

# AcceptanceNG: An Acceptance Testing Tool for Flex Applications



Rodrigo Almeida  
Rafael Barros  
Iuri de Silvio  
Eduardo Guerra

# Contexto

## P2D: necessidade de criação de testes de aceitação para aplicações Flex e AIR.

- A execução manual dos testes é custosa e demanda muito tempo.
- FlexUnit: não permite parar a execução para o carregamento dos componentes.
- Outras ferramentas suportam apenas testes com aplicações com Flex e não com AIR.

# Tipos de Teste em Ambientes Ágeis

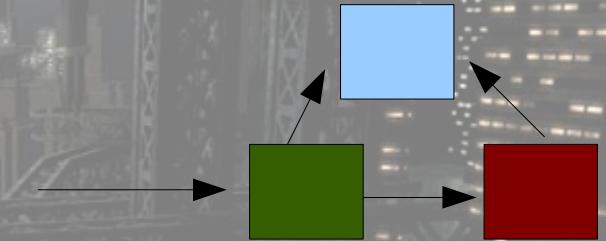
## Teste de Unidade

?



## Teste de Integração

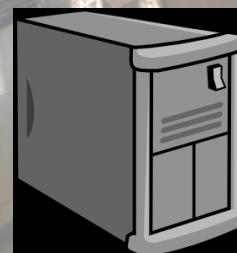
?



## Teste de Aceitação



?



## Teste de Regressão



?

# Testes de Aceitação



- Criados sob a perspectiva do cliente e preferencialmente por ele.
- Necessidade de automação, pois tomam muito tempo no ciclo de desenvolvimento.
- Deve levar em conta somente os efeitos percebidos pelo cliente.

# Arquiteturas de Execução

## IN-PROC

- A interação com a aplicação com o aplicativo ocorre no mesmo processo que ele está executando.
- Executa um aplicativo diferente do aplicativo final, adicionando incerteza aos testes.
- Permite acesso ao estado interno da aplicação.

## OUT-OF-PROC

- A interação ocorre em um processo separado do aplicativo sendo testado.
- Simula uma interação externa de um cliente real do sistema.
- Permite acesso apenas ao que é visualizado pelo cliente na aplicação.

# Ferramentas Existentes para Flex e AIR

## Selenium

- ✓ Out-of-proc
- ✓ Open-source
- ✗ Não testa AIR

## HP/Mercury QuickTest

- ✓ Out-of-proc
- ✓ Grava interação com cliente
- ✗ Não testa AIR

## FlexMonkey

- ✓ Testa aplicações AIR
- ✓ Grava interação com cliente
- ✗ In-proc
- ✗ Pode exigir intervenção manual impedindo uma automação de 100% dos testes

# AcceptanceNG

- Ferramenta para o teste de aplicações AIR através de código Java.
- Utiliza TestNG e Spring (pode usar funcionalidades como dependência de testes e injeção de dependências).
- Arquitetura in-proc (preciso incluir algumas bibliotecas para tornar a aplicação testável pela ferramenta).
- Baseado no Selenium e com idéias “inspiradas” no Ruby on Rails.

# Estrutura Interna do AcceptanceNG

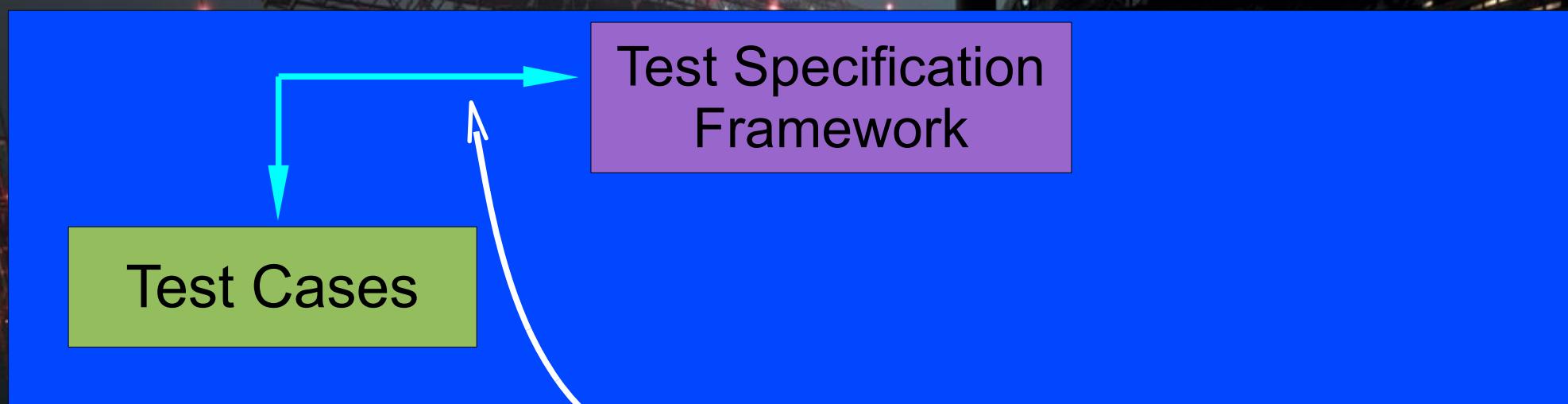
AIR Application  
under Test

Automation  
Library

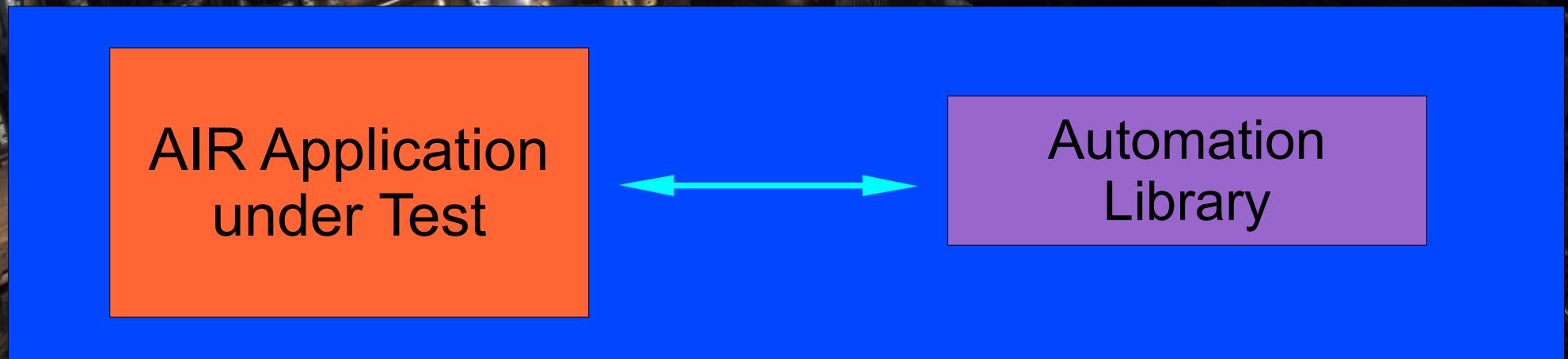


A Automation  
Library deve ser  
incluída junto com  
a aplicação para  
tornar possível  
simular a interação  
com a interface.

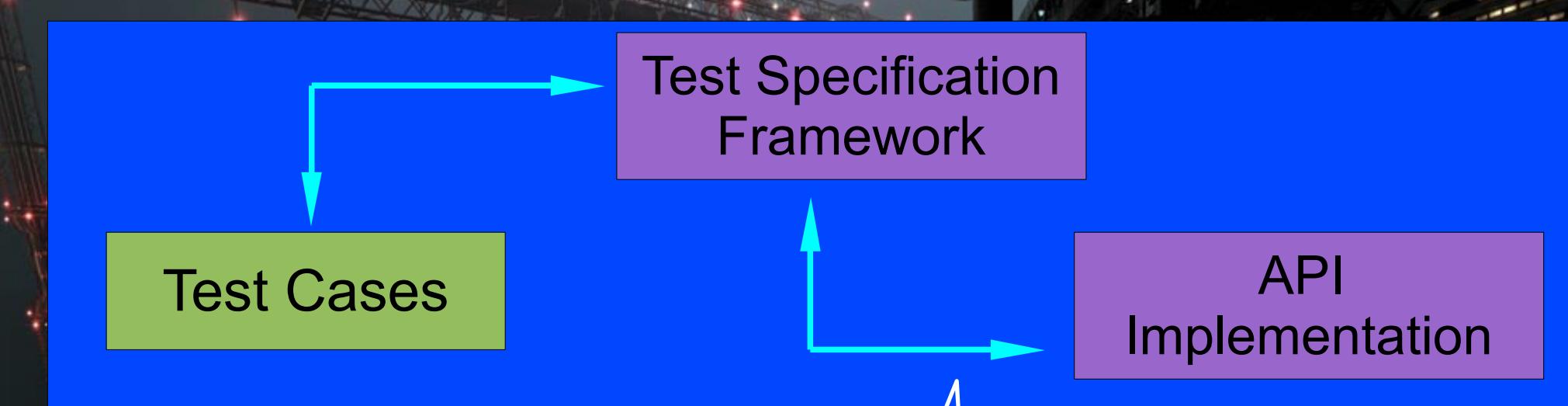
# Estrutura Interna do AcceptanceNG



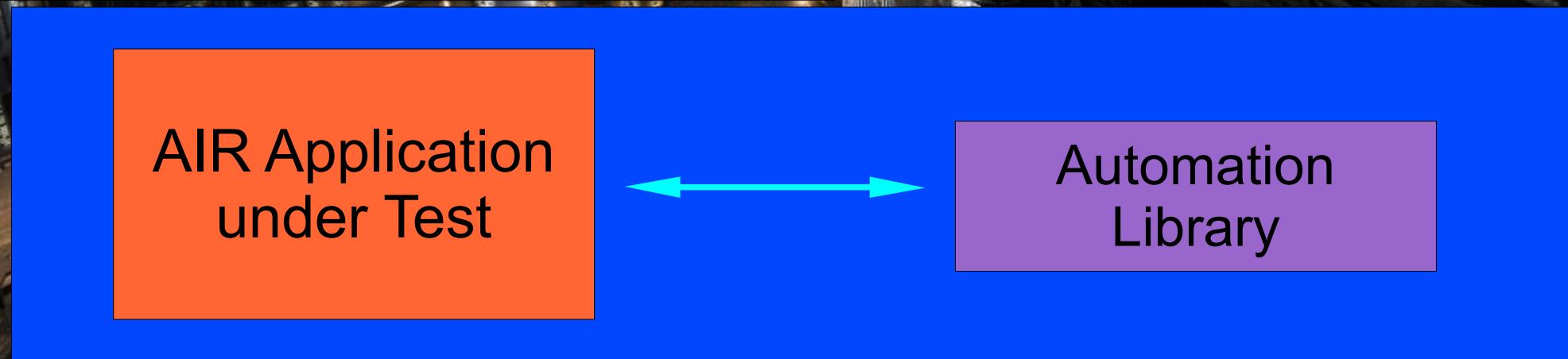
Define como testes devem ser estruturados e como deve-se interagir com a interface.



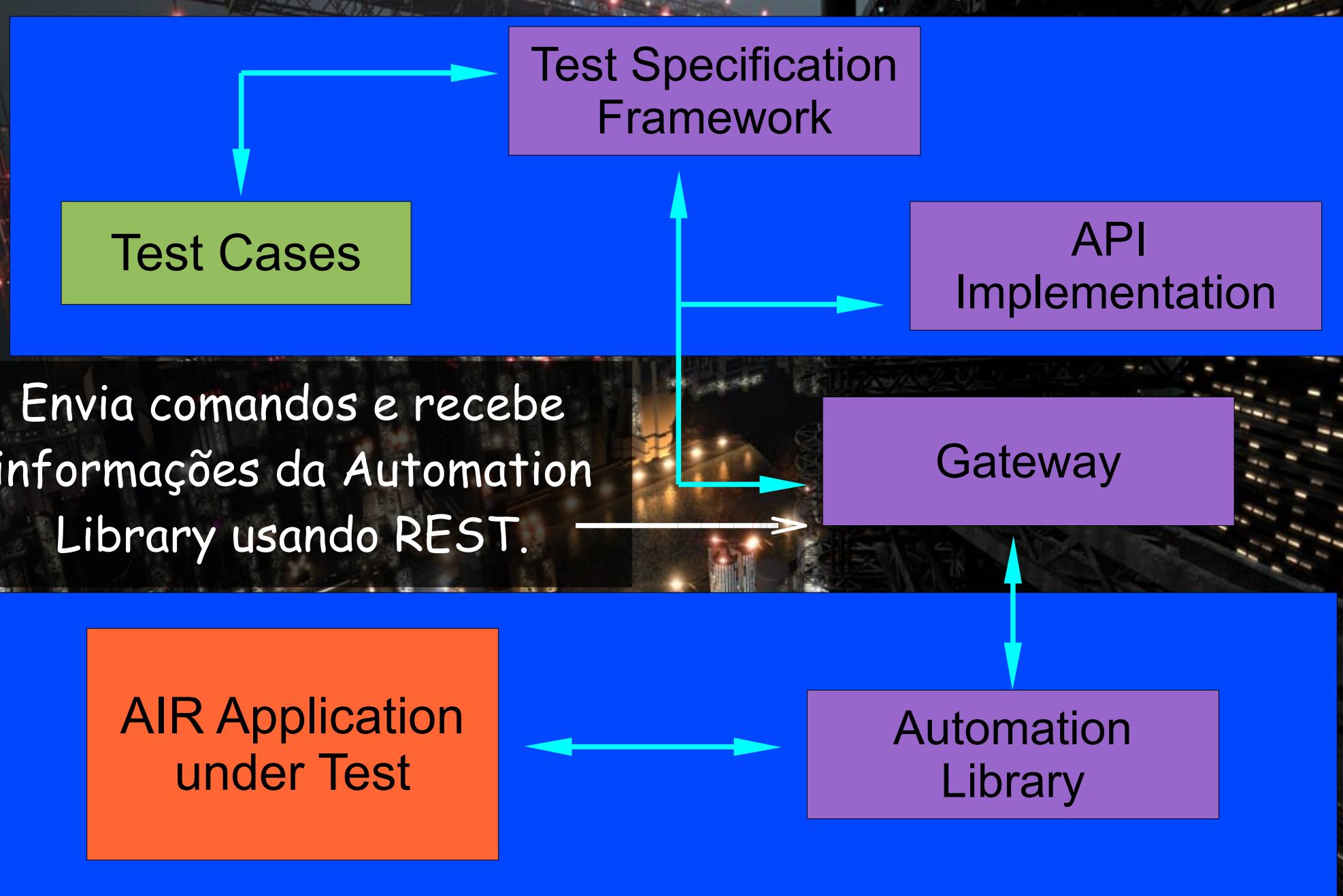
# Estrutura Interna do AcceptanceNG



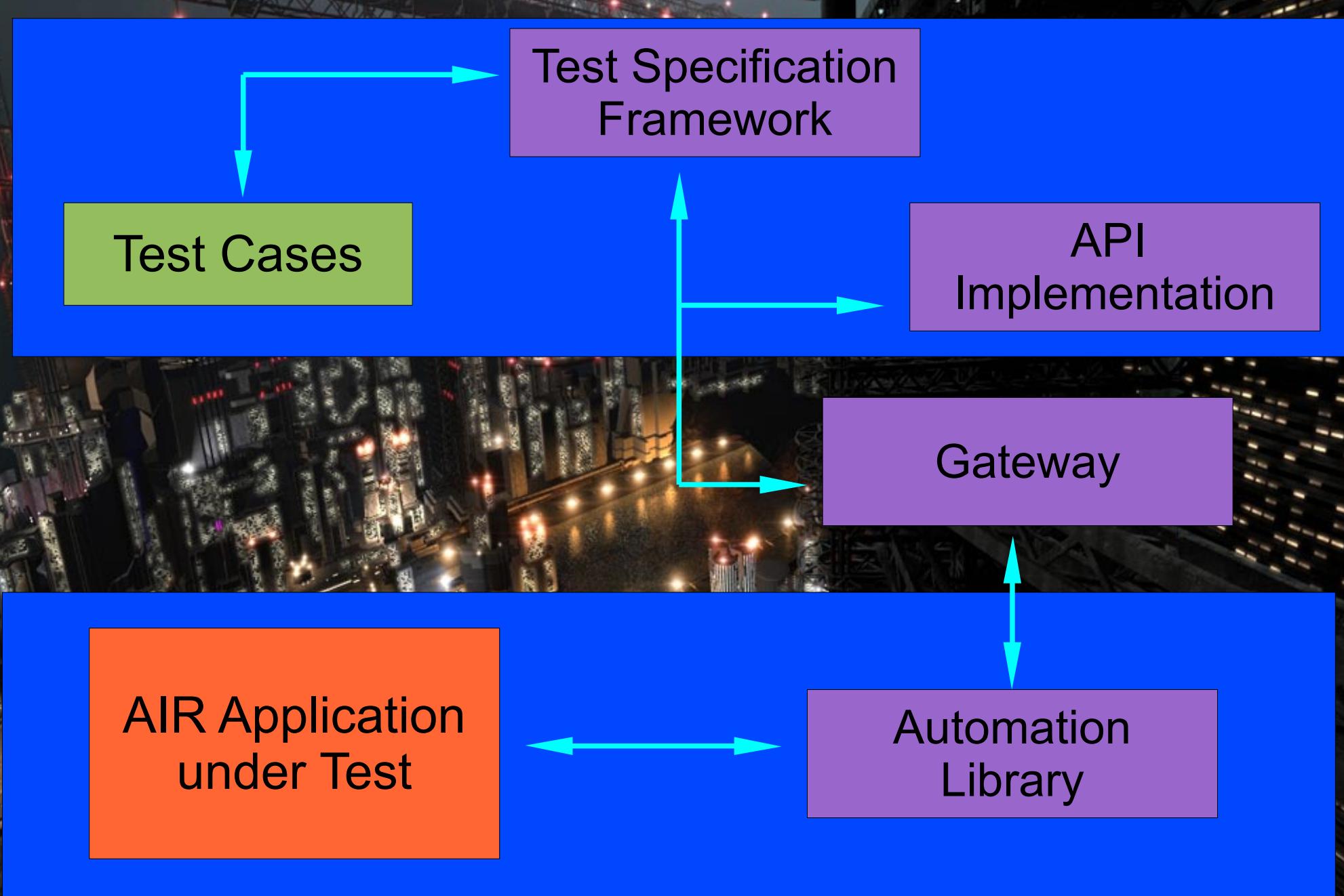
Traduz os comandos em Java  
para os comandos da  
Automation Library.



# Estrutura Interna do AcceptanceNG



# Estrutura Interna do AcceptanceNG



# Configurações

Para a execução dos testes é preciso configurar a aplicação para ser executada usando beans do Spring.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans ...some declarations from spring ...>
  <bean id="swfLauncher" class="br.com.p2d....launcher.AIRLauncher">
    <property name="swfUrl" value="bin-debug/financeTest.swf" />
  </bean>
  <bean id="api" class="br.com.p2d....mala.APIImplementationFactory" />
  <bean id="gatewayFactory" class="br.com.p2d....rpg.RPGFactory">
    <property name="swfLauncher" ref="swfLauncher" />
  </bean>
</beans>
```

# Exemplo de Test Case

Define qual arquivo de configuração deve ser utilizado.

```
@ContextConfiguration(locations={"/META-INF/ApplicationContext.xml"})
public class SendABugReportTest extends FlexAcceptanceTestCase
{
    @Test
    public void sendABugReport() throws Throwable
    {
        // Ensure there is enough time for the App to retrieve
        // available modules from the FeedbackService
        Thread.sleep(3000);
        button("btGoToBugReport").click();
    }
}
```

Esse comando é traduzido para um comando entendido pela Automation Library.

# AcceptanceNG – Outras Funcionalidades

- Possibilita a execução de testes remotamente, visto que a execução dos testes é demorada e dificulta o uso da máquina para outras tarefas (deixa muito lenta).
- Dá suporte a execução de outras tarefas, como a inicialização de servidores e inserção de informações em bancos de dados.



## Demo Video

# Utilização do AcceptanceNG na P2D

- A ferramenta começou a ser utilizada efetivamente em Janeiro de 2010.
- Existe um módulo com cerca de 40 casos de uso inteiramente testado com AcceptanceNG.
- Novos módulos e módulos já existentes estão começando a ser testados usando a ferramenta.

O uso da ferramenta gerou um feedback para equipe de desenvolvimento muito mais rápido que a equipe que fazia os testes de forma manual. Essa agilidade faz valer o investimento para automação dos testes.

# Conclusões

O AcceptanceNG atendeu os requisitos de testes de aceitação para aplicações RIA e foi utilizado com sucesso em um ambiente real na P2D.

- Exige uma configuração inicial trabalhosa, mas depois disso é simples de usar.
- Integrou-se bem com o Scrum (automação de testes de regressão permitem iterações menores).
- Futuramente pretende-se dar suporte a geração de testes direto da interface, a um maior número de componentes e a uma integração melhor com o Maven.



# Obrigado!



# Perguntas?



*guerraem@gmail.com*  
Twitter: @emguerra