

# Detecção de Violações de SLA em Coreografias de Serviços Web

**Candidato**

Victoriano Alfonso Phocco Diaz

**Orientador**

Daniel Macêdo Batista

Instituto de Matemática e Estatística  
Departamento de Ciência da Computação  
Universidade de São Paulo

Março de 2013

# Roteiro

- 1 Conceitos Básicos
- 2 Problema
- 3 QoS e Monitoramento em Coreografias de Serviços Web
- 4 Trabalhos Relacionados
- 5 Proposta
- 6 Experimentos e Resultados
- 7 Conclusões e Trabalhos Futuros

## 1 Conceitos Básicos

## 2 Problema

## 3 QoS e Monitoramento em Coreografias de Serviços Web

## 4 Trabalhos Relacionados

## 5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias

## 6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS
- Monitoramento

## 7 Conclusões e Trabalhos Futuros

# Serviço Web

Definição pela W3C [W3C,2004]:

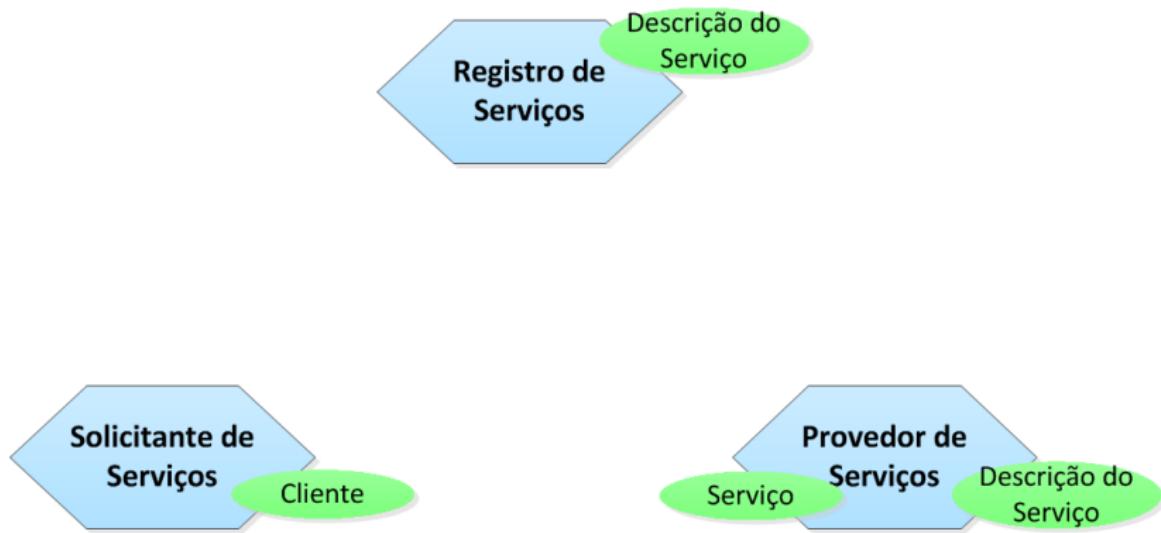
## Serviço Web

*A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.*

## SOA (Arquitetura Orientada a Serviços)

É um estilo de arquitetura de software cujo princípio fundamental prega que as funcionalidades implementadas pelas aplicações devem ser disponibilizadas na forma de serviços [SOA, 2006].

## SOA (2/2)



**Figura:** Triângulo da SOA (baseado em [W3C, 2002])

## SOA (2/2)

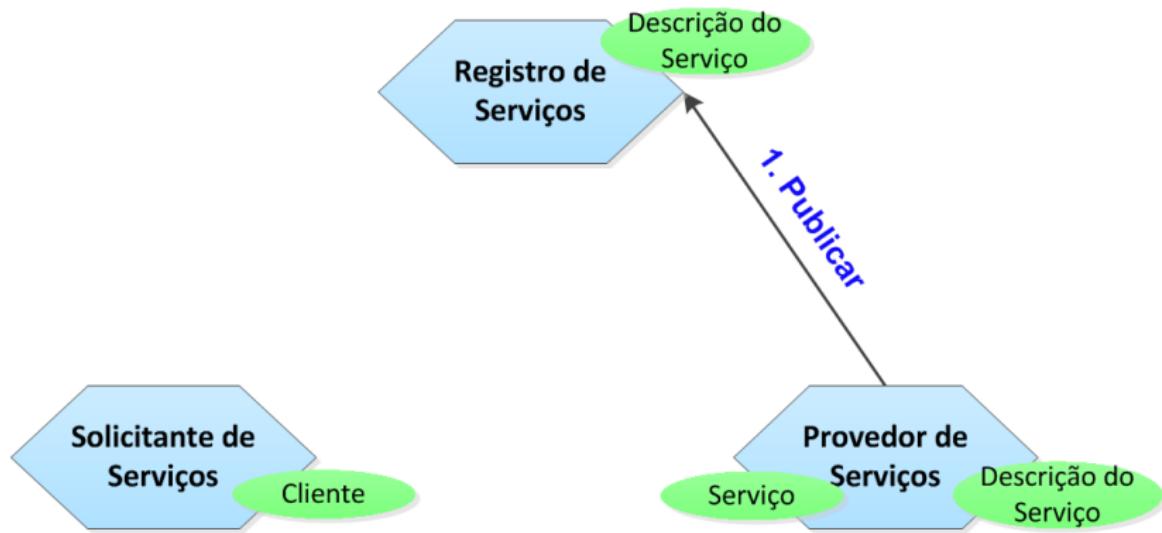


Figura: Triângulo da SOA (baseado em [W3C, 2002])

## SOA (2/2)

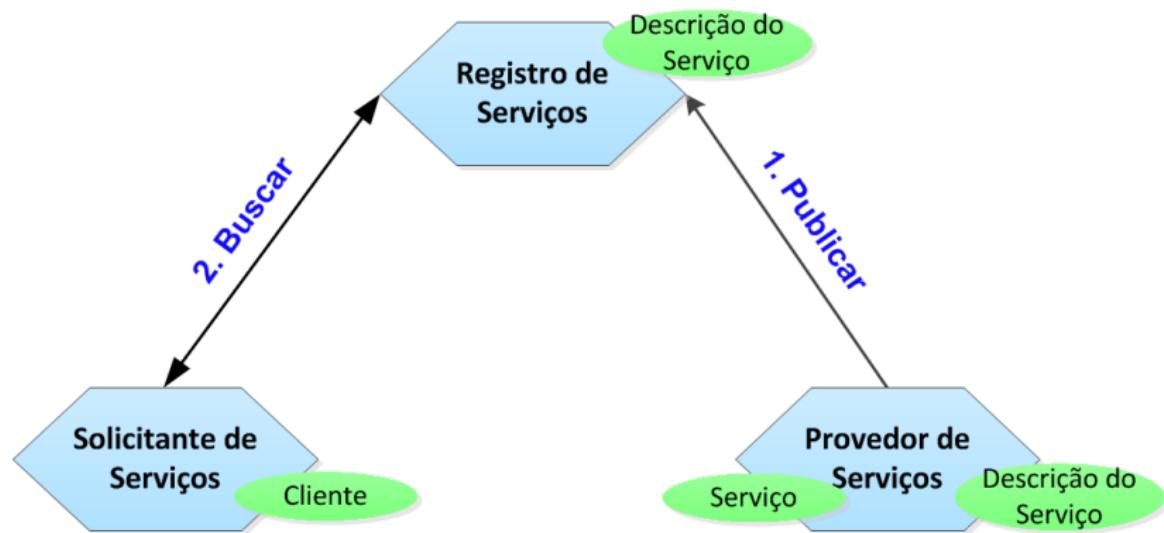


Figura: Triângulo da SOA (baseado em [W3C, 2002])

## SOA (2/2)

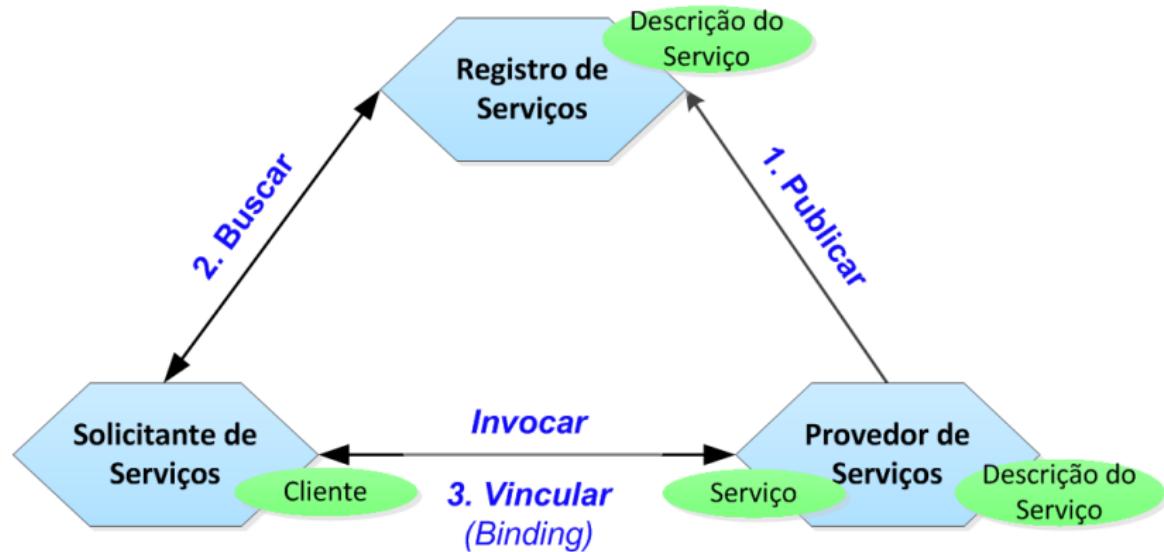


Figura: Triângulo da SOA (baseado em [W3C, 2002])

## SOA (2/2)



Figura: Triângulo da SOA (baseado em [W3C, 2002])

## SOC (Computação Orientada a Serviços)

É um novo paradigma de computação que utiliza serviços como blocos básicos de construção para suportar o desenvolvimento rápido, de baixo custo e de fácil composição de aplicações distribuídas heterogêneas [Papazoglou et al., 2006].

Elementos Chave:

- Serviços.
- SOA.
- Composição de Serviços .
- QoS .

# Orquestração de Serviços

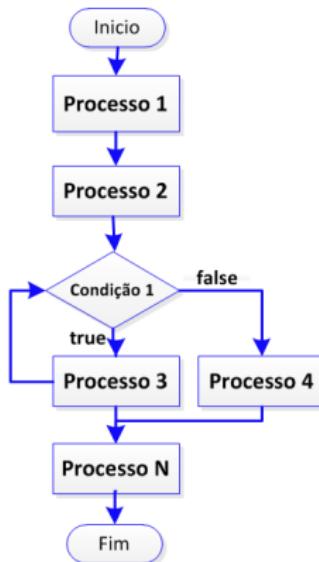


Figura: Orquestração de serviços

# Orquestração de Serviços

## Orquestração de Serviços

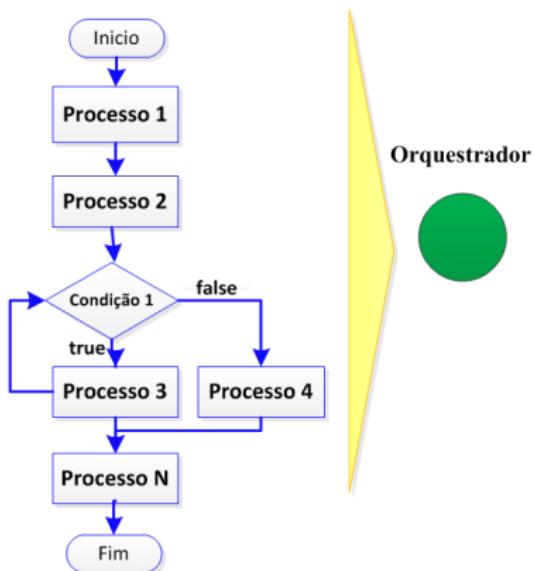


Figura: Orquestração de serviços

# Orquestração de Serviços

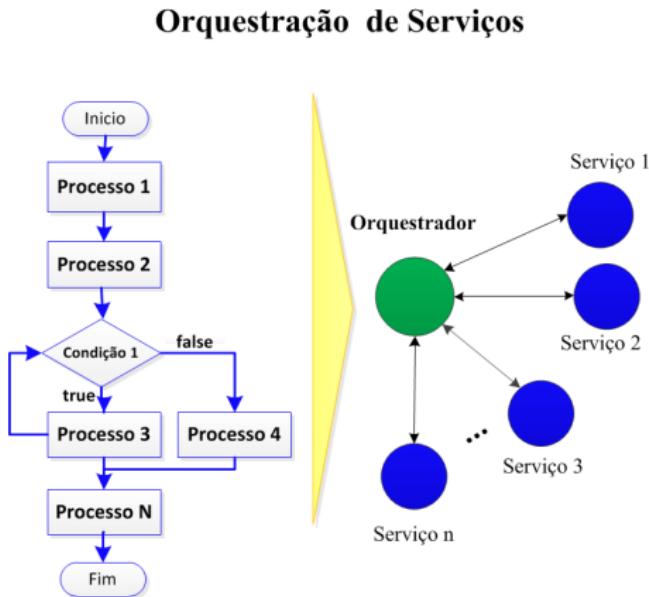


Figura: Orquestração de serviços

# Coreografia de Serviços

## Coreografia de Serviços

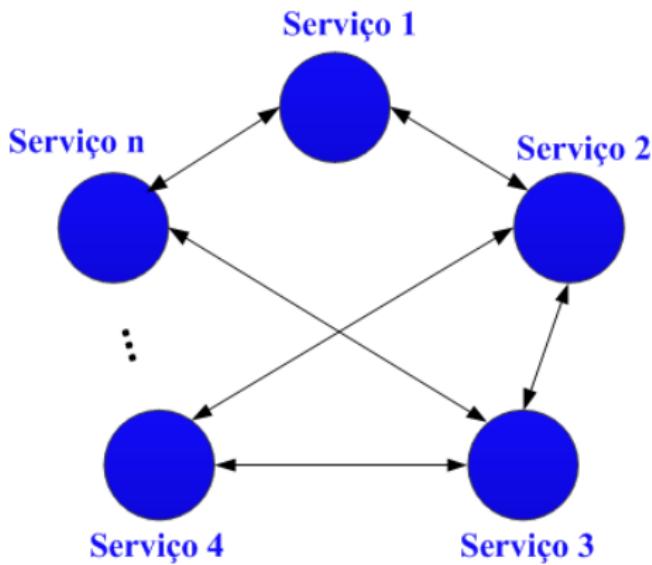
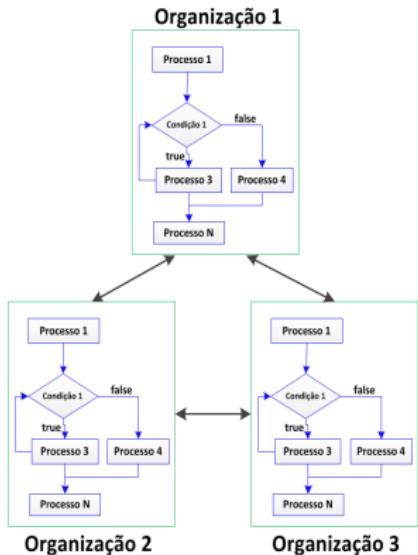


Figura: Coreografia de serviços

# Coreografia de Serviços

## Processos de Negócio Inter-Organizacionais



## Coreografia de Serviços

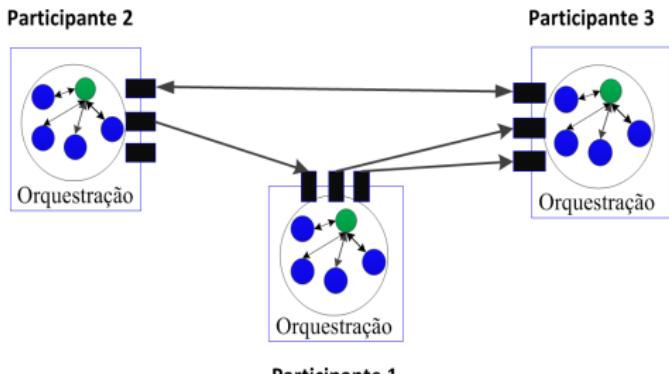
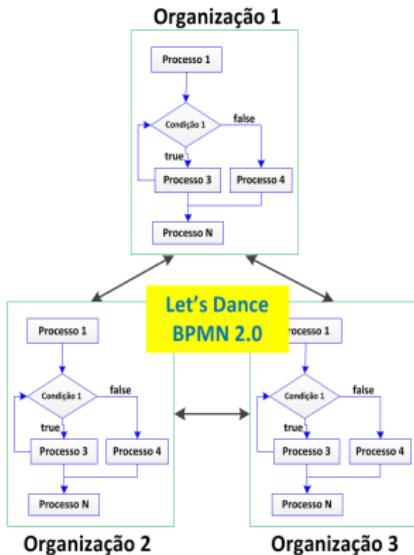


Figura: Coreografia de serviços

# Coreografia de Serviços

## Processos de Negócio Inter-Organizacionais



## Coreografia de Serviços

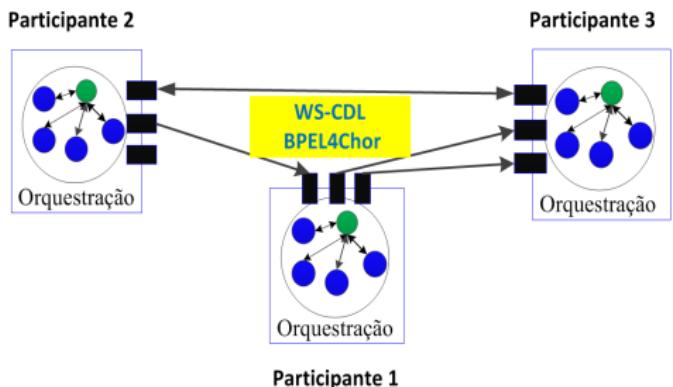


Figura: Coreografia de serviços

# Coreografia de Processos

- Uma Coreografia também é um processo.

# Coreografia de Processos

- Uma Coreografia também é um processo.
- **BPMN** (Business Process Model and Notation) é um padrão para modelagem de processo de negócios.
- **BPMN** suporta modelagem de coreografias.

# Coreografia de Processos

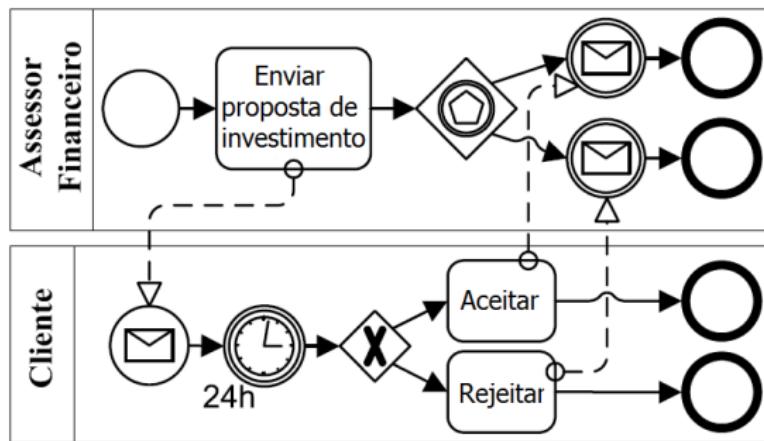
- Uma Coreografia também é um processo.
- **BPMN** (Business Process Model and Notation) é um padrão para modelagem de processo de negócios.
- **BPMN** suporta modelagem de coreografias.
- Duas abordagens de modelagem:
  - ▶ Modelo de Interconexão

# Coreografia de Processos

- Uma Coreografia também é um processo.
- **BPMN** (Business Process Model and Notation) é um padrão para modelagem de processo de negócios.
- **BPMN** suporta modelagem de coreografias.
- Duas abordagens de modelagem:
  - ▶ Modelo de Interconexão
  - ▶ Modelo de Interação

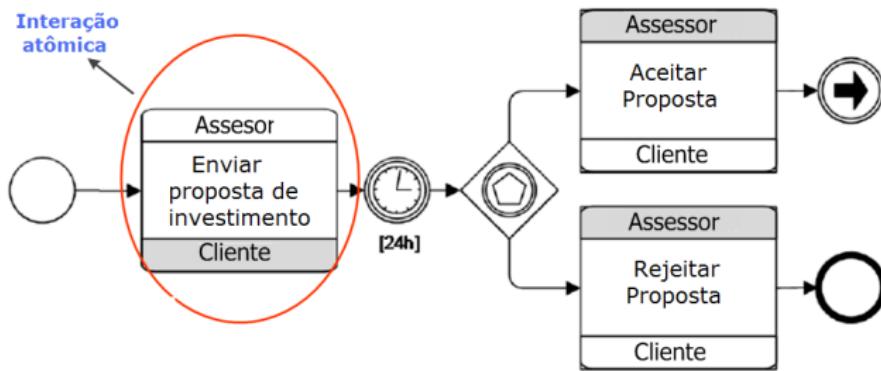
# Modelo de Interconexão

- Vistas públicas interconectadas.
- Uso de atividades de processos comuns.
- Colaborações em BPMN 2.



# Modelo de Interação

- Interações **capturadas globalmente**.
- Blocos básicos de construção: **interações atômicas** entre participantes.
- Suportado a partir do BPMN 2.



1 Conceitos Básicos

2 Problema

3 QoS e Monitoramento em Coreografias de Serviços Web

4 Trabalhos Relacionados

5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias

6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS
- Monitoramento

7 Conclusões e Trabalhos Futuros

# Objetivos

## Objetivo principal

- Detectar violações de SLAs em coreografias de serviços web.

## Objetivos secundários

- Propor uma técnica para definir SLAs baseada em restrições probabilísticas de QoS.
- Propor e implementar uma técnica de monitoramento “não intrusivo” de coreografias de serviços Web usando SLAs.
- Avaliar o desempenho das propostas.

# Justificativa

- Importância da **coreografia** de serviços Web.

# Justificativa

- Importância da **coreografia** de serviços Web.
- **QoS** é um fator importante na adaptação, seleção, otimização, composição na SOC.

# Justificativa

- Importância da **coreografia** de serviços Web.
- **QoS** é um fator importante na adaptação, seleção, otimização, composição na SOC.
- **Monitoramento** é uma base para a reação (adaptação, reconfiguração, renegociação, etc).
  - ▶ Detecção de falhas e violações de SLA.

# Justificativa

- Importância da **coreografia** de serviços Web.
- **QoS** é um fator importante na adaptação, seleção, otimização, composição na SOC.
- **Monitoramento** é uma base para a reação (adaptação, reconfiguração, renegociação, etc).
  - ▶ Detecção de falhas e violações de SLA.
- **Contratos probabilísticos** refletem melhor o comportamento dinâmico dos **atributos de QoS** dos serviços Web.

# Problema a ser resolvido

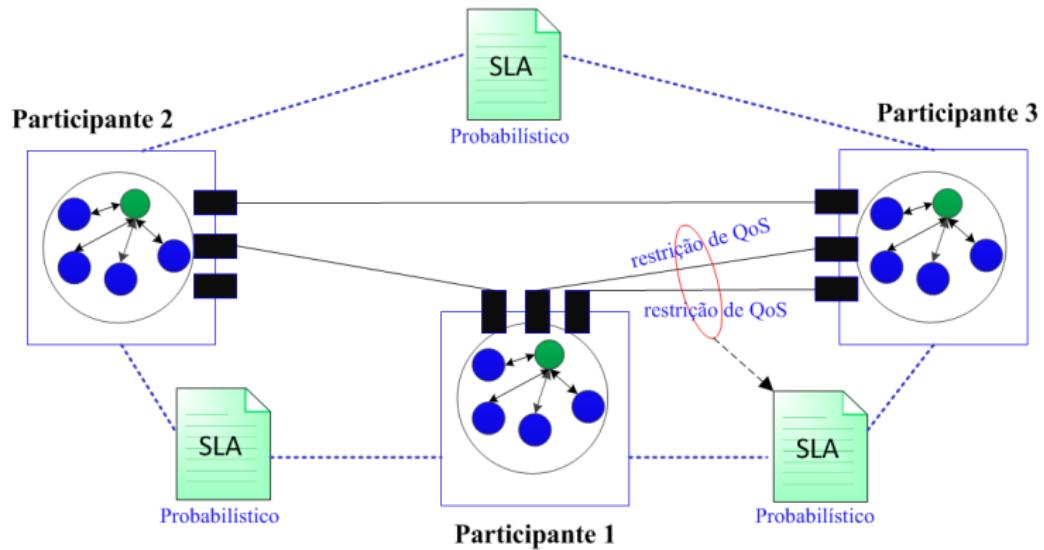


Figura: Problema a ser resolvido

1 Conceitos Básicos

2 Problema

3 QoS e Monitoramento em Coreografias de Serviços Web

4 Trabalhos Relacionados

5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias

6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS
- Monitoramento

7 Conclusões e Trabalhos Futuros

# Problema a ser resolvido

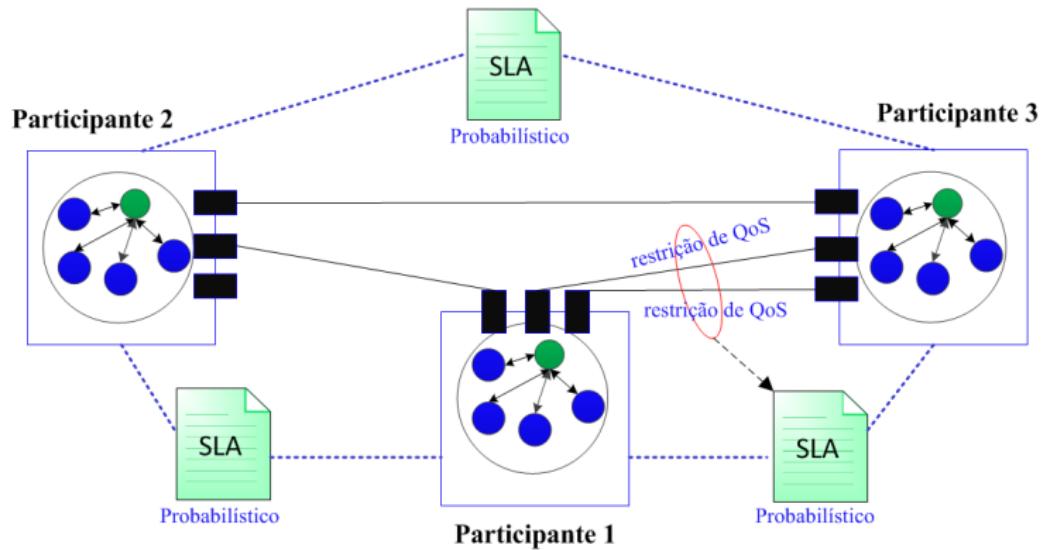


Figura: Problema a ser resolvido

- Qualidade de Serviço : QoS.
- **Funcionalidade/serviço** = Quais operações o sistema executa.
  - ▶ Exemplo: compra de passagens de avião.
- **QoS/Característica Não Funcional** = Quão bem o sistema executa os serviços.
  - ▶ Exemplo: O tempo médio de resposta é 2 segundos.
- Importante em Composição de Serviços : **QoS-aware Composition**.

# Qualidade de Serviço

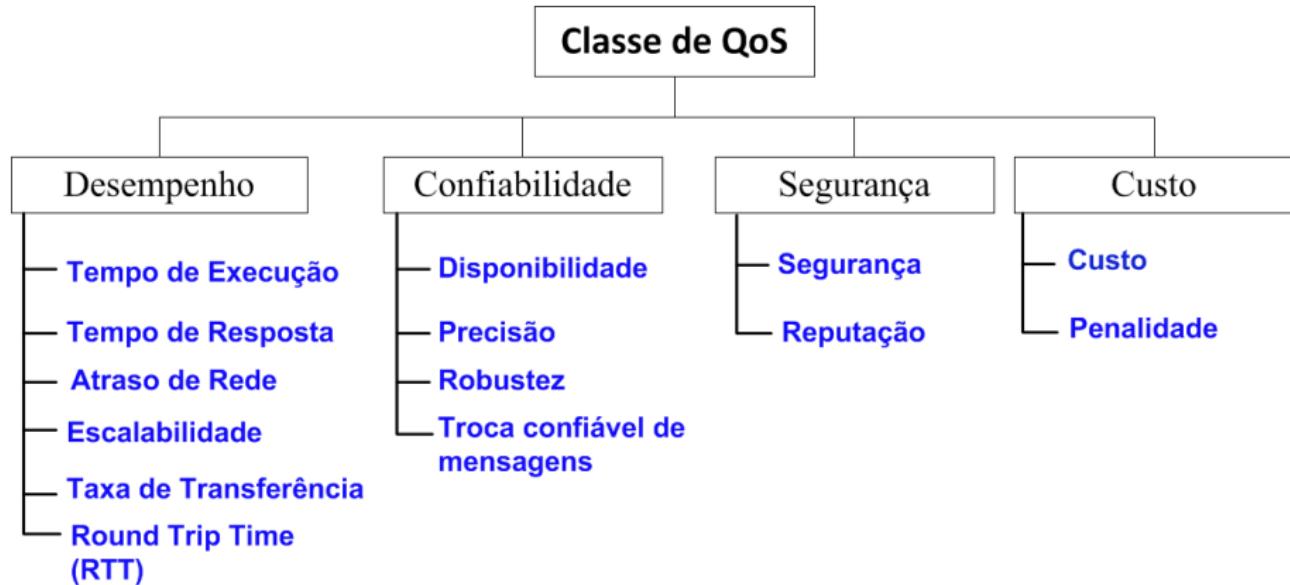


Figura: Taxonomia de atributos de QoS [Rosenberg et al.,2006]

# Cálculo de QoS

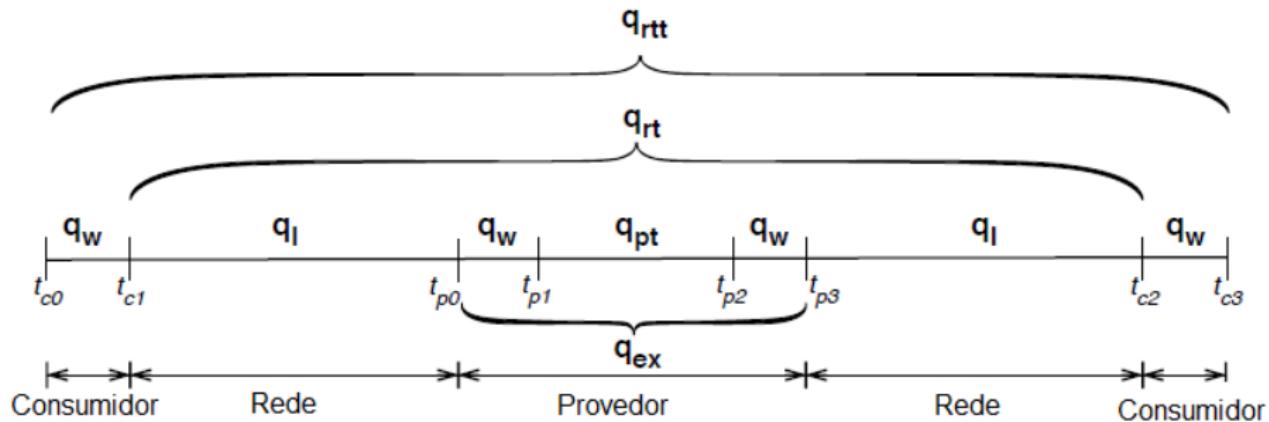


Figura: Instantes de tempo na utilização de um serviço Web [Michlmayr et al., 2009]

# Cálculo de QoS

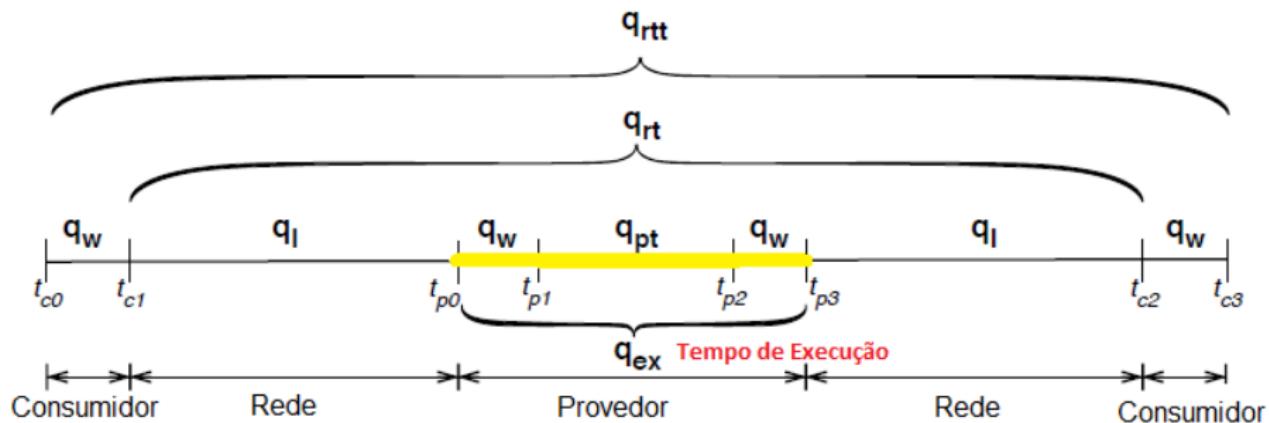


Figura: Instantes de tempo na utilização de um serviço Web

# Cálculo de QoS

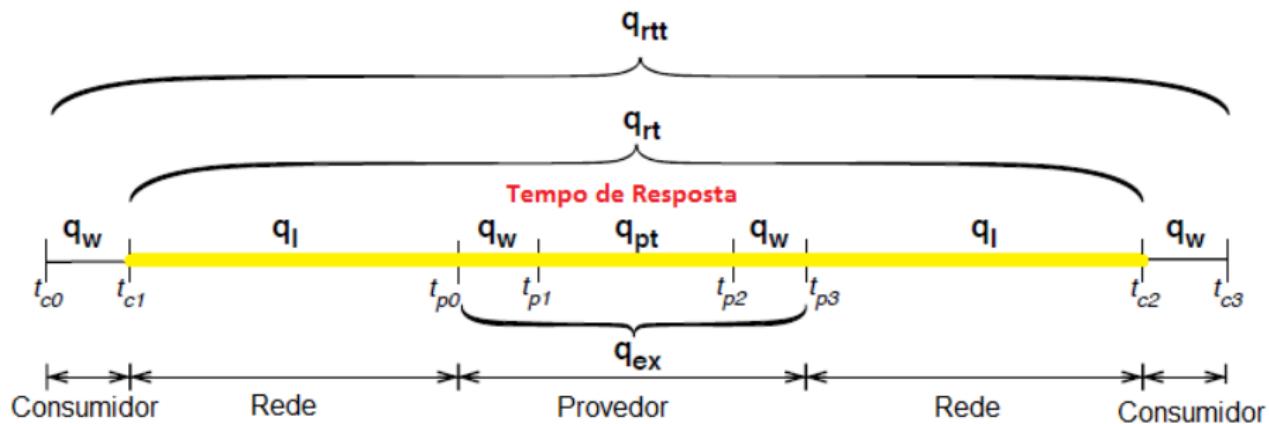


Figura: Instantes de tempo na utilização de um serviço Web

# Cálculo de QoS

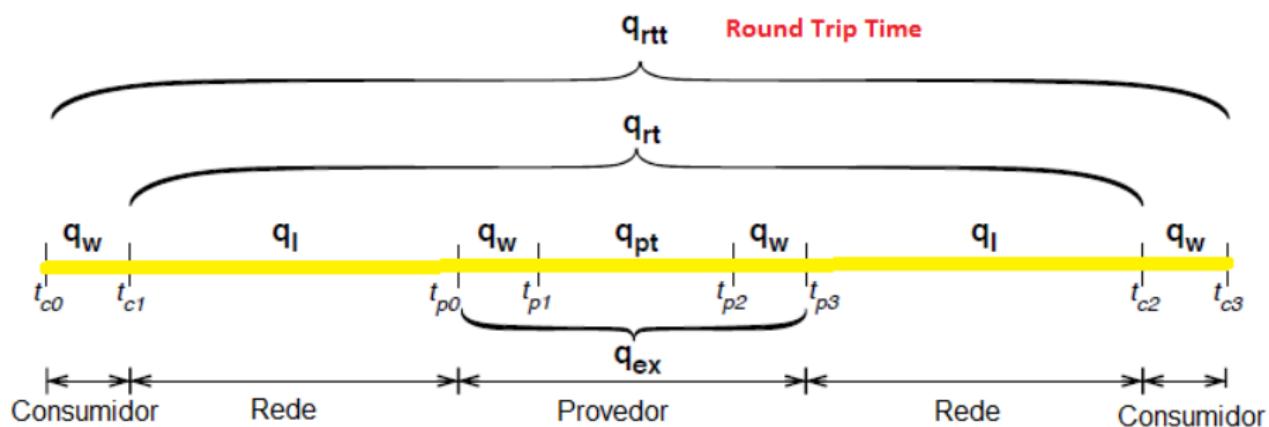


Figura: Instantes de tempo na utilização de um serviço Web

- **Contrato** = Acordo formal entre uma ou mais partes, define requisitos e garantias das partes.

- **Contrato** = Acordo formal entre uma ou mais partes, define requisitos e garantias das partes.
- **SLA** = Contrato que envolve requisitos e garantias de QoS.

- **Contrato** = Acordo formal entre uma ou mais partes, define requisitos e garantias das partes.
- **SLA** = Contrato que envolve requisitos e garantias de QoS.
- **Um SLA consiste de:**
  - ▶ Partes
  - ▶ Operações do serviço:
  - ▶ Obrigações:

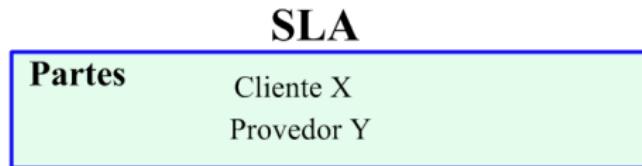
- **Contrato** = Acordo formal entre uma ou mais partes, define requisitos e garantias das partes.
- **SLA** = Contrato que envolve requisitos e garantias de QoS.
- **Um SLA consiste de:**
  - ▶ Partes
  - ▶ Operações do serviço:
    - ★ Operações
  - ▶ Obrigações:

- **Contrato** = Acordo formal entre uma ou mais partes, define requisitos e garantias das partes.
- **SLA** = Contrato que envolve requisitos e garantias de QoS.
- **Um SLA consiste de:**
  - ▶ Partes
  - ▶ Operações do serviço:
    - ★ Operações
    - ★ **Parâmetros de SLA:** define as métricas de QoS envolvidas.
  - ▶ Obrigações:

- **Contrato** = Acordo formal entre uma ou mais partes, define requisitos e garantias das partes.
- **SLA** = Contrato que envolve requisitos e garantias de QoS.
- **Um SLA consiste de:**
  - ▶ Partes
  - ▶ Operações do serviço:
    - ★ Operações
    - ★ **Parâmetros de SLA:** define as métricas de QoS envolvidas.
  - ▶ Obrigações:
    - ★ **Garantias de QoS (objetivos ou restrições).**

- **Contrato** = Acordo formal entre uma ou mais partes, define requisitos e garantias das partes.
- **SLA** = Contrato que envolve requisitos e garantias de QoS.
- **Um SLA consiste de:**
  - ▶ Partes
  - ▶ Operações do serviço:
    - ★ Operações
    - ★ **Parâmetros de SLA:** define as métricas de QoS envolvidas.
  - ▶ Obrigações:
    - ★ **Garantias de QoS (objetivos ou restrições).**
    - ★ Ações a serem tomadas se as garantias forem descumpridas (**reação**).

# Exemplo de SLA



**Figura:** Um exemplo simples de um SLA

# Exemplo de SLA

SLA	
<b>Partes</b>	Cliente X Provedor Y
<b>Operações</b>	Comprar Passagem
<b>Parâmetros de SLA</b>	
Tempo de Resposta Disponibilidade	
<b>Métricas de QoS</b>	
Tempo de Resposta, ms, número real	

Figura: Um exemplo simples de um SLA

# Exemplo de SLA

SLA	
<b>Partes</b>	Cliente X Provedor Y
<b>Operações</b>	Comprar Passagem
<b>Parâmetros de SLA</b>	Tempo de Resposta Disponibilidade
<b>Métricas de QoS</b>	Tempo de Resposta, ms, número real
<b>Garantias</b>	Tempo de Resposta < 10 ms Disponibilidade > 90%
<b>Ações caso descumprimento</b>	Notificação Renegociação

Figura: Um exemplo simples de um SLA

# Agregação de QoS

- Processo de obter o valor cumulativo da QoS da composição a partir dos valores de QoS dos seus serviços componentes.
- Não existe solução geral.
- **Depende do atributo de QoS e do modelo de composição.**
- Abordagens:
  - ▶ Somas, Máximos, Mínimos, Médias, etc.
  - ▶ Analíticas: Redes de Petri, Redes de Fila, etc.
  - ▶ Heurísticas: Algoritmos Genéticos.
  - ▶ **Simulação.**

# Exemplo de Agregação de QoS

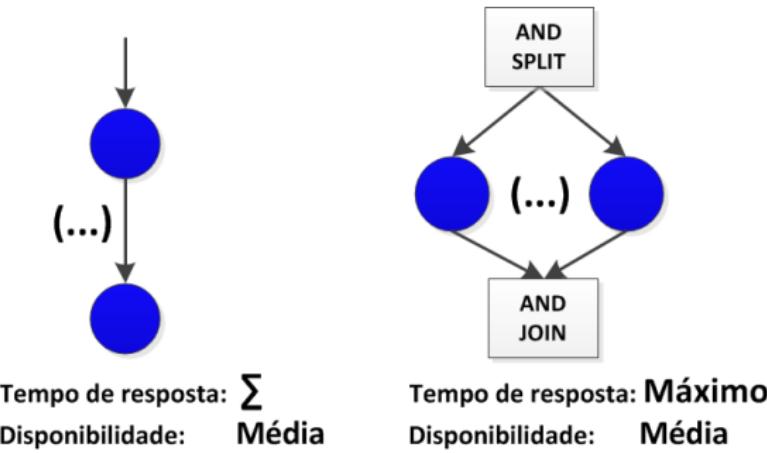


Figura: Exemplo de Agregação de QoS

# Contratos Rígidos

- Os contratos são tipicamente realizados em base a **restrições rígidas** (*hard contracts*):

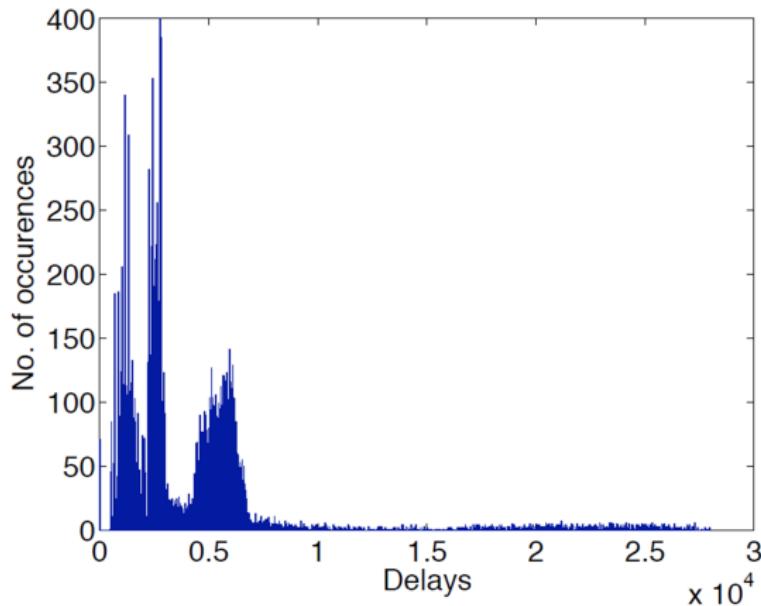
# Contratos Rígidos

- Os contratos são tipicamente realizados em base a **restrições rígidas** (*hard contracts*):
  - ▶ Tempo de resposta < 10 ms.

# Contratos Rígidos

- Os contratos são tipicamente realizados em base a **restrições rígidas** (*hard contracts*):
  - ▶ Tempo de resposta < 10 ms.
- Contratos rígidos não refletem o comportamento dinâmico da QoS dos serviços Web.

# Comportamento dinâmico de atributos de QoS



**Figura:** Tempos de resposta de 20,000 chamadas de um serviço [Rosario et al., 2008]

# Contratos Não Rígidos

- Contratos não rígidos (*soft contracts*):
  - ▶ **Tempo de resposta < 10 ms, em 95% dos casos.**

Desse jeito, não é possível compor esse tipo de restrições ou contratos, isto é, composição de restrições.

# Contratos Não Rígidos

- Contratos não rígidos (*soft contracts*):
  - ▶ **Tempo de resposta < 10 ms, em 95% dos casos.**  
Desse jeito, não é possível compor esse tipo de restrições ou contratos, isto é, composição de restrições.
- **Solução:** contratos probabilísticos não rígidos (*probabilistic soft contracts*).
  - ▶ **Para cada parâmetro de QoS (tempo de resposta). Eu ofereço sua distribuição de probabilidade e garanto que não será pior do que isso.**

# Contratos Não Rígidos

- Contratos não rígidos (*soft contracts*):
  - ▶ **Tempo de resposta < 10 ms, em 95% dos casos.**  
Desse jeito, não é possível compor esse tipo de restrições ou contratos, isto é, composição de restrições.
- **Solução:** contratos probabilísticos não rígidos (*probabilistic soft contracts*).
  - ▶ **Para cada parâmetro de QoS (tempo de resposta). Eu ofereço sua distribuição de probabilidade e garanto que não será pior do que isso.**
- As **restrições probabilísticas** podem ser compostas.
  - ▶ Existem algumas abordagens para orquestração.
  - ▶ **Não existem abordagens para coreografias .**
  - ▶ **Tratam somente tempo de resposta .**

# Monitoramento baseado em QoS

Responsabilidades:

- Mede e calcula valores de métricas de QoS, também inclui **agregação** de valores dos atributos de QoS.
- Verifica se existe violação de alguma restrição de QoS.
- Monitoramento de Coreografias deve ser “não intrusivo” .

# Abordagens de Monitoramento

Monitoramento Intrusivo: **Instrumentação**

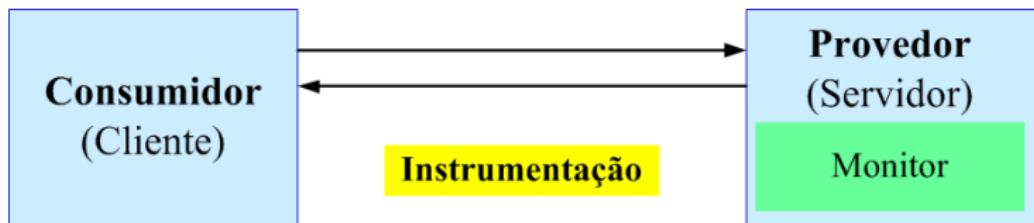


Figura: Monitoramento por Instrumentação

# Abordagens de Monitoramento

Monitoramento Não Intrusivo: **Interceptação**



Figura: Monitoramento por Interceptação

# Abordagens de Monitoramento

Monitoramento Não Intrusivo: **Probe-Request**

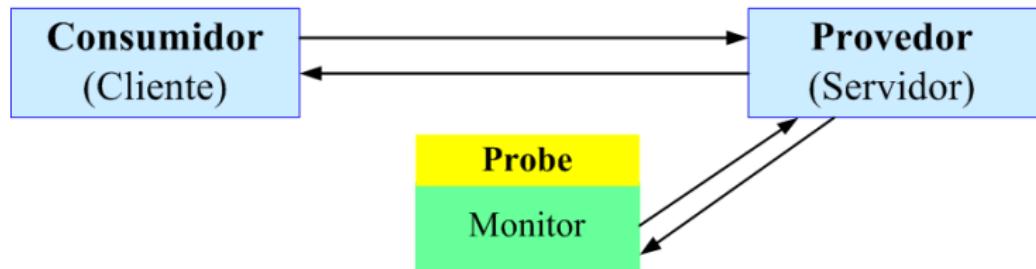


Figura: Monitoramento mediante Probe-Request

# Abordagens de Monitoramento

Monitoramento Não Intrusivo: **Sniffing**

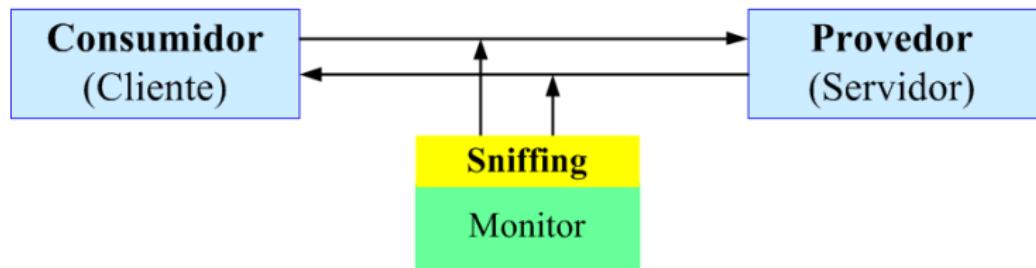


Figura: Monitoramento mediante sniffing

# Camadas no Monitoramento



Figura: Camadas do Monitoramento

# Camadas no Monitoramento

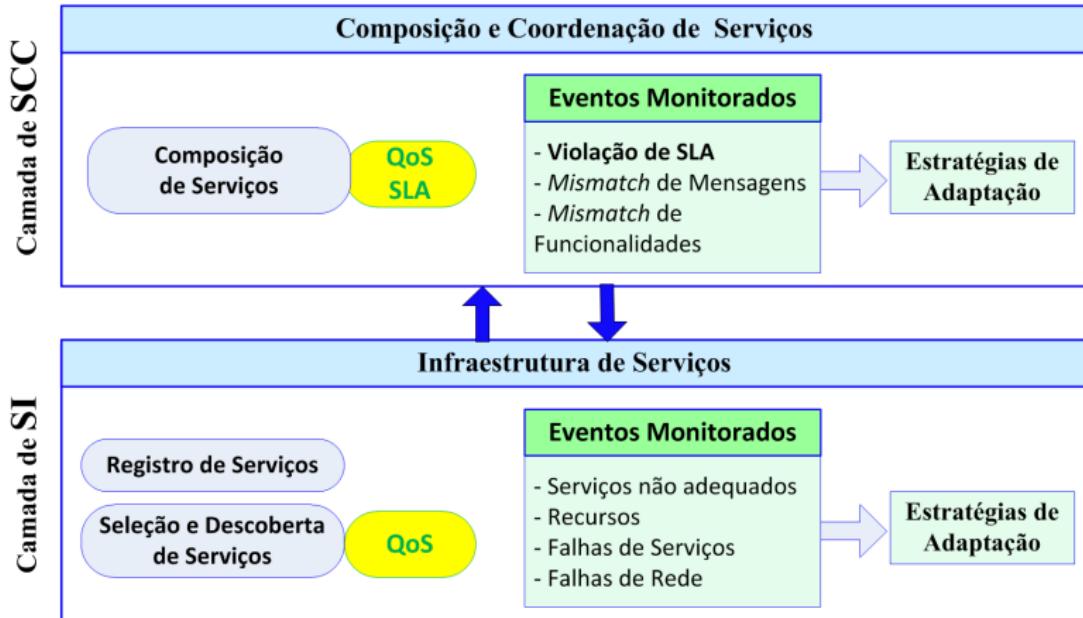
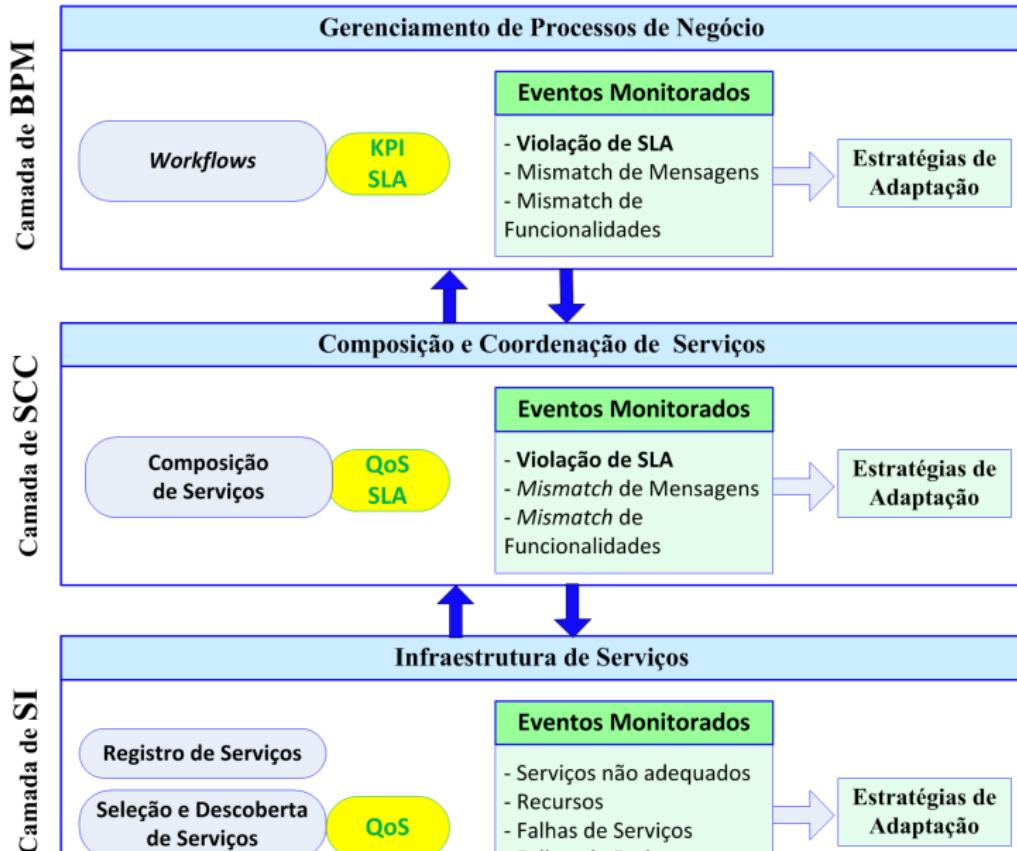
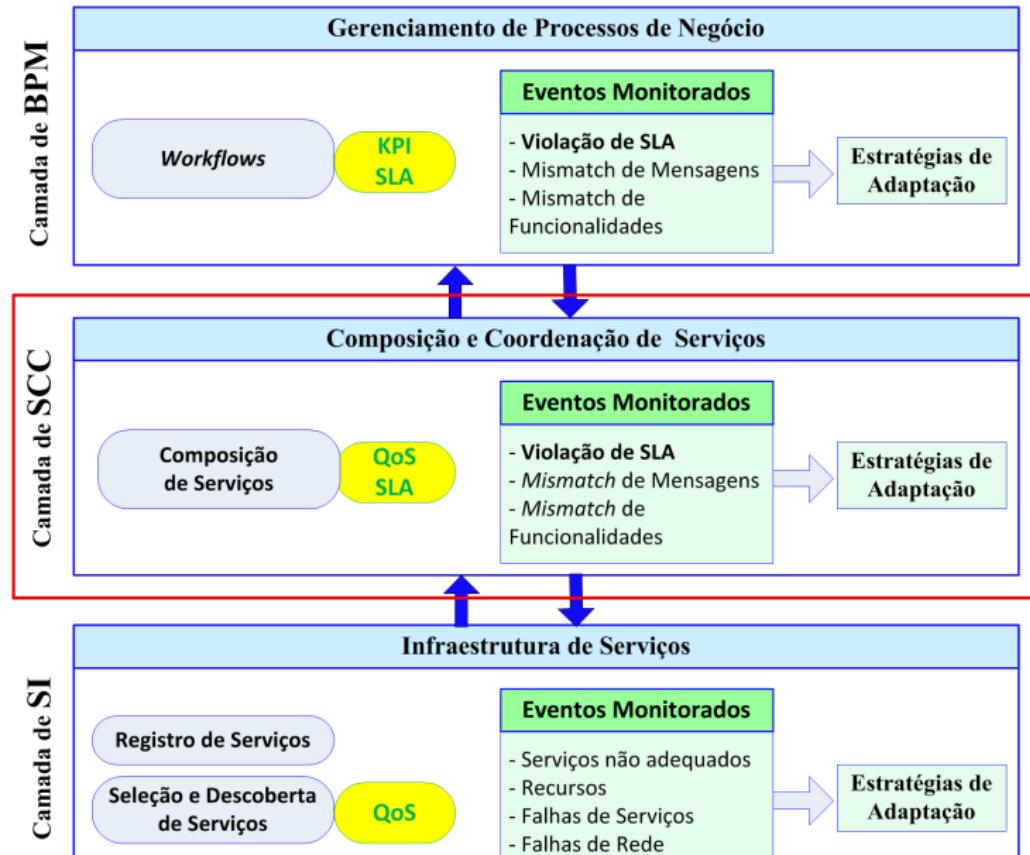


Figura: Camadas do Monitoramento

# Camadas no Monitoramento



# Camadas no Monitoramento



# QoS multi-camada em coreografias de serviços Web

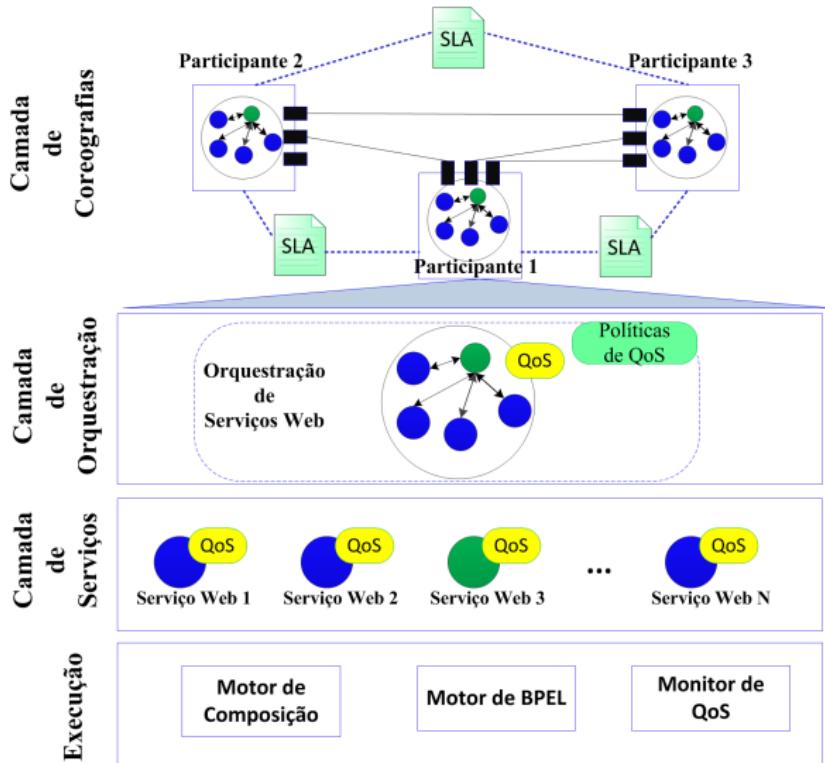


Figura: Integração de QoS e SLA multi-camada em coreografias

1 Conceitos Básicos

2 Problema

3 QoS e Monitoramento em Coreografias de Serviços Web

4 Trabalhos Relacionados

5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias

6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS
- Monitoramento

7 Conclusões e Trabalhos Futuros

# Monitoramento de Coreografias de Serviços Web Baseado em QoS

- (Xiangpeng et al., 2007), (Pandey and Chaudhary, 2008) e (Pandey, 2010) :
  - ▶ Métodos formais para especificar QoS em coreografias.
  - ▶ **Focado somente na linguagem.**

# Monitoramento de Coreografias de Serviços Web Baseado em QoS

- (Xiangpeng et al., 2007), (Pandey and Chaudhary, 2008) e (Pandey, 2010) :
  - ▶ Métodos formais para especificar QoS em coreografias.
  - ▶ **Focado somente na linguagem.**
- (Wetzstein et al., 2010):
  - ▶ Monitoramento de processos inter-organizacionais.
  - ▶ **Foco em KPIs e não em QoS.**

# Monitoramento de Coreografias de Serviços Web Baseado em QoS

- (Xiangpeng et al., 2007), (Pandey and Chaudhary, 2008) e (Pandey, 2010) :
  - ▶ Métodos formais para especificar QoS em coreografias.
  - ▶ **Focado somente na linguagem.**
- (Wetzstein et al., 2010):
  - ▶ Monitoramento de processos inter-organizacionais.
  - ▶ **Foco em KPIs e não em QoS.**
- (UI Haq et al., 2010) :
  - ▶ Agregações hierárquicas de SLAs em coreografias.
  - ▶ Framework baseado em regras.
  - ▶ **Foco em processos de negócio e KPIs.**

# Monitoramento de Coreografias de Serviços Web Baseado em QoS

- (Xiangpeng et al., 2007), (Pandey and Chaudhary, 2008) e (Pandey, 2010) :
  - ▶ Métodos formais para especificar QoS em coreografias.
  - ▶ **Focado somente na linguagem.**
- (Wetzstein et al., 2010):
  - ▶ Monitoramento de processos inter-organizacionais.
  - ▶ **Foco em KPIs e não em QoS.**
- (UI Haq et al., 2010) :
  - ▶ Agregações hierárquicas de SLAs em coreografias.
  - ▶ Framework baseado em regras.
  - ▶ **Foco em processos de negócio e KPIs.**
- (Xia et al., 2009) :
  - ▶ Predição analítica de QoS em coreografias em **WSCI**.
  - ▶ **Foco na linguagem.**

# Monitoramento de Coreografias de Serviços Web Baseado em QoS

- (Xiangpeng et al., 2007), (Pandey and Chaudhary, 2008) e (Pandey, 2010) :
  - ▶ Métodos formais para especificar QoS em coreografias.
  - ▶ **Focado somente na linguagem.**
- (Wetzstein et al., 2010) :
  - ▶ Monitoramento de processos inter-organizacionais.
  - ▶ **Foco em KPIs e não em QoS.**
- (UI Haq et al., 2010) :
  - ▶ Agregações hierárquicas de SLAs em coreografias.
  - ▶ Framework baseado em regras.
  - ▶ **Foco em processos de negócio e KPIs.**
- (Xia et al., 2009) :
  - ▶ Predição analítica de QoS em coreografias em **WSCI**.
  - ▶ **Foco na linguagem.**
- (Rosenberg, 2009) :
  - ▶ Modelo multi-camada de QoS para coreografias de serviços Web.
  - ▶ **Sem uma técnica para estabelecer SLAs.**
  - ▶ **Foco em restrições rígidas.**

# Monitoramento de Serviços Usando SLAs Probabilísticos

- (Rosario et al., 2008) e (Rosario et al., 2009)
  - ▶ Método de Monte-Carlo para definir SLAs em orquestrações.
  - ▶ **Foco no tempo de resposta.**
  - ▶ Monitoramento de SLAs probabilísticos em orquestrações de serviços usando Orc.

# Monitoramento de Serviços Usando SLAs Probabilísticos

- (Rosario et al., 2008) e (Rosario et al., 2009)
  - ▶ Método de Monte-Carlo para definir SLAs em orquestrações.
  - ▶ **Foco no tempo de resposta.**
  - ▶ **Monitoramento de SLAs probabilísticos em orquestrações de serviços usando Orc.**
- (Zheng et al., 2010):
  - ▶ A Função Densidade de Probabilidade (PDF) da QoS.
  - ▶ Demonstrações por meio de simulações.
  - ▶ **Foco no tempo de resposta e não trata SLAs.**

# Simulador e Infraestrutura de Coreografias

- WS-CDL+ (Z. Kang et al., 2007)
  - ▶ motor de execução de coreografias para WS-CDL.
  - ▶ É apenas um protótipo.

# Simulador e Infraestrutura de Coreografias

- WS-CDL+ (Z. Kang et al., 2007)
  - ▶ motor de execução de coreografias para WS-CDL.
  - ▶ É apenas um protótipo.
- Pi4SOA (X. Zhou et al. 2006):
  - ▶ Modelagem e verificação de coreografias.
  - ▶ Foco na modelagem e simulações para verificar inconsistências na especificação.

# Simulador e Infraestrutura de Coreografias

- WS-CDL+ (Z. Kang et al., 2007)
  - ▶ motor de execução de coreografias para WS-CDL.
  - ▶ É apenas um protótipo.
- Pi4SOA (X. Zhou et al. 2006):
  - ▶ Modelagem e verificação de coreografias.
  - ▶ Foco na modelagem e simulações para verificar inconsistências na especificação.
- Open Knowledge (A. Barker et al. 2009):
  - ▶ Rodar sistemas distribuídos em uma arquitetura P2P.
  - ▶ Roda coreografias de maneira limitada por conta do LCC.
- BPEL gold (L. Engler 2009):
  - ▶ Propõe o conceito de *Choreography-aware Service Bus* (CSB).
  - ▶ Propõe a BPELgold: uma linguagem específica da implementação.
  - ▶ A implementação é um protótipo.

1 Conceitos Básicos

2 Problema

3 QoS e Monitoramento em Coreografias de Serviços Web

4 Trabalhos Relacionados

## 5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias

6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS
- Monitoramento

7 Conclusões e Trabalhos Futuros

## 1 Conceitos Básicos

## 2 Problema

## 3 QoS e Monitoramento em Coreografias de Serviços Web

## 4 Trabalhos Relacionados

## 5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias

## 6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS
- Monitoramento

## 7 Conclusões e Trabalhos Futuros

# Contribuições

- Desenvolvimento de um simulador de coreografias com suporte de QoS.

# Contribuições

- Desenvolvimento de um simulador de coreografias com suporte de QoS.
- Mecanismos para definir requisitos de QoS em coreografias de serviços Web. Para tanto utilizaram-se duas abordagens:
  - ▶ **Analítica:** Usando Redes de Petri Estocásticas Generalizadas (GSPN) como representação intermediária da especificação de uma coreografia.
  - ▶ **Simulações:** Por meio de avaliações de desempenho e modelos de falha usando o simulador de coreografias.

# Contribuições

- Desenvolvimento de um simulador de coreografias com suporte de QoS.
- Mecanismos para definir requisitos de QoS em coreografias de serviços Web. Para tanto utilizaram-se duas abordagens:
  - ▶ **Analítica:** Usando Redes de Petri Estocásticas Generalizadas (GSPN) como representação intermediária da especificação de uma coreografia.
  - ▶ **Simulações:** Por meio de avaliações de desempenho e modelos de falha usando o simulador de coreografias.
- Definição de restrições probabilísticas de QoS entre os serviços de uma coreografia de serviços Web.

# Contribuições

- Desenvolvimento de um simulador de coreografias com suporte de QoS.
- Mecanismos para definir requisitos de QoS em coreografias de serviços Web. Para tanto utilizaram-se duas abordagens:
  - ▶ **Analítica:** Usando Redes de Petri Estocásticas Generalizadas (GSPN) como representação intermediária da especificação de uma coreografia.
  - ▶ **Simulações:** Por meio de avaliações de desempenho e modelos de falha usando o simulador de coreografias.
- Definição de restrições probabilísticas de QoS entre os serviços de uma coreografia de serviços Web.
- Proposta de um mecanismo de monitoramento de coreografias para detectar violações de SLA com restrições de QoS probabilísticas.

# Contribuições

- Desenvolvimento de um simulador de coreografias com suporte de QoS.
- Mecanismos para definir requisitos de QoS em coreografias de serviços Web. Para tanto utilizaram-se duas abordagens:
  - ▶ **Analítica:** Usando Redes de Petri Estocásticas Generalizadas (GSPN) como representação intermediária da especificação de uma coreografia.
  - ▶ **Simulações:** Por meio de avaliações de desempenho e modelos de falha usando o simulador de coreografias.
- Definição de restrições probabilísticas de QoS entre os serviços de uma coreografia de serviços Web.
- Proposta de um mecanismo de monitoramento de coreografias para detectar violações de SLA com restrições de QoS probabilísticas.
- **Atributos de QoS:** tempo de resposta, tempo de processamento, largura de banda e latência de rede.

# Etapas

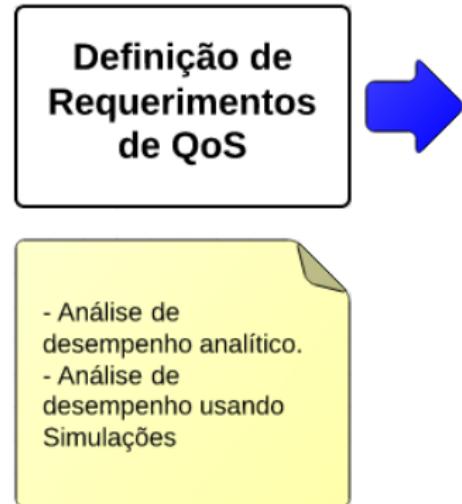


Figura: Etapas para atingir o monitoramento de coreografias

# Etapas

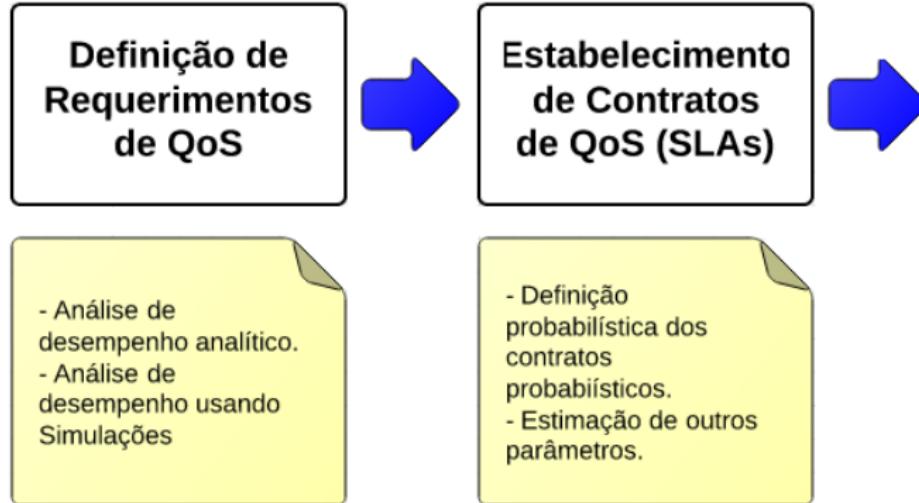


Figura: Etapas para atingir o monitoramento de coreografias

# Etapas



Figura: Etapas para atingir o monitoramento de coreografias

# Elementos BPMN 2 Suportados

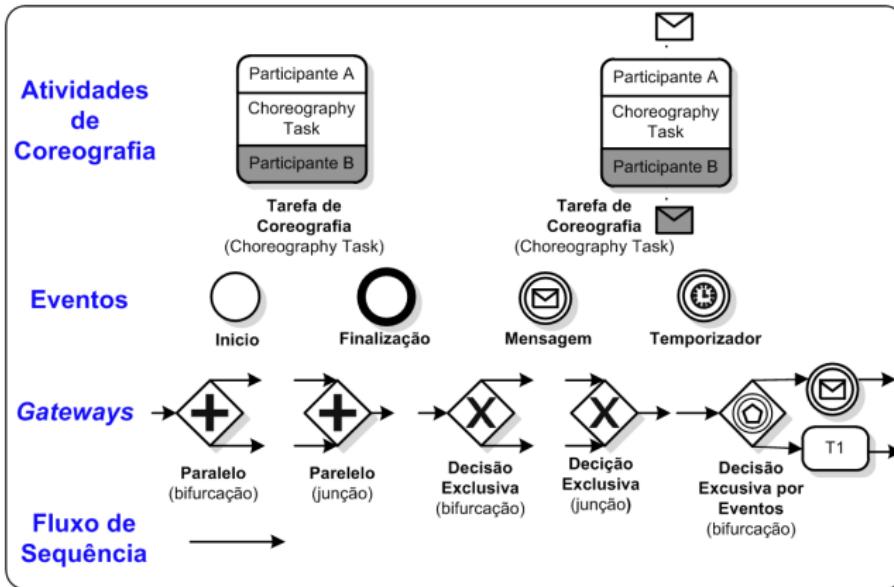
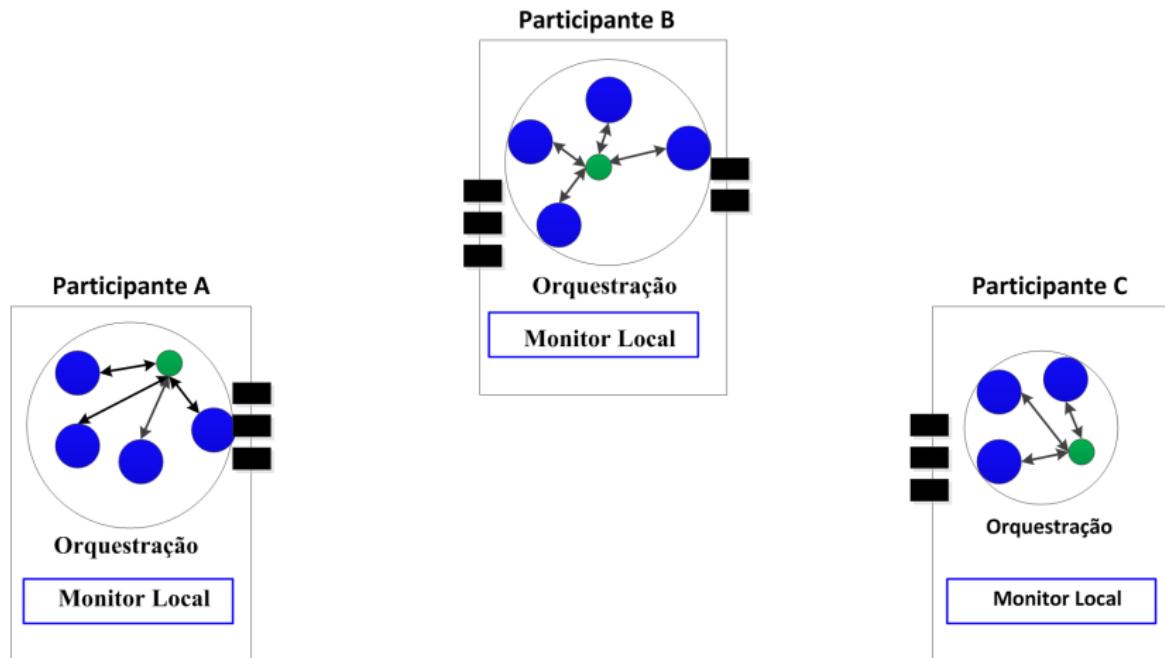


Figura: Elementos BPMN suportados.

# Arquitetura do monitoramento para detectar violações de SLA



**Figura:** Perspectiva Geral do monitoramento proposto

# Arquitetura do monitoramento para detectar violações de SLA

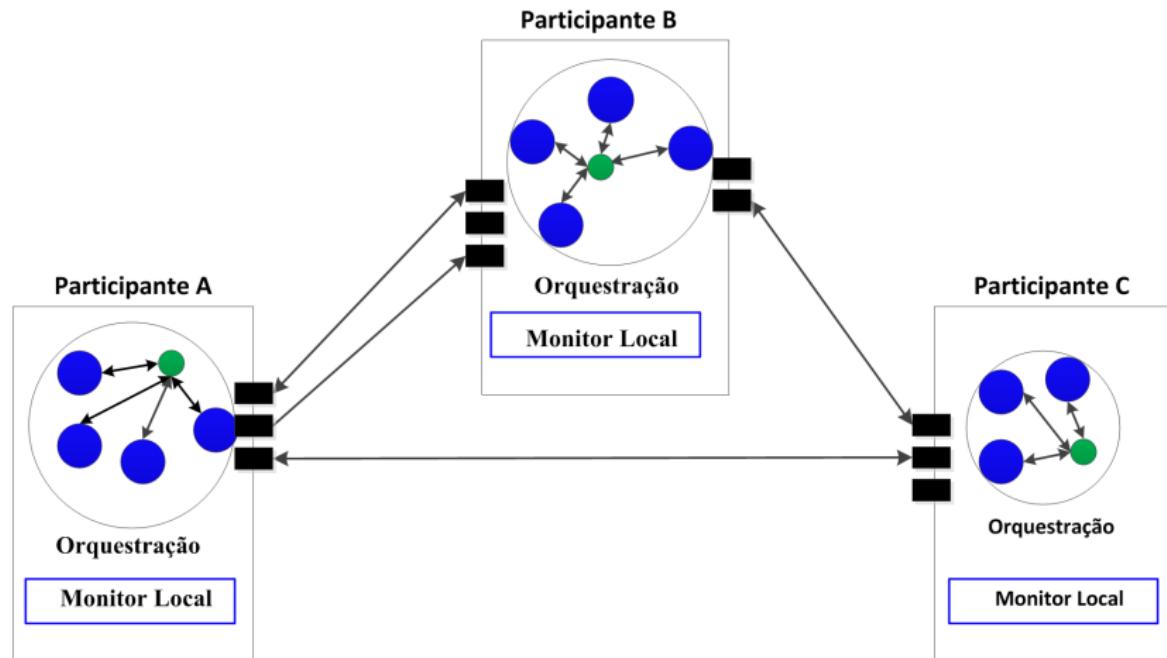
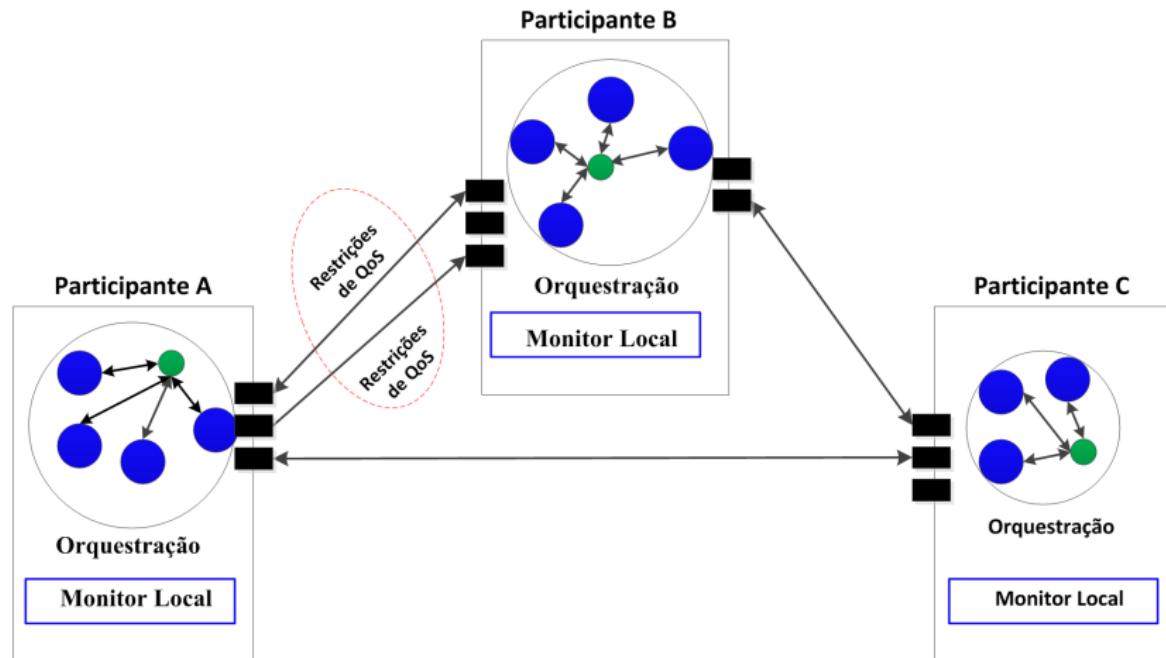


Figura: Perspectiva Geral do monitoramento proposto

# Arquitetura do monitoramento para detectar violações de SLA



**Figura:** Perspectiva Geral do monitoramento proposto

# Arquitetura do monitoramento para detectar violações de SLA

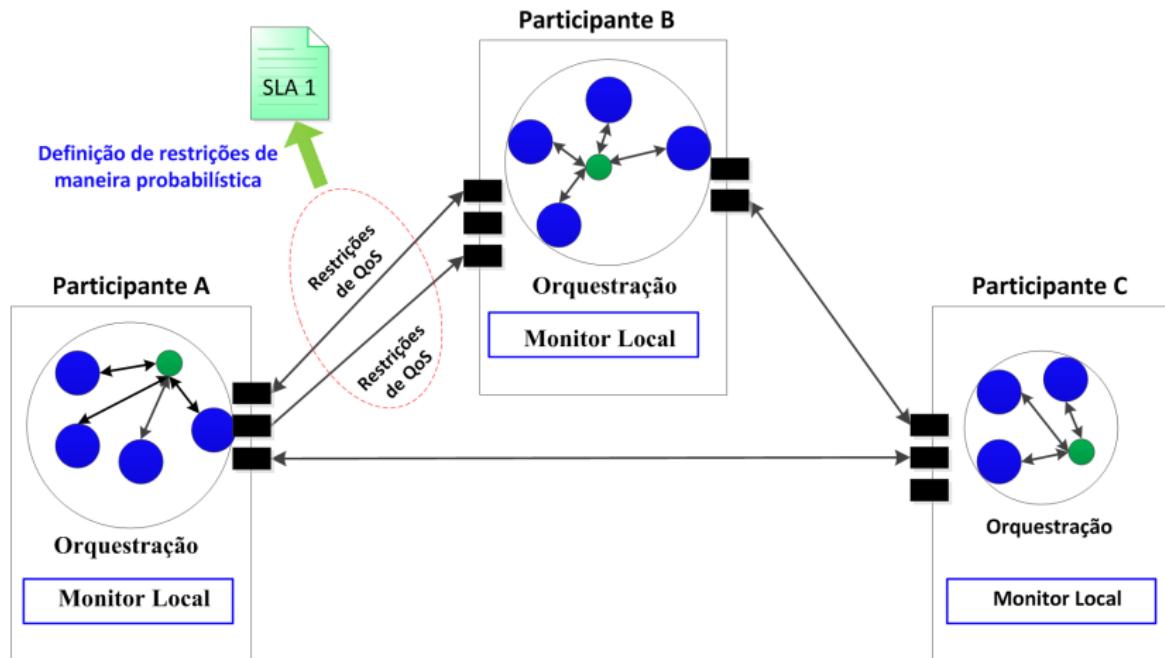


Figura: Perspectiva Geral do monitoramento proposto

# Arquitetura do monitoramento para detectar violações de SLA

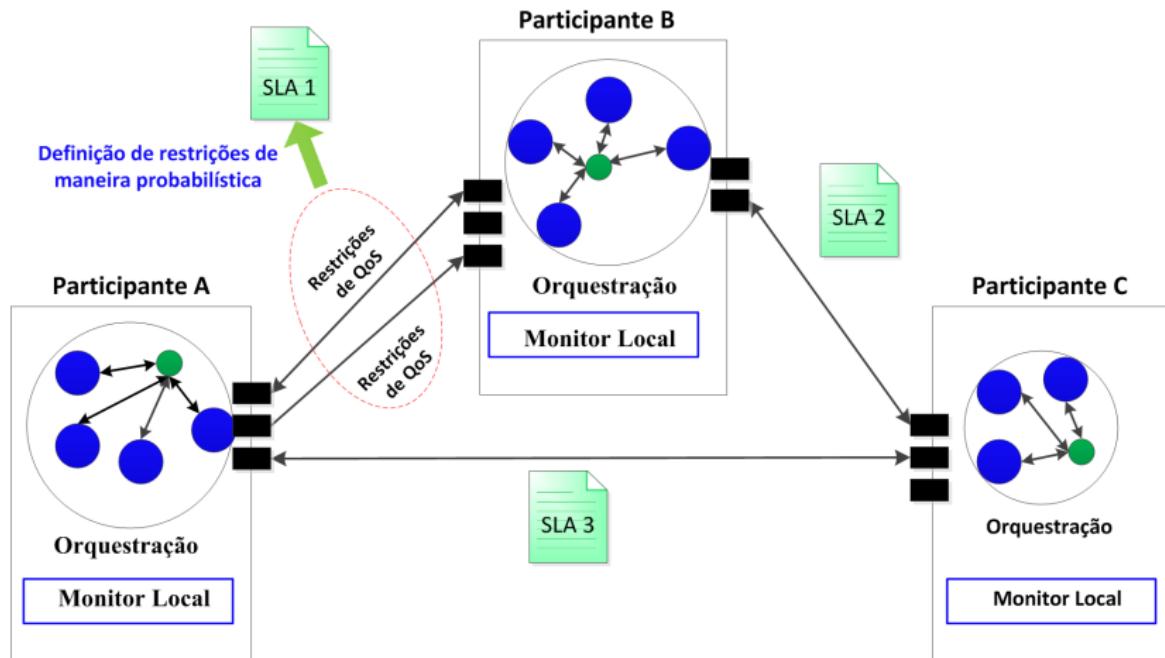
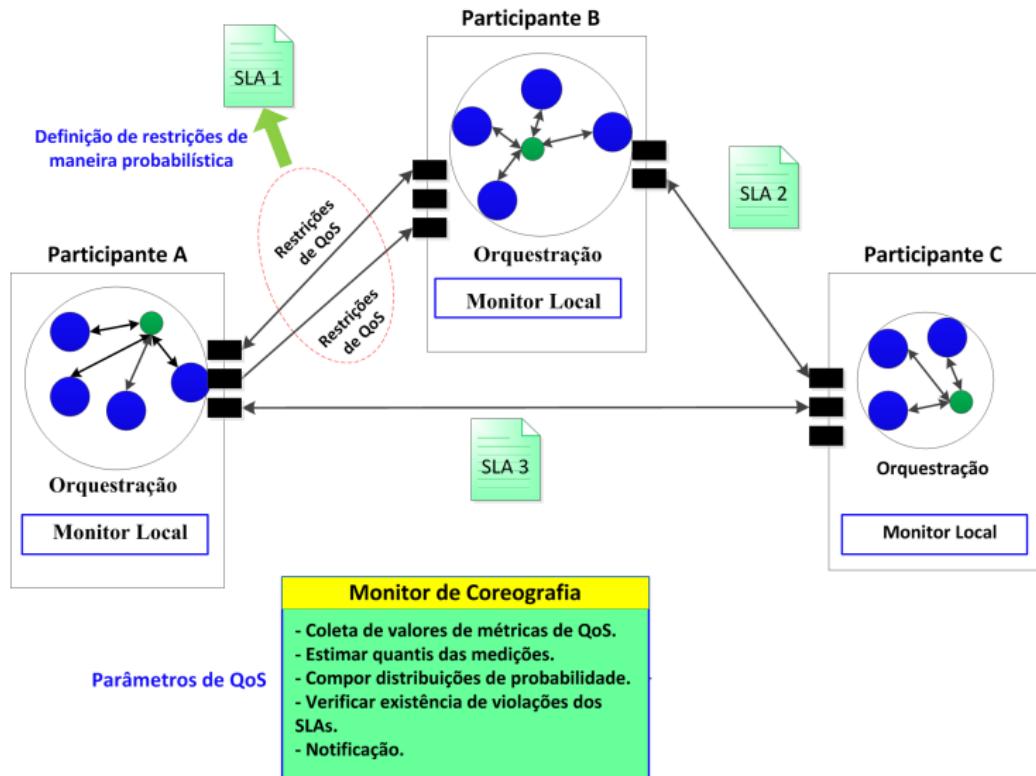
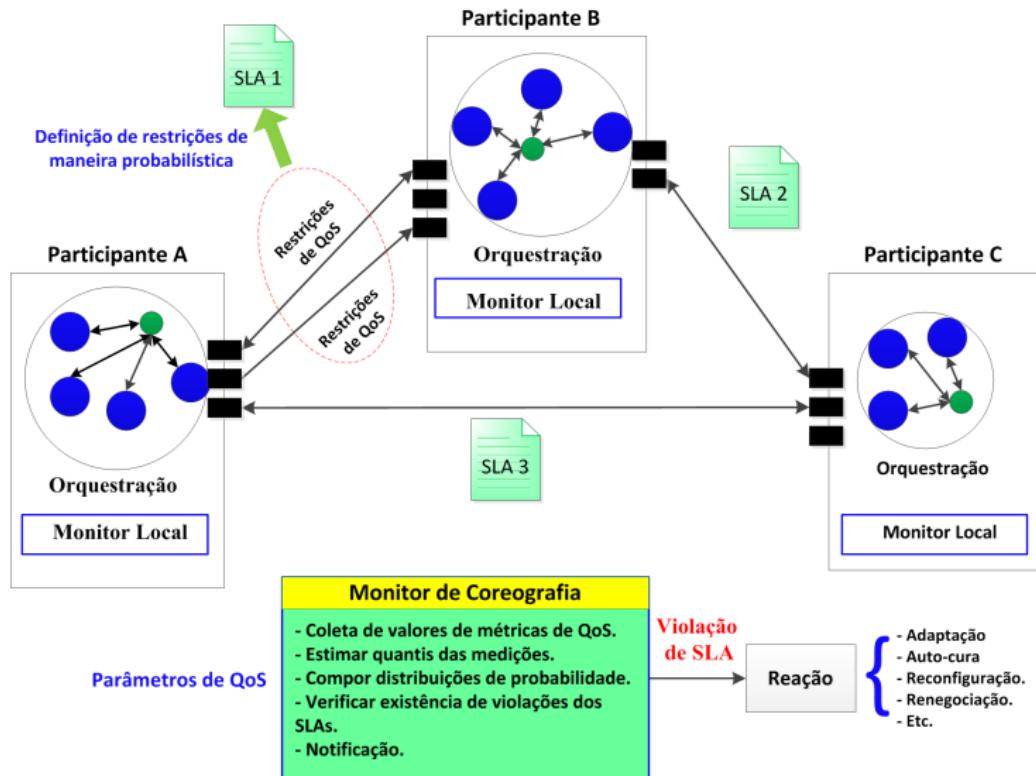


Figura: Perspectiva Geral do monitoramento proposto

# Arquitetura do monitoramento para detectar violações de SLA



# Arquitetura do monitoramento para detectar violações de SLA



# Modelo de QoS (I)

- Atributos de QoS:

# Modelo de QoS (I)

- Atributos de QoS:
  - ▶ Serviço : **tempo de processamento, tempo de resposta, .**

# Modelo de QoS (I)

- Atributos de QoS:
  - ▶ Serviço : **tempo de processamento, tempo de resposta, .**
  - ▶ Rede : largura de banda, atraso e **erros de comunicação.**

# Modelo de QoS (I)

- Atributos de QoS:
  - ▶ Serviço : **tempo de processamento, tempo de resposta, .**
  - ▶ Rede : largura de banda, atraso e **erros de comunicação.**
  - ▶ Mensagem : **integridade da mensagem.**

# Modelo de QoS (II)

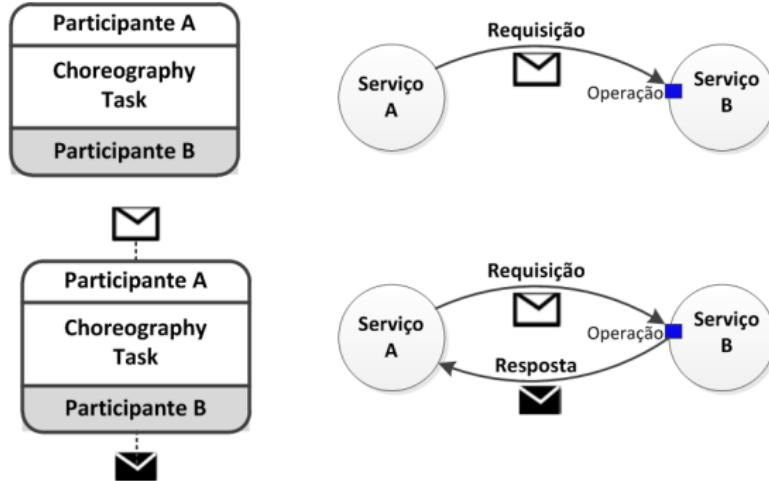


Figura: Interação de serviços a partir de interações atômicas do BPMN2.

# Modelo de QoS (II)

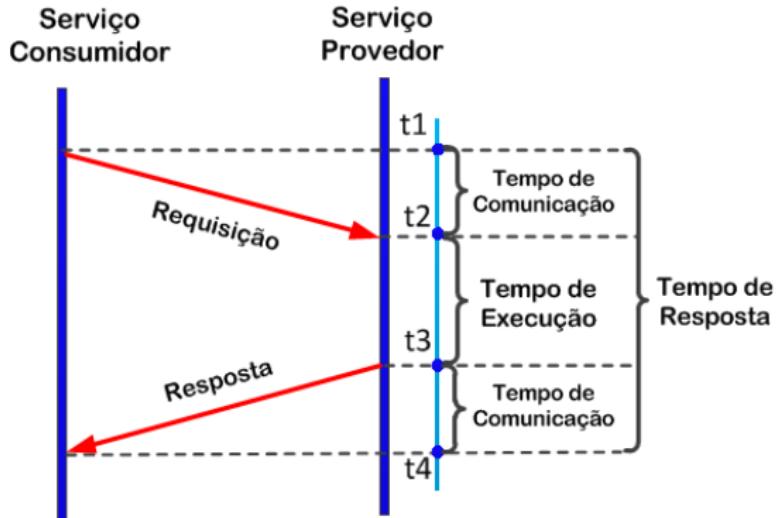


Figura: Atributos de QoS em uma interação com um serviço Web.

## 1 Conceitos Básicos

## 2 Problema

## 3 QoS e Monitoramento em Coreografias de Serviços Web

## 4 Trabalhos Relacionados

## 5 Proposta

- Visão Geral
- **Definição de requisitos de QoS**
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias

## 6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS
- Monitoramento

## 7 Conclusões e Trabalhos Futuros

# Definição de requisitos de QoS: Analiticamente

- ① **Mapeamento** de uma coreografia para uma GSPN ( Rede Petri Estocástica Generalizada).

# Definição de requisitos de QoS: Analiticamente

- ① **Mapeamento** de uma coreografia para uma GSPN ( Rede Petri Estocástica Generalizada).
- ② **Configurações** na GSPN obtida.

# Definição de requisitos de QoS: Analiticamente

- ① **Mapeamento** de uma coreografia para uma GSPN ( Rede Petri Estocástica Generalizada).
- ② **Configurações** na GSPN obtida.
- ③ **Simulações** dos cenários.

# Mapeamento de BPMN para GSPN (I)

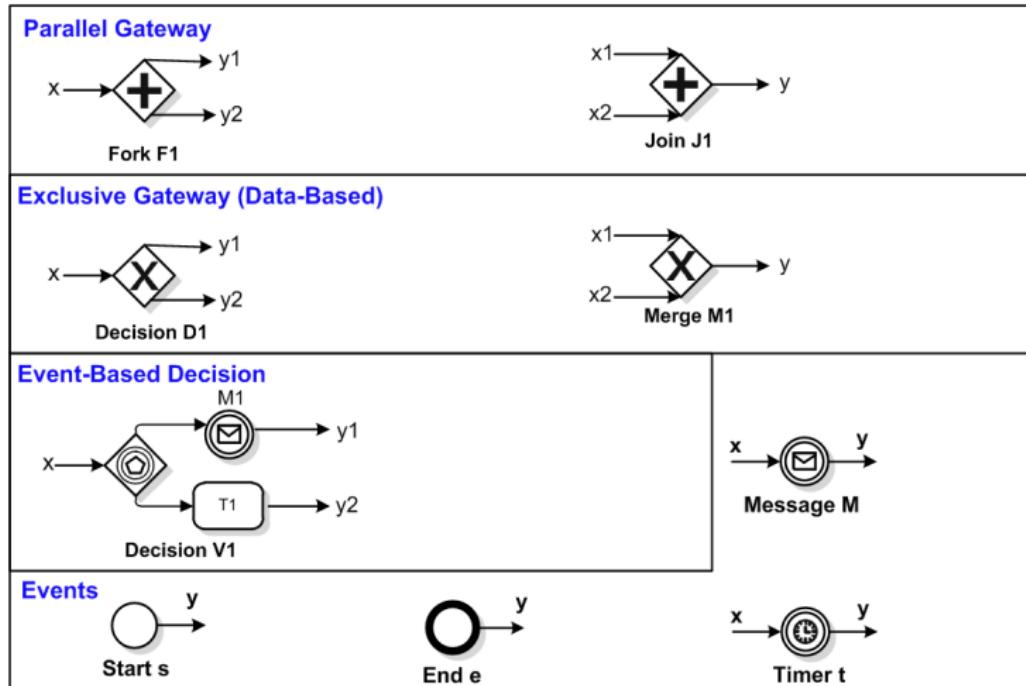


Figura: Mapeamento de eventos e gateways para módulos GSPN.

# Mapeamento de BPMN para GSPN (I)

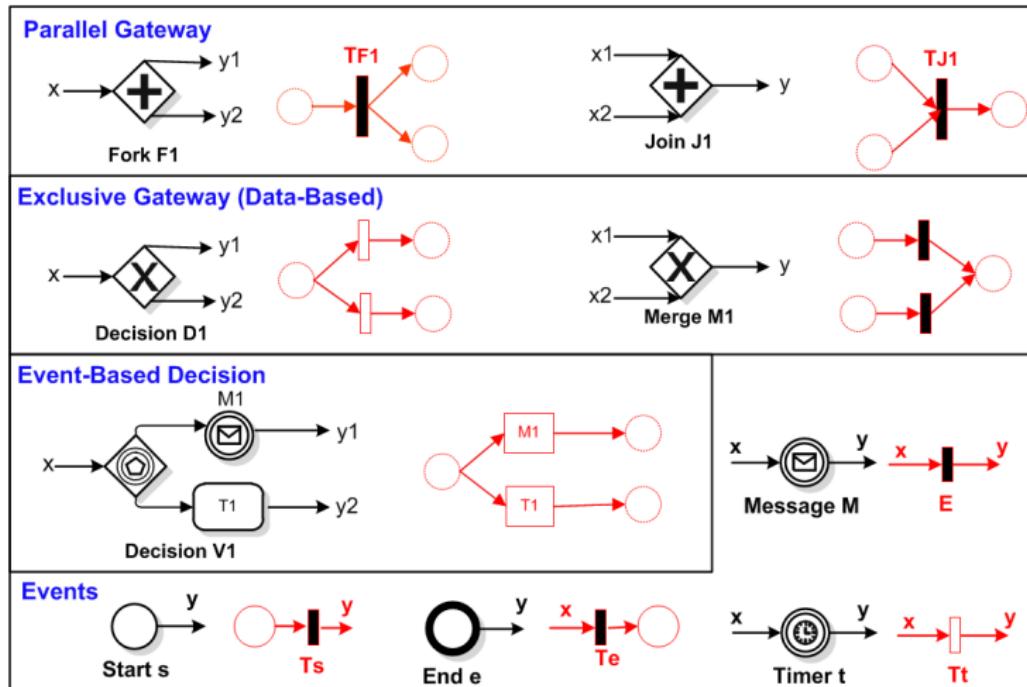
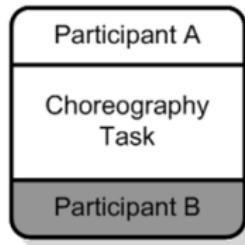


Figura: Mapeamento de eventos e gateways para módulos GSPN.

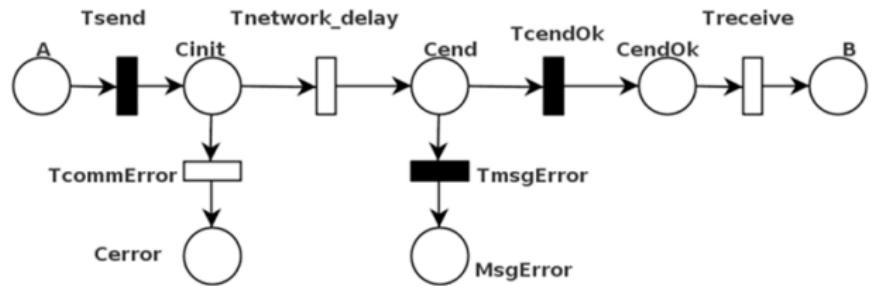
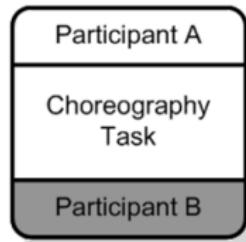
# Mapeamento de BPMN para GSPN (II)



**A) Interação em BPMN2**

**B) Mapeamento para GSPN com QoS**

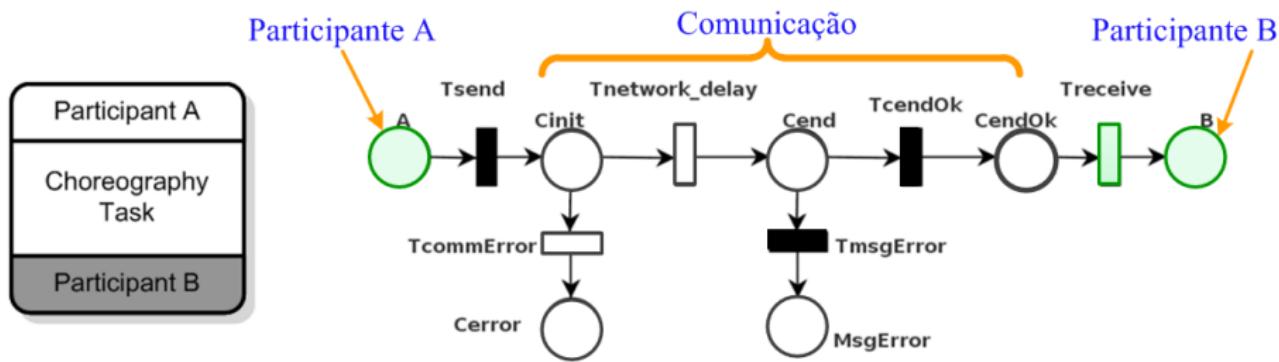
# Mapeamento de BPMN para GSPN (II)



A) Interação em BPMN2

B) Mapeamento para GSPN com QoS

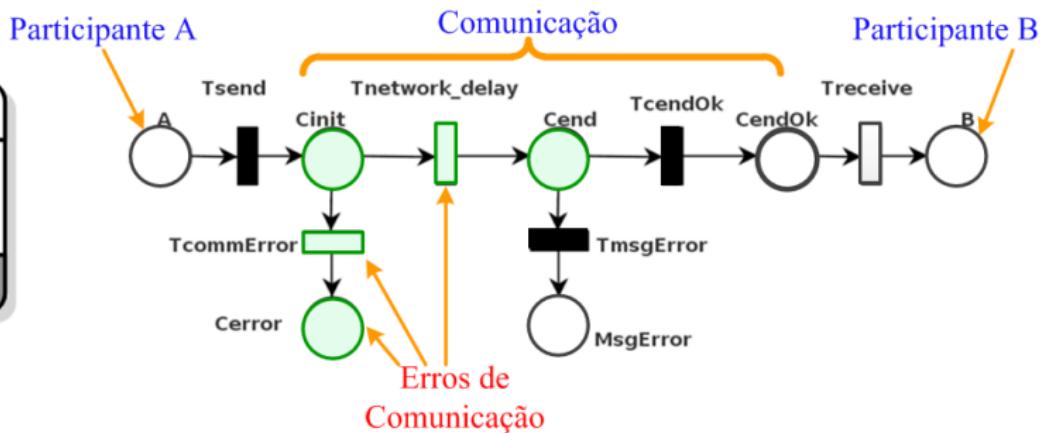
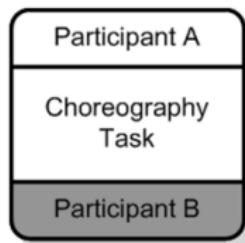
# Mapeamento de BPMN para GSPN (II)



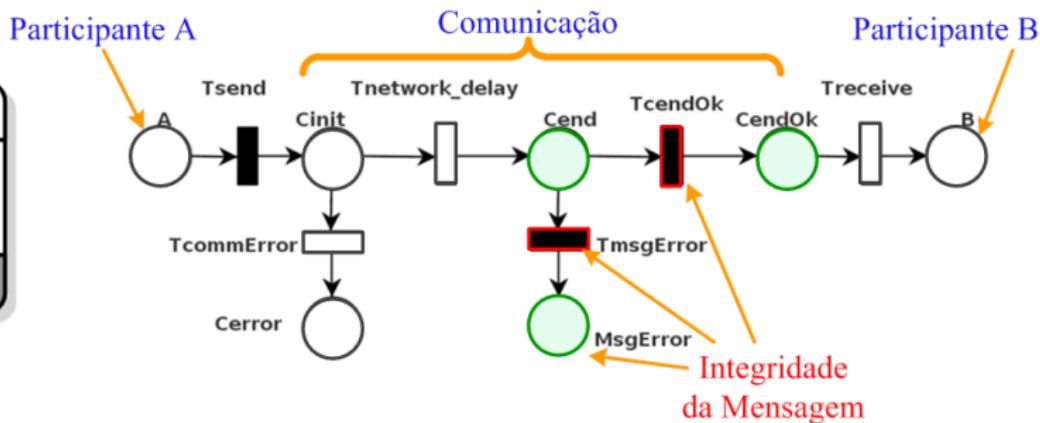
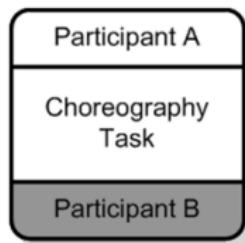
A) Interação em BPMN2

B) Mapeamento para GSPN com QoS

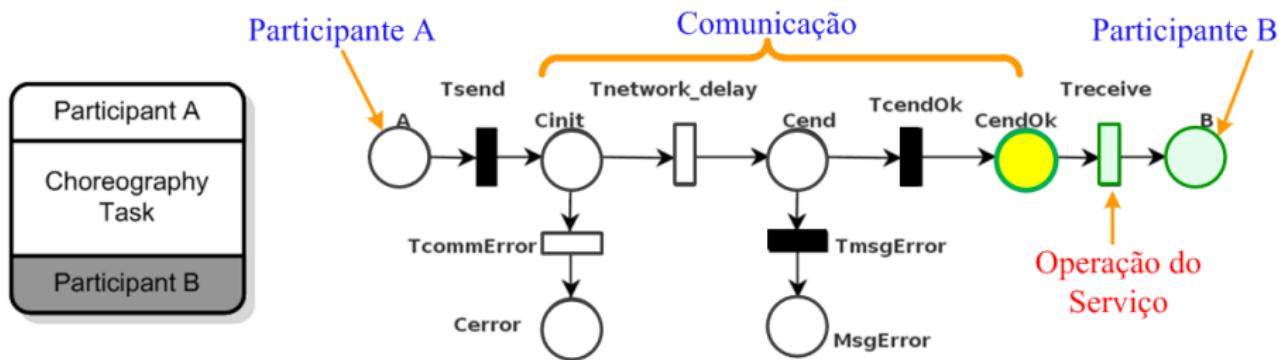
# Mapeamento de BPMN para GSPN (II)



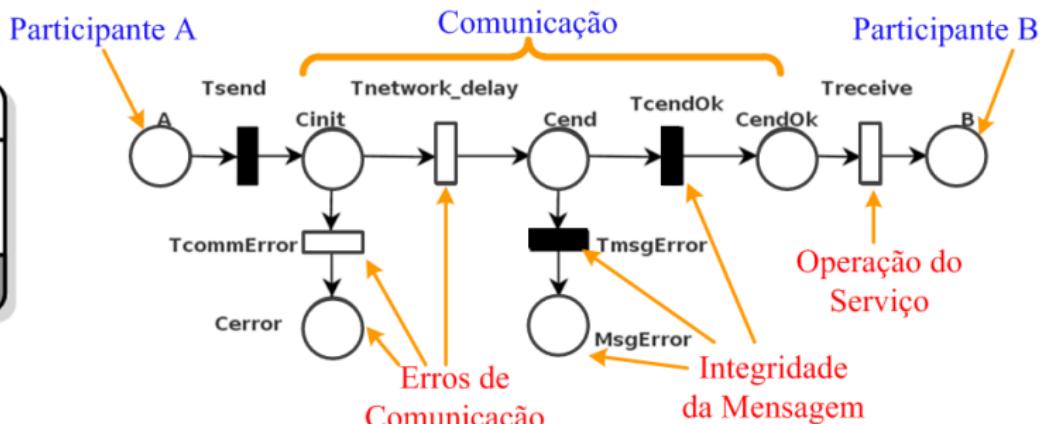
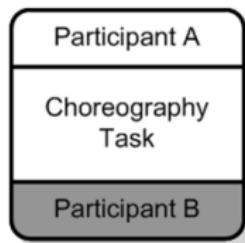
# Mapeamento de BPMN para GSPN (II)



# Mapeamento de BPMN para GSPN (II)



# Mapeamento de BPMN para GSPN (II)



# Algoritmo de Mapeamento (I)

---

## Algorithm 1 \*

---

Mapeamento de uma coreografia em BPMN 2.0 para uma GSPN com suporte de QoS

**Input:** Process Choreography  $PC = (\mathcal{O}, \mathcal{A}, \mathcal{E}, \mathcal{G}, \mathcal{T}, \{e^S\}, \mathcal{E}^I, \{e^E\}, \mathcal{E}^{IM}, \mathcal{E}^{IT}, \mathcal{G}^F, \mathcal{G}^J, \mathcal{G}^X, \mathcal{G}^M, \mathcal{G}^V, \mathcal{F})$  in BPMN 2.0.

# Algoritmo de Mapeamento (I)

---

## Algorithm 2 \*

---

Mapeamento de uma coreografia em BPMN 2.0 para uma GSPN com suporte de QoS

**Input:** Process Choreography  $PC = (\mathcal{O}, \mathcal{A}, \mathcal{E}, \mathcal{G}, \mathcal{T}, \{e^S\}, \mathcal{E}^I, \{e^E\}, \mathcal{E}^{IM}, \mathcal{E}^{IT}, \mathcal{G}^F, \mathcal{G}^J, \mathcal{G}^X, \mathcal{G}^M, \mathcal{G}^V, \mathcal{F})$  in BPMN 2.0.

**Output:** Generalized Stochastic Petri Net  $GSPN_{QoS}$ .

# Algoritmo de Mapeamento (I)

## Algorithm 3 \*

Mapeamento de uma coreografia em BPMN 2.0 para uma GSPN com suporte de QoS

**Input:** Process Choreography  $PC = (\mathcal{O}, \mathcal{A}, \mathcal{E}, \mathcal{G}, \mathcal{T}, \{e^S\}, \mathcal{E}^I, \{e^E\}, \mathcal{E}^{IM}, \mathcal{E}^{IT}, \mathcal{G}^F, \mathcal{G}^J, \mathcal{G}^X, \mathcal{G}^M, \mathcal{G}^V, \mathcal{F})$  in BPMN 2.0.  
**Output:** Generalized Stochastic Petri Net  $GSPN_{QoS}$ .

$CT_i \in \mathcal{T}$ ,  $G_j \in \mathcal{G}$  and  $E_k \in \mathcal{E}$ . where  $i, j, k \in \mathbb{N}$ .

$PNQoS(CT_i)$ ,  $PNQoS(G_j)$ ,  $PNQoS(E_k)$  are functions return a GSPN according to mapping rules.

$\oplus$  as the operator composition that returns other GSPN.

```
 $GSPN_{QoS} \leftarrow$  Empty Petri Net
For  $CT_i \in \mathcal{T}$  Do
     $GSPN_{QoS} \leftarrow GSPN_{QoS} \oplus PNQoS(CT_i)$ 
    Add a arrival timed Transition at beginning of the  $GSPN_{QoS}$ .
End
For  $G_j \in \mathcal{G}$  Do
     $GSPN_{QoS} \leftarrow GSPN_{QoS} \oplus PN(G_j)$ 
End
For  $E_k \in \mathcal{E}$  Do
     $GSPN_{QoS} \leftarrow GSPN_{QoS} \oplus PN(E_k)$ 
End
Add a starting Place and immediate Transition at the beginning of the  $GSPN_{QoS}$ .
Add a ending Place and immediate Transition at the end of the  $GSPN_{QoS}$ .
```

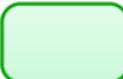
# Algoritmo de Mapeamento (II)

## 1) Regras de Mapeamento

Elemento BPMN	Módulo GSPN
Atividade de Coreografia	
Eventos	
<i>Gateways</i>	

# Algoritmo de Mapeamento (II)

## 1) Regras de Mapeamento

Elemento BPMN	Módulo GSPN
Atividade de Coreografia	
Eventos	
Gateways	

## 2) Composição

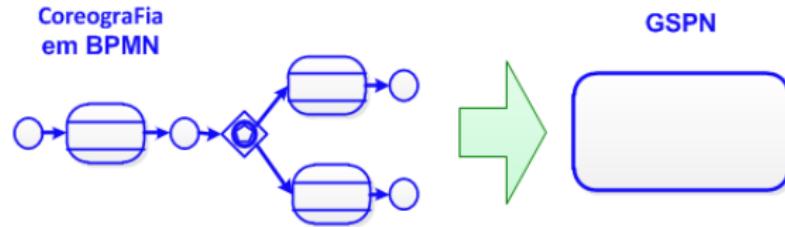


Operador de Composição



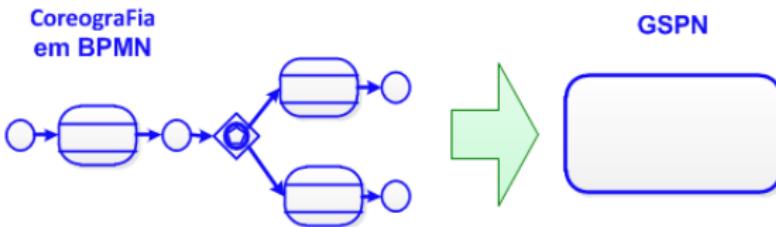
# Algoritmo de Mapeamento (II)

## 3) Substituir e compor

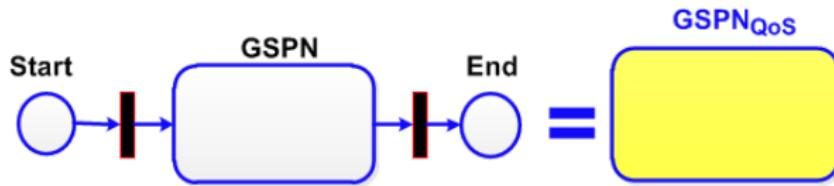


# Algoritmo de Mapeamento (II)

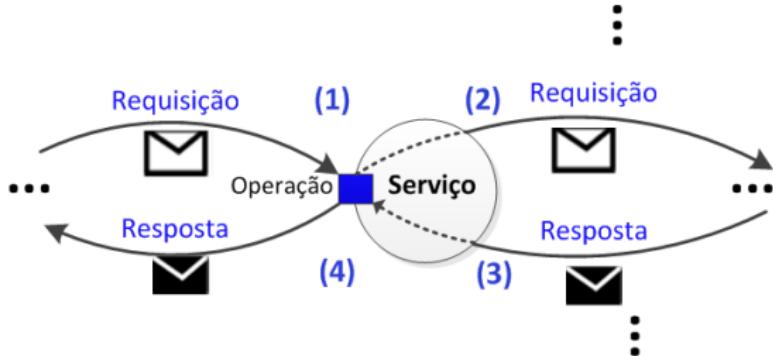
## 3) Substituir e compor



## 4) Reduzir e adicionar elementos finais

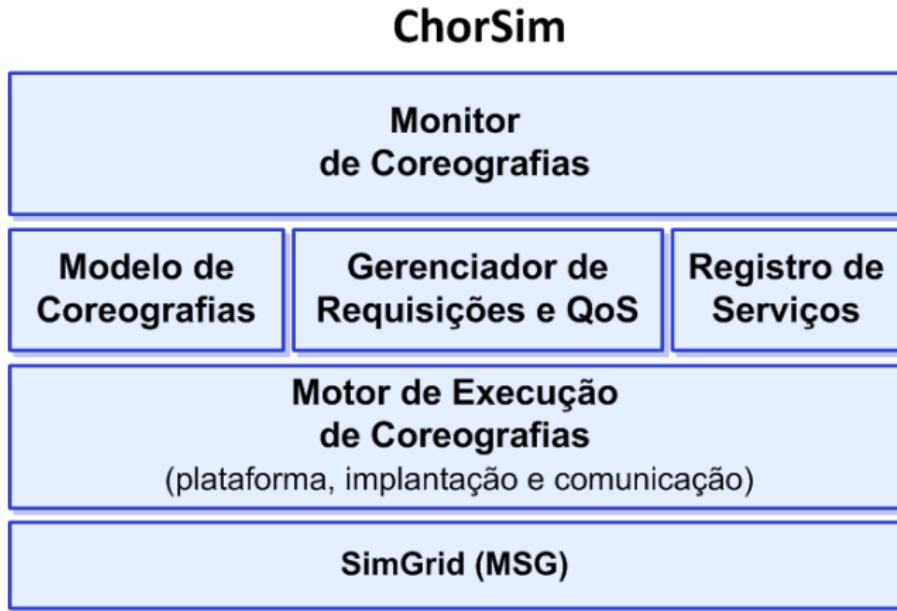


# ChorSim: Simulador de Coreografias



**Figura:** Atributos de QoS calculados em um evento dado. (1) Recebendo requisições de um cliente ou serviço. (2) enviando requisições para um outro serviço. (3) recebendo resposta de um outro serviço (dependência). (4) enviando resposta para um cliente ou serviço solicitador.

# ChorSim: Simulador de Coreografias



**Figura:** Arquitetura do simulador de coreografias.

## 1 Conceitos Básicos

## 2 Problema

## 3 QoS e Monitoramento em Coreografias de Serviços Web

## 4 Trabalhos Relacionados

## 5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico**
- Monitoramento de coreografias

## 6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS
- Monitoramento

## 7 Conclusões e Trabalhos Futuros

# Estabelecimento de SLAs Probabilísticos

$$F_S(x) = P(\delta_S \leq x)$$

é a função distribuição acumulada (fda) de um parâmetro de QoS  $\delta$  do serviço  $S$ .

$$F_S(x) = P(\delta_S \leq x)$$

é a função distribuição acumulada (fda) de um parâmetro de QoS  $\delta$  do serviço  $S$ .

## ① Condições Iniciais:

- ▶ Configurar a plataforma e a implantação no ChorSim.
- ▶ Estimar a função de distribuição acumulada  $F_{S_i}$  para cada serviço  $S_i$ .

# Estabelecimento de SLAs Probabilísticos

$$F_S(x) = P(\delta_S \leq x)$$

é a função distribuição acumulada (fda) de um parâmetro de QoS  $\delta$  do serviço  $S$ .

## ① Condições Iniciais:

- ▶ Configurar a plataforma e a implantação no ChorSim.
- ▶ Estimar a função de distribuição acumulada  $F_{S_i}$  para cada serviço  $S_i$ .

## ② Simulação usando ChorSim:

- ① Para cada invocação de um serviço  $s_i$  na interação de um participante provedor A com um outro participante cliente B, um valor do parâmetro de QoS  $q$  é obtido a partir da simulação em ChorSim.

# Estabelecimento de SLAs Probabilísticos

$$F_S(x) = P(\delta_S \leq x)$$

é a função distribuição acumulada (fda) de um parâmetro de QoS  $\delta$  do serviço  $S$ .

## ① Condições Iniciais:

- ▶ Configurar a plataforma e a implantação no ChorSim.
- ▶ Estimar a função de distribuição acumulada  $F_{S_i}$  para cada serviço  $S_i$ .

## ② Simulação usando ChorSim:

- ① Para cada invocação de um serviço  $s_i$  na interação de um participante provedor A com um outro participante cliente B, um valor do parâmetro de QoS  $q$  é obtido a partir da simulação em ChorSim.
- ② **Agregação:** Estimar o valor do parâmetro de QoS do serviço composto  $S$  a partir dos valores obtidos no passo anterior usando o ChorSim.

# Estabelecimento de SLAs Probabilísticos

$$F_S(x) = P(\delta_S \leq x)$$

é a função distribuição acumulada (fda) de um parâmetro de QoS  $\delta$  do serviço  $S$ .

## ① Condições Iniciais:

- ▶ Configurar a plataforma e a implantação no ChorSim.
- ▶ Estimar a função de distribuição acumulada  $F_{S_i}$  para cada serviço  $S_i$ .

## ② Simulação usando ChorSim:

- ① Para cada invocação de um serviço  $s_i$  na interação de um participante provedor A com um outro participante cliente B, um valor do parâmetro de QoS  $q$  é obtido a partir da simulação em ChorSim.
- ② **Agregação:** Estimar o valor do parâmetro de QoS do serviço composto  $S$  a partir dos valores obtidos no passo anterior usando o ChorSim.
- ③ Rodar as simulações dos passos 2.1 e 2.2 várias vezes, o suficiente para estimar empiricamente  $F_S$ .

# Estabelecimento de SLAs Probabilísticos

$$F_S(x) = P(\delta_S \leq x)$$

é a função distribuição acumulada (fda) de um parâmetro de QoS  $\delta$  do serviço  $S$ .

## ① Condições Iniciais:

- ▶ Configurar a plataforma e a implantação no ChorSim.
- ▶ Estimar a função de distribuição acumulada  $F_{S_i}$  para cada serviço  $S_i$ .

## ② Simulação usando ChorSim:

- ① Para cada invocação de um serviço  $s_i$  na interação de um participante provedor A com um outro participante cliente B, um valor do parâmetro de QoS  $q$  é obtido a partir da simulação em ChorSim.
  - ② **Agregação:** Estimar o valor do parâmetro de QoS do serviço composto  $S$  a partir dos valores obtidos no passo anterior usando o ChorSim.
  - ③ Rodar as simulações dos passos 2.1 e 2.2 várias vezes, o suficiente para estimar empiricamente  $F_S$ .
- ④ A partir de  $F_S$  se selecionam quantis para definir o contrato.

1 Conceitos Básicos

2 Problema

3 QoS e Monitoramento em Coreografias de Serviços Web

4 Trabalhos Relacionados

5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias**

6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS
- Monitoramento

7 Conclusões e Trabalhos Futuros

# Monitoramento

- $F_S$ : Distribuição de probabilidade do contrato.
- $\Delta$ : Um conjunto finito de amostras dos valores de um parâmetro de QoS do serviço  $S$ .
- $F'_S$ : Distribuição de Probabilidade após a agregação usando ChorSim.
- $\lambda$ : Zona de Tolerância.

$$F'_{S,\Delta}(x) = \frac{|\{\delta, \delta \in \Delta \leq x\}|}{|\Delta|} \quad (1)$$

---

<sup>1</sup>Rosario et al., 2009

- $F_S$ : Distribuição de probabilidade do contrato.
- $\Delta$ : Um conjunto finito de amostras dos valores de um parâmetro de QoS do serviço  $S$ .
- $F'_S$ : Distribuição de Probabilidade após a agregação usando ChorSim.
- $\lambda$ : Zona de Tolerância.

$$F'_{S,\Delta}(x) = \frac{|\{\delta, \delta \in \Delta \leq x\}|}{|\Delta|} \quad (1)$$

$$\exists x \in R^+ : F'_{S,\Delta}(x) < F_S(x) \quad (2)$$

---

<sup>1</sup>Rosario et al., 2009

# Monitoramento

- $F_S$ : Distribuição de probabilidade do contrato.
- $\Delta$ : Um conjunto finito de amostras dos valores de um parâmetro de QoS do serviço  $S$ .
- $F'_S$ : Distribuição de Probabilidade após a agregação usando ChorSim.
- $\lambda$ : Zona de Tolerância.

$$F'_{S,\Delta}(x) = \frac{|\{\delta, \delta \in \Delta \leq x\}|}{|\Delta|} \quad (1)$$

$$\exists x \in R^+ : F'_{s,\Delta}(x) < F_s(x) \quad (2)$$

$$\sup_{x \in R^+} (F'_{s,\Delta}(x) - F_s(x)) \geq \lambda^1 \quad (3)$$

---

<sup>1</sup>Rosario et al., 2009

# Detecção de Violações de SLA (I)

**Problema:** Dominância estocástica

$$H_0 : \forall x, F_S(x) \geq F'_S(x)$$

*contra*

$$H_1 : \exists x, F_S(x) < F'_S(x)$$

# Detecção de Violações de SLA (I)

**Problema:** Dominância estocástica

$$H_0 : \forall x, F_S(x) \geq F'_S(x)$$

*contra*

$$H_1 : \exists x, F_S(x) < F'_S(x)$$

**Solução:** *One-sided two-sample Kolmogorov-Smirnov test*

$$[D, p] = kstest(X_{contract}, X_{monitoring}, KS_{side}) \quad (4)$$

## Detecção de Violações de SLA (II)

Então, para a detecção de violações usam-se:

$$[D^+, p^+] = kstest(X_{contract}, X_{monitoring}, greater)$$

$$[D^-, p^-] = kstest(X_{contract}, X_{monitoring}, less)$$

## Detecção de Violações de SLA (II)

Então, para a detecção de violações usam-se:

$$[D^+, p^+] = kstest(X_{contract}, X_{monitoring}, greater)$$

$$[D^-, p^-] = kstest(X_{contract}, X_{monitoring}, less)$$

Portanto, para que um conjunto de amostras  $X_{monitoring}$  cumpra o contrato uma regra baseada em  $p^+$  e  $D$  deve ser definida:

## Detecção de Violações de SLA (II)

Então, para a detecção de violações usam-se:

$$[D^+, p^+] = kstest(X_{contract}, X_{monitoring}, greater)$$

$$[D^-, p^-] = kstest(X_{contract}, X_{monitoring}, less)$$

Portanto, para que um conjunto de amostras  $X_{monitoring}$  cumpra o contrato uma regra baseada em  $p^+$  e  $D$  deve ser definida:

$$\text{verify}(X_{monitoring}) = \begin{cases} \text{true,} & \text{se } p^+ \geq \alpha \wedge D^+ < \lambda \\ \text{false,} & \text{de outra maneira} \end{cases}$$

## 1 Conceitos Básicos

## 2 Problema

## 3 QoS e Monitoramento em Coreografias de Serviços Web

## 4 Trabalhos Relacionados

## 5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias

## 6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS
- Monitoramento

## 7 Conclusões e Trabalhos Futuros

- 1 Conceitos Básicos
- 2 Problema
- 3 QoS e Monitoramento em Coreografias de Serviços Web
- 4 Trabalhos Relacionados
- 5 Proposta
  - Visão Geral
  - Definição de requisitos de QoS
  - Estabelecimento do contrato probabilístico
  - Monitoramento de coreografias
- 6 Experimentos e Resultados
  - **Definição de Requisitos de QoS Analiticamente**
  - Definição de requisitos de QoS usando ChorSim
  - Estabelecimento do contrato de QoS
  - Monitoramento
- 7 Conclusões e Trabalhos Futuros

# Cenário de Coreografia para a abordagem analítica

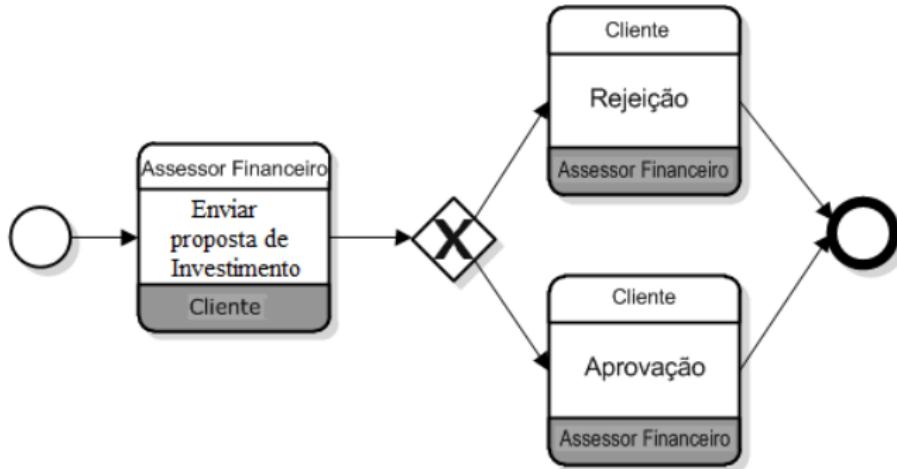


Figura: Exemplo de modelo de interação de coreografias, oferta de investimento.

# Mapeamento

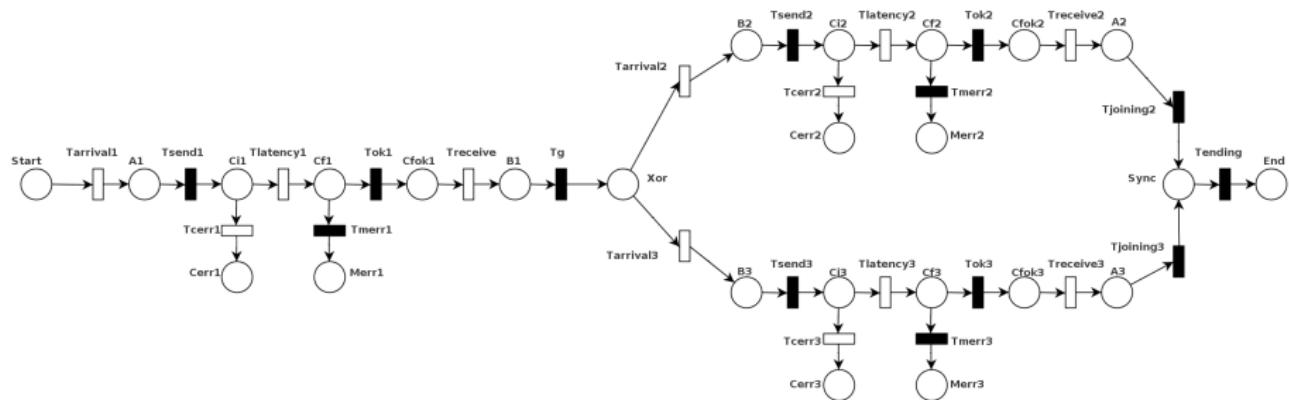
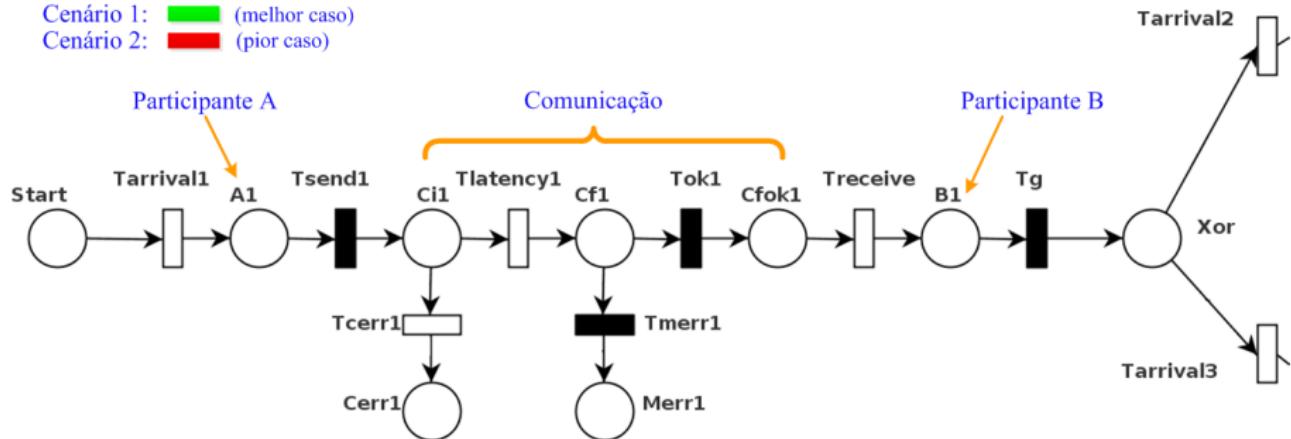


Figura: GSPN obtida após o mapeamento.

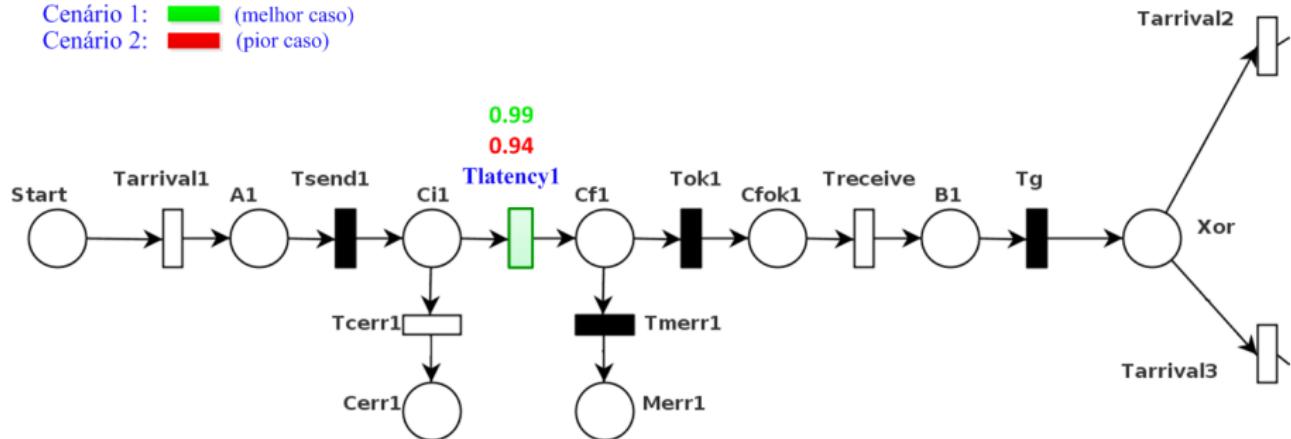
# Configuração (I)

Cenário 1: (melhor caso)  
Cenário 2: (pior caso)



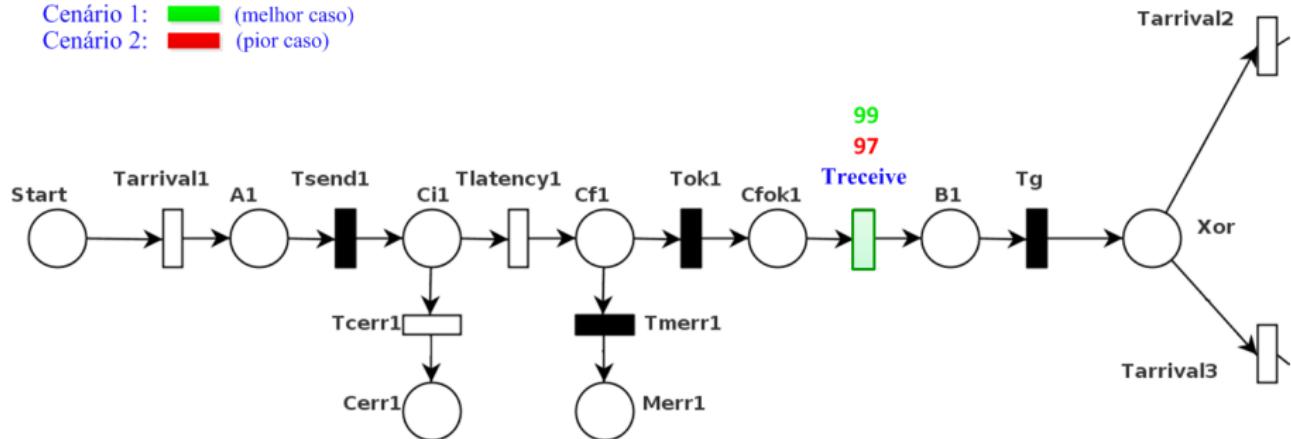
# Configuração (I)

Cenário 1: █ (melhor caso)  
Cenário 2: █ (pior caso)



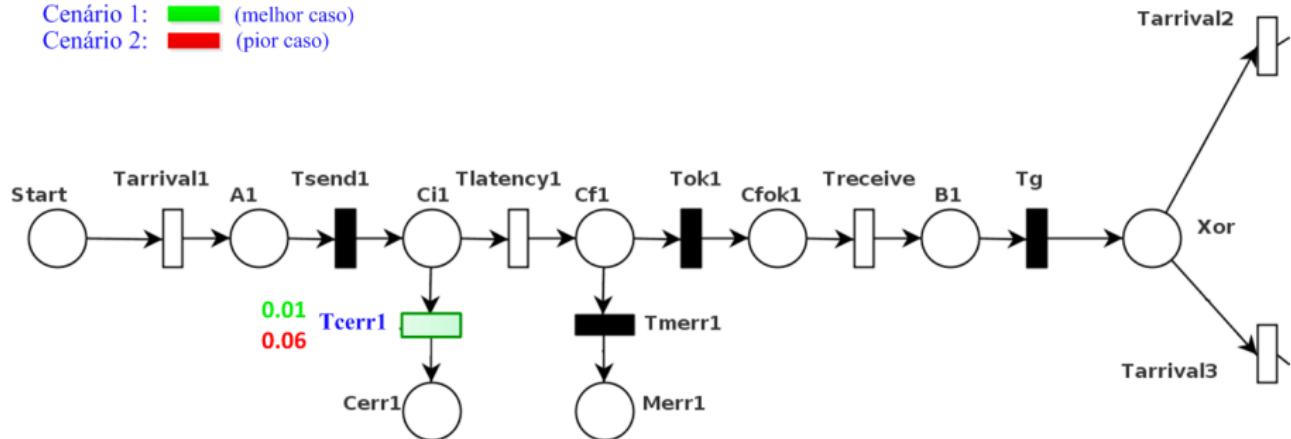
# Configuração (I)

Cenário 1: █ (melhor caso)  
Cenário 2: █ (pior caso)



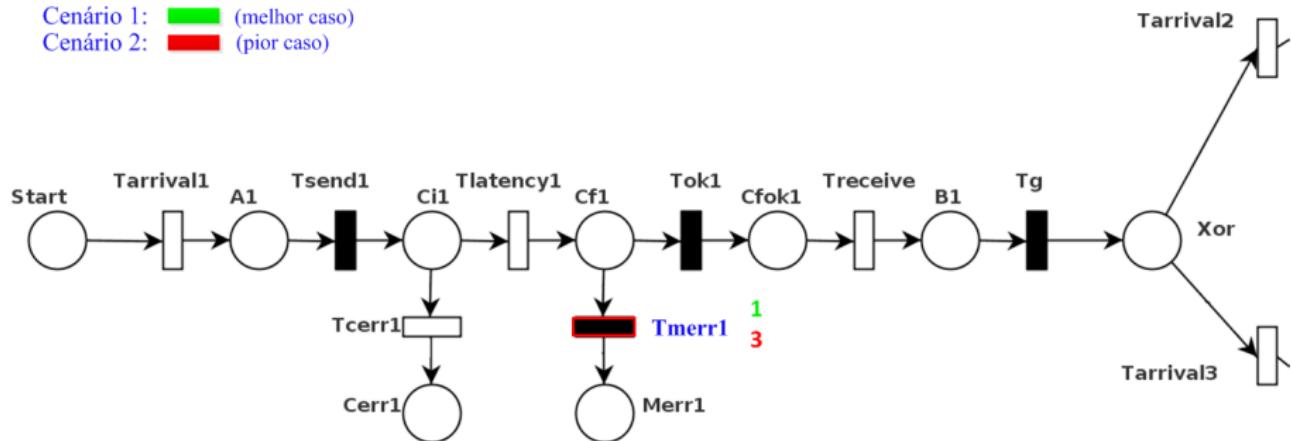
# Configuração (I)

Cenário 1: ■ (melhor caso)  
Cenário 2: ■ (pior caso)



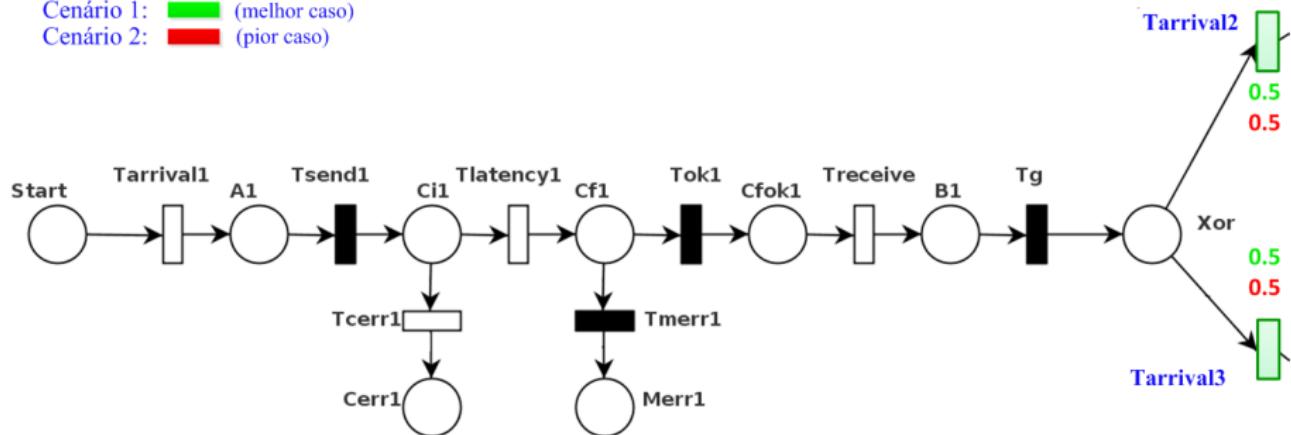
# Configuração (I)

Cenário 1: █ (melhor caso)  
Cenário 2: █ (pior caso)



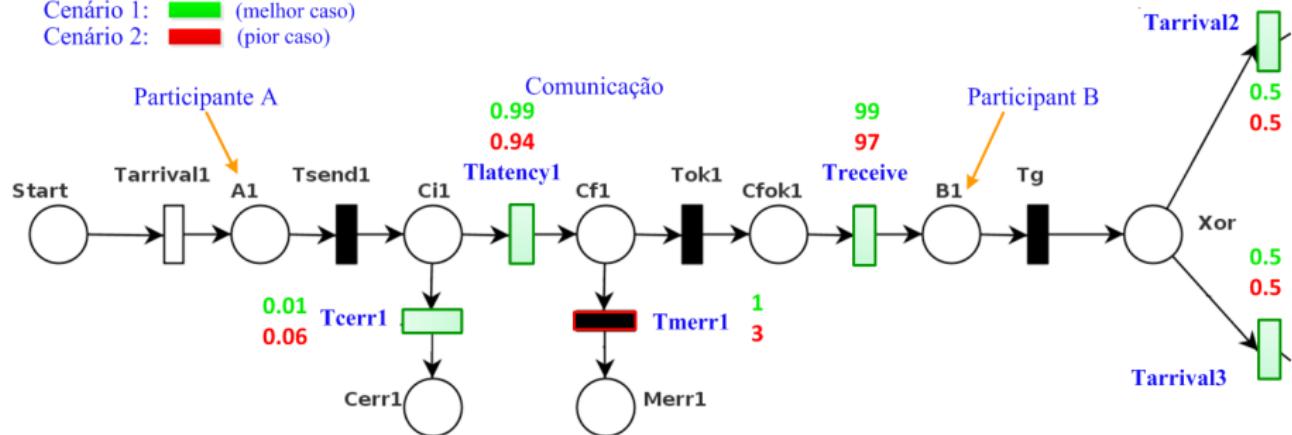
# Configuração (I)

Cenário 1: █ (melhor caso)  
Cenário 2: █ (pior caso)



# Configuração (I)

Cenário 1: (melhor caso)  
Cenário 2: (pior caso)



## Configuração (II)

Tabela: Pesos dos cenários 1 e 2

Transição	Pesos	
	Cenário 1	Cenário 2
$T_{latency1}, T_{latency2}, T_{latency3}$	0.99	0.94
$T_{cerr1}, T_{cerr2}, T_{cerr3}$	0.01	0.06
$T_{receive}, T_{receive2}, T_{receive3}$	99	97
$T_{merr1}, T_{merr2}, T_{merr3}$	1	3
$T_{arrival2}, T_{arrival3}$	0.5	0.5

# Simulações na GSPN

- A ferramenta **Pipe2** foi usado para **modelar** e **simular** a **GSPN**.

# Simulações na GSPN

- A ferramenta **Pipe2** foi usado para **modelar** e **simular** a **GSPN**.
- **1 token = 1 instância de coreografia.**
- 100 **tokens** são considerados para cada cenários na posição de inicio (**Start**).
- 100 **instâncias concorrentes** foram executadas (*multiple-server semantic*).

# Simulações na GSPN

- A ferramenta **Pipe2** foi usado para **modelar** e **simular** a **GSPN**.
- **1 token = 1 instância de coreografia.**
- 100 **tokens** são considerados para cada cenários na posição de inicio (**Start**).
- 100 **instâncias concorrentes** foram executadas (*multiple-server semantic*).
- 1500 disparos e 10 replicações.
- Nível de confiança de 95%.

# Resultados

Tabela: Resultados das simulações

Posição	Número médio de tokens (%)	
	Cenário 1	Cenário 2
<i>Start</i>	35.28	40.15
<i>End</i>	41.95	38.78
$M_{err1}$	0.39	0.91
$M_{err2}$	0.00	0.93
$M_{err3}$	0.00	0.66
$C_{err1}$	0.74	2.94
$C_{err2}$	0.00	0.00
$C_{err3}$	0.78	0.16
$C_{i1}$	8.32	8.90
$C_{i2}$	0.63	0.69
$C_{i3}$	0.75	8.90

# Resultados

Tabela: Resultados das simulações

Posição	Número médio de tokens (%)	
	Cenário 1	Cenário 2
Erros no formato da mensagem	Start	35.28
	End	41.95
	$M_{err1}$	0.39
	$M_{err2}$	0.00
	$M_{err3}$	0.00
	$C_{err1}$	0.74
	$C_{err2}$	0.00
	$C_{err3}$	0.78
	$C_{i1}$	8.32
	$C_{i2}$	0.63
	$C_{i3}$	0.75
		8.90

**Instâncias perdidas:**

- 1.52%
- 3.10%

# Resultados

Tabela: Resultados das simulações

	Posição	Número médio de tokens (%)	
		Cenário 1	Cenário 2
Erros no formato da mensagem	Start	35.28	40.15
	End	41.95	38.78
	$M_{err1}$	0.39	0.91
Erros de Comunicação	$M_{err2}$	0.00	0.93
	$M_{err3}$	0.00	0.66
	$C_{err1}$	0.74	2.94
	$C_{err2}$	0.00	0.00
	$C_{err3}$	0.78	0.16
	$C_{i1}$	8.32	8.90
	$C_{i2}$	0.63	0.69
	$C_{i3}$	0.75	8.90

**Instâncias perdidas:**

■ 1.52% (verde)

■ 3.10% (vermelho)

**Instâncias perdidas:**

■ 0.39% (verde)

■ 2.50% (vermelho)

# Resultados

Tabela: Resultados das simulações

	Posição	Número médio de tokens (%)	
		Cenário 1	Cenário 2
Erros no formato da mensagem	Start	35.28	40.15
	End	41.95	38.78
	$M_{err1}$	0.39	0.91
Erros de Comunicação	$M_{err2}$	0.00	0.93
	$M_{err3}$	0.00	0.66
	$C_{err1}$	0.74	2.94
Gargalhos	$C_{err2}$	0.00	0.00
	$C_{err3}$	0.78	0.16
	$C_{i1}$	8.32	8.90
	$C_{i2}$	0.63	0.69
	$C_{i3}$	0.75	8.90

**Instâncias perdidas:**

■ 1.52% ■ 3.10%

**Instâncias perdidas:**

■ 0.39% ■ 2.50%

## 1 Conceitos Básicos

## 2 Problema

## 3 QoS e Monitoramento em Coreografias de Serviços Web

## 4 Trabalhos Relacionados

## 5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias

## 6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- **Definição de requisitos de QoS usando ChorSim**
- Estabelecimento do contrato de QoS
- Monitoramento

## 7 Conclusões e Trabalhos Futuros

# Cénario de coreografia para a abordagem com ChorSim

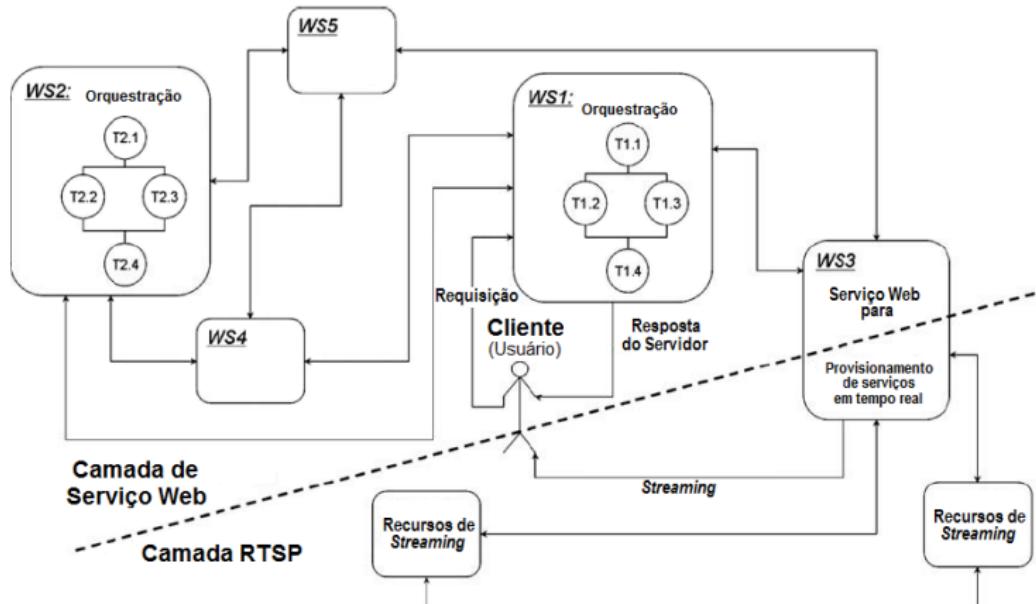


Figura: Coreografia de serviços da aplicação de CDN

# Configuração das simulações em ChorSim (I)

## Objetivo

Analizar o comportamento do **tempo de resposta total** do serviço composto  $WS_1$  em função do **tamanho da resposta** de  $WS_1$  e diferentes valores de **largura de banda**.

# Configuração das simulações em ChorSim (I)

## Objetivo

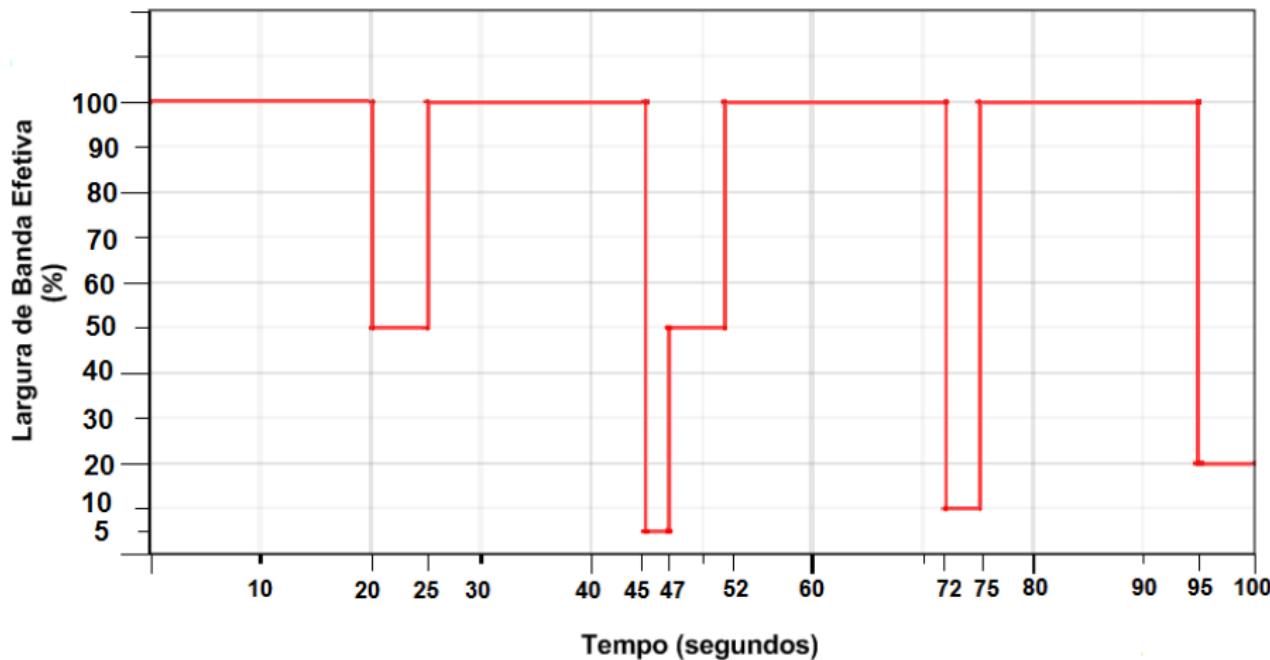
Analizar o comportamento do **tempo de resposta total** do serviço composto  $WS_1$  em função do **tamanho da resposta** de  $WS_1$  e diferentes valores de **largura de banda**.

- **Cenário 1:** Larguras de banda uniformes.
- **Cenário 2:** Larguras de banda variáveis.

# Configuração das simulações em ChorSim (II)

## Cenario 2:

Largura de Banda Efetiva Vs Tempo (Periodo=100s)



# Configuração das simulações em ChorSim (III)

Tabela: Configuração de valores dos atributos de QoS nas requisições

Requisições	Largura de banda	Tamanho da requisição	Latência	# requisições
Cliente a WS <sub>1</sub>	1Mbps	1.95MB	0.002s	1 a 10
WS <sub>1</sub> a WS <sub>3</sub>	1Mbps	5.47MB	0.002s	1 a 10
WS <sub>3</sub> a WS <sub>5</sub>	1Mbps	5.47MB	0.002s	1 a 10

# Configuração das simulações em ChorSim (III)

Tabela: Configuração de valores dos atributos de QoS nas requisições

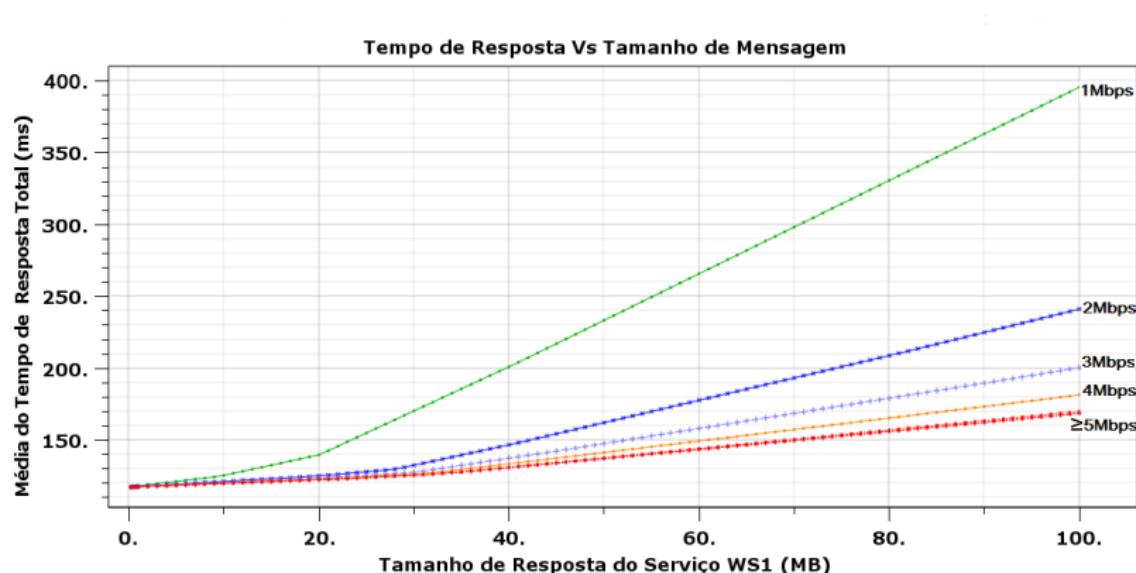
Requisições	Largura de banda	Tamanho da requisição	Latência	# requisições
Cliente a WS <sub>1</sub>	1Mbps	1.95MB	0.002s	1 a 10
WS <sub>1</sub> a WS <sub>3</sub>	1Mbps	5.47MB	0.002s	1 a 10
WS <sub>3</sub> a WS <sub>5</sub>	1Mbps	5.47MB	0.002s	1 a 10

Tabela: Configuração de valores dos atributos de QoS nas respostas

Respostas	Largura de banda	Tamanho de resposta	Latência	timeout
WS <sub>1</sub> a Cliente	1Mbps a 16Mbps	1KB a 100MB	0.002s	1000s
WS <sub>3</sub> a WS <sub>1</sub>	20Mbps	8MB	0.002s	1000s
WS <sub>5</sub> a WS <sub>3</sub>	40Mbps	200MB	0.002s	1000s

# Resultados das simulações usando ChorSim

## Cenário 1



**Figura: Cenario 1:** Tempo médio de resposta total da coreografia em função do tamanho de resposta do serviço  $WS_1$  com larguras de banda de  $1Mbps$  até  $16Mbps$

# Resultados das simulações usando ChorSim

## Cenário 2

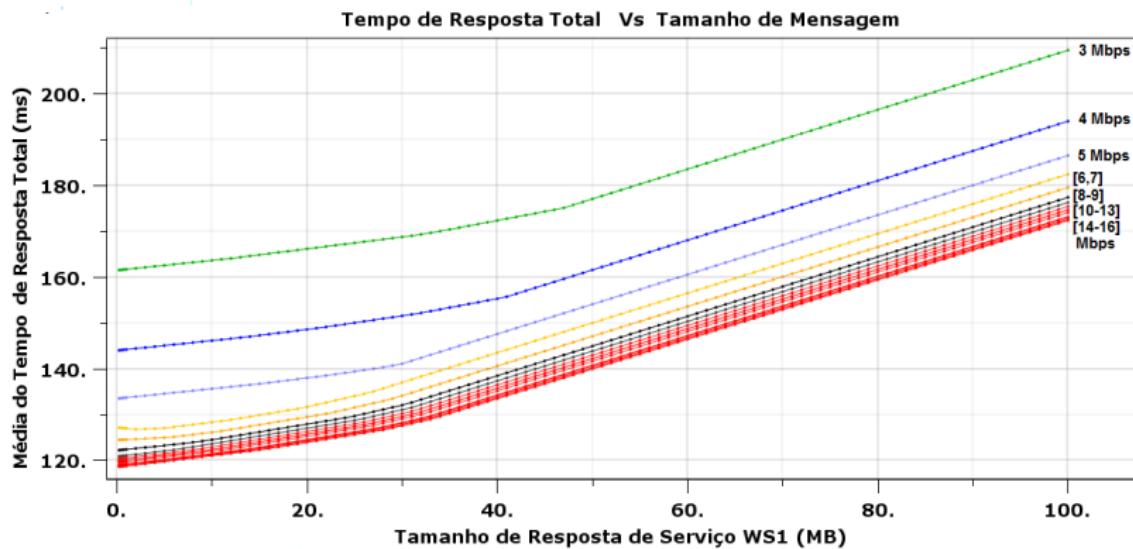


Figura: Cenário 2: Tempo médio de resposta total da coreografia em função do tamanho de resposta do serviço  $WS_1$ . A largura de banda varia de 1Mbps até 16Mbps

1 Conceitos Básicos

2 Problema

3 QoS e Monitoramento em Coreografias de Serviços Web

4 Trabalhos Relacionados

5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias

6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS**
- Monitoramento

7 Conclusões e Trabalhos Futuros

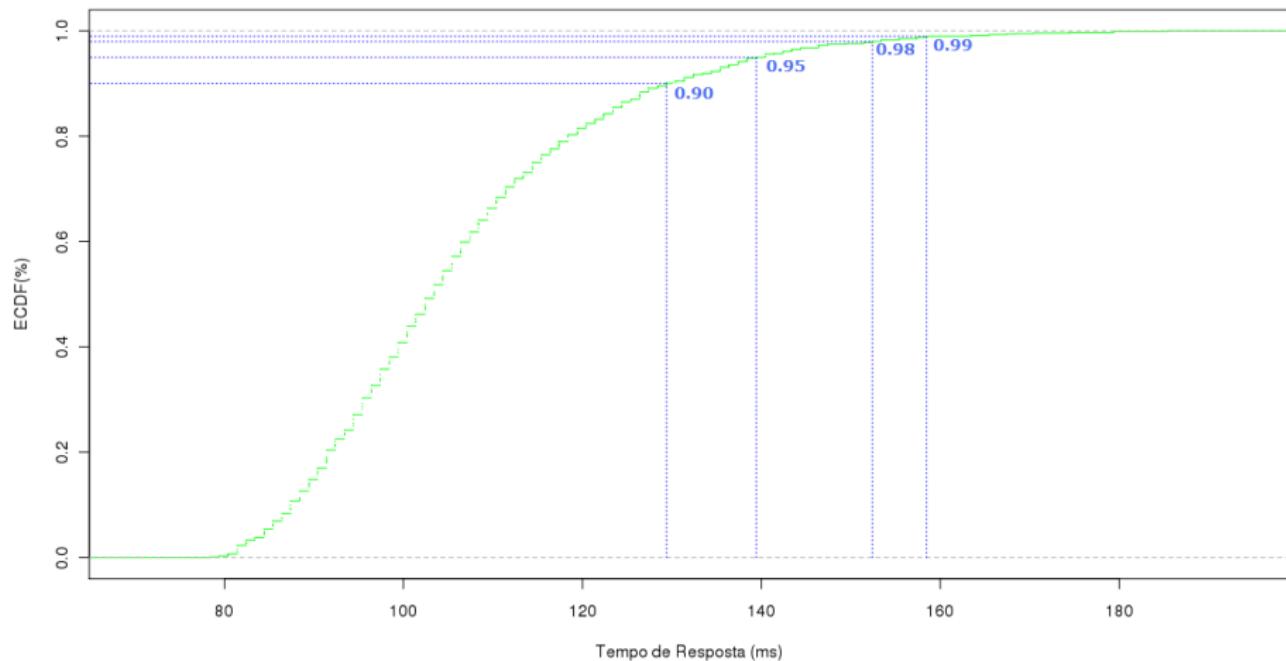
# Estabelecimento do contrato probabilístico (I)

**Tabela:** Configuração das taxas de degradação dos serviços para obter o contrato do tempo de resposta para o serviço composto  $WS_1$

Serviço	Distribuição	Taxa de degradação ( $\lambda$ )
$WS_1$	Exponencial	1/10000
$WS_3$	Exponencial	1/10000
$WS_5$	Exponencial	1/10000

# Estabelecimento do contrato probabilístico (II)

Distribuição de Probabilidade Empírica (ECDF) do Contrato



**Figura:** Distribuição de probabilidade empírica (ECDF) do contrato do serviço  $WS_1$  com base nos tempos de resposta

1 Conceitos Básicos

2 Problema

3 QoS e Monitoramento em Coreografias de Serviços Web

4 Trabalhos Relacionados

5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias

6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS
- Monitoramento**

7 Conclusões e Trabalhos Futuros

# Configuração do Monitoramento

$$\text{verify}(X_{monitoring}) = \begin{cases} \text{true}, & \text{se } p^+ \geq 0.05 \wedge D^+ < 0.15 \\ \text{false}, & \text{de outra maneira} \end{cases} \quad (5)$$

# Configuração do Monitoramento

$$\text{verify}(X_{\text{monitoring}}) = \begin{cases} \text{true}, & \text{se } p^+ \geq 0.05 \wedge D^+ < 0.15 \\ \text{false}, & \text{de outra maneira} \end{cases} \quad (5)$$

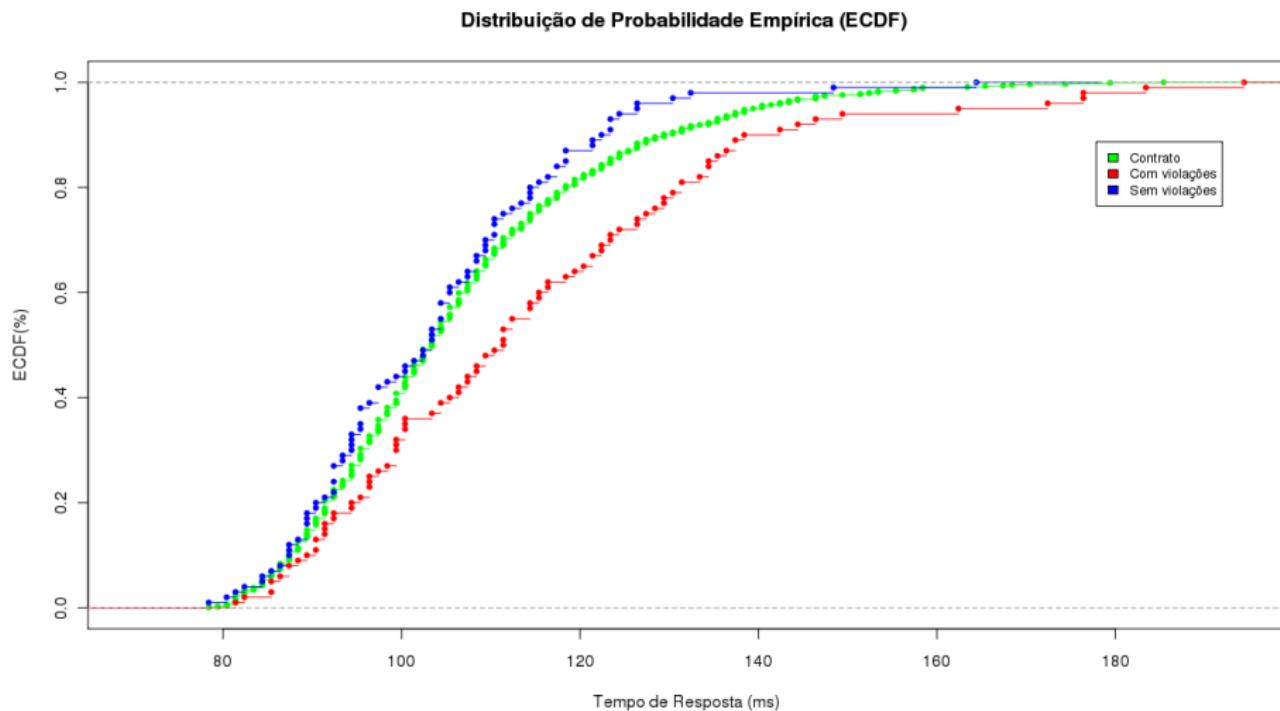
- $X_{\text{monitoring}}$ : Conjunto de amostras dos tempos de resposta do serviço composto  $WS_1$ .
- Um monitoramento sequencial e *on-line* precisa computar  $N$  amostras que se sobrepõem.
- Conjunto de janelas de amostras a monitorar:  
 $\{1, \dots, N\}, \{p, \dots, p + N\}, \dots, \{mp, \dots, mp + N\} \dots$   
 $p \leq N$  e  $m = 1, 2, \dots$
- Configuração:  $N = 100$ , e o desvio  $p = 1$ .

# Cenários para o monitoramento

**Tabela:** Estabelecimento das taxas de degradação de tempo de processamento dos serviços para os cenários.

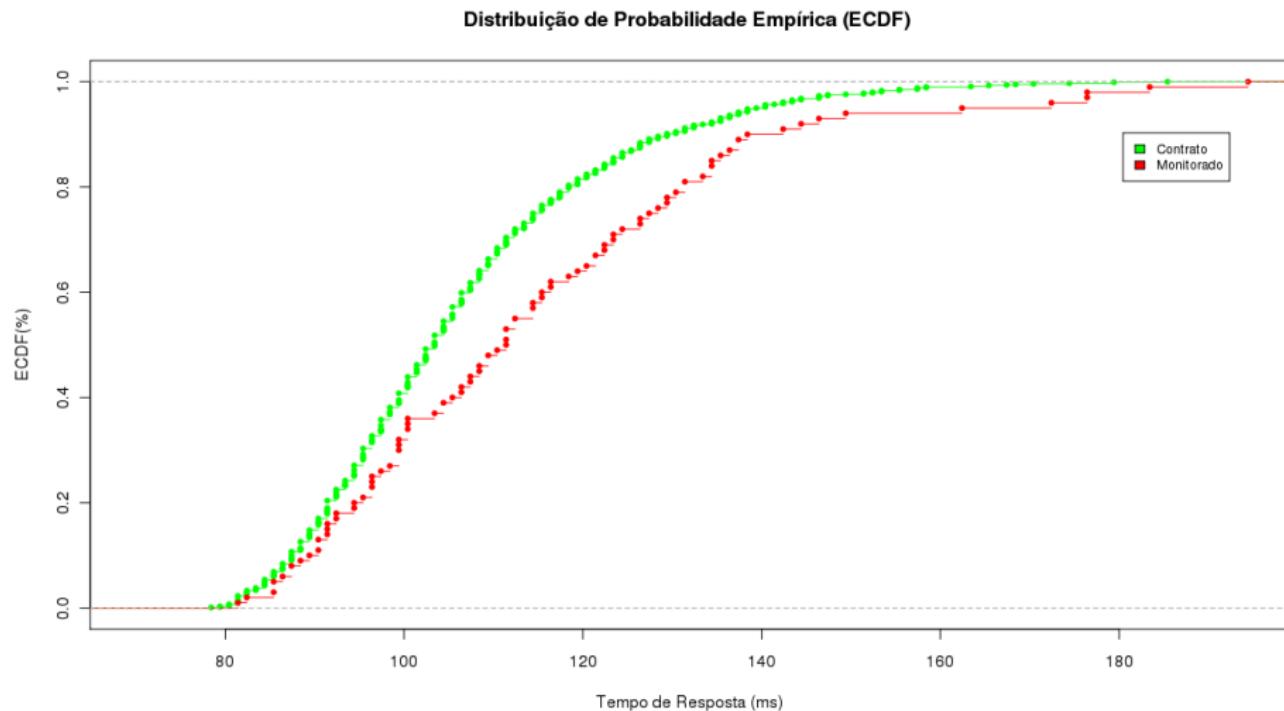
<b>Cenário</b>	<b>Taxa de degradação (<math>\lambda</math>)</b>		
	$WS_1$	$WS_3$	$WS_5$
Cenário 1	1/13000	1/12500	1/11500
Cenário 2	1/12000	1/11000	1/12000
Cenário 3	1/10500	1/10000	1/10500
Cenário 4	1/9000	1/10000	1/9000
<b>Contrato</b>	1/10000	1/10000	1/10000

# Resultados da detecção de violações de SLA (I)



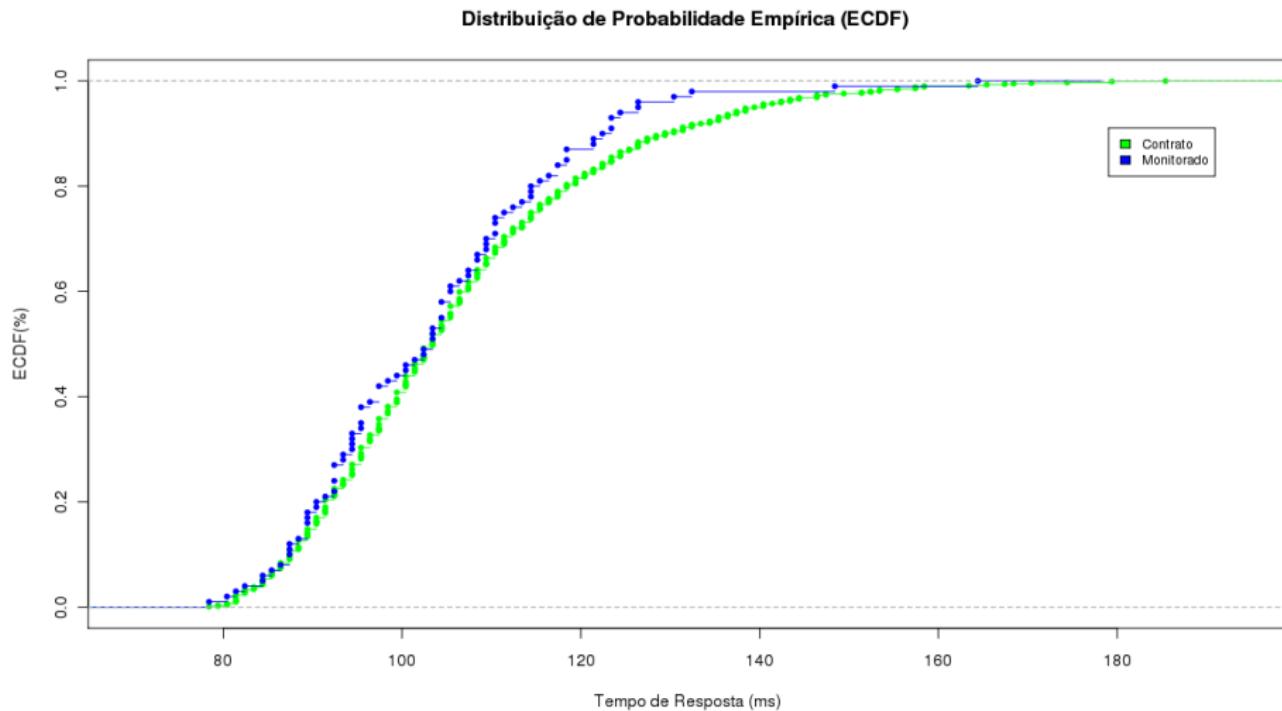
**Figura:** Comparação do ECDF do contrato com o ECDF do cenário 1 e 4. O tamanho das amostras dos cenários é 100.

# Resultados da detecção de violações de SLA (I)



**Figura:** Comparação do ECDF do contrato com o ECDF do cenário 1

# Resultados da detecção de violações de SLA (I)



**Figura:** Comparação do ECDF do contrato com o ECDF do cenário 4

# Resultados da detecção de violações de SLA (II) - *Online*

## Cenário 1

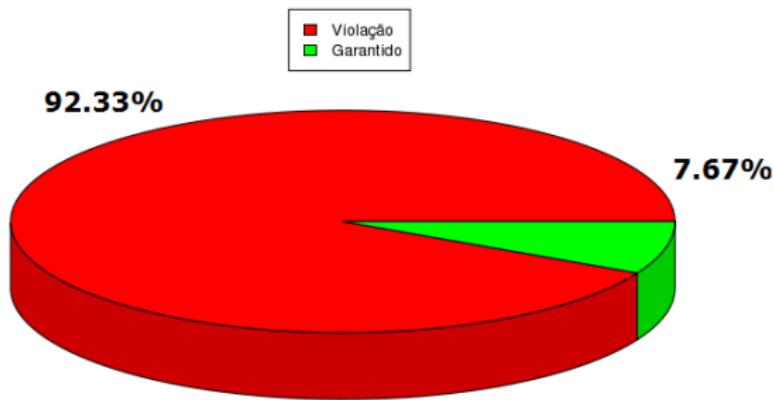
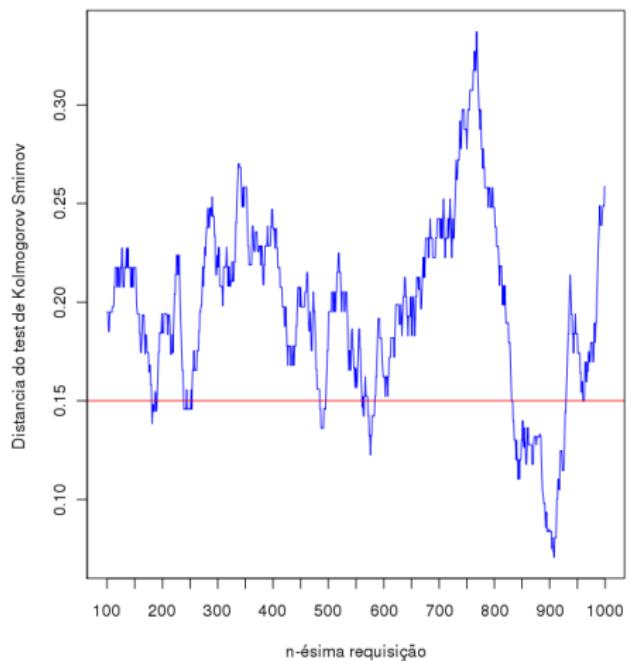


Figura: Violações de SLA para o cenário 1.

# Resultados da detecção de violações de SLA (II) - *Online*

## Cenário 1

Monitoramento: Distância de Kolmogorov



Monitoramento: P+ value de Kolmogorov

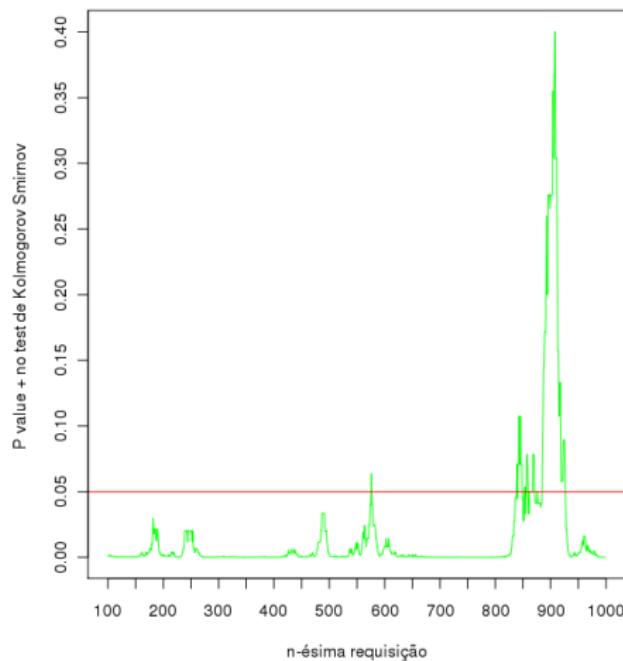


Figura: Monitoramento e detecção de violações de SLA para o cenário 1.

# Resultados da detecção de violações de SLA

## Cenário 2

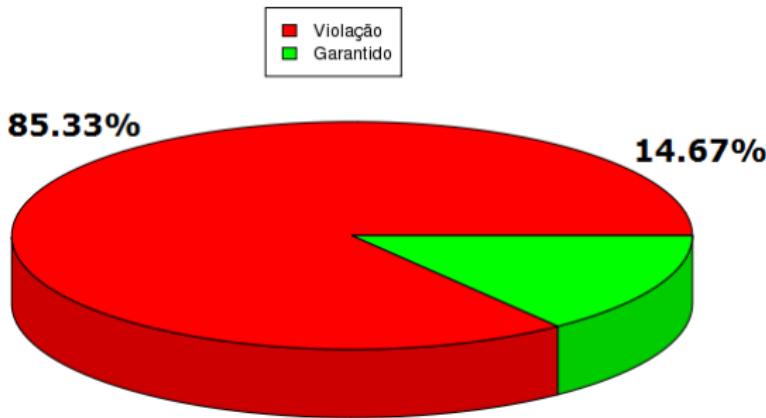
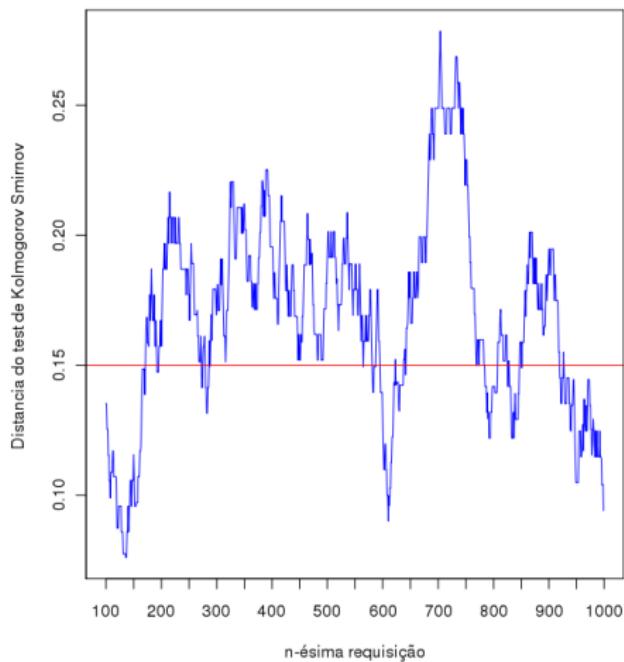


Figura: Violações de SLA para o cenário 2.

# Resultados da detecção de violações de SLA

## Cenário 2

Monitoramento: Distância de Kolmogorov



Monitoramento: P+ value de Kolmogorov

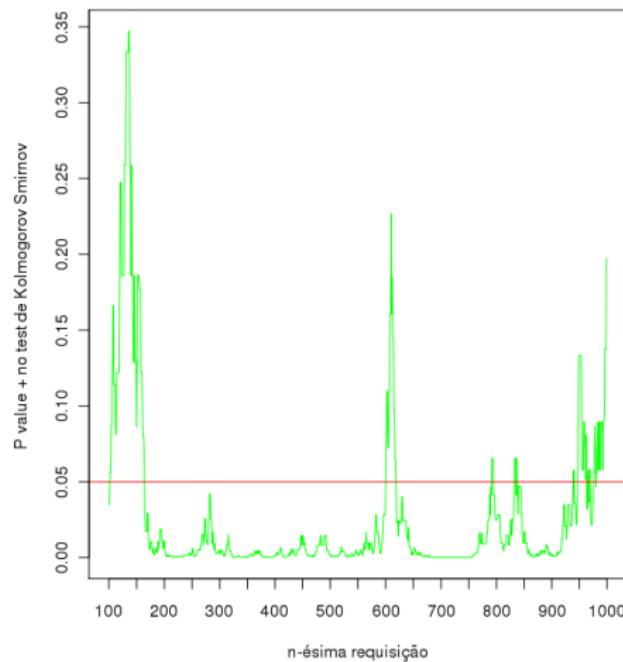


Figura: Monitoramento e detecção de violações de SLA para o cenário 2

# Resultados da detecção de violações de SLA

## Cenário 3

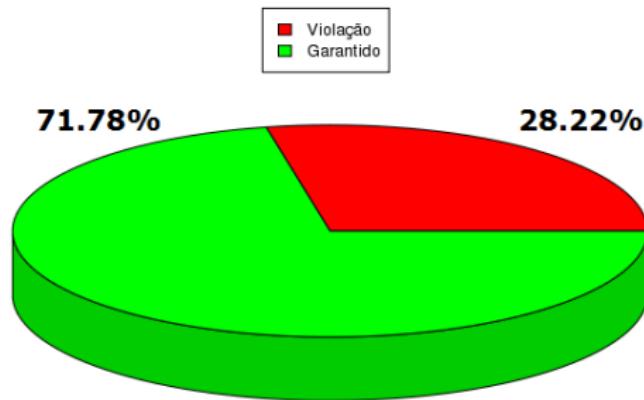
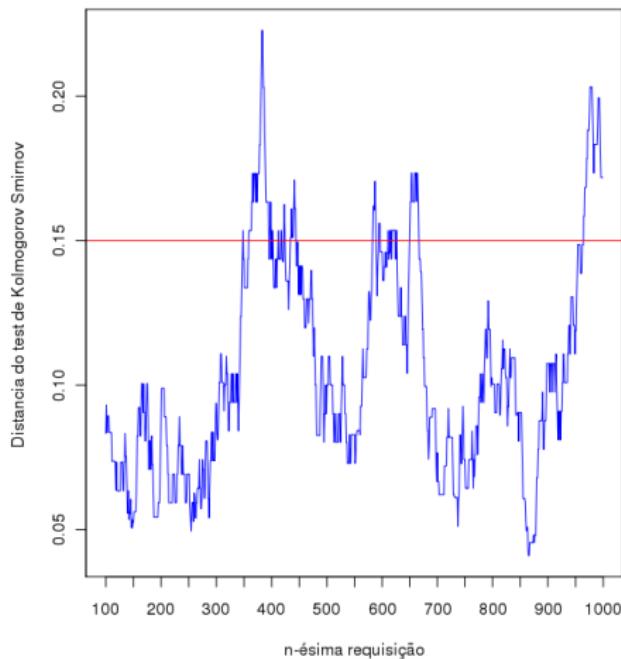


Figura: Violações de SLA para o cenário 3.

# Resultados da detecção de violações de SLA

## Cenário 3

Monitoramento: Distância de Kolmogorov



Monitoramento: P+ value de Kolmogorov

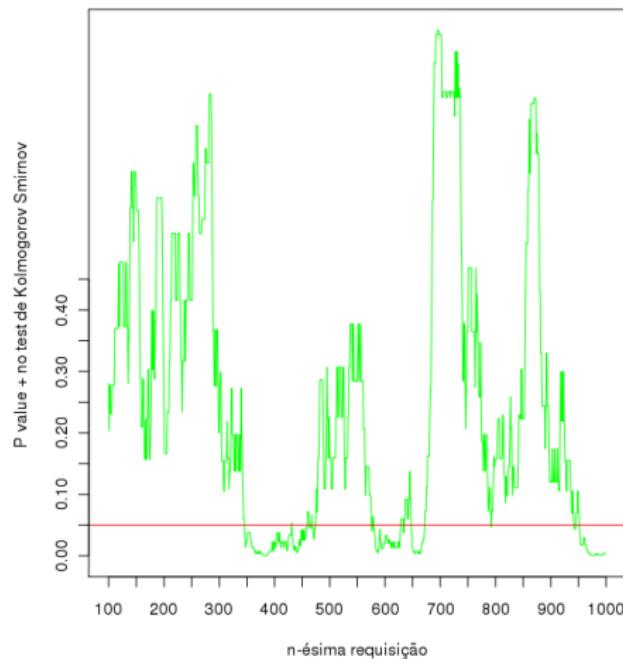


Figura: Monitoramento e detecção de violações de SLA para o cenário 3.

# Resultados da detecção de violações de SLA

## Cenário 4

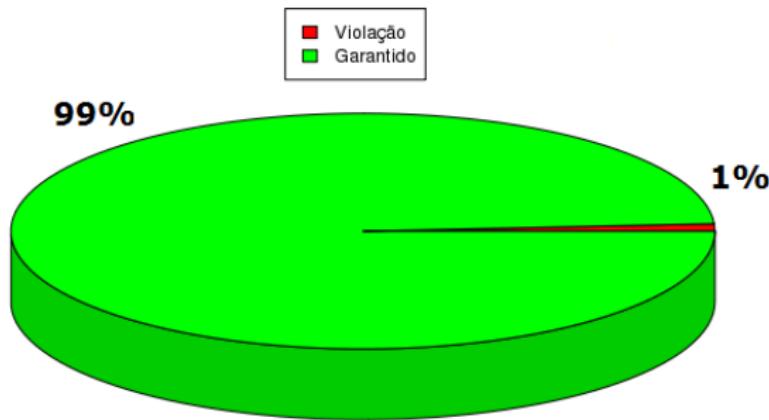
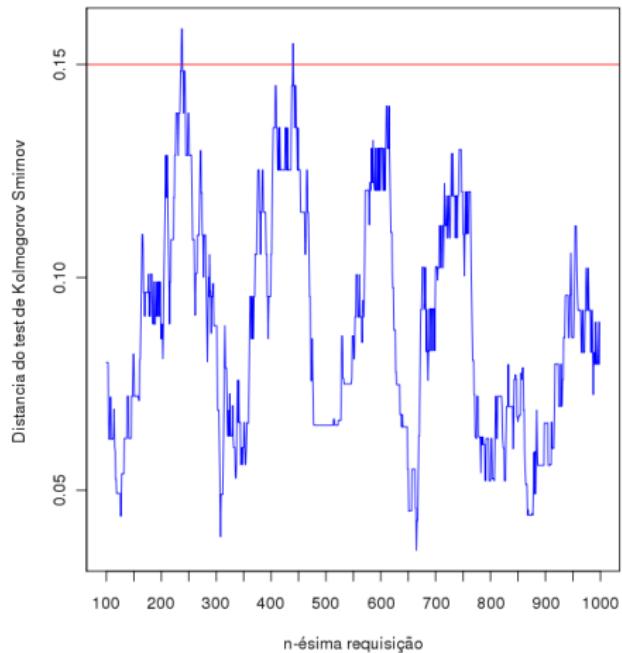


Figura: Violações de SLA para o cenário 4.

# Resultados da detecção de violações de SLA

## Cenário 4

Monitoramento: Distância de Kolmogorov



Monitoramento: P+ value de Kolmogorov

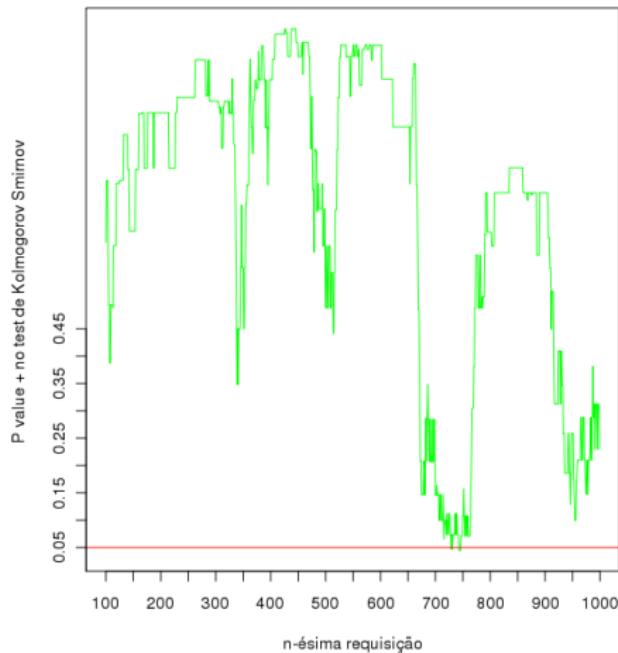


Figura: Monitoramento e detecção de violações de SLA para o cenário 4.

## 1 Conceitos Básicos

## 2 Problema

## 3 QoS e Monitoramento em Coreografias de Serviços Web

## 4 Trabalhos Relacionados

## 5 Proposta

- Visão Geral
- Definição de requisitos de QoS
- Estabelecimento do contrato probabilístico
- Monitoramento de coreografias

## 6 Experimentos e Resultados

- Definição de Requisitos de QoS Analiticamente
- Definição de requisitos de QoS usando ChorSim
- Estabelecimento do contrato de QoS
- Monitoramento

## 7 Conclusões e Trabalhos Futuros

# Conclusões (I)

- Foram propostas abordagens e técnicas para definição de requisitos de QoS, estabelecimento de contratos probabilísticos e monitoramento com a detecção de violações de contrato em coreografias de serviços Web.
- O modelo de interação em BPMN2 foi utilizado.
- Construi-se um simulador para suportar *enactment* de coreografias de serviços Web com suporte de QoS.
- Na definição de requisitos, propôs-se uma metodologia usando mapeamento de coreografia para GSPN. Além disso, apresentou-se avaliações de desempenho usando o ChorSim.

## Conclusões (II)

- Apresentou-se uma abordagem para estabelecer contratos probabilísticos usando ChorSim.
- O monitor foi desenvolvido de modo a suportar um monitoramento *online*.
- Foi proposta uma técnica de dominância estocástica para detectar violações de contratos probabilísticos. A técnica está baseada no teste de *Kolmogorov-Smirnov* de apenas um lado.
- Os resultados mostram que na regra para detectar violações o *p-value* pode ser suficiente.

# Trabalhos Futuros

- Suporte de mais atributos de QoS de desempenho e outros como confiabilidade, segurança, entre outros, em todas as etapas.
- Suporte de mais elementos de coreografia de serviços do BPMN2 em todas as etapas.
- Pesquisa em substituição dinâmica, adaptação dinâmica, reconfiguração, autocura, entre outro, baseada em QoS para coreografias de serviços Web.
- Suporte de definição probabilística de atributos de QoS como largura de banda e latência de rede no ChorSim.
- Avaliações das simulações com implementações reais de coreografias.
- Levar em consideração exemplos ou cenários reais e/ou complexos.

# Envolvimento



**Baile Project**  
Enabling Scalable Cloud Service Choreographies

# Obrigado!