

SOFTWARE AS A SERVICE (SAAS) E SERVICE-ORIENTED ARCHITECTURE (SOA)

ENGENHARIA DE SISTEMAS DE INFORMAÇÃO

Daniel Cordeiro

18 de agosto de 2017

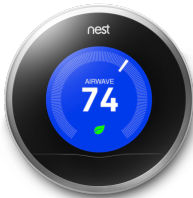
Escola de Artes, Ciências e Humanidades | EACH | USP

O QUE ELES TÊM EM COMUM?



Chrome

INTRODUCING
amazon echo



Software “embalado”

- cliente específico disponibilizado como um binário, upgrades frequentes
 - precisam trabalhar com muitos tipos de hardware diferentes, S.O., bibliotecas, ...
 - difíceis de manter
 - precisa de muito teste de compatibilidade em cada lançamento
- Alternativa: aplicação centralizada em servidor, *thin client*
 - busca, e-mail, comércio eletrônico, redes sociais, vídeos, etc.
 - mais recentemente engloba ferramentas de produtividade (Documentos Google/Office 365), finanças, IDEs (Codenvy)
- Diversos pesquisadores acreditam que esse seja o futuro dos softwares (eu inclusive)

POR QUE SAAS > SWS ?

1. Com SaaS não há mais preocupações sobre características de hardware ou S.O.
2. Dados são armazenados de forma segura e confiável nos servidores
3. É fácil fazer com que grupos interajam com os mesmos dados
4. Se a quantidade de dados for muito grande ou se mudar com muita frequência, é mais fácil manter uma única cópia centralizada
5. Uma cópia do software em um único ambiente de hardware/S.O.
⇒ ausência de problemas de compatibilidades ⇒ é possível fazer testes beta de novas funcionalidades em 1% dos usuários
6. Uma cópia ⇒ simplicidade nos upgrades para desenvolvedores e evita requisições de upgrades dos usuários

Vantagens

- Usa capacidades extras de hardware não disponíveis em HTML 5
- *Pode ser* mais rápido... ou não — muitos são apps HTML 5 que rodam em um “contêiner” nativo
- Sua marca está na tela principal do usuário
 - mas você também pode conseguir isso com o uso dos favoritos

Desvantagens

- Mais difíceis de manter
- Upgrades voltam a se tornar um problema do usuário

Pergunta:

Será que podemos projetar um software de forma que possamos recombina r módulos independentes para oferecer a muitos apps sem envolver muita programação?

[Amazon CEO Jeff Bezos] realized long before the vast majority of Amazonians that Amazon needs to be a platform.

— Steve Yegge, Googler, ex-Amazon, em seu blog em 2011

MENSAGEM DO CEO EM 2002: A AMAZON DEVE USAR SOA!

1. *Todas as equipes, de agora em diante, exibirão seus dados e funcionalidade através de interfaces dos serviços.*
2. *As equipes devem se comunicar entre si através dessas interfaces.*
3. *Não haverá nenhuma outra forma de comunicação permitida entre processos: nenhuma conexão direta, nenhuma leitura direta do estoque de dados de outras equipes, nenhum modelo de memória compartilhada e nenhuma excessão de qualquer outra natureza. A única comunicação permitida é através das chamadas às interface dos serviços através da rede.*
4. *Não importa de quais tecnologias elas se utilizam. HTTP, CORBA, Pub/Sub, protocolos customizados – não importa. O Bezos não se importa.*

MENSAGEM DO CEO EM 2002: A AMAZON DEVE USAR SOA!

1. *Todas as equipes, de agora em diante, exibirão seus dados e funcionalidade através de interfaces dos serviços.*
2. *As equipes devem se comunicar entre si através dessas interfaces.*
3. *Não haverá nenhuma outra forma de comunicação permitida entre processos: nenhuma conexão direta, nenhuma leitura direta do estoque de dados de outras equipes, nenhum modelo de memória compartilhada e nenhuma excessão de qualquer outra natureza. A única comunicação permitida é através das chamadas à interface dos serviços através da rede.*
4. *Não importa de quais tecnologias elas se utilizam. HTTP, CORBA, Pub/Sub, protocolos customizados – não importa. O Bezos não se importa.*

MENSAGEM DO CEO EM 2002: A AMAZON DEVE USAR SOA!

1. *Todas as equipes, de agora em diante, exibirão seus dados e funcionalidade através de interfaces dos serviços.*
2. *As equipes devem se comunicar entre si através dessas interfaces.*
3. *Não haverá nenhuma outra forma de comunicação permitida entre processos: nenhuma conexão direta, nenhuma leitura direta do estoque de dados de outras equipes, nenhum modelo de memória compartilhada e nenhuma excessão de qualquer outra natureza. A única comunicação permitida é através das chamadas à interface dos serviços através da rede.*
4. *Não importa de quais tecnologias elas se utilizam. HTTP, CORBA, Pub/Sub, protocolos customizados – não importa. O Bezos não se importa.*

MENSAGEM DO CEO EM 2002: A AMAZON DEVE USAR SOA!

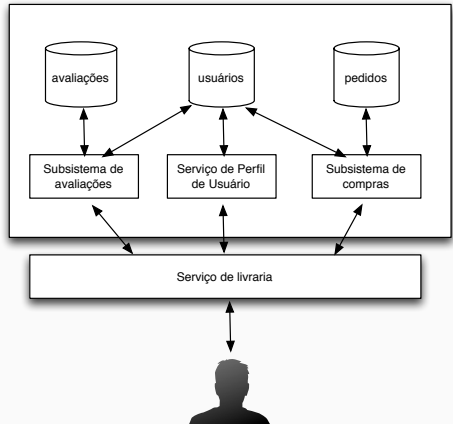
1. *Todas as equipes, de agora em diante, exibirão seus dados e funcionalidade através de interfaces dos serviços.*
2. *As equipes devem se comunicar entre si através dessas interfaces.*
3. *Não haverá nenhuma outra forma de comunicação permitida entre processos: nenhuma conexão direta, nenhuma leitura direta do estoque de dados de outras equipes, nenhum modelo de memória compartilhada e nenhuma excessão de qualquer outra natureza. A única comunicação permitida é através das chamadas às interface dos serviços através da rede.*
4. *Não importa de quais tecnologias elas se utilizam. HTTP, CORBA, Pub/Sub, protocolos customizados – não importa. O Bezos não se importa.*

5. *Todas as interfaces dos serviços, sem exceção, devem ser projetadas a partir do zero para serem exteriorizadas. Isto é, a equipe precisa planejar e projetar para ser capaz de exibir a interface para desenvolvedores do mundo exterior. Sem exceções.*
6. *Qualquer um que não fizer isso será despedido.*
7. *Obrigado; Tenham um bom dia!*

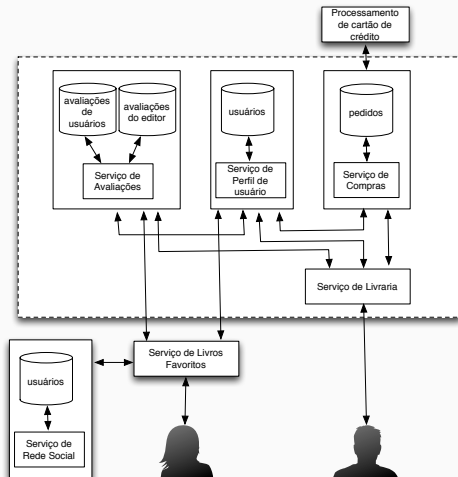
5. *Todas as interfaces dos serviços, sem exceção, devem ser projetadas a partir do zero para serem exteriorizadas. Isto é, a equipe precisa planejar e projetar para ser capaz de exibir a interface para desenvolvedores do mundo exterior. Sem exceções.*
6. *Qualquer um que não fizer isso será despedido.*
7. *Obrigado; Tenham um bom dia!*

5. *Todas as interfaces dos serviços, sem exceção, devem ser projetadas a partir do zero para serem exteriorizadas. Isto é, a equipe precisa planejar e projetar para ser capaz de exibir a interface para desenvolvedores do mundo exterior. Sem exceções.*
6. *Qualquer um que não fizer isso será despedido.*
7. *Obrigado; Tenham um bom dia!*

- Subsistemas internos podem compartilhar dados diretamente
- Todos os subsistemas acessados por uma única API



- Subsistemas independentes, como se estivessem em *datacenters* diferentes
- Pode ser re combinado para criar novos serviços (Serviço de Livros Favoritos)



Qual afirmação abaixo é uma desvantagem de SOA se comparado ao projeto como silo?

- SOA pode ser mais difícil de depurar & ajustar
- SOA resulta em menos produtividade no desenvolvimento
- A complexidade de SOA é um problema para times pequenos
- SOA é mais caro para implantar que silo porque mais servidores são necessários para atender uma mesma carga de trabalho

COMPUTAÇÃO EM NUVEM

QUAL O HARDWARE IDEAL PARA SAAS?

- Amazon, Google, Microsoft... desenvolveram hardware para executar SaaS
- O que eles usam? Mainframes? Supercomputadores?
- Como que desenvolvedores independentes de software podem construir aplicativos SaaS e serem competitivos sem investir em hardware especializado com fazem as grandes empresas?

SaaS dependem de 3 coisas da infraestrutura:

1. **Comunicação**

Permite os seus consumidores interagirem com o serviço

2. **Escalabilidade** (*scalability*)

Flutuação na demanda + possibilidade de adicionar mais serviços para lidar com crescimento rápido dos usuários

3. **Confiabilidade** (*dependability*)

Serviço & comunicação disponíveis 24x7

Aglomerados de computadores (*clusters*): computadores comuns (*commodities*) conectados por *switches* Ethernet comuns

1. Mais escalável do que servidores convencionais
2. Mais barato do que servidores convencionais
 - 20x mais barato que um grande servidor equivalente
3. Confiabilidade obtida com uso extensivo de redundância
4. Poucos operadores para milhares de servidores
 - Seleção cuidadosa de hardware/software idênticos
 - Monitores de máquinas virtuais simplificam a operação

- Aglomerados cresceram de 1.000 servidores para 100.000 graças a demanda dos clientes para apps SaaS
- Economia de escala derrubaram o preço de grandes *datacenters* em fatores de 3x a 8x
 - Comprar, abrigar, operar 100.000 vs. 1.000 computadores
- *Datacenters* tradicionais usavam 10% – 20%
- Ganham \$ oferecendo computação utilitária (pague apenas pelo que usa) a preços menores do que os que os clientes conseguiriam

COMPUTAÇÃO UTILITÁRIA / PLATAFORMAS PÚBLICAS DE COMPUTAÇÃO EM NUVEM

- Oferecem poder computacional, armazenamento, comunicação por centavos/hora
- Não há vantagens ao aumentar a escala:
1.000 computadores @ 1 hora
= 1 computador @ 1.000 horas
- Usuário da nuvem tem ilusão de escalabilidade infinita
 - existem tantos computadores quanto você puder pagar
- Maiores exemplos: Amazon Web Services, Google App Engine, Microsoft Azure, UOL Cloud e Locaweb

EXEMPLOS DE PREÇOS NA AWS

Região: Leste dos EUA (Norte da Virgínia)

	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GB)	Uso do Linux/UNIX
Uso geral – Geração atual					
t2.nano	1	Variável	0.5	Somente EBS	\$0.0065 por hora
t2.micro	1	Variável	1	Somente EBS	\$0.013 por hora
t2.small	1	Variável	2	Somente EBS	\$0.026 por hora
t2.medium	2	Variável	4	Somente EBS	\$0.052 por hora
t2.large	2	Variável	8	Somente EBS	\$0.104 por hora
m4.large	2	6.5	8	Somente EBS	\$0.12 por hora
m4.xlarge	4	13	16	Somente EBS	\$0.239 por hora
m4.2xlarge	8	26	32	Somente EBS	\$0.479 por hora
m4.4xlarge	16	53.5	64	Somente EBS	\$0.958 por hora
m4.10xlarge	40	124.5	160	Somente EBS	\$2.394 por hora
m3.medium	1	3	3.75	1 x 4 SSD	\$0.067 por hora
m3.large	2	6.5	7.5	1 x 32 SSD	\$0.133 por hora
m3.xlarge	4	13	15	2 x 40 SSD	\$0.266 por hora

- 72º lugar na lista TOP 500 dos supercomputadores mais rápidos do mundo em 2012
 - 532 computadores, 17000 cores = 240 TeraFLOPS
 - \$ 1300 por hora
 - 240º lugar na lista de jun/2016
- FarmVille no AWS
 - antes, o maior jogo online tinha 5M usuários
 - 4 dias: 1M; 2 meses: 10M; 9 meses: 75M
- IBM Watson: 90 servidores IBM Power 750
 - 3,5 GHz 8 cores/servidor
 - 90 por \$ 2,4 / hora = \$ 200/hora

Qual afirmação sobre *datacenters* privados vs. plataformas de computação utilitária públicas (ex: AWS) é falsa?

- *Datacenters* privados não são compartilhados por múltiplas companhias/competidores
- *Datacenters* privados podem ser a única opção para algumas aplicações que seguem muitas regulamentações
- *Datacenters* privados são inerentemente mais seguros do que plataformas públicas
- *Datacenters* privados poderiam ter o mesmo custo de plataformas públicas se eles usassem o mesmo tipo de hardware e software