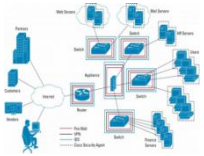
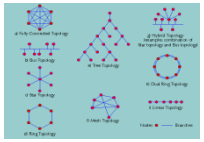




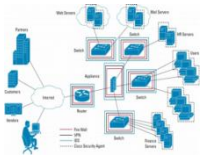
Modelos e Arquiteturas de Sistemas Computacionais

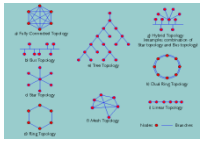
DAS5316 – Integração de Sistemas Corporativos

Prof. Ricardo J. Rabelo

UFSC – Universidade Federal de Santa Catarina
DAS – Departamento de Automação e Sistemas

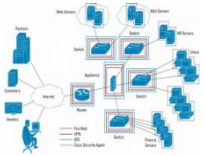
SUMÁRIO

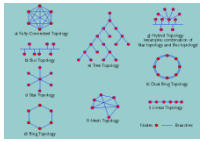




- Importância da definição da Arquitetura
- Tipos & Modelos de Arquiteturas
- Caracterização das Arquiteturas
- Vantagens e Limitações das Arquiteturas
- Comentários Finais

IMPORTÂNCIA DA DEFINIÇÃO DA ARQUITETURA

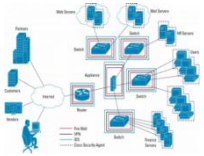


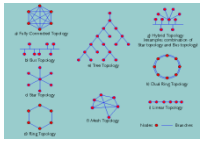


Quando da definição das estratégias de integração dos vários sistemas de uma empresa, é essencial que se conheçam como estes sistemas encontram-se implantados.

No escopo desta disciplina e do engenheiro de automação, interessa saber as arquiteturas típicas existentes e com as quais um dado sistema a ser integrado deverá ter que lidar para poder interoperar com os demais (instalados na própria empresa, ou em outras empresas).

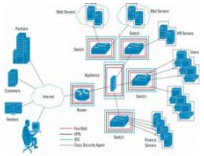
IMPORTÂNCIA DA DEFINIÇÃO DA ARQUITETURA

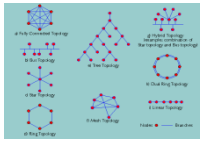




Importante esclarecer que não se trata de reapresentar conteúdos de redes e de sistemas distribuídos, mas, na ótica do especificador e projetista da aplicação de automação, analisar o impacto que a opção ou imposição de uso de uma dada arquitetura tem em termos de integração.

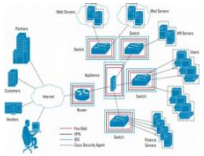
IMPORTÂNCIA DA DEFINIÇÃO DA ARQUITETURA

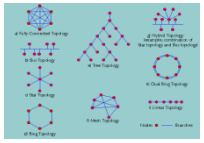




Esta definição impacta a *qualidade* do projeto global de sistemas, até porque uma mesma empresa, ou uma mesma ferramenta, pode ter que conviver e interoperar com sistemas disponibilizados em mais do que um tipo de arquitetura.

IMPORTÂNCIA DA DEFINIÇÃO DA ARQUITETURA

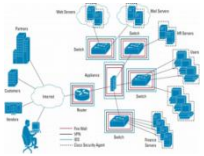


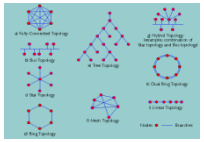


Essa qualidade geral pode ser vista sob algumas dimensões – ***não funcionais*** – de análise, ...

... isto é, de aspectos que não têm a ver diretamente com o que cada aplicação / sistema faz, mas sim com o como ele deverá estar integrado e, a partir disso, com o quão rápido, custoso, seguro, etc. o sistema integrado executará suas funções.

IMPORTÂNCIA DA DEFINIÇÃO DA ARQUITETURA

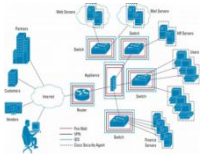


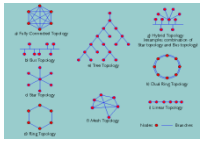


Neste sentido ...

- tanto a estratégia considerada tecnicamente mais adequada de integração pode esbarrar em custos financeiros e de segurança, por exemplo, ...;
- ... como a necessidade de se atingir certos requisitos funcionais e não funcionais de uma dada aplicação pode incorrer na necessidade de se modificar a arquitetura (ou parte dela) de sistemas da empresa.

IMPORTÂNCIA DA DEFINIÇÃO DA ARQUITETURA

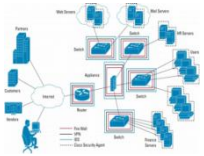


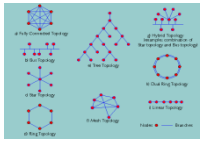


Por exemplo, ...

Qual deverá ser o melhor modelo de licença de software para o sistema ERP ? Por Licença de Software ou por Pagamento por Acesso ? Como deve ser concebida a abordagem de interoperação se parte dos sistemas com quais o ERP terá que trocar informações está num computador central, tipo *mainframe*, e outra parte está em sistemas *Peer-to-Peer* (P2P), instalados em múltiplos servidores ? Qual é o impacto da decisão em termos de necessidade de grande disponibilidade dos sistemas ?

TIPOS & MODELOS DE ARQUITETURAS

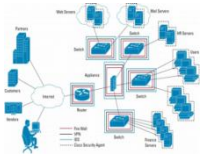


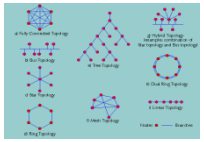


Do ponto de vista de como uma dada aplicação enxerga as outras com as quais deverá interoperar, existem quatro modelos básicos:

- ***Centralizado;***
- ***Cliente / Servidor "Clássico";***
- ***Provedores de Serviços de Aplicação (ASP);***
- ***Software-como-um-Serviço (SaaS) e Nuvem.***

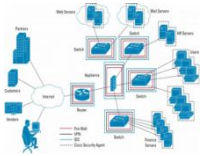
Esses quatro modelos diferentes não são mutuamente exclusivos, podendo coexistir numa mesma empresa ou arquitetura global de integração.

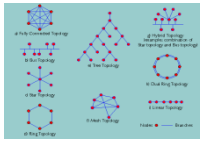




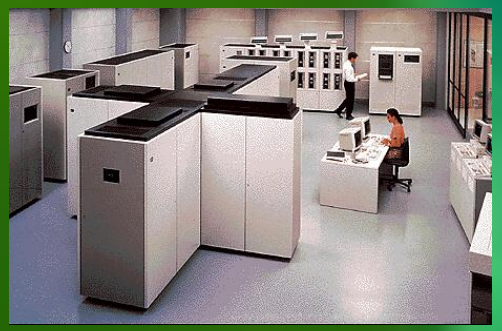
MODELO CENTRALIZADO

MODELO CENTRALIZADO



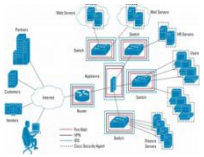


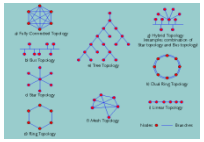
O modelo Centralizado é aqui representado pelo seu extremo, que é a situação onde os sistemas da empresa (ou parte deles) estão implantados em *mainframes* ou em alguns poucos servidores de grande poder de processamento.



Os *mainframes* são computadores de grande porte que centralizam a hospedagem e a execução de aplicações, assim como a persistência de dados, e são ainda comuns em algumas empresas de grande porte.

MODELO CENTRALIZADO



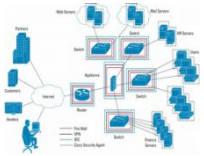


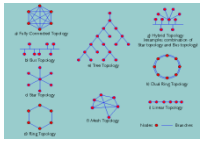
Mainframe & BD

rede

cliente

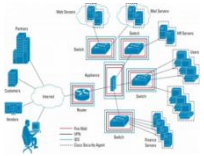
MODELO CENTRALIZADO

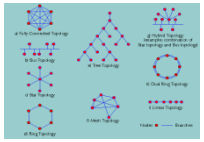




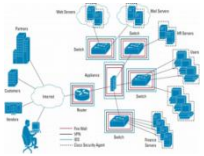
- Neste modelo, todo o processamento é central e existe um repositório de dados também central, tipicamente chamado de banco de dados corporativo.
- Não há processamento no lado dos clientes, que usam estações ou terminais passivos (“burros”).
- O acesso às aplicações e ao banco de dados é feito através de uma requisição, via rede local ou VPN, usualmente fazendo uso de uma Intranet ou Portal da empresa.
- ...

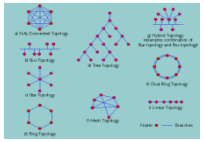
MODELO CENTRALIZADO





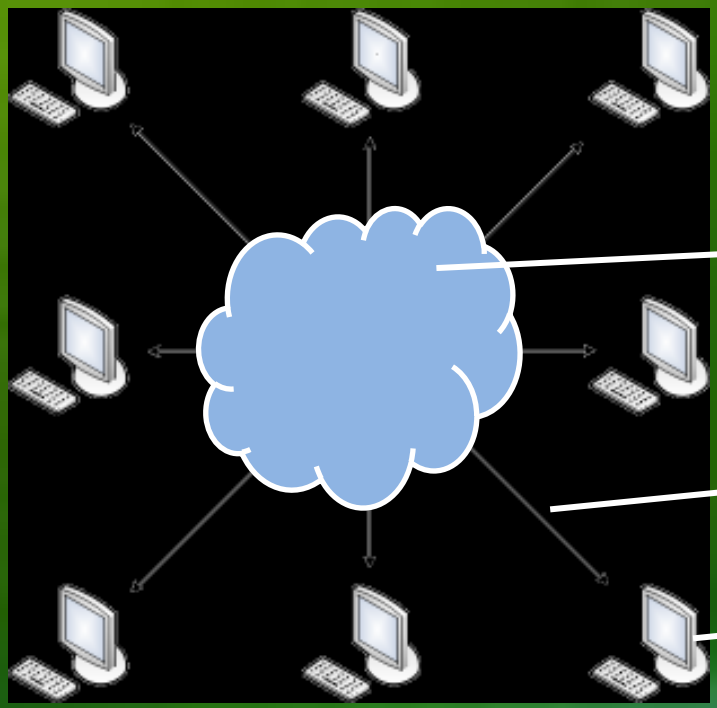
- Os aspectos de segurança e demais de QoS são preponderantemente determinados pelo computador central e pela banda & hardware da rede.
- Os sistemas acessados são pagos usualmente na forma de licença única e multiplicada pelo número de terminais clientes. O conceito de software livre e gratuito praticamente não se coloca.
- A integração de um sistema heterogêneo com uma aplicação do *mainframe* dá-se ou de forma indireta via Banco de Dados, ou de forma semi-direta, através de uma requisição (procedimento “batch”) para o *mainframe*.

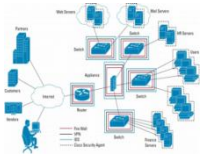


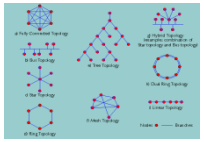


MODELO CENTRALIZADO

- A Computação em Nuvem (*Cloud Computing*) não deixa de ser equivalente ao modelo centralizado; porém, na Nuvem, é logicamente centralizado e com protocolos de comunicação e ferramentas de acesso diferentes.

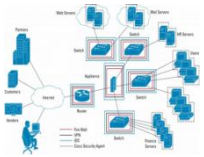


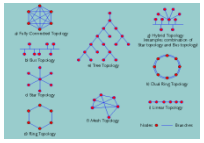




MODELO CLIENTE : SERVIDOR “CLÁSSICO”

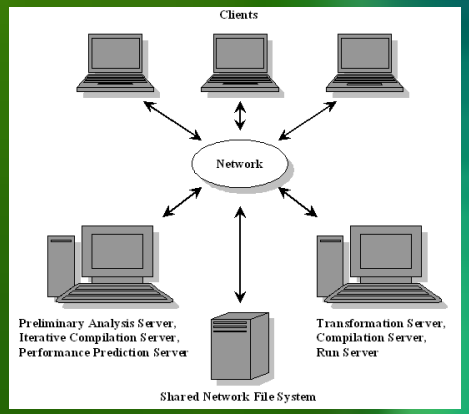
MODELO CLIENTE : SERVIDOR “CLÁSSICO”





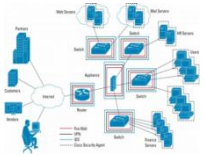
O modelo Cliente : Servidor é atualmente o mais comum, embora, do ponto de vista conceitual de redes, esteja presente em todos os modelos de comunicação hoje usados.

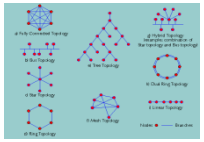
A perspectiva que aqui se quer enfatizar é a noção de que o processamento não é totalmente centralizado em um único computador central, mas sim distribuído por vários servidores.

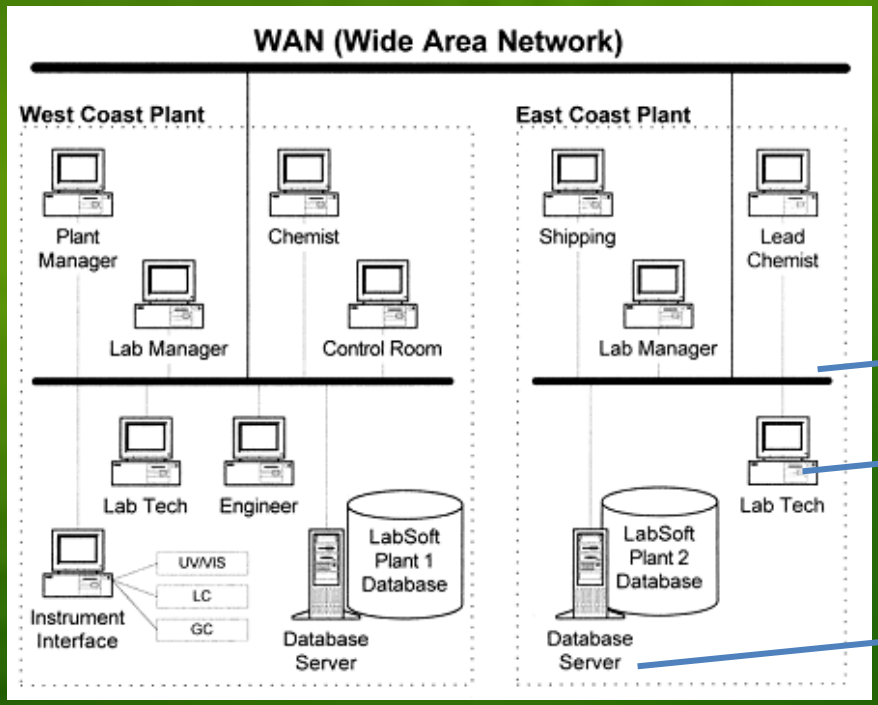


Por consequencia, a hospedagem e a execução de aplicações, assim como a persistência de dados, é descentralizada.

MODELO CLIENTE : SERVIDOR “CLÁSSICO”





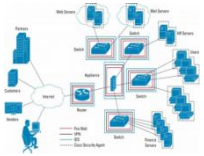


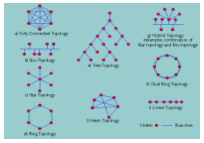
rede

cliente

servidor

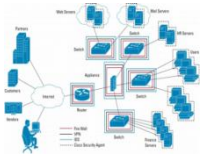
MODELO CLIENTE : SERVIDOR “CLÁSSICO”

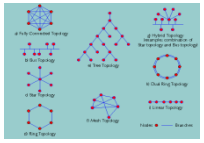




- Neste modelo, o processamento é distribuído e podem existir um central ou vários repositórios de dados, cada um devidamente conhecido sobre sua função (servidor de impressão, servidor de arquivos, servidor de modelos de produtos, etc.) mas transparente às aplicações.
- Há também processamento no lado das aplicações-cliente, que usam os serviços do servidor para dar sequencia às suas execuções / processos de negócios.
- O acesso às aplicações e ao banco de dados é também feito através de uma requisição, via rede local ou VPN, fazendo uso de Internet, Intranet ou Portal da empresa.
- ...

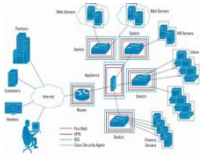
MODELO CLIENTE : SERVIDOR “CLÁSSICO”

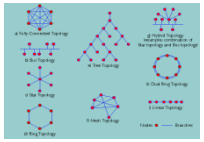




- Os aspectos de segurança e demais de QoS são determinados pela qualidade dos servidores e pela banda & hardware da rede.
- Os sistemas acessados são pagos usualmente na forma de licença única por servidor, e multiplicada pelo número de terminais clientes. O uso de software livre é comum, portanto potencialmente gratuito.
- A integração de um sistema heterogêneo com uma aplicação de um servidor dá-se ou de forma indireta via Banco de Dados, ou de forma direta, através de uma requisição de serviço para o servidor (via API, RPC, RMI, etc.).

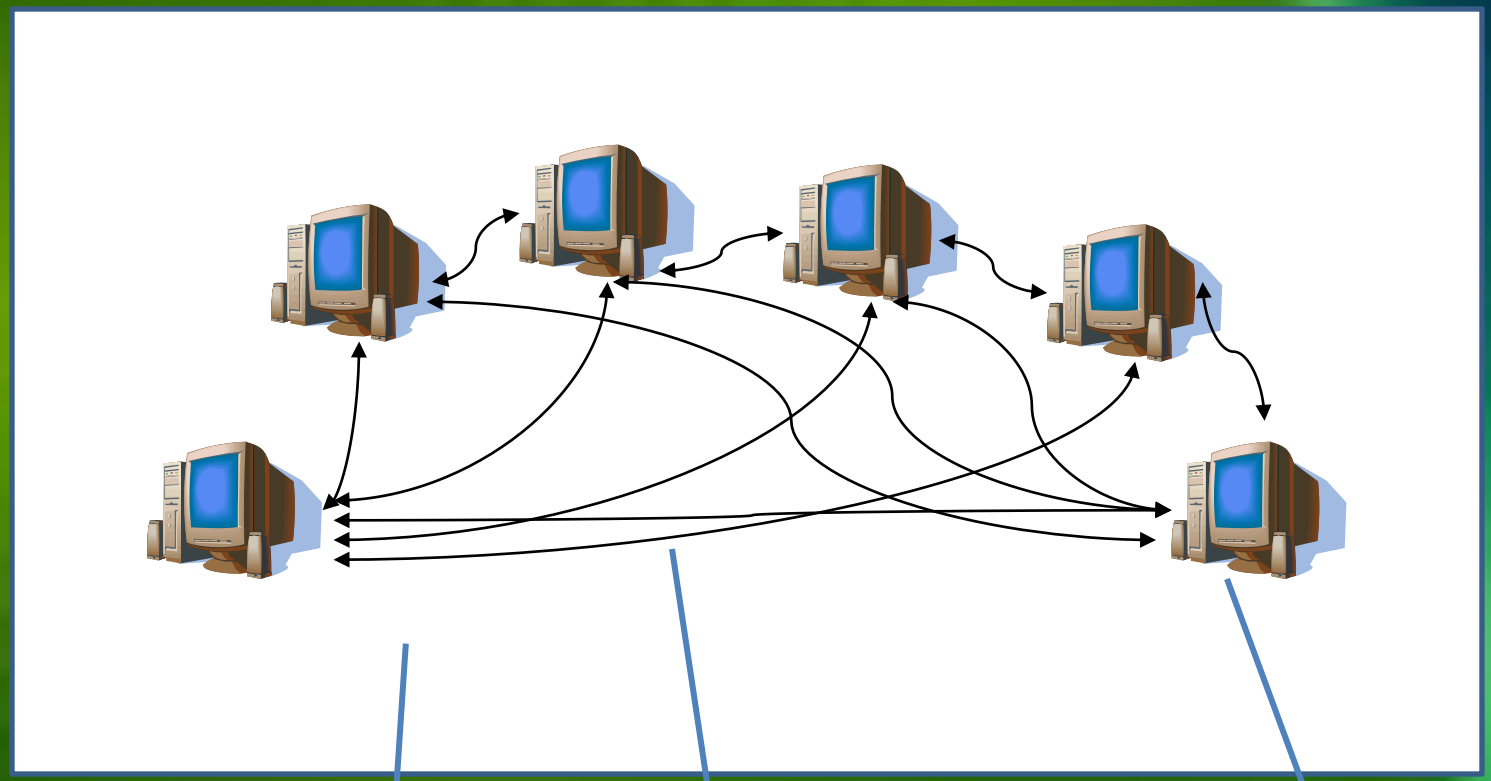
MODELO CLIENTE : SERVIDOR P2P





- Uma importante variante atual deste modelo clássico são os sistemas baseados no modelo P2P (*Peer-to-Peer*).
- Neste modelo, o conceito clássico de cliente:servidor é estendido no sentido de que todos os nós são, ao mesmo tempo, clientes e servidores. É muito usado para uma melhor distribuição de conteúdo e para compartilhamento de arquivos.
- A requisição do serviço é usualmente feita via plataformas (*middlewares*) e protocolos especialmente desenhados para P2P, que abstraem a localização dos servidores que têm toda ou parte da informação desejada.

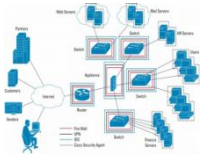
MODELO CLIENTE : SERVIDOR P2P

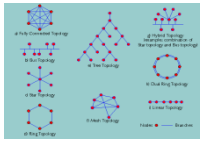


Plataforma P2P
(instalada em cada nó)

Rede & Protocolos P2P

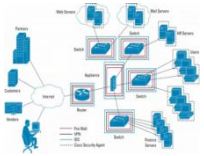
Cliente & Servidor

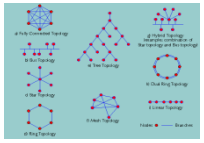




MODELO ASP

MODELO ASP

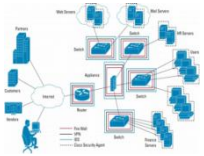


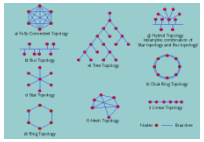


O Modelo ASP (*Application Service Provider - Provedor de Serviços de Aplicação*) é um modelo mais recente que se baseia na idéia da disponibilização de aplicações inteiras **via Internet**, que ficam fora da empresa, e que são completamente gerenciadas por uma empresa terceira.

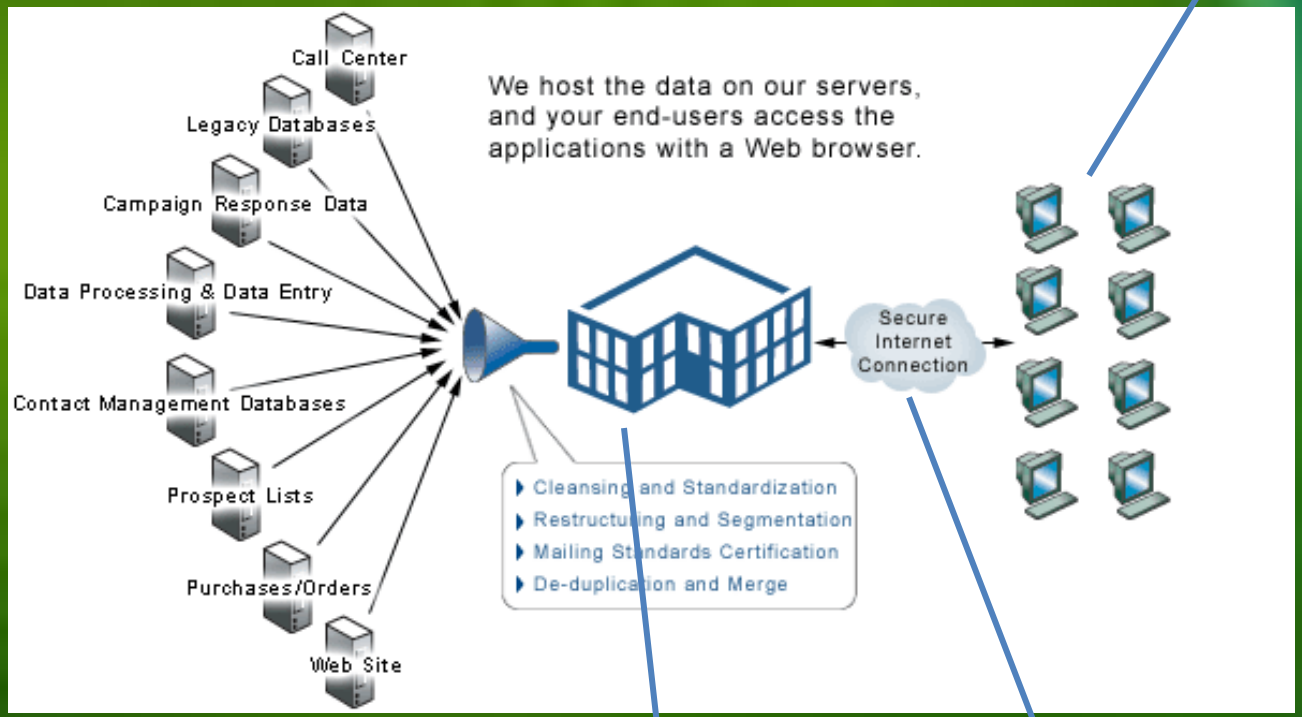
A empresa cliente não compra a licença do software e tão pouco o hospeda localmente, mas simplesmente o "aluga", pagando pelo seu uso, via rede.

MODELO ASP





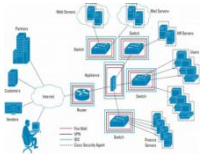
clientes

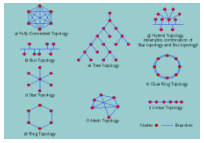


provedor (servidor)

rede

MODELO ASP



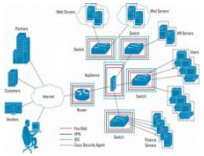


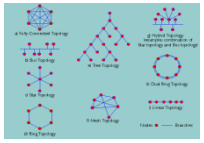
Sua arquitetura basicamente contempla um servidor que provê o acesso à aplicação. Na maior parte dos casos, são aplicações baseadas em web, onde basta ao cliente ter acesso à rede e a um navegador.

Baseado em um contrato de utilização (sob vários modelos de negócio) – SLA *(Service Level Agreement)* – todo o armazenamento e acesso a aplicações é terceirizado (*outsourcing*), assim como o cumprimento dos aspectos de QoS acordados.

A empresa-gestora (o provedor) fica totalmente responsável pela segurança, atualização de versões, manutenção, treinamento, etc., da aplicação.

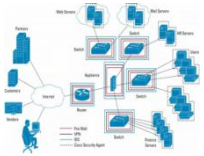
MODELO ASP

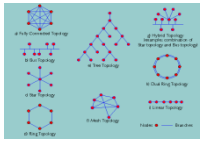




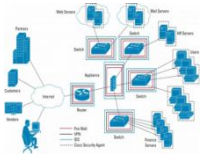
- Neste modelo, o processamento é logicamente centralizado (depende de como o provedor implanta a aplicação). O provedor também pode ser o responsável por toda hospedagem dos dados, de forma que o repositório de dados da empresa pode ser igualmente centralizado (embora de forma transparente ao usuário).
- Há também processamento no lado das aplicações-cliente, que usam os serviços do servidor para dar sequencia às suas execuções / processos de negócio. Porém, aqui tipicamente o usuário está imerso na aplicação, apenas com o fato de que toda a navegação pelas funcionalidades da aplicação e transações disparadas acabam sendo feitas via rede, e executadas no provedor.
- ...

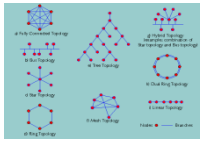
MODELO ASP





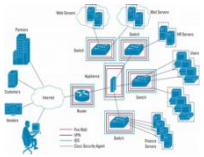
- Os aspectos de segurança e demais de QoS são determinados pela qualidade dos servidores do provedor, pela banda contratada & hardware da rede.
- Rígida observância dos SLAs.
- Os sistemas acessados são pagos usualmente na forma de licença única por servidor, e multiplicada pelo número de terminais clientes. No mundo corporativo, o uso de software livre não é comum neste modelo.
- A integração de um sistema heterogêneo com uma aplicação de um servidor ASP dá-se ou de forma indireta via Banco de Dados, ou de forma direta, através de uma requisição para o servidor. Porém, neste modelo, as requisições chegarão e serão gerenciadas pelo provedor, e não pela empresa-cliente.

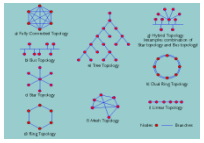




MODELO SAAS

MODELO SaaS

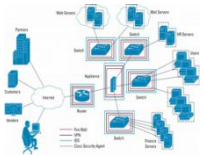


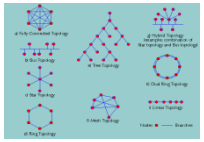


O Modelo SaaS (*Software-as-a-Service / Software-como-um-Serviço*) é o modelo mais recente. Ele se baseia na idéia da disponibilização de **serviços especializados de software**, relativamente pequenos, (e não mais de pacotes inteiros, muito grandes) via Internet, que ficam fora da empresa, e que são também completamente gerenciados por uma empresa terceira.

A empresa cliente não compra licença alguma de software e tão pouco o hospeda localmente, mas simplesmente invoca o serviço, acessando-o *on-demand* (apenas quando o requer/usa) e paga pelo seu uso.

MODELO SaaS





Cliente

Serviços que o cliente contratou:

- Financeiro
- Folha de pagamento

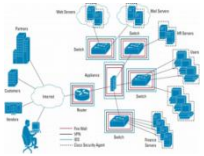
Internet

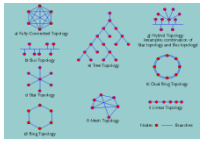
Provedor

Serviços que o provedor oferece:

- Financeiro
- Folha de pagamento
- Administração de frota
- Previsão do tempo
- Gerenciamento Hoteleiro
- Mapas
- Localização
- Reserva de Carros
- Locação de Carros

MODELO SaaS



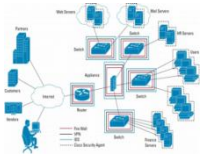


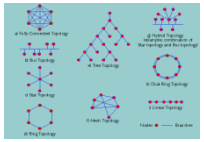
Sua arquitetura basicamente contempla um servidor que provê o acesso aos serviços. Na maior parte dos casos, são serviços baseados em web, onde basta ao cliente ter acesso à rede e a um navegador.

Também baseados em SLAs, todo o armazenamento e acesso aos serviços é terceirizado, assim como o cumprimento dos aspectos de QoS acordados.

A empresa-gestora (o provedor) fica totalmente responsável pela segurança, atualização de versões, manutenção, treinamento, etc., do serviço.

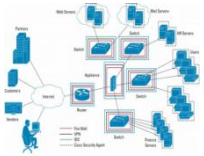
MODELO SaaS

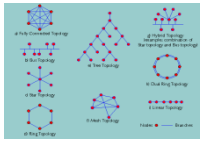




- Neste modelo, o processamento é também logicamente centralizado (depende de como o provedor implanta a aplicação). O provedor pode ser o responsável por toda hospedagem dos dados, de forma que o repositório de dados da empresa pode ser igualmente centralizado (embora de forma transparente ao usuário), e acessado como um serviço, e não indiretamente via uma aplicação.
- Há também processamento no lado das aplicações-cliente, que usam os serviços do servidor para dar sequencia às suas execuções. Porém, aqui o usuário normalmente não está imerso na aplicação (como no ASP), mas sim usando serviços específicos, desacoplados dos demais serviços.
- ...

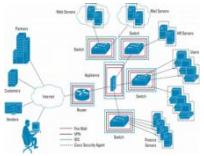
MODELO SaaS

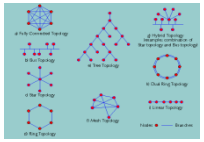


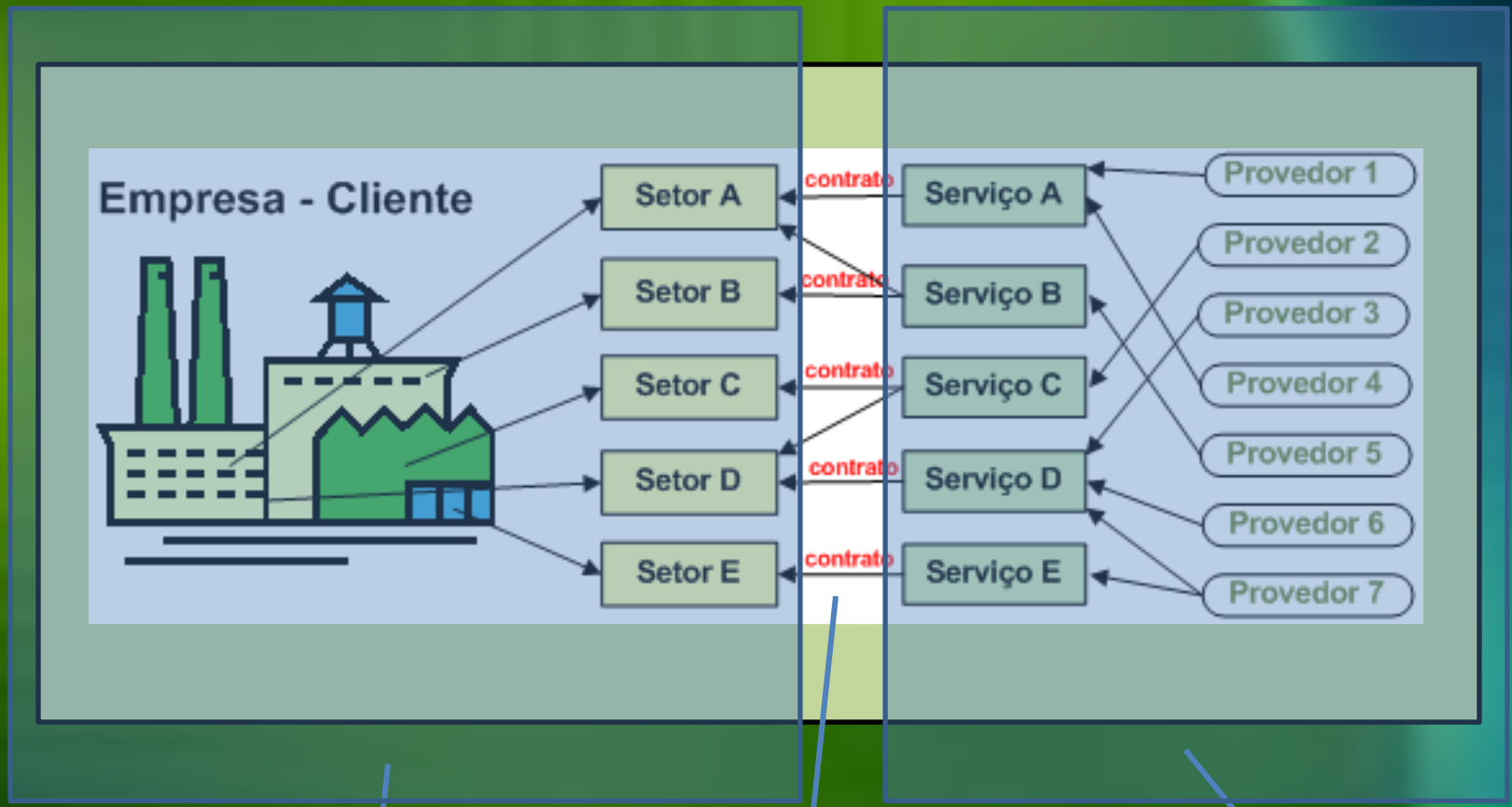


- Os aspectos de segurança e demais de QoS são determinados pela qualidade dos servidores do provedor, pela banda contratada & hardware da rede.
- Rígida observância dos SLAs.
- Os serviços acessados são pagos de acordo com o uso (*pay-per-use*). O uso de software livre por empresas pode se aplicar neste modelo.
- A integração de um sistema heterogêneo com um serviço de um servidor SaaS dá-se de forma direta, através de uma requisição para o servidor. Caberá ao cliente, ou invocar um dado serviço de dentro da aplicação e gerenciar o resultado, ou simplesmente receber o resultado direto no seu navegador.

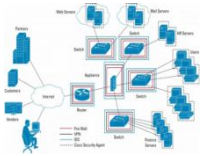
MODELO SaaS

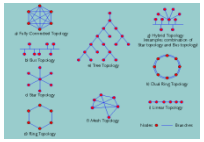





Empresa - Cliente
Setor A
Setor B
Setor C
Setor D
Setor E
contrato
contrato
contrato
contrato
contrato
Serviço A
Serviço B
Serviço C
Serviço D
Serviço E
Provedor 1
Provedor 2
Provedor 3
Provedor 4
Provedor 5
Provedor 6
Provedor 7
cliente
rede
Múltiplos provedores (servidores)

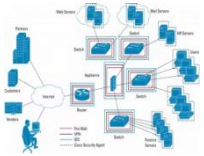
MODELO SaaS : SOA

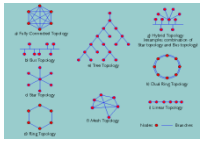


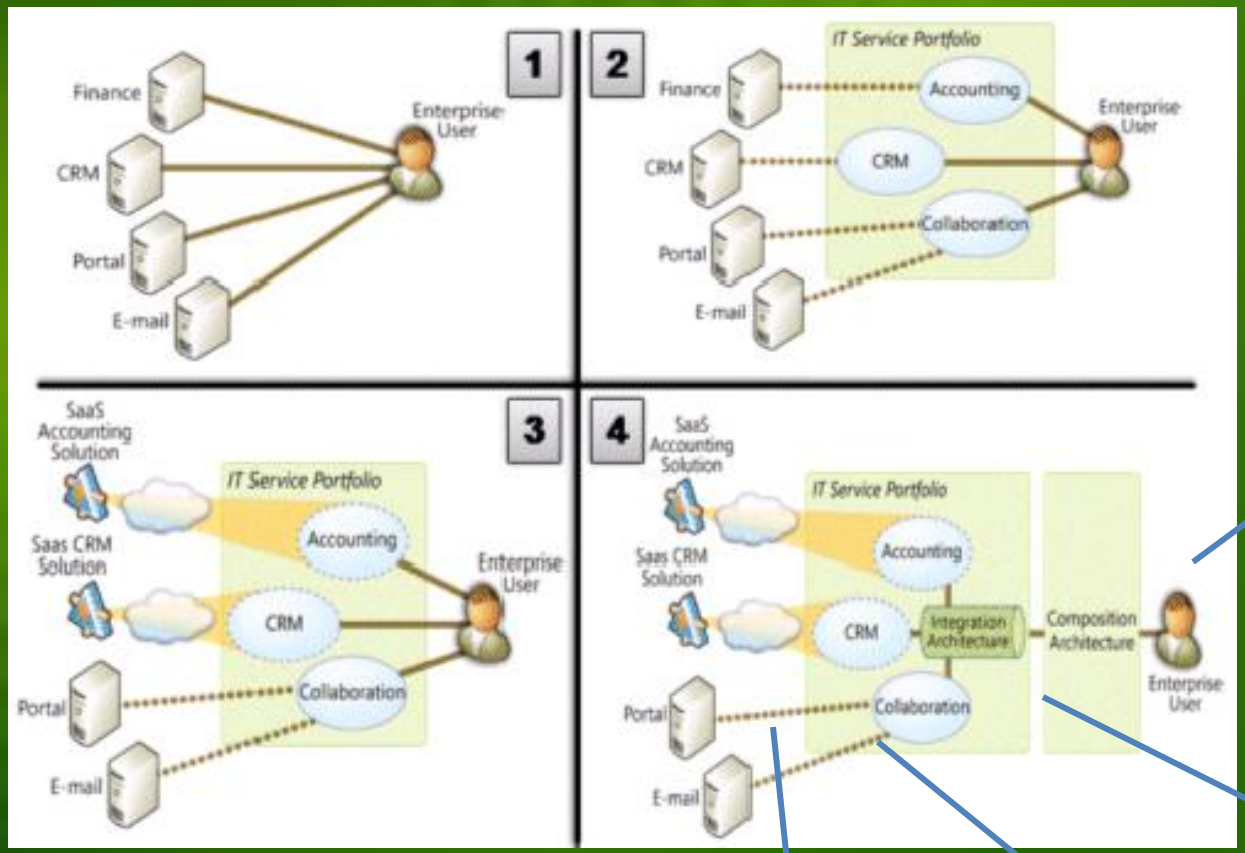


- Um importante uso atual do modelo SaaS é o seu direcionamento para aplicações **SOA** (*Service Oriented Architecture / Arquitetura Orientada a Serviços*).
- Neste modelo, o conceito original de SaaS é alargado, no sentido de que os serviços de software passam a ser disponibilizados como **funcionalidades de baixa granularidade, altamente reutilizáveis**, para compor aplicações no lado do cliente.
- A integração da aplicação com um serviço SaaS dá-se de forma direta, através de uma requisição para o servidor. Caberá ao cliente encontrar e invocar um dado serviço de dentro da aplicação e gerenciar o resultado.

MODELO SaaS : SOA







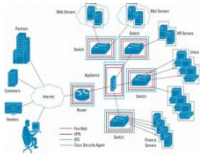
cliente

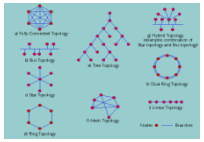
Aplicação SOA

rede

provedor (servidor)

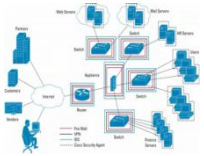
MODELO SaaS : SOA

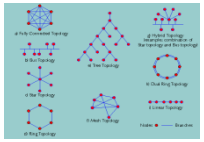




- É a atual “febre” do momento.
- Porém, a sua potencialidade tem sido pouquíssima utilizada ainda na prática. Na maior parte dos casos, a própria empresa-cliente atua como o provedor de serviços (i.e. ela é que tem e gerencia o repositório de serviços) e, assim, a aplicação-cliente não precisa *descobrir* onde está o provedor que tem o serviço desejado.
- As empresas que, de fato, usam a figura de uma empresa-provedora externa, acessam os devidos serviços apenas dessa empresa, e não de outras que eventualmente tenham serviços equivalentes.

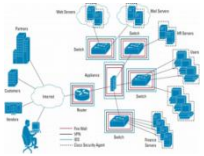
MODELO SaaS : SOA

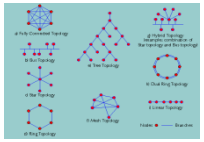




- A complexidade de integração / interoperação de aplicações numa perspectiva SaaS ou SOA deve ser vista sob duas óticas:
  - Do ponto de vista estritamente tecnológico, a interoperação tende a ser menos complexa (exceto a semântica), pois as TIs de suporte usadas são majoritariamente baseadas em padrões atualmente muito utilizados (ex. XML, SOAP e web services), o que diminui os problemas de interoperabilidade (mas não (!) os resolve, pois muitos desses padrões não são interopéráveis ente si).
  - ...

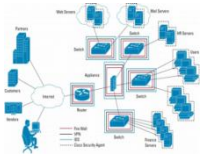
MODELO SaaS : SOA

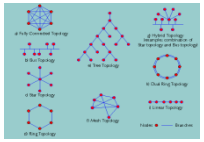




- Por outro lado, se o modelo SaaS permite uma maior flexibilidade / independência de um único provedor de software (podendo-se tentar até usar o melhor serviço do momento, sem estar preso a um contrato fixo e de longa duração), a composição (e posterior execução da aplicação SOA com base em serviços SaaS pode ser altamente (!) complexa.

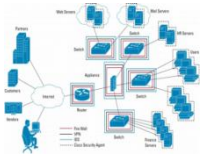
  Isto por que há que criar uma camada de ligação entre os vários serviços (gerados por vários provedores diferentes), tratar dos seus argumentos, semânticas, etc., e isso, em termos de estado da prática, tem que ser feito manualmente, programando-se.

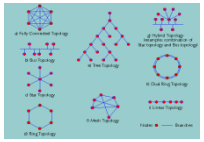




QUADRO COMPARATIVO

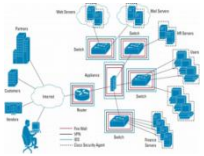
COMPARATIVO RESUMIDO DE CARACTERÍSTICAS

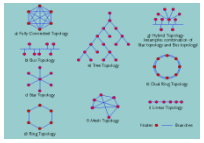




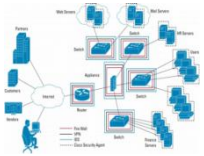
	Centralizada	C/S	ASP	SaaS
Disponibilidade	Muito Alta	Média/Alta	Alta	Alta
Desempenho	Muito Alto	Média/Alta	Média/Alta	Média/Alta
Segurança	Muito Alta	Média/Alta	Média/Alta	Média/Alta
Escalabilidade	Baixa	Alta	Média/Alta	Muito Alta
Tolerância Faltas	Baixa	Alta	Média/Alta	Alta / Muito Alta
UsoTempo Real	Baixa	Alta	Média	Média/Baixa
Contrato e $	Licença, Alto	Licença, de Baixo a Alto	Por Uso, de Médio a Alto	Por Uso, de Baixo a Médio
Complexidade Integração	Alta	Baixa	Média	Média / Altíssima

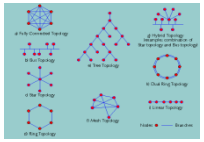
COMENTÁRIOS FINAIS





- Os quatro modelos apresentados não foram discutidos sob as óticas de Redes de Computadores ou Sistemas Distribuídos, mas sim em como o engenheiro de automação, ao especificar um sistema ou ter que analisar a abordagem de integração, deve enxergar o cenário de aplicações e de arquiteturas existentes na empresa.
- Por definição, não existe “o” modelo mais certo.
- Os quatro modelos têm características diferentes e, como tal, para cada caso / empresa, apresentará prós e contras. O mais importante é decidir qual é o melhor para ela.
- Os quatro modelos podem coexistir numa mesma empresa, nos seus variados níveis.
- A complexidade e custos de integração aumentam proporcionalmente de acordo com heterogeneidade dos modelos existentes.





FIM