

Teste como Serviço (TaaS) na Computação em Nuvem

Ricardo Ramos de Oliveira ICMC-USP

E-mail: ricardoramos@icmc.usp.br

Orientador: Prof. Dr. Adenilso da Silva Simao



Apresentação

Ricardo Ramos de Oliveira



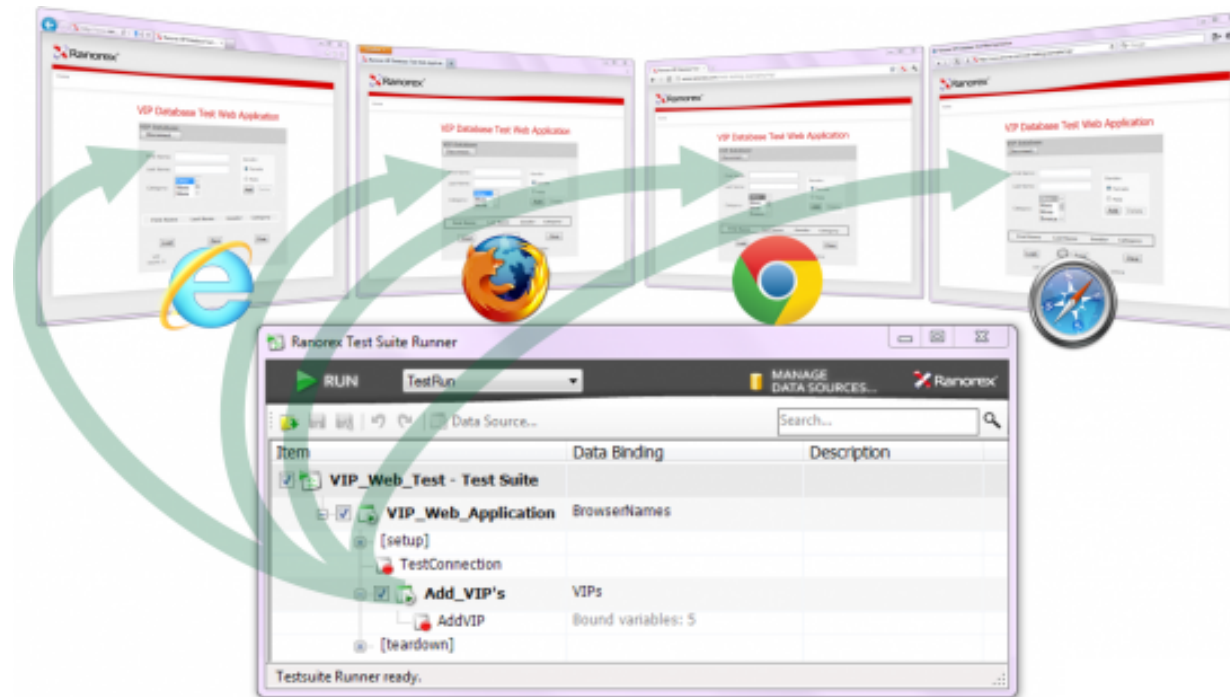
- Doutorando em Ciência da Computação
ICMC-USP
São Carlos - SP
- Professor de Computação
IFSULDEMINAS
Poços de Caldas - MG
- Áreas de Pesquisa:
Teste como Serviço (TaaS)
Computação em Nuvem

Agenda

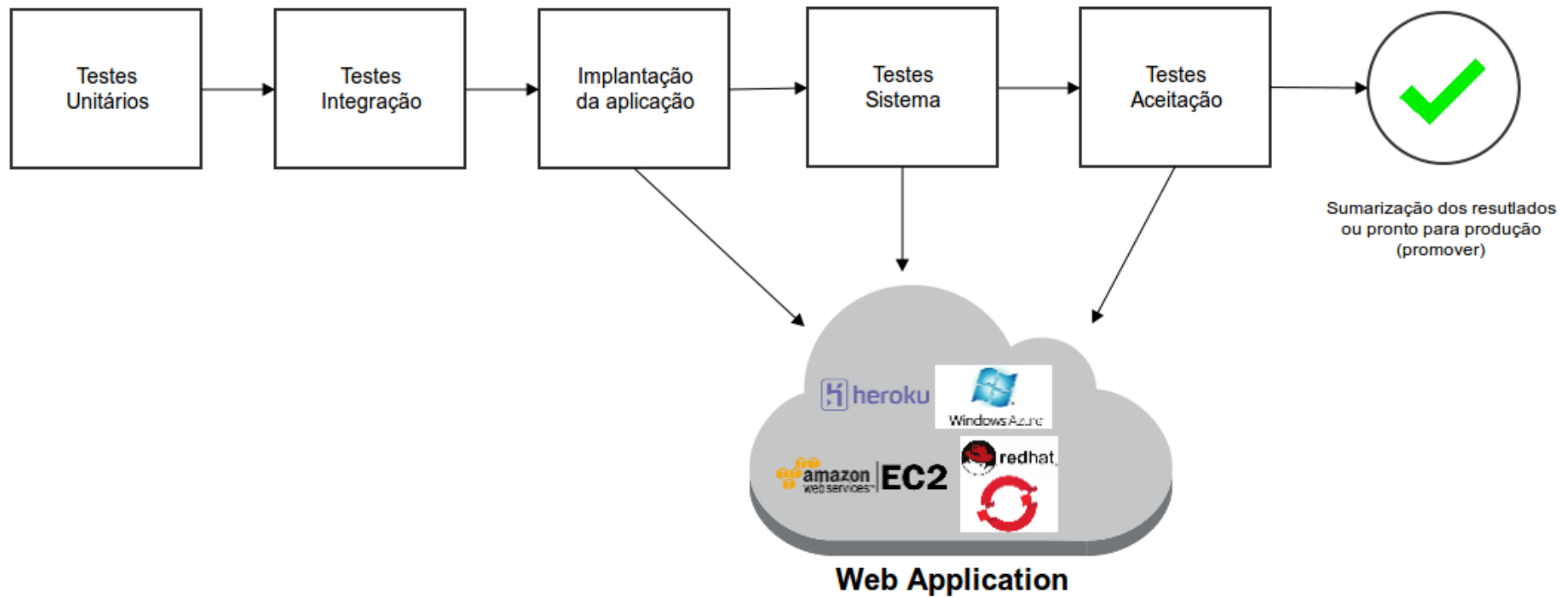
- 1 – Testes de Aceitação do Usuário (UAT)
- 2 – Automatização de Testes com o Selenium e o JUnit
- 3 – Computação em Nuvem
- 4 - Teste como Serviço (TaaS) na Nuvem
- 5 - Projeto de Doutorado

Teste de Aceitação do Usuário (UAT)

Test GUI



Fases de Teste de Software



Teste de Aceitação do Usuário (UAT)

- **Satisfazer os requisitos e processos de negócios conforme as necessidades dos usuários;**
- **Manual x Automatizado**
 - Manual - ↑ custo, tempo, esforço e muito tedioso;
 - Automatizado - ↑ preciso, eficaz, eficiente e confiável.
- Um dos **principais objetivos** do teste de software é **automatizar o máximo** possível as tarefas de teste;

Teste de Aceitação do Usuário (UAT)

- Principais características dos testes UAT (Test GUI):
 - 1 - Teste ponta a ponta (end-to-end - E2E);
 - 2 - Regras de Negócio (Não só a interface)
 - 3 - Defeitos de regressão (Desenvolvimento X Implantação);
 - 4 – Simular as ações dos usuários;
 - 5 – Testes mais próximos da aplicação em produção;

Teste de Aceitação do Usuário (UAT)

- Principais características dos testes UAT (Test GUI):
 - 6 – Permite encontrar erros arquiteturais e
 - 7 - Problemas de configuração ou do ambiente de teste;
 - 8 – A falta de testes *cross-browser* pode resultar em problemas de funcionalidade.

Automatização de testes com o Selenium e JUnit





JUnit

Selenium



Selenium Webdriver 2.0

- Criado pela ThoughtWorks – 2004
- O que é o Selenium?
 - Conjunto de ferramentas de software **open source**:
 - **Selenium IDE** – plugin Firefox 
 - **Selenium Webdriver** – API de automação de navegadores 
- Automatização dos testes de aplicações web

Selenium WebDriver 2.0

- **Versões do Selenium Webdriver:**
- Selenium 1.0 RC - Primeira versão
- **Selenium 2.0 – Webdriver**
- Appium mobile - camada intermediária entre o webdriver e uma aplicação mobile (Webdriver para mobile)

Selenium WebDriver 2.0

- **Principais Características**
- Selenium é baseado em JavaScript executa a partir do navegador;
- Operações de localização de elementos de interface do usuário;
- O usuário pode **acompanhar a execução dos testes** e conferir os resultados ao mesmo tempo em que instrui os navegadores para testar as aplicações web.

Selenium Webdriver 2.0

- Funcionamento básico do Selenium:



JUnit

JUnit

- JUnit é um ***framework open source*** para escrever testes automatizados em Java;
- Facilita o desenvolvimento e execução de testes;
- Já vem instalado e configurado nas principais IDEs;
- O JUnit também pode ser utilizado para a automatização de testes de aceitação do usuário (UAT).

- Exemplos de Teste com o JUnit

```
@Test
public void testeSomaComAssertEquals() {
    int valorEsperado = 2;
    int valorRetornado = calculadora.somar(1, 1);
    assertEquals(valorEsperado, valorRetornado);
}
```

```
@Test
public void testeSomaComAssertTrue() {
    int valorEsperado = 2;
    int valorRetornado = calculadora.somar(1, 1);
    assertTrue(valorEsperado == valorRetornado);
}
```

- Exemplos de Teste com o JUnit

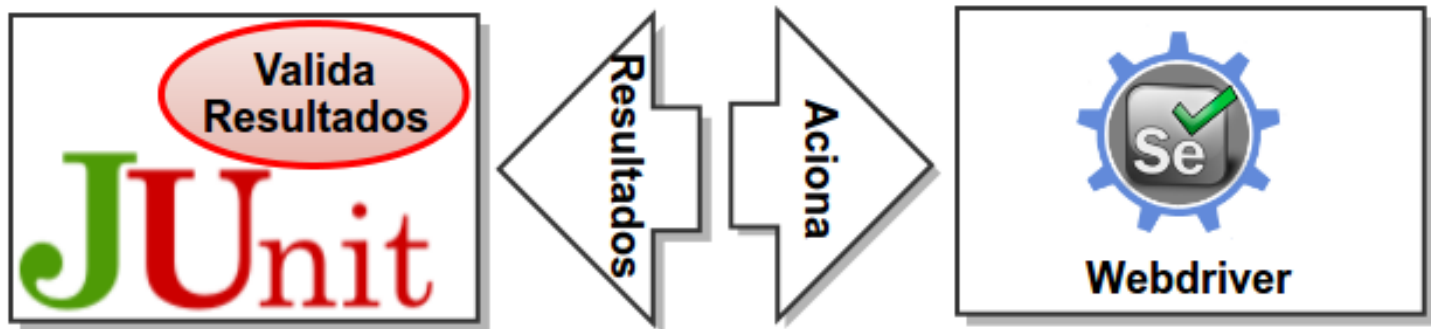
```
@Test
public void testeSomaComAssertFalse() {
    int valorEsperado = 2;
    int valorRetornado = calculadora.somar(1, 1);
    assertFalse(valorEsperado != valorRetornado);
}
```

```
@Test
public void testeSomaComAssertThat() {
    int valorEsperado = 2;
    int valorRetornado = calculadora.somar(1, 1);
    assertThat(valorRetornado, is(valorEsperado));
}
```

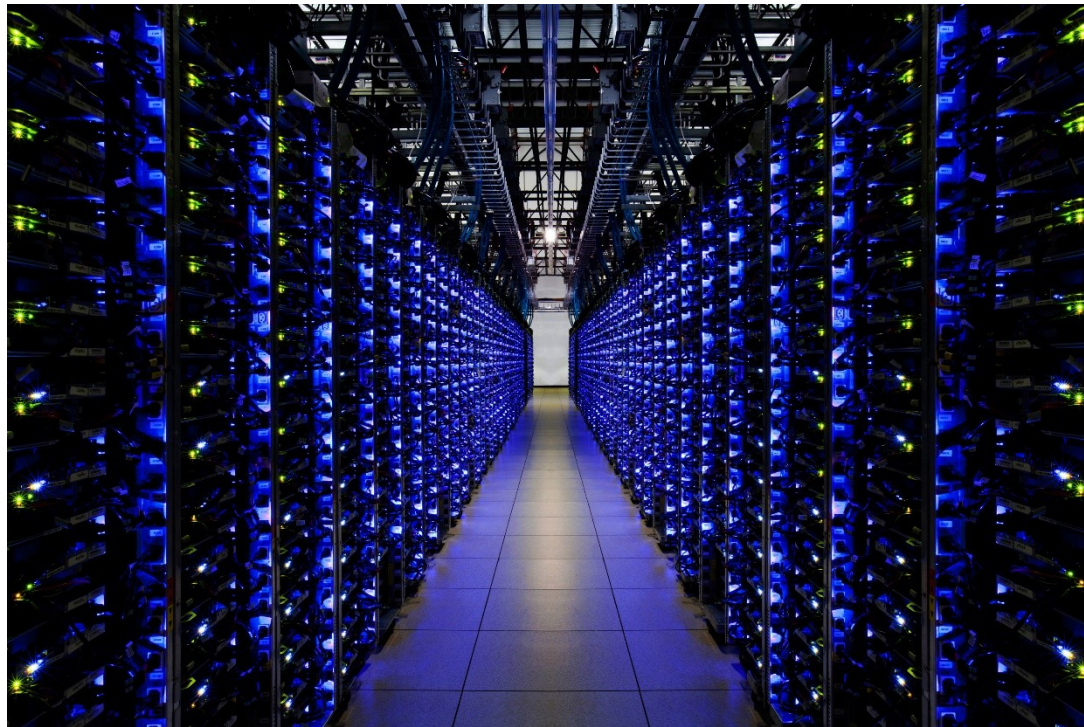
Selenium e JUnit

- **Comunicação entre o JUnit e o Selenium**

Execução dos Testes pelo JUnit por meio do Selenium



Computação em Nuvem



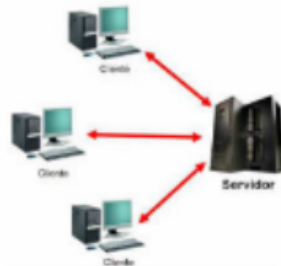
Computação em Nuvem

Do mainframe viemos, ao mainframe voltaremos...

Década de 50



Década de 80



Thin Client - Antigo

Década de 90



Computação em Nuvem



Fornecida e Usada

Thin Client - Atual

Computação em Nuvem



- Um dos componentes chave da computação em nuvem é a virtualização;
- Abstrai as características físicas do hardware (Máquinas Virtuais).

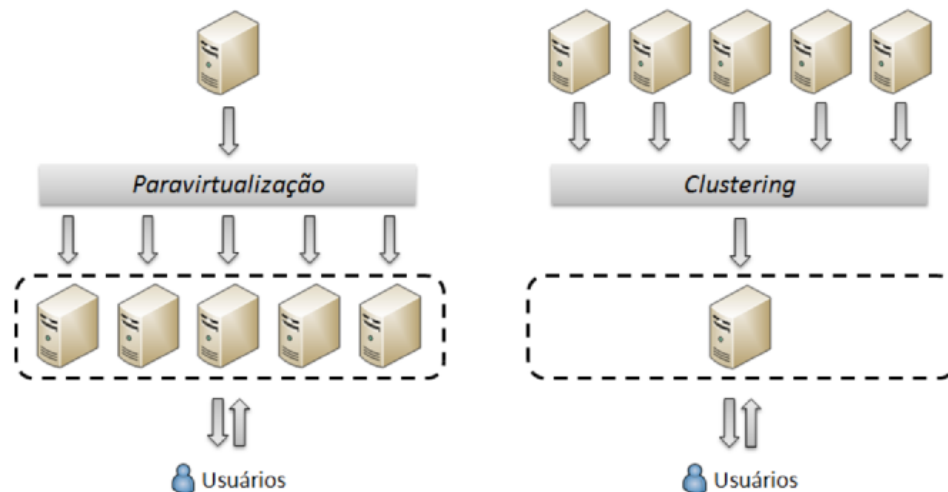


Figura: Virtualizacao: a) paravirtualização e b) clustering

Computação em Nuvem



■ Computação em Nuvem

- Uma das principais vantagens: Escalabilidade;
- Contratar recurso sob demanda: alocação e desalocação; (De forma dinâmica sem a intervenção de um ser humano...)
- Pagamento por uso (pay-per-use).

OBS: É por meio da elasticidade que eu consigo reduzir os custos e otimizar a utilização dos recursos computacionais

Computação em Nuvem



■ Tipos de Escalabilidade

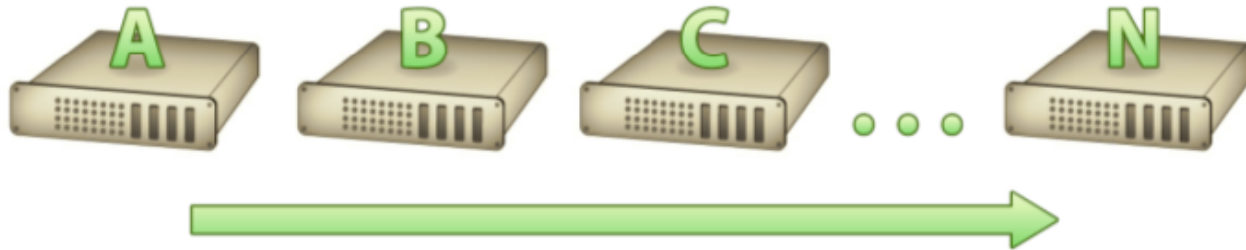


Figura: Exemplo de Escalabilidade Horizontal

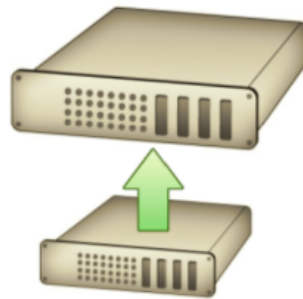


Figura: Exemplo de Escalabilidade Vertical

Computação em Nuvem



■ Computação em Nuvem

- 3 modelos de serviço e 4 abordagens de implantação. Os modelos de serviço são compostos por:

- Software como Serviço (SaaS)
- Plataforma como Serviço (PaaS)
- Infraestrutura como Serviço (IaaS)

Computador Pessoal



Termo XaaS – Everything as a Service: Database as a Service, Framework as a Service, Hardware as a Service, MDE as a Service...

Computação em Nuvem



■ Computação em Nuvem

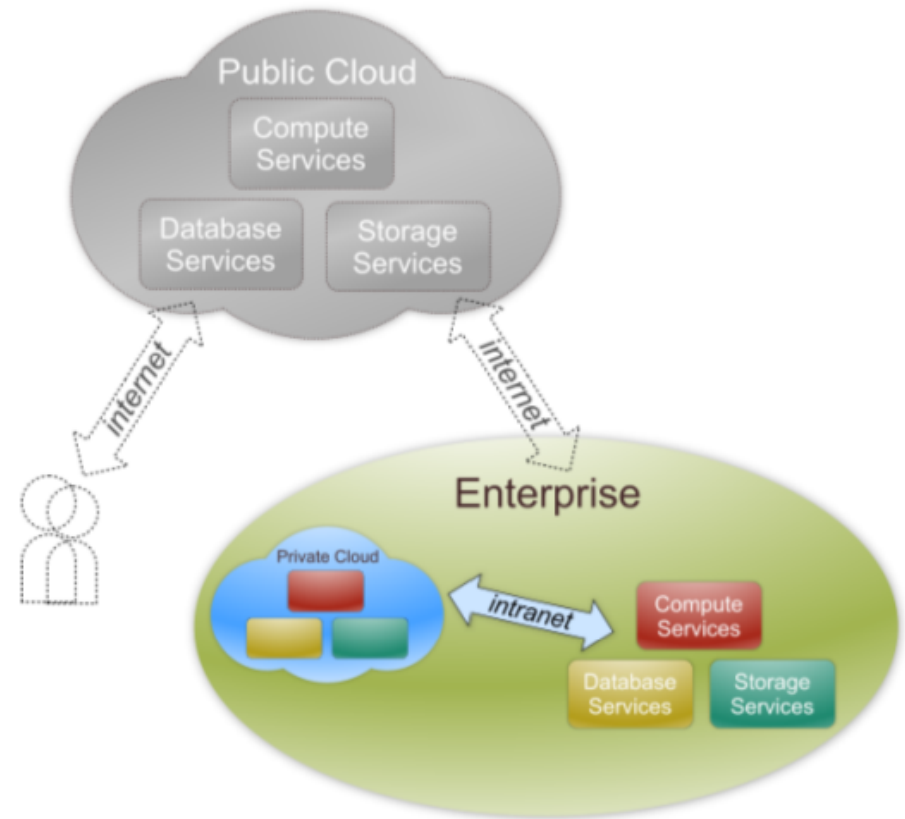
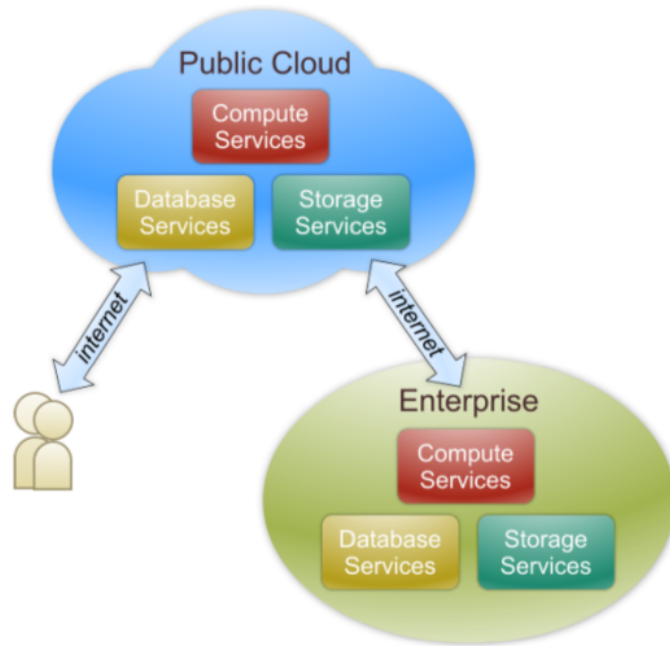


Figura: Exemplos de modelos de serviços na computação em nuvem

Computação em Nuvem



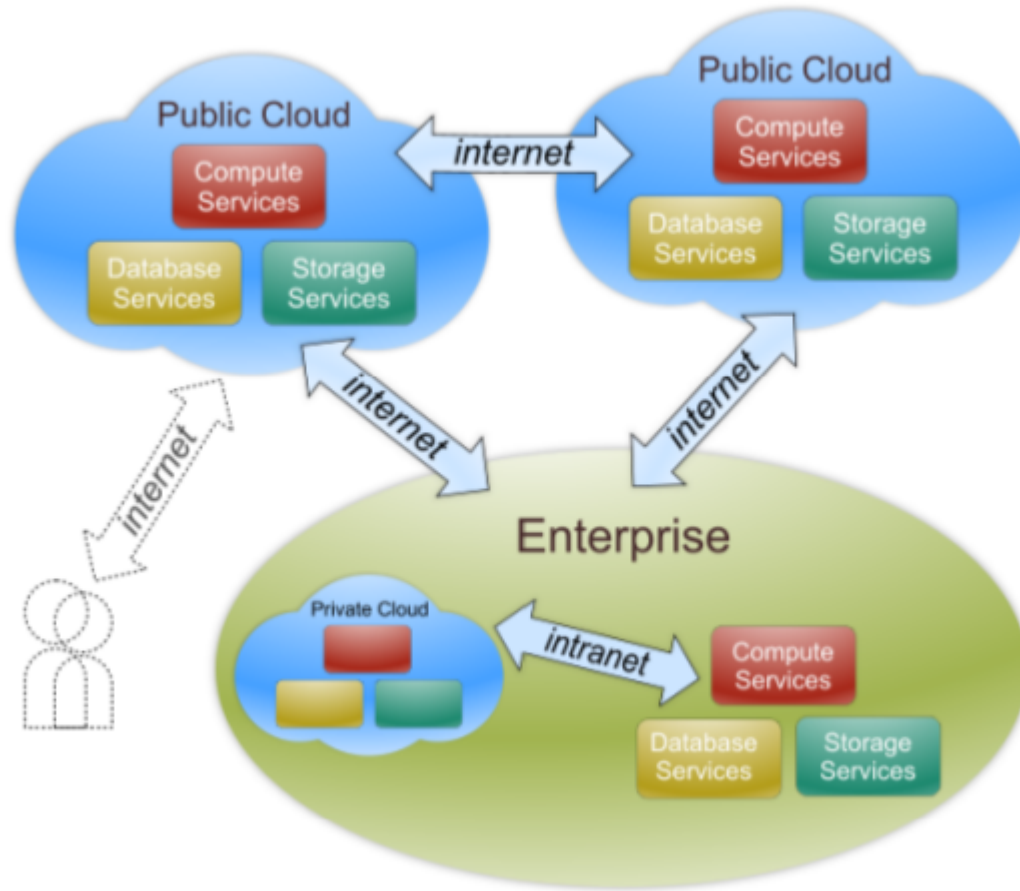
■ Tipos de Nuvem – Modelo de Implantação



Computação em Nuvem



■ Tipos de Nuvem – Modelo de Implantação



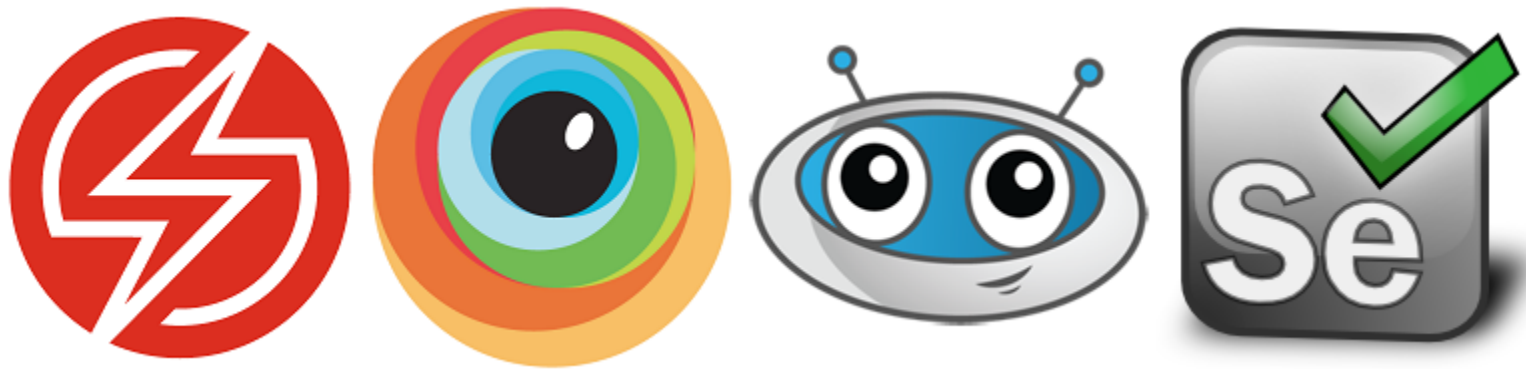
Computação em Nuvem



■ Tipos de Nuvem – Modelos de Implantação



Teste como Serviço (TaaS) na Computação em Nuvem



Contexto



Figura: Diferentes Plataformas e *Browsers*

Contexto

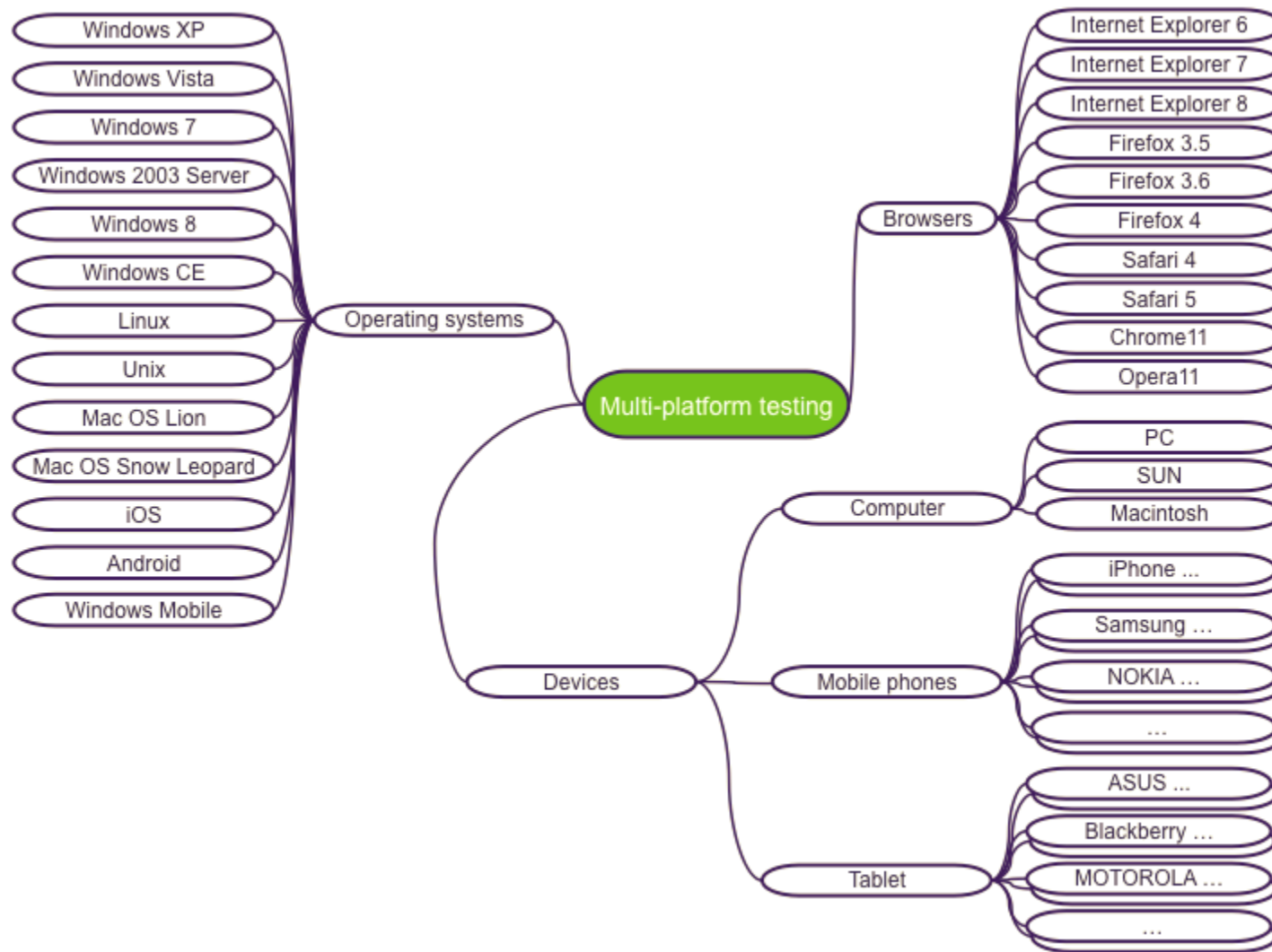


Figura: Teste de Múltiplas Plataformas

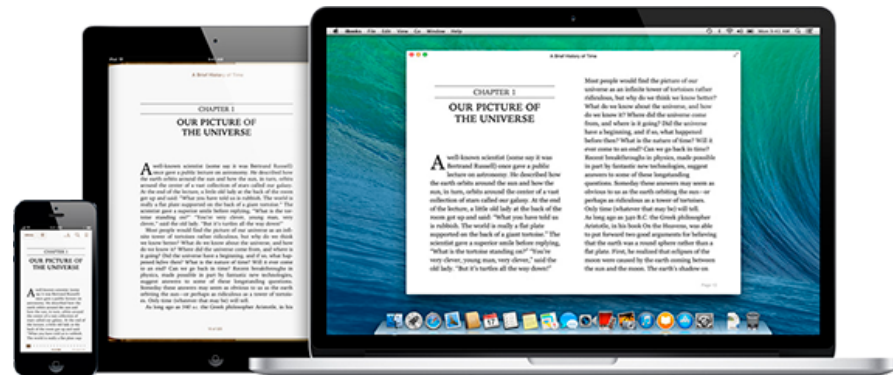
O que é TaaS?



- O Teste como Serviço (TaaS) é definido como uma ferramenta de teste que é oferecida como um serviço para verificação e validação das funcionalidades do sistema por meio da internet;
- Realizar a fase de “testes” do processo de desenvolvimento de software na nuvem;
- Economia entre 50% a 70% do custo com testes de software.

O que é TaaS?

- Reduzir os custos dos testes multiplataforma;
- Computação em Nuvem (Otimização dos recursos);
- Teste como Serviço (TaaS)
 - ❑ Executa testes multiplataformas
 - ❑ Infraestrutura de Nuvem



Vantagens do TaaS



- As principais vantagens do uso do modelo de Teste como Serviço (TaaS) são:
 - 1 - Redução de custos (com infraestrutura servidores, memória e processamento);
 - 2 - Ambiente de teste elástico;
 - 3 - Testes automatizados sob demanda;
 - 4 - Alta disponibilidade;
 - 5 - Modelo de pagamento por uso (*pay-per-use*).
(Conta de água e energia elétrica)
-

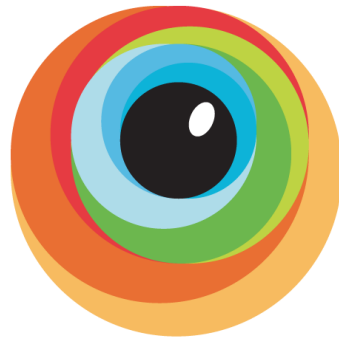
Desvantagens do TaaS



- As principais desvantagens do (TaaS) são:
 - 1 - Problemas de segurança
 - 2 - Ausência de padrões
 - 3 - Mau planejamento
 - 4 - Desperdício de recurso

Tipos de TaaS

- Existem vários tipos de TaaS (Funcionais e Não Funcionais):



Testa a parte funcional do *layout* e não o *layout* propriamente dito).

 **applitools**eyes

 **BlazeMeter**[™]
THE LOAD TESTING CLOUD

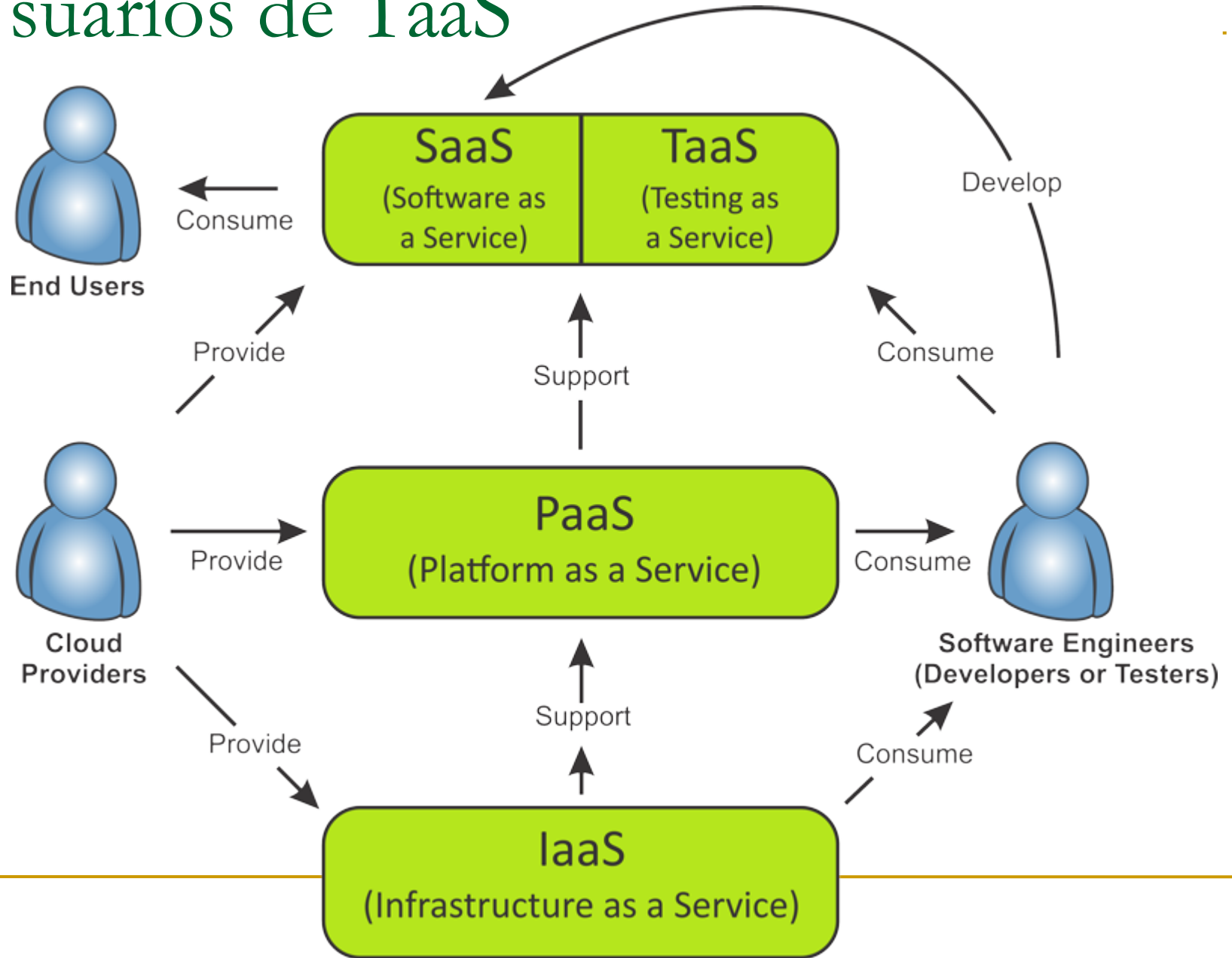


LOAD MANAGER



LOAD IMPACT

Usuários de TaaS



Funcionamento do TaaS

Teste como Serviço (TaaS)

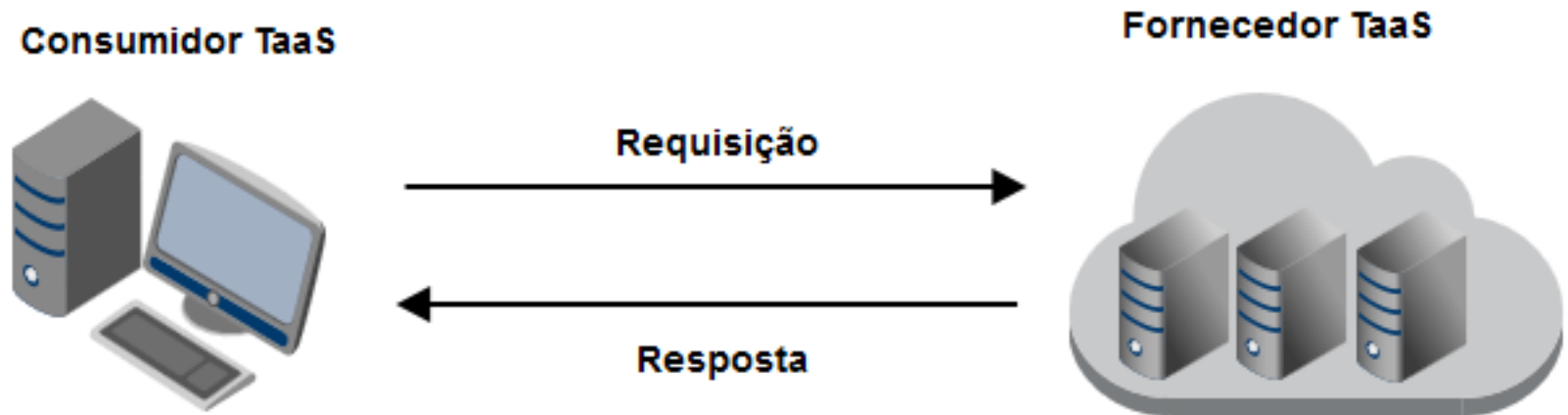
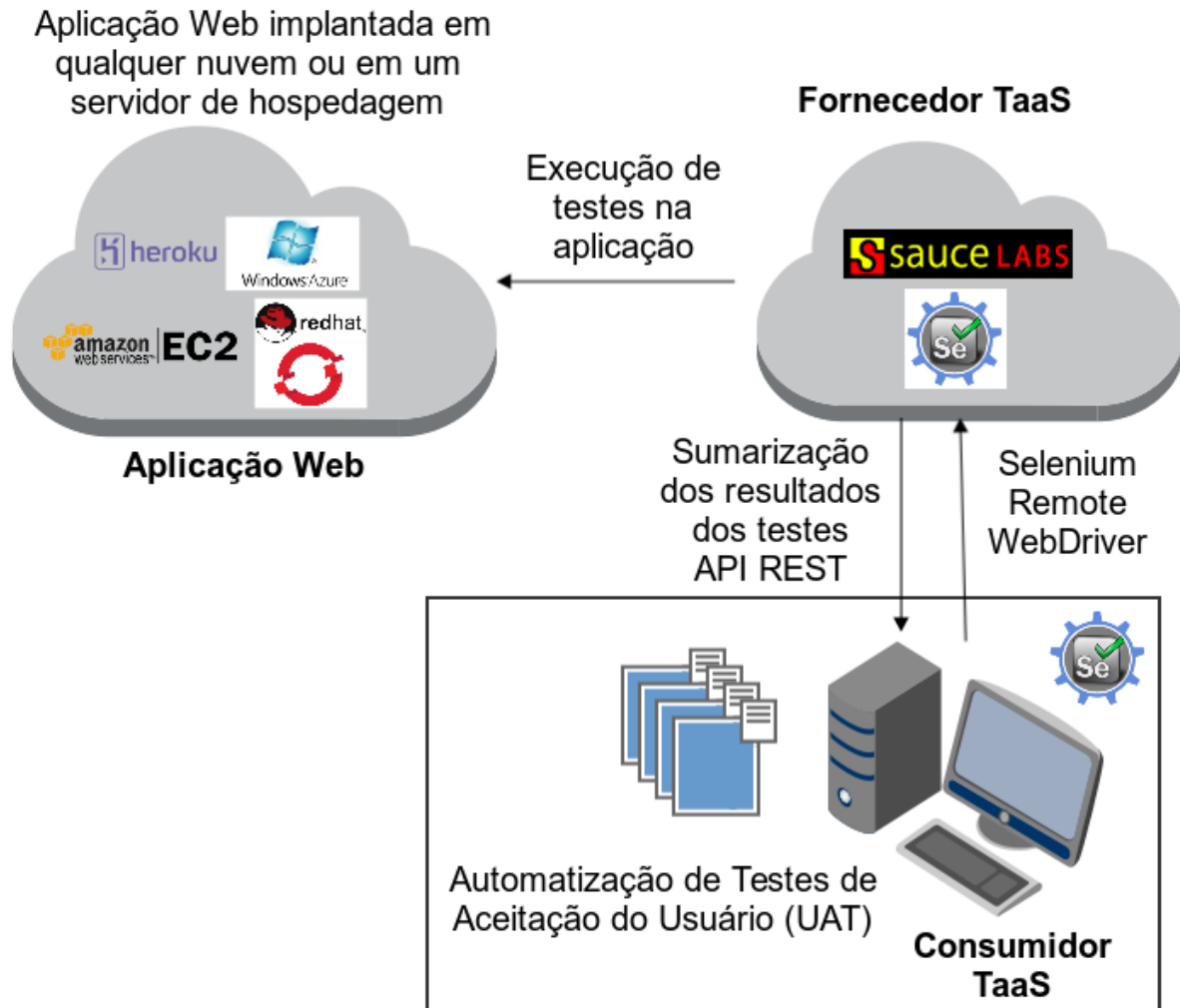


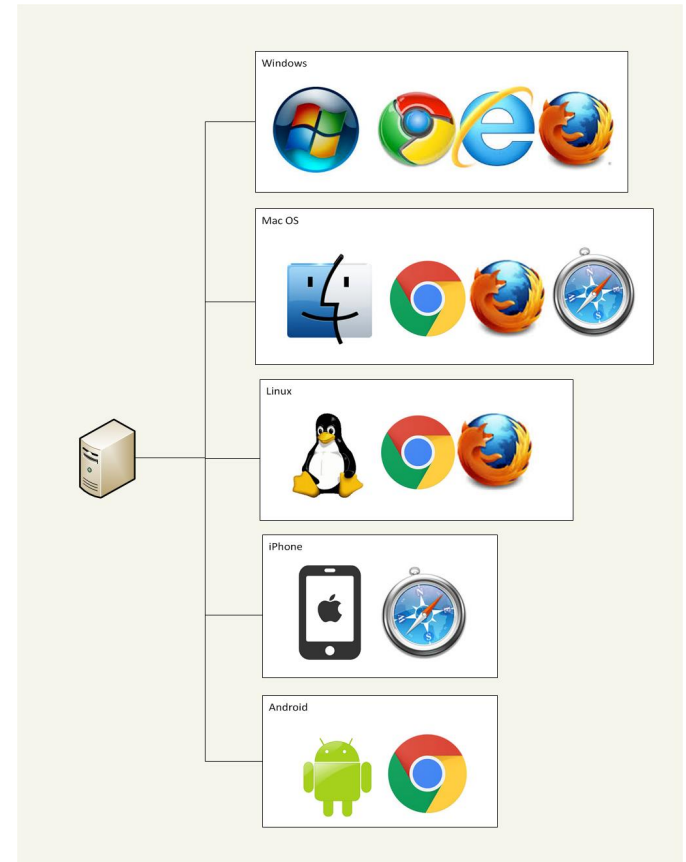
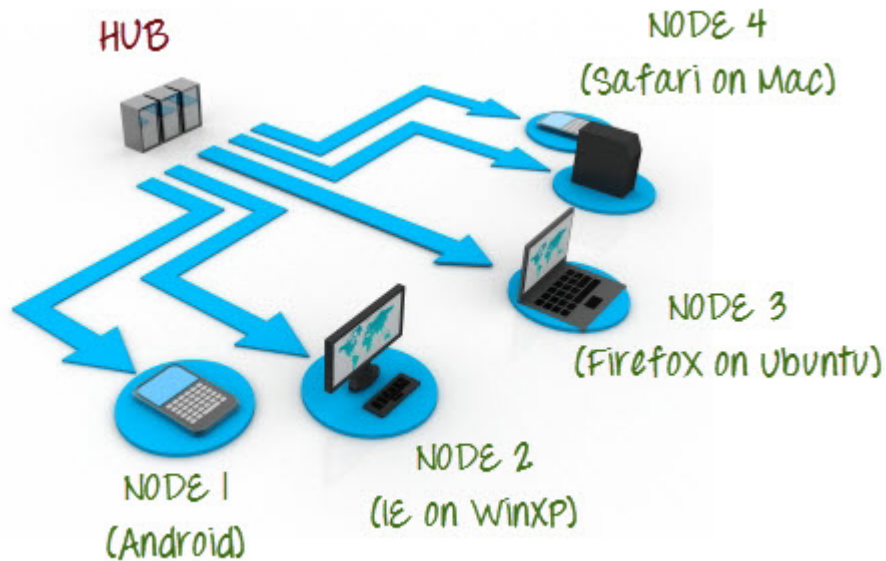
Figura: Interação Fornecedor-Consumidor (TaaS)

Funcionamento do TaaS



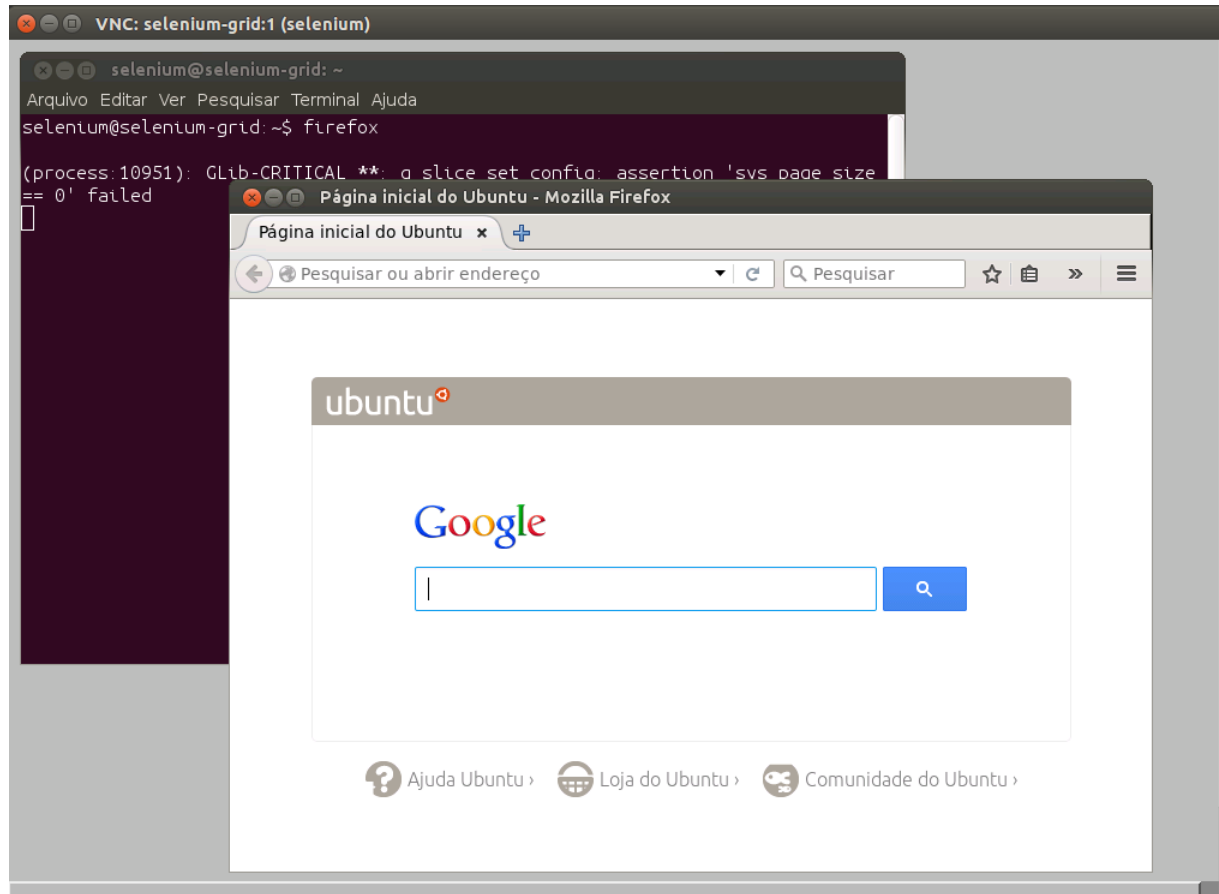
Funcionamento do TaaS

- Os TaaS utilizam o Selenium Grid



Funcionamento do TaaS

- Selenium Grid – ICMC-USP (Core TaaS)



<http://www.labes.icmc.usp.br:6144/grid/console>

Características dos TaaSs

- Características de TaaS na Nuvem
 - Aplicações Web Tradicionais X Aplicações em Nuvem;
 - Independente de metodologias de desenvolvimento
 - **Teste Sequencial** X Teste Paralelo
 - Máquinas virtuais concorrentes
 - Testes Escaláveis e Pay-as-you-test
- Fornecedores utilizados no curso:
 - Sauce Labs, Browserstack e Testingbot

Projeto de Doutorado



Projeto de Doutorado

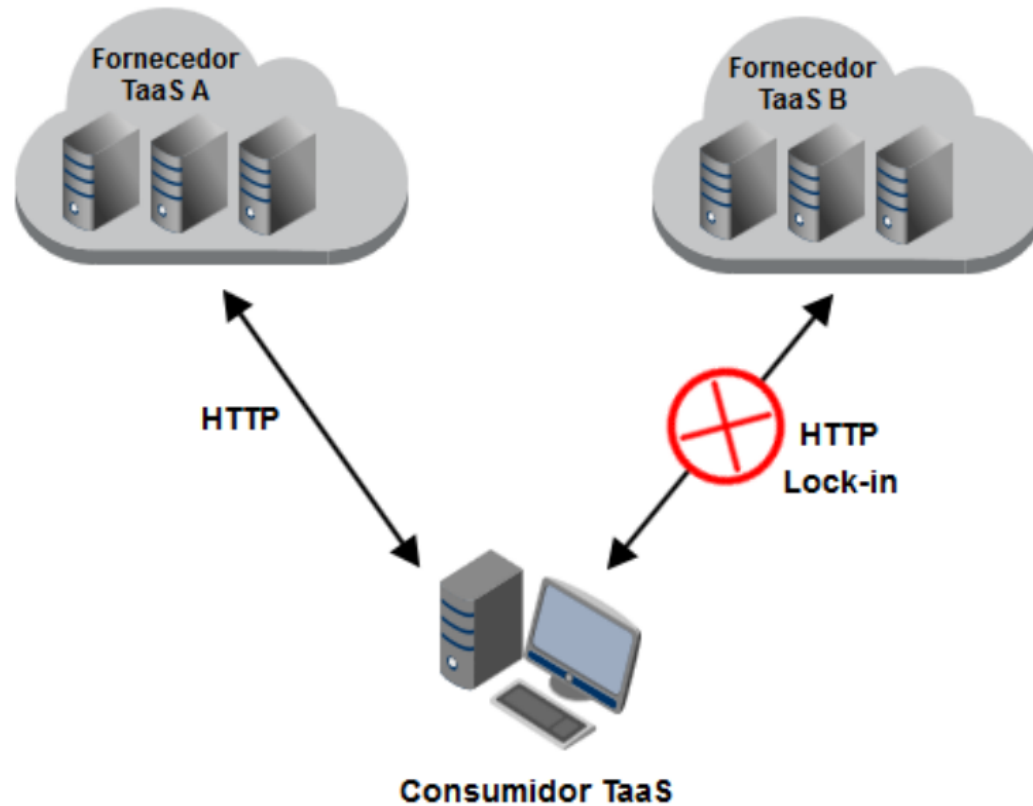


- **Problema de Pesquisa:**
- O Problema de ***vendor lock-in***
 - Aprisionamento do usuário;
 - Falta de **padronização** e **interoperabilidade** entre os fornecedores de TaaS;
 - Clientes - Dificuldade em migrar os testes;

Projeto de Doutorado



- ***Vendor lock-in***



Projeto de Doutorado



■ ***Vendor lock-in***

- Um dos 10 maiores problemas enfrentados na Computação em Nuvem;
 - *A view of cloud computing* (Michael Armbrust)
 - *Above the clouds: A berkeley view of cloud computing* (Michael Armbrust)
- Afeta o Teste como Serviço (TaaS)
- Maiores obstáculos para a adoção da nuvem.

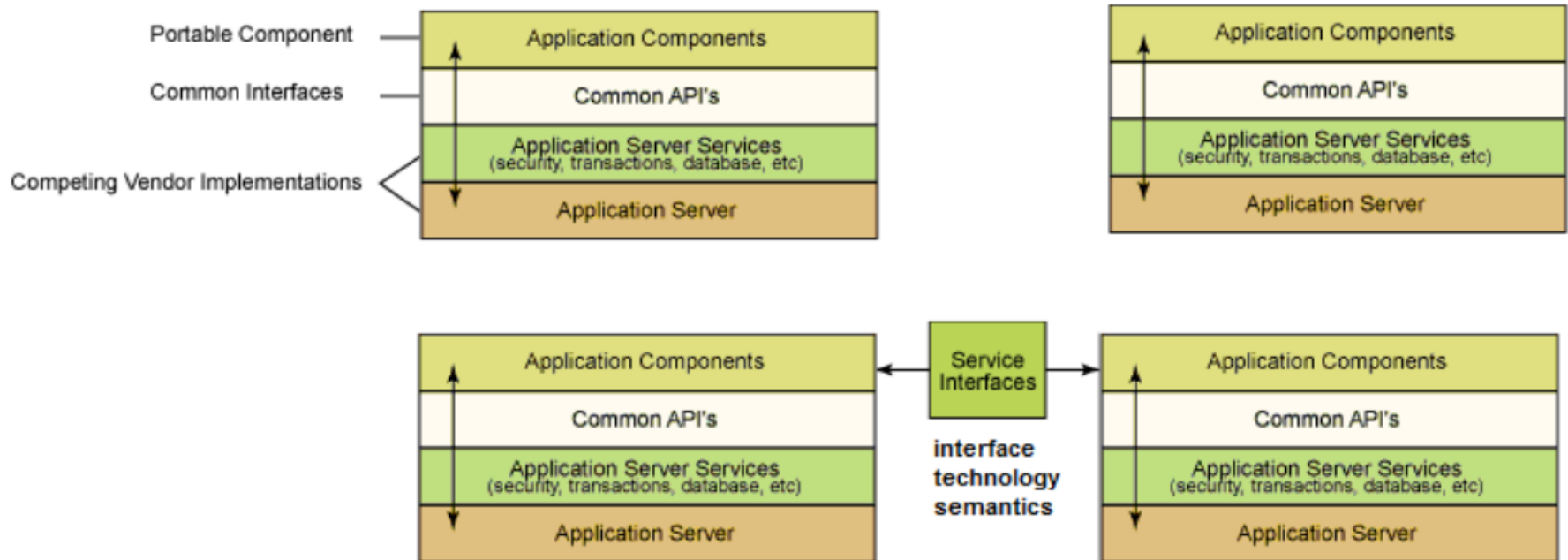
Projeto de Doutorado

■ Objetivo do Projeto de Doutorado

- Investigar uma possível solução para o problema de **lock-in** no contexto de TaaS;
- Reduzir o esforço e custo;
- Favorecer a interoperabilidade e portabilidade;
- Favorecer a manutenibilidade

Projeto de Doutorado

■ Interoperabilidade e Portabilidade



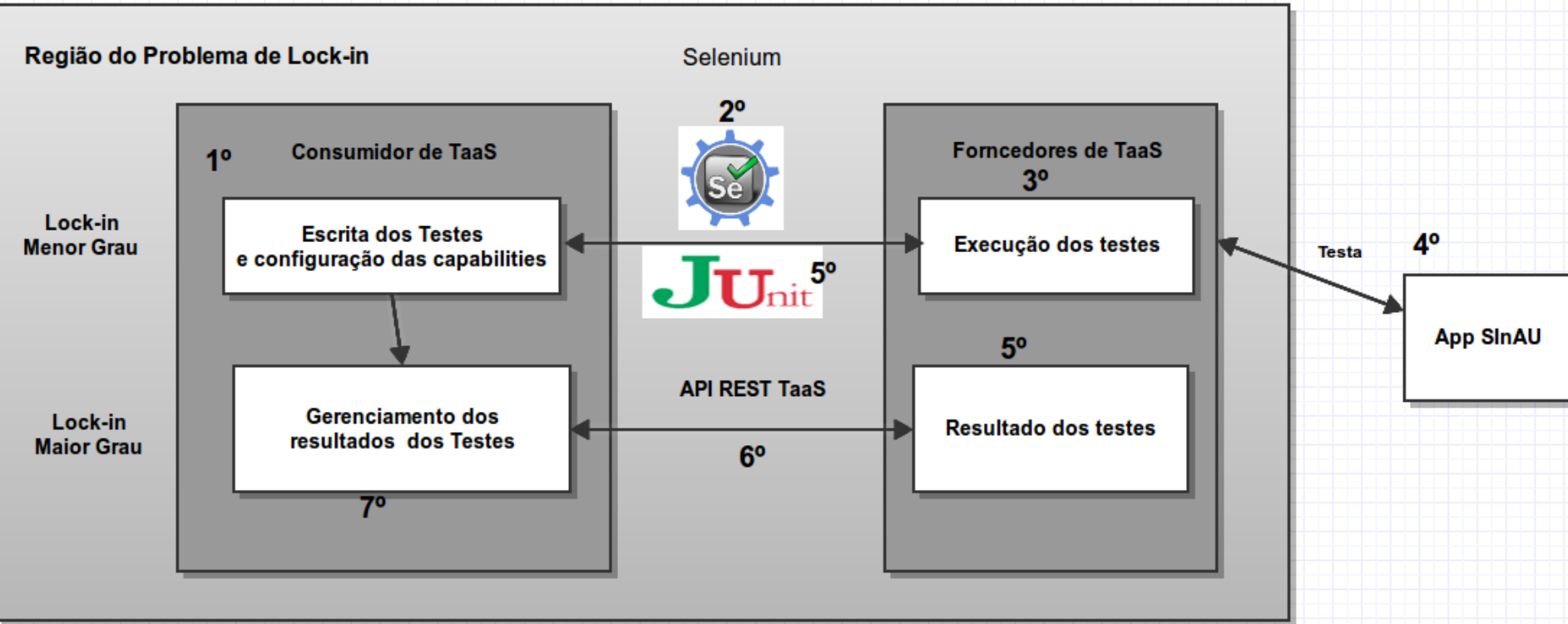
Projeto de Doutorado

■ Questões de Pesquisa do Doutorado:

- 1 - Como solucionar o reduzir os impactos ocasionados pelo problema de lock-in no contexto de Teste como Serviço (TaaS)?
- 2 – Quais as etapas do processo de teste na nuvem são afetadas pelo problema de lock-in?

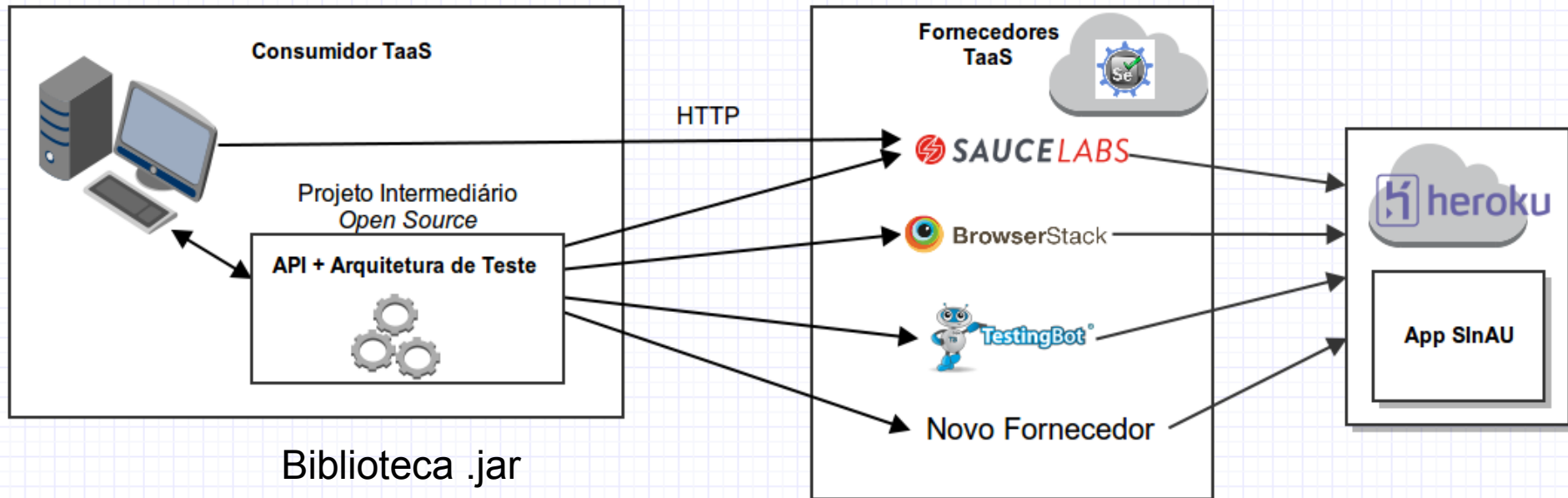
Projeto de Doutorado

Etapas do Processo de Teste na Nuvem





Projeto de Doutorado

Abordagem Proposta



Projeto de Doutorado

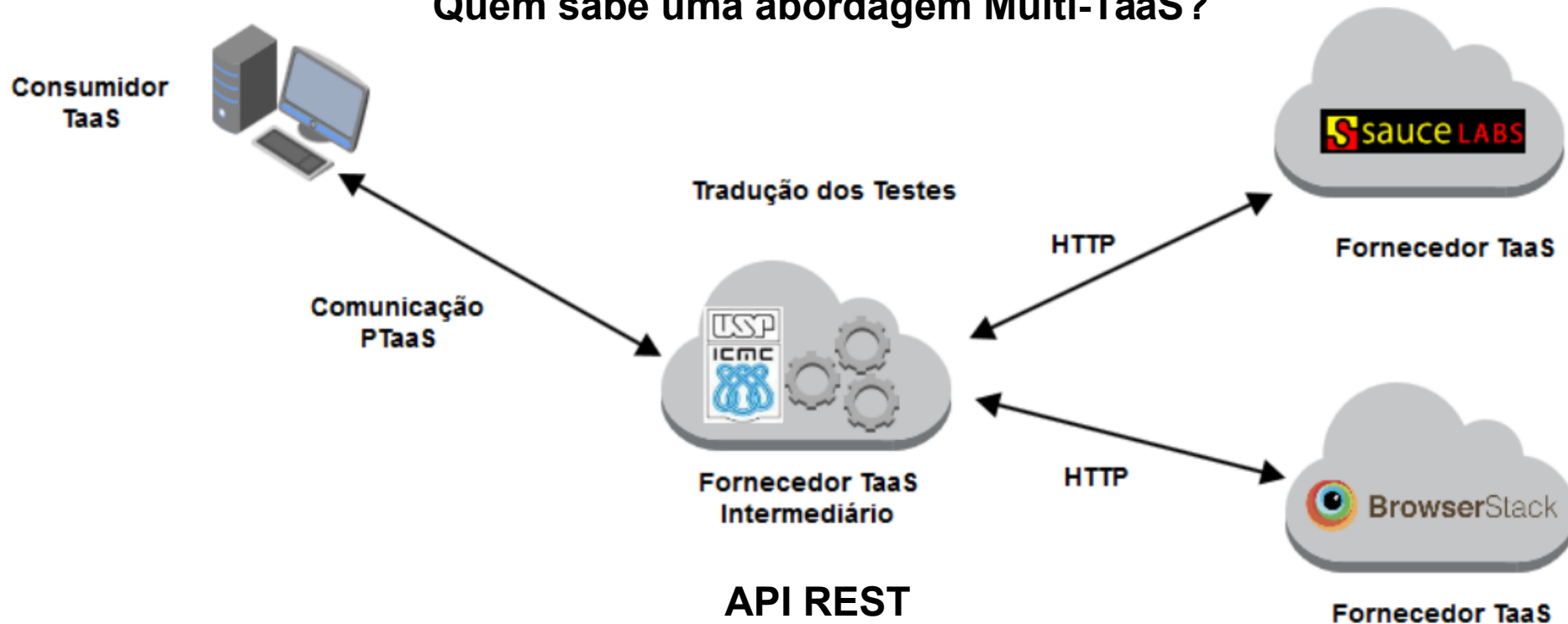
Abordagem Proposta

Especificação do Projeto  Novos TaaS no geral	1 - Padroniza o gerenciamento dos resultados dos testes por meio de relatórios, screenshots, videos... 2 - Especifica como um TaaS deve ser (Documentação da API REST)
Especificação do Selenium 	1 - Padroniza a escrita dos Testes 2 - Padroniza a execução dos Testes

Projeto de Doutorado

Trabalhos Futuros

Quem sabe uma abordagem Multi-TaaS?



Publicações

- Artigo: ***Impact of the Vendor Lock-in Problem on Testing as a Service (TaaS)*** – Aceito para publicação
- Congresso – Vancouver no Canadá:
 - IEEE International Conference on Cloud Engineering 2017

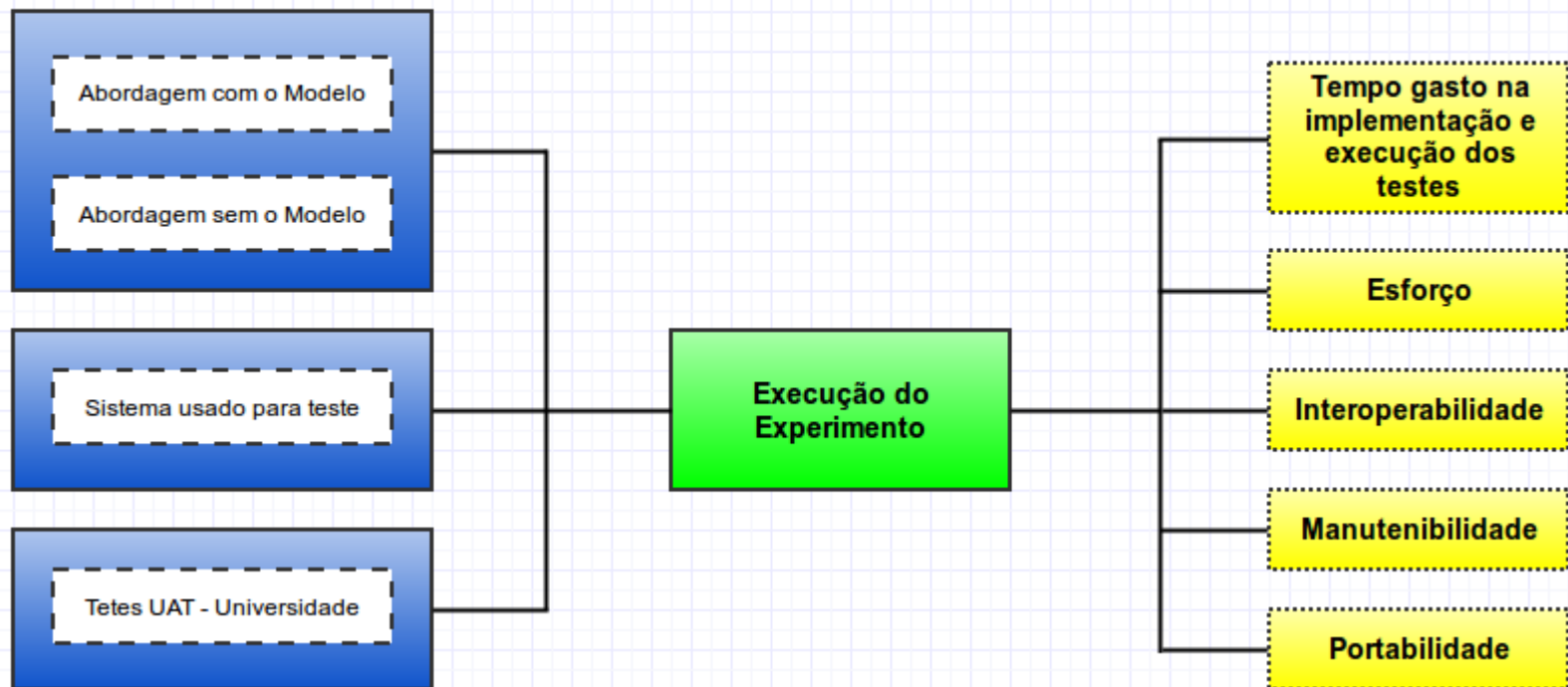
Publicações – Em andamento

- Artigo: ***A Research Roadmap of Vendor Lock-in Solutions to Testing as a Service (TaaS) - Journal***
- Lacunas de Pesquisa:
 - 1 - Evidências empíricas
 - 2 - Teste Estrutural (Cobertura de Código...)
 - 3 - Testes Baseados em Modelos
 - 4 - Testes Multi-tenancy
 - 5 - Testes Paralelos na Nuvem
 - 6 - Abordagens Semânticas para o Teste na Nuvem
 - 7 - Testes envolvendo Federação, Multi-cloud e Intercloud

Publicações – Em andamento

■ Experimento do Doutorado

Lock-in de Execução dos Testes



Legenda:



Fator



Tratamento

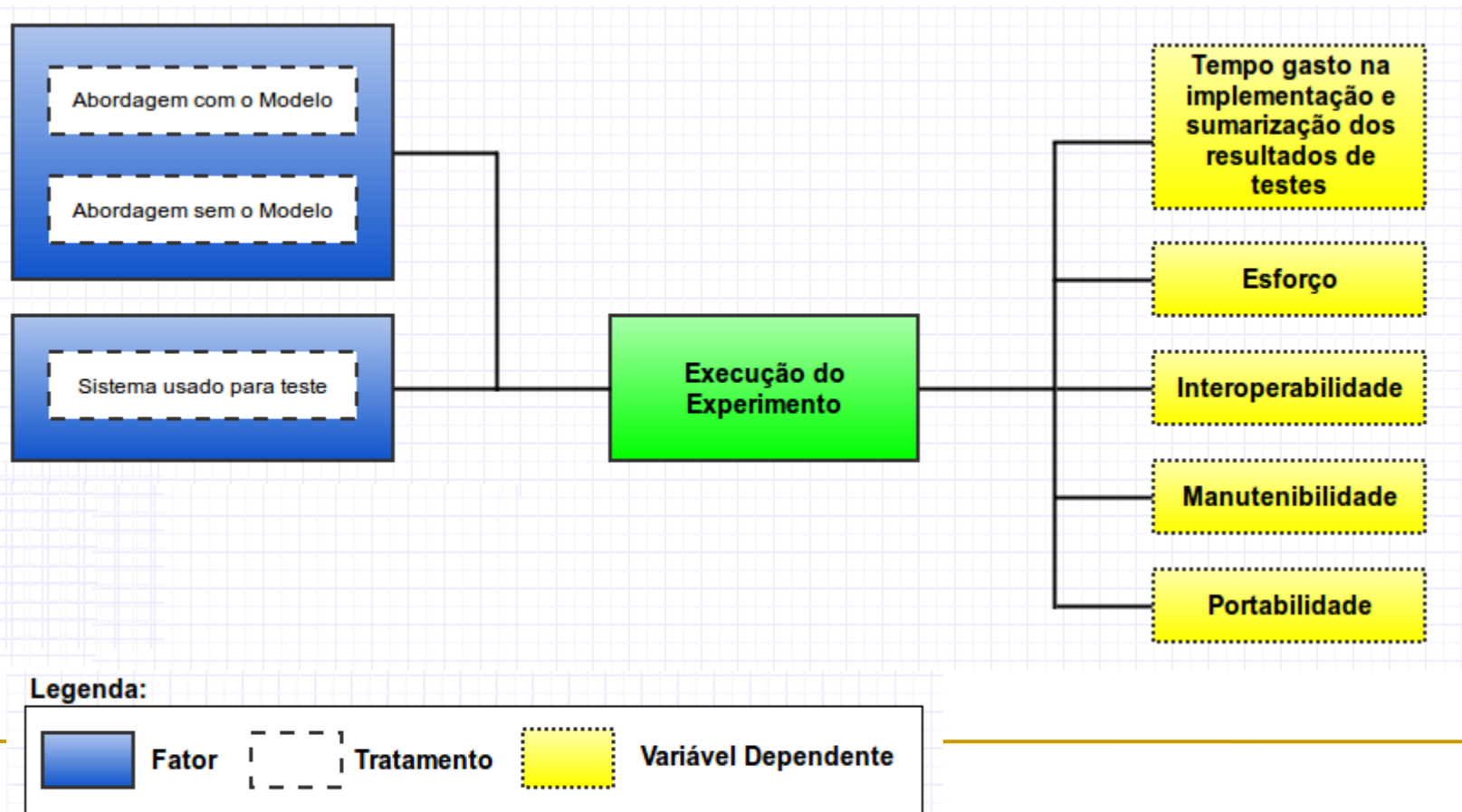


Variável Dependente

Publicações – Em andamento

■ Experimento do Doutorado

Lock-in de Gerenciamento e Sumarização dos Resultados de Testes



Hipóteses

Esforço

Hipótese Nula (H0): Esforço da migração (com modelo) = Esforço da migração (sem modelo)

Hipótese Alternativa (H1): Esforço da migração (com modelo) < Esforço da migração (sem modelo)

Hipótese Alternativa (H2): Esforço da migração (com modelo) > Esforço da migração (sem modelo)

Verificar o quanto as diferenças entre as hipóteses é significativa...

Hipóteses

Interoperabilidade

Hipótese Nula (H0): Interoperabilidade (com modelo) = Interoperabilidade (sem modelo)

Hipótese Alternativa (H1): Interoperabilidade (com modelo) > Interoperabilidade (sem modelo)

Hipótese Alternativa (H2): Interoperabilidade (com modelo) < Interoperabilidade (sem modelo)

Hipóteses

Manutenibilidade

Hipótese Nula (H_0): Manutenibilidade (com modelo) =
Manutenibilidade (sem modelo)

Hipótese Alternativa (H_1): Manutenibilidade (com modelo) >
Manutenibilidade (sem modelo)

Hipótese Alternativa (H_2): Manutenibilidade (com modelo) <
Manutenibilidade (sem modelo)

Hipóteses

Portabilidade

Hipótese Nula (H0): Portabilidade (com modelo) = Portabilidade (sem modelo)

Hipótese Alternativa (H1): Portabilidade (com modelo) > Portabilidade (sem modelo)

Hipótese Alternativa (H2): Portabilidade (com modelo) < Portabilidade (sem modelo)

Projeto de Doutorado

- Mostrar os testes de aceitação com a arquitetura de teste na nuvem
- Executar os testes sequencialmente no Sauce Labs, Browserstack e Testingbot
- Executar os testes paralelamente no Sauce Labs, Browserstack e Testingbot

Perguntas?

- “Habilidade é o que você é capaz de fazer.
Motivação determina o que você faz.
Atitude determina a qualidade do que você faz”
(Lou Holtz)

Obrigado!

Ricardo Ramos de Oliveira
ricardoramos@icmc.usp.br