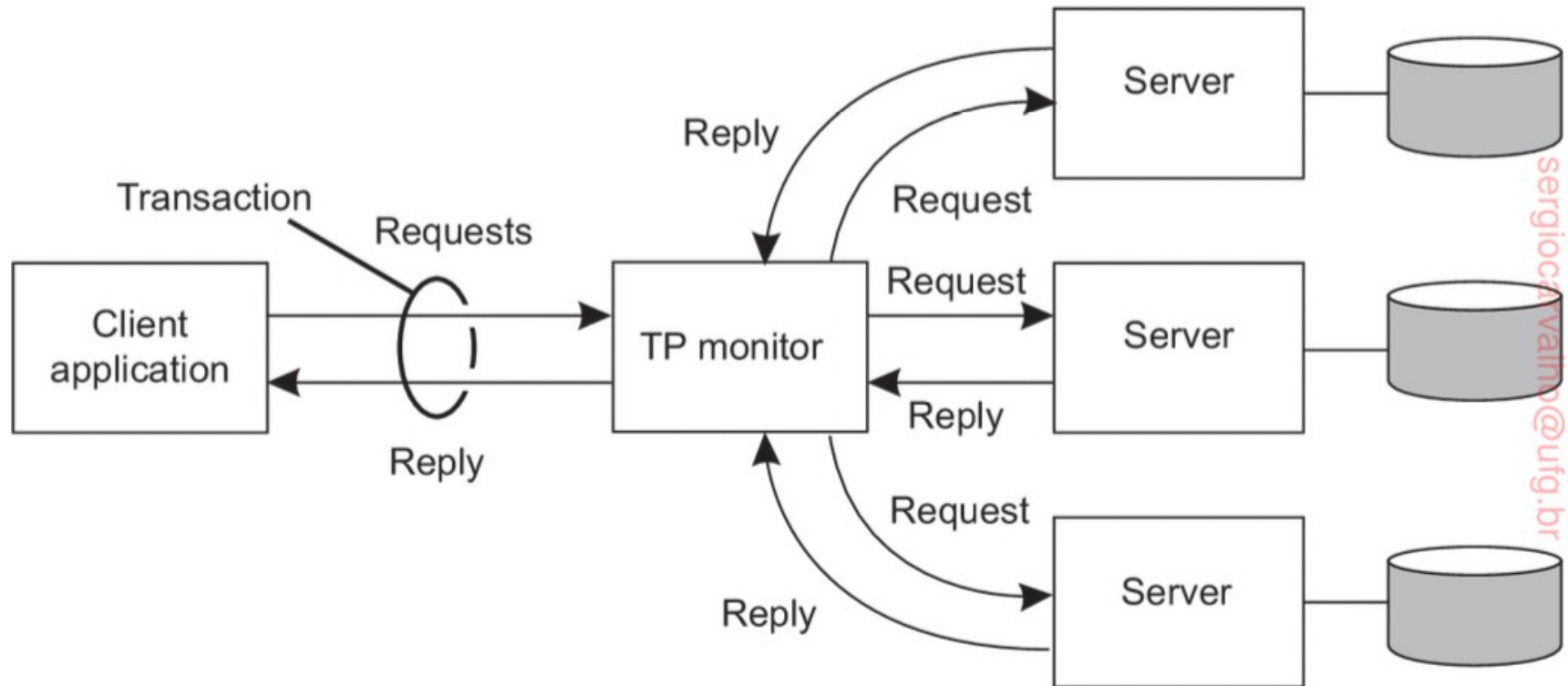


Figure 1.12: The role of a TP monitor in distributed systems.

Integração de Aplicações Empresariais (EAI)

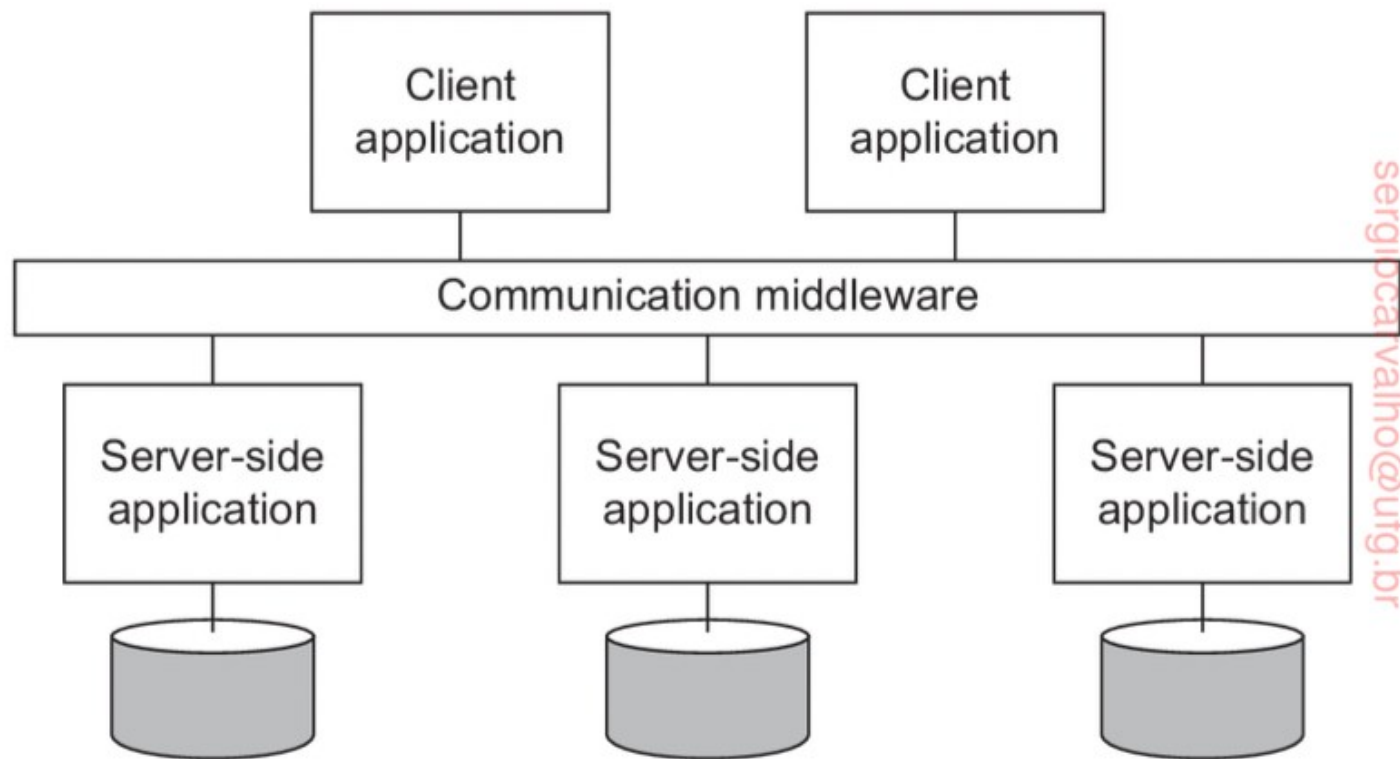


Figure 1.13: Middleware as a communication facilitator in enterprise application integration.

Principais elementos na definição de Sistemas Distribuídos

- Infraestrutura para:
 - aplicações
 - serviços
- Estrutura física
 - múltiplos computadores conectados em rede
- Autonomia
- Sem memória compartilhada
 - sistema fracamente acoplado
- Sem relógio global
- Comunicação através de passagem de mensagens
 - tipicamente assíncronas
- Cooperação e coordenação

Exemplo de Sistema Distribuído: A Web

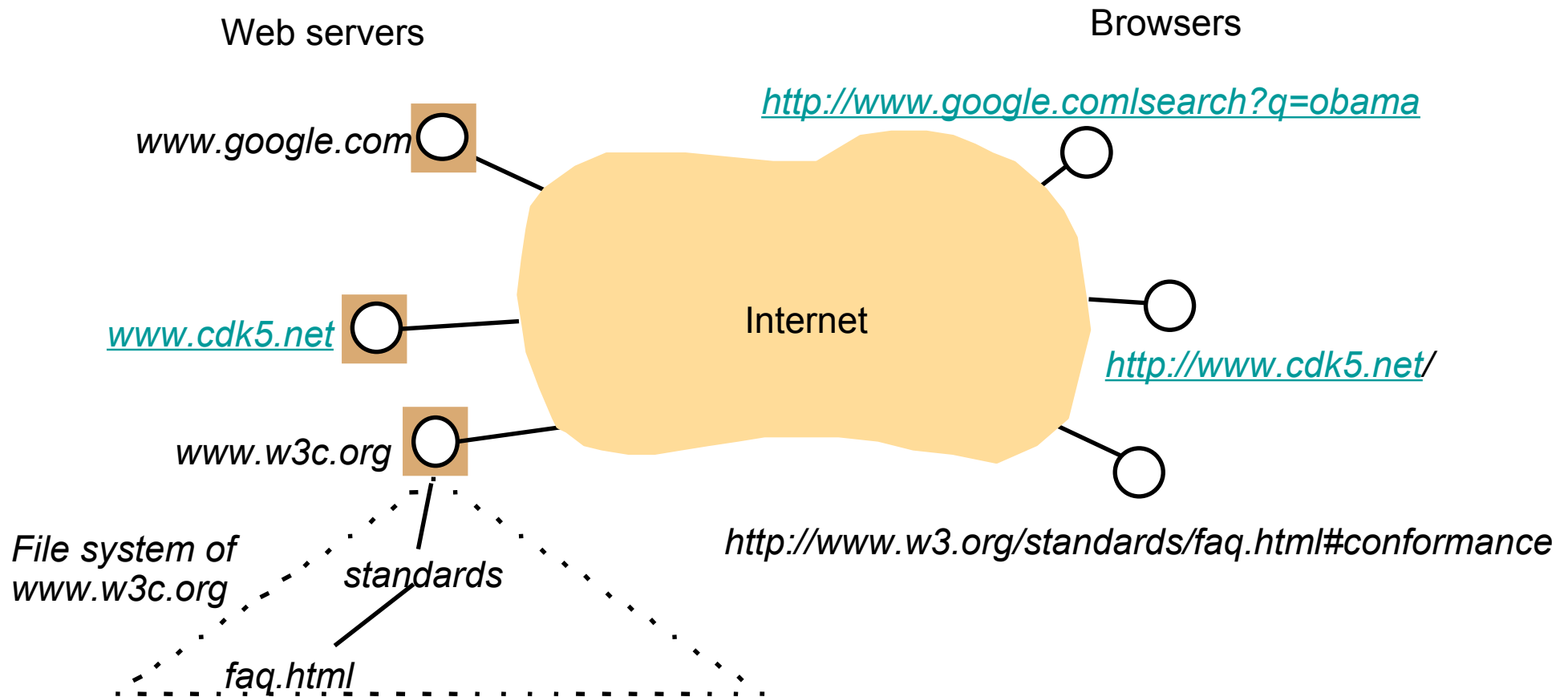
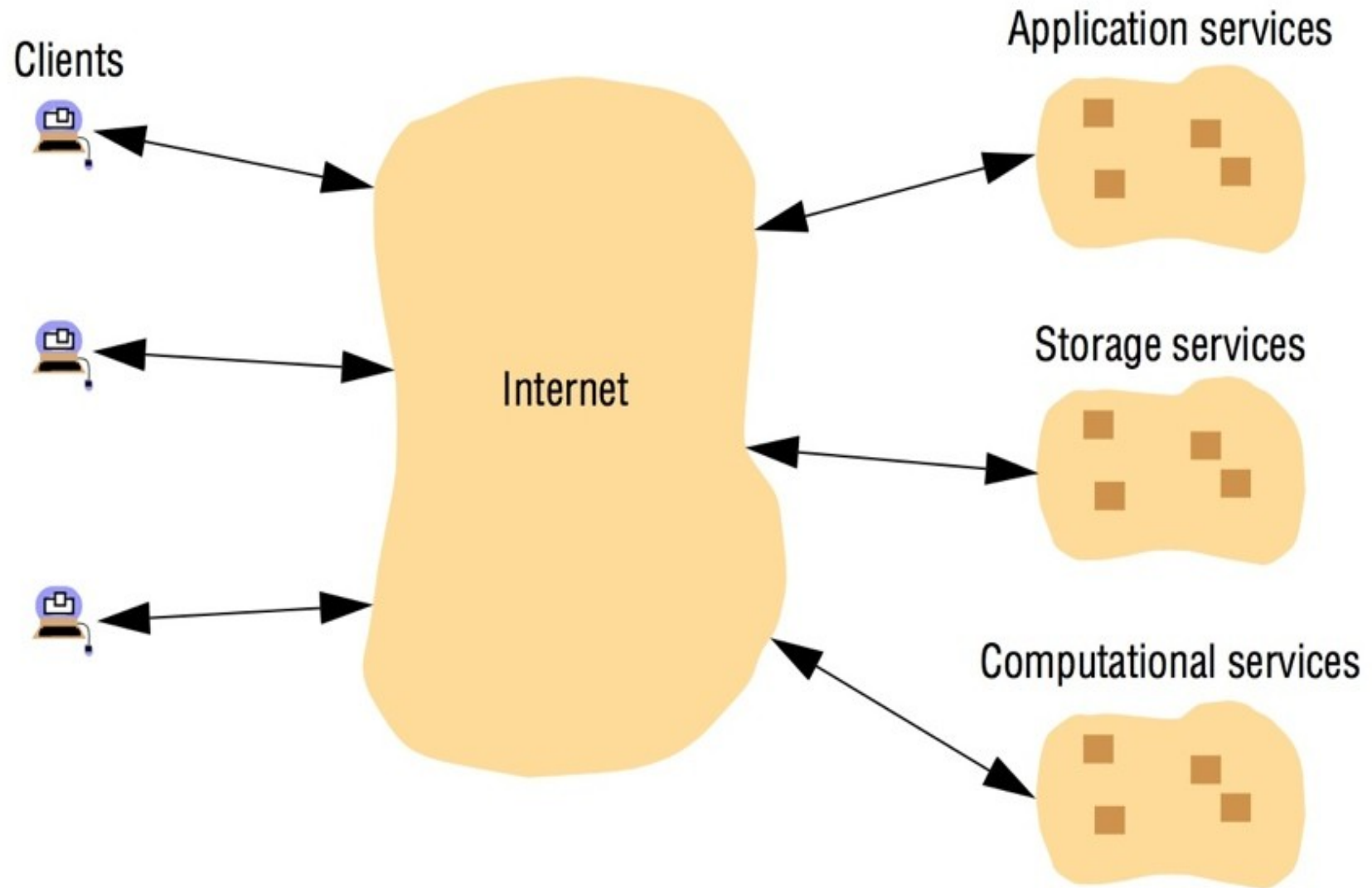


Figure 1.5

Cloud computing



Tendências

- Rede em todos os lugares e a Internet
- computação móvel e ubíqua
- computação em grade
- redes de sensores
- Computação distribuída como uma *utility* (computação em grade e em nuvem)
- Model-Driven Middleware
- Adaptive and Reflective Middleware
- IoT
- Autonomic Computing
- Systems of Systems

Model-Driven Middleware

Providing the ability for users to define their own applications in the form of high-level models that are executed by a middleware-based execution engine

This approach is enabled by restricting the scope to a particular application domain

Each domain has its own modeling language and the middleware is engineered to support that language

Use of models at runtime

- to describe end-user applications
- to describe the internal state, behavior and structure of the middleware



Adaptive Middleware

Middleware that is capable of dynamically adapting its structure and behavior to meet changing user requirements and varying conditions of the runtime environment

Reflection: the system maintains a representation of itself

- This self-representation can be manipulated at runtime and is causally connected with the system itself – if one changes, the other changes accordingly



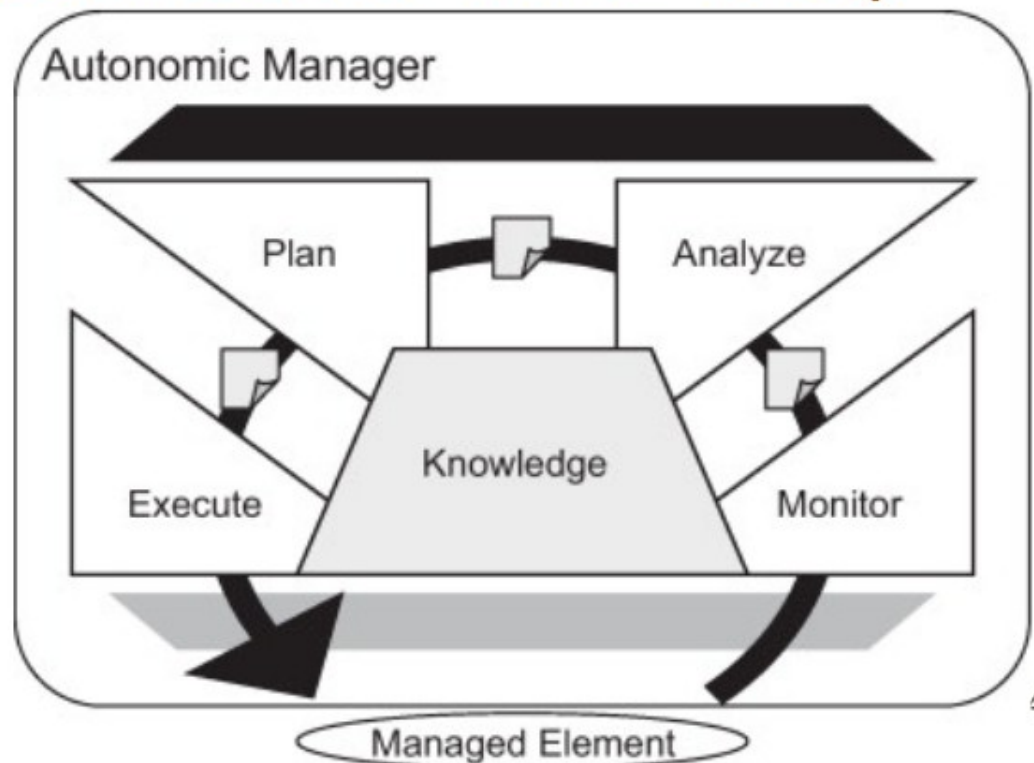
Autonomic Computing

System components are capable of self-management

- self-healing, self-optimization, self-configuration, self-adaptation, self-protection (self-* properties)

Autonomic computing architecture: MAPE-K loop

- Monitoring
- Analysis
- Planning
- Execution
- Knowledge



IoT

Consequence of research and development in the areas of ubiquitous computing, ubiquitous networking, embedded systems, wireless sensor networks, software-intensive systems

Internet of Things: focus on objects with unique IDs and Internet presence

Cyber-physical Systems: focus on the seamless integration of physical and digital infrastructures

Essentially two views of the same concept

Systems of Systems

Systems of Systems: complex systems that can be composed, either statically or on demand, from a combination of other independent systems in order to accomplish tasks that require more complex capabilities than the simple sum of the constituent systems

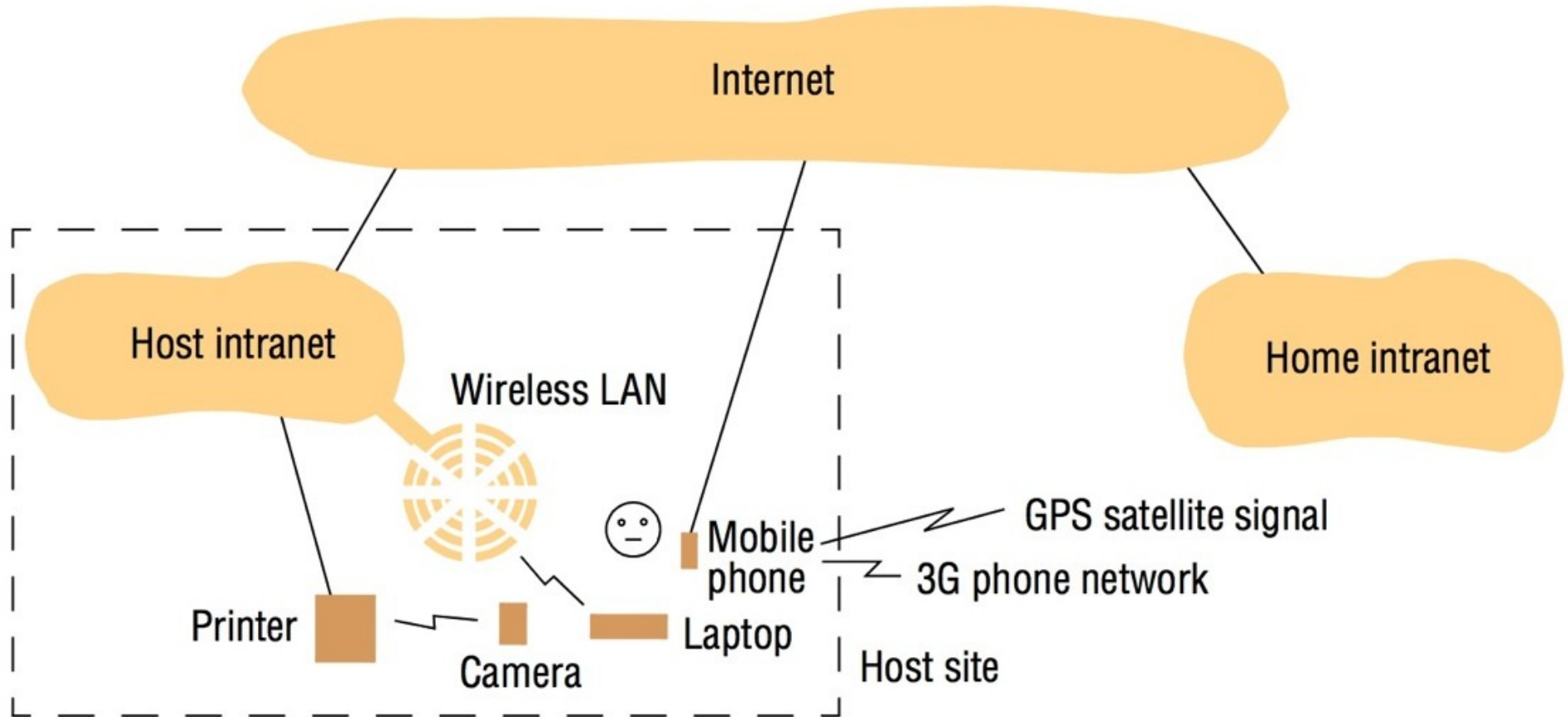


Principais Desafios em SD

Desafios que um desenvolvedor deve considerar no desenvolvimento de um sistema distribuído

- Heterogeneidade
- Abertura (openness)
- Segurança
- Escalabilidade
- Tratamento de falhas
- Concorrência
- Transparência

Exemplo de ambiente heterogêneo



Escalabilidade

- Um sistema distribuído é escalável se ele permanece funcionando adequadamente à medida que novos componentes ou parâmetros (p.ex. usuários) são acrescentados.
- Escalabilidade sempre é estabelecida
 - Em função de um componente ou parâmetro particular
 - Dentro de um limite considerado aceitável para um cenário.
- Em geral, é usado como critério de comparação entre sistemas, ao invés de valor absoluto.

Escalabilidade

- Limitações à escalabilidade

Conceito	Exemplo
Serviços centralizados	Um único servidor para todos os usuários
Dados centralizados	Uma única agenda on-line
Algoritmos centralizados	Execução de roteamento baseada em informação completa de todas as rotas

Tratamento de falhas

- Qualquer processo, computador ou rede pode falhar independentemente de outros.
- Todos os componentes precisam estar **conscientes da sua dependência** de outros que podem falhar e **tratar cada falha apropriadamente**.
- Técnicas comuns:
 - Detectar falhas
 - Mascarar falhas
 - Tolerar falhas
 - Recuperação de falhas
 - Redundância

Concorrência

- Em SDs, serviços e aplicações podem ser compartilhados com clientes ao mesmo tempo
- O que acontece caso mais de um cliente tente modificar uma estrutura de dados compartilhada na rede?
- Sistema deve ser capaz de lidar com essa situação e manter o componente compartilhado em estado consistente (*sincronização*)
- Exemplo: acesso distribuído a BD

Transparências de distribuição

- Tornar *invisíveis* as complicações geradas pela distribuição:
 - para o programador de aplicações distribuídas
 - para o usuário
 - para o administrador do sistema
- Principais tipos de transparência
 - acesso, localização, falha, migração, relocação, replicação, persistência, transação, concorrência

Principais transparências de distribuição

- **Acesso:** recursos locais e remotos são utilizados com as mesma operações
- **Localização:** recursos são acessados sem o conhecimento da sua localização física ou de rede
- **Concorrência:** diversos processos operam concorrentemente sobre um mesmo recurso compartilhado sem interferência entre eles
- **Replicação:** múltiplas instâncias de recursos são mantidas sem que seja necessário conhecimento dos usuários ou programadores

Principais transparências de distribuição

- **Falha:** falhas são mantidas escondidas dos programadores e usuários, que acessam com sucesso os recursos
- **Mobilidade:** permite a movimentação de recursos ou clientes na rede sem que isso afete o funcionamento do sistema
- **Desempenho:** sistema é reconfigurado quando a sua carga aumenta
- **Escala:** permite ao sistema ser expandido em escala, sem que seja necessário a modificação da estrutura ou algoritmos.

Aplicações

- Em princípio, qualquer aplicação convencional pode ser portada com sucesso para um ambiente de sistema distribuído
- Algumas áreas de aplicação emergentes
 - multimídia distribuída
 - espaços ativos (smart spaces, smart cities,...)
 - disseminação de informações
 - computação móvel e ubíqua
 - computação em grade/nuvem
 - redes de sensores
 - IoT/loE

Leitura

Capítulo 1 do Distributed Systems, 5th edition,
George Coulouris

Capítulo 1 do Distributed Systems, 2th edition,
Tanenbaum

Créditos

Prof. Sérgio T. Carvalho
sergio@inf.ufg.br