

Detection of SLA Violations in Web Service Choreography

Student

Victoriano Alfonso Phocco Diaz

Advisor

Daniel Macedo Batista

Co-Advisor

Marco Dimas Gubitoso

MSc. Qualification Exam

Mathematics and Statistics Institute (IME)
Computer Science Department
University of Sao Paulo

18/10/2011

Agenda

- 1 Problem
- 2 Objectives
- 3 QoS and Monitoring on Web Service Choreography
- 4 Related Works
- 5 Contributions
- 6 Purpose

- 1 Problem
- 2 Objectives
- 3 QoS and Monitoring on Web Service Choreography
- 4 Related Works
- 5 Contributions
- 6 Purpose

Web Service

W3C definition [W3C,2004]:

Web Service

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 (1/2)

SOA (Software Oriented Architecture)

SOA can be defined as an architectural style promoting the concept of business-aligned enterprise service as the fundamental unit of designing, building, and composing enterprise business solutions. Multiple patterns, defining design, implementations, and deployment of the SOA solutions, complete this style. [IBM, 2007].

SOA (2/2)

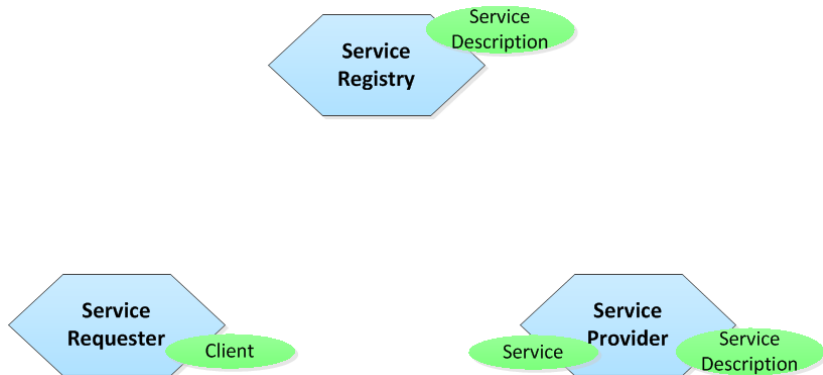


Figure: SOA triangle (based on [W3C, 2002])

SOA (2/2)

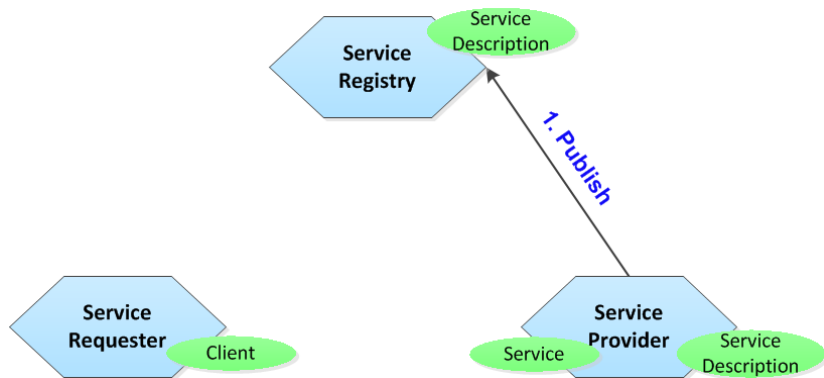


Figure: SOA triangle (based on [W3C, 2002])

SOA (2/2)

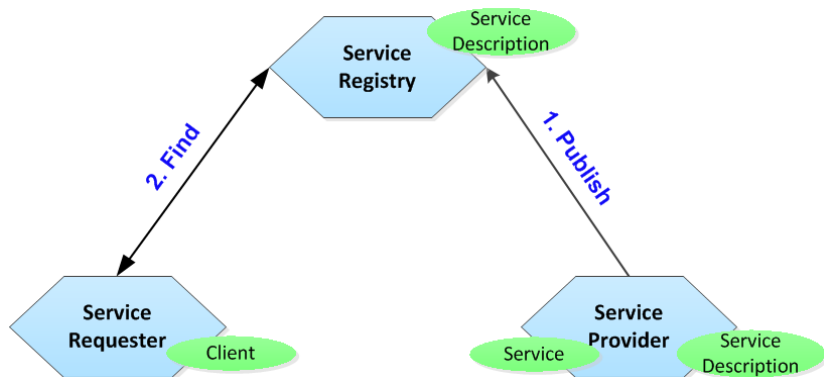


Figure: SOA triangle (based on [W3C, 2002])

SOA (2/2)

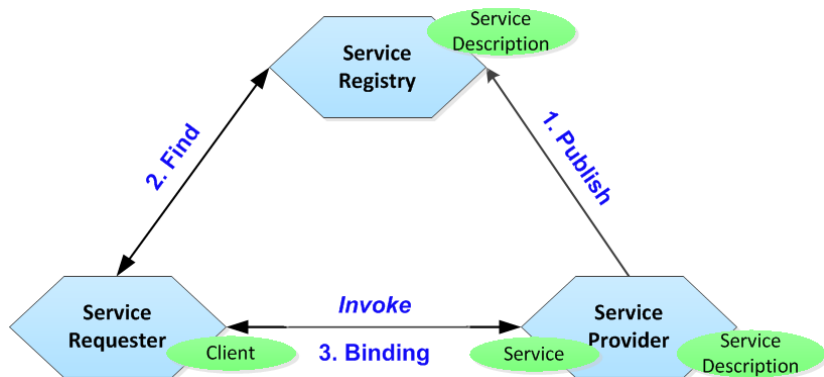


Figure: SOA triangle (based on [W3C, 2002])

SOA (2/2)

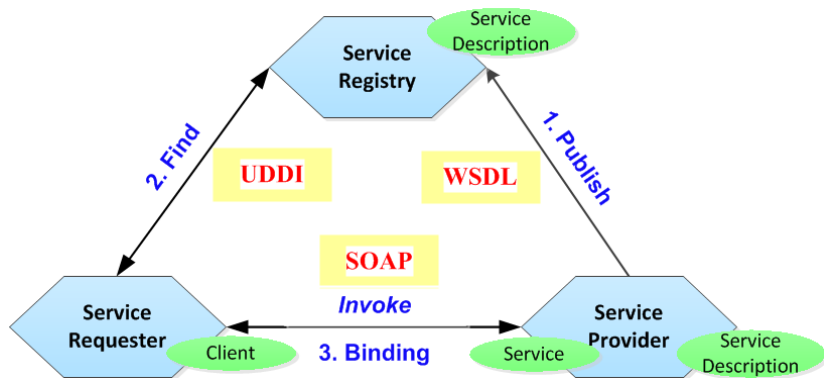


Figure: SOA triangle (based on [W3C, 2002])

SOC (Service Oriented Computing)

It is a new computing paradigm that utilizes services as the basic constructs to support the development of rapid, low-cost and easy composition of distributed applications even in heterogeneous environments. [Papazoglou et al., 2006].

Key elements:

- Services.
- SOA.
- Service Composition.
- ...

Service Composition

- **Composite Service:** A service built from other services. A composite service is also another service.
- **Service Composition:** Process to obtain composite services combining and linking basic services.
- Approaches:

Service Composition

- **Composite Service:** A service built from other services. A composite service is also another service.
- **Service Composition:** Process to obtain composite services combining and linking basic services.
- Approaches:
 - ▶ **Service Orchestration**.

Service Composition

- **Composite Service:** A service built from other services. A composite service is also another service.
- **Service Composition:** Process to obtain composite services combining and linking basic services.
- Approaches:
 - ▶ **Service Orchestration** .
 - ▶ **Service Choreography** .

Service Orchestration

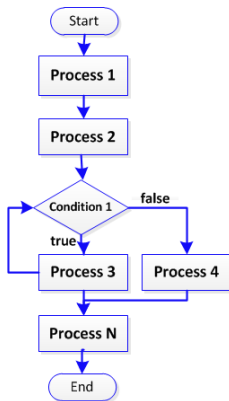


Figure: Service Orchestration

Service Orchestration

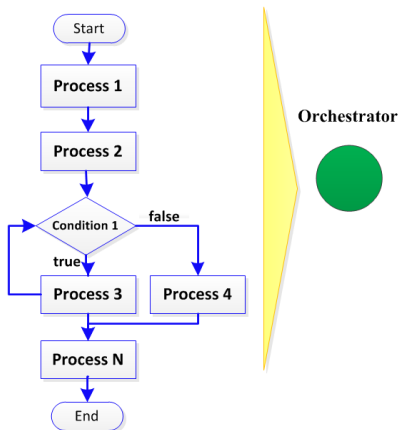


Figure: Service Orchestration

Service Orchestration

Service Orchestration

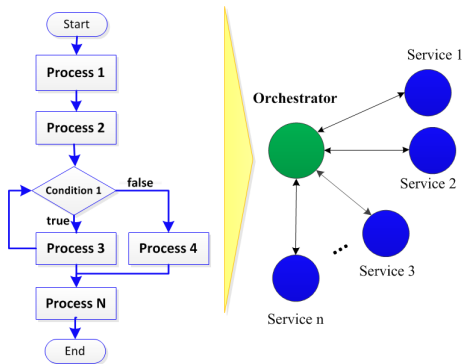


Figure: Service Orchestration

Service Choreography

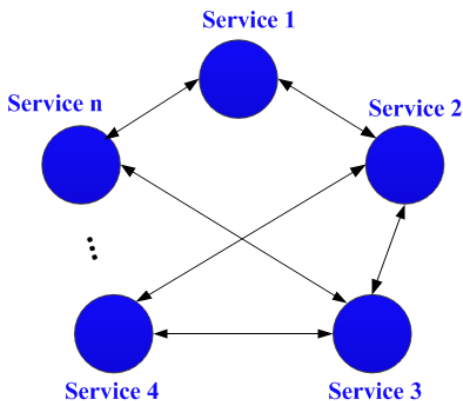


Figure: Service Choreography

Service Choreography

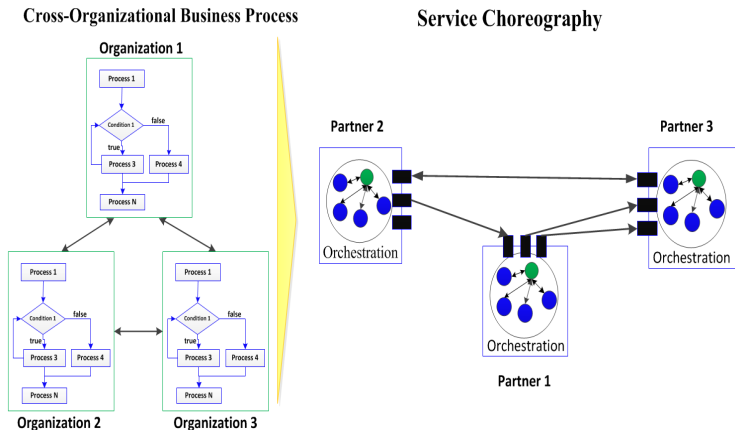


Figure: Service Choreography

Service Choreography

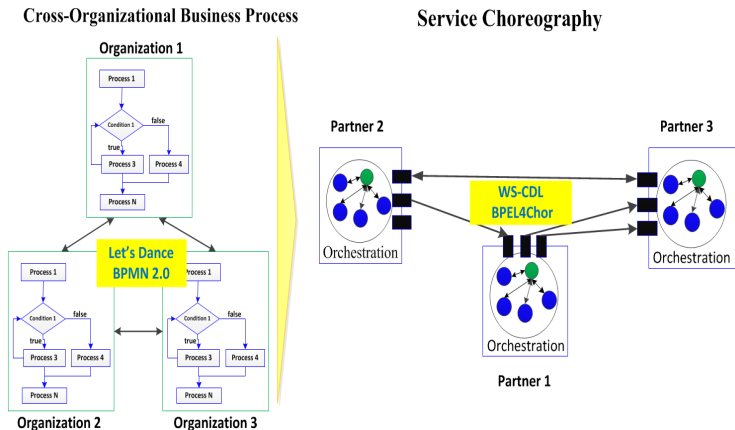


Figure: Service Choreography

Problem to solve

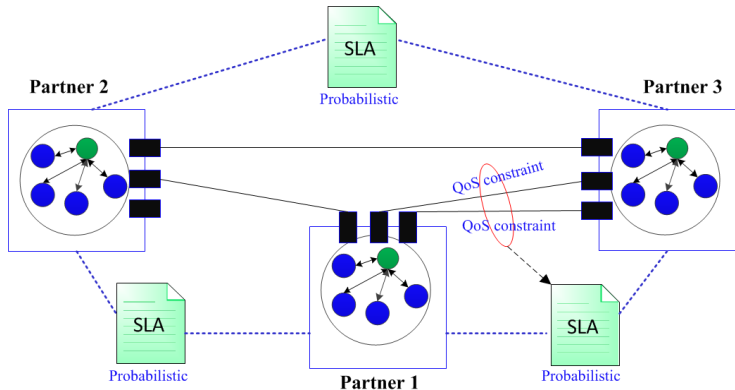


Figure: Problem to solve

- 1 Problem
- 2 Objectives
- 3 QoS and Monitoring on Web Service Choreography
- 4 Related Works
- 5 Contributions
- 6 Purpose

Main Objective

- To detect violations of probabilistic SLAs on Web Service Choreography.

Secondary Objectives

- To propose a technique in order to define SLAs based on probabilistic constraints of QoS.
- To propose and implement a non-intrusive monitoring technique of Web service choreography using SLAs.
- Performance evaluation.

Motivation

- Importance of Web service **choreography**.

Motivation

- Importance of Web service **choreography**.
- **QoS** is key factor to adaptation, selection, optimization, composition into SOC.

Motivation

- Importance of Web service **choreography**.
- **QoS** is key factor to adaptation, selection, optimization, composition into SOC.
- **Monitoring** is a basis for a reaction (adaptation, reconfiguration, renegotiation, etc).
 - ▶ Failure and violations detection of SLA.

Motivation

- Importance of Web service **choreography**.
- **QoS** is key factor to adaptation, selection, optimization, composition into SOC.
- **Monitoring** is a basis for a reaction (adaptation, reconfiguration, renegotiation, etc).
 - ▶ Failure and violations detection of SLA.
- **Probabilistic Contracts** better reflect the dynamic behavior of the QoS attributes of Web services.

- 1 Problem
- 2 Objectives
- 3 QoS and Monitoring on Web Service Choreography
- 4 Related Works
- 5 Contributions
- 6 Purpose

Problem to solve

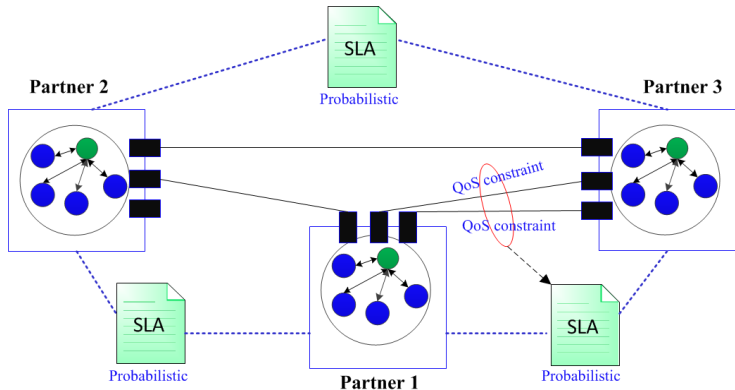


Figure: Problem to solve

Quality of Service

- Quality of Service : QoS.
- **Functional/service** = Which operations a system performs.
 - ▶ Example: Buy airline tickets.
- **Non-Functional QoS/Characteristic** = How a system performs services.
 - ▶ Example: Mean response time less than 2 seconds.
- Important in Service Composition : **QoS-aware Composition**.

Quality of Services

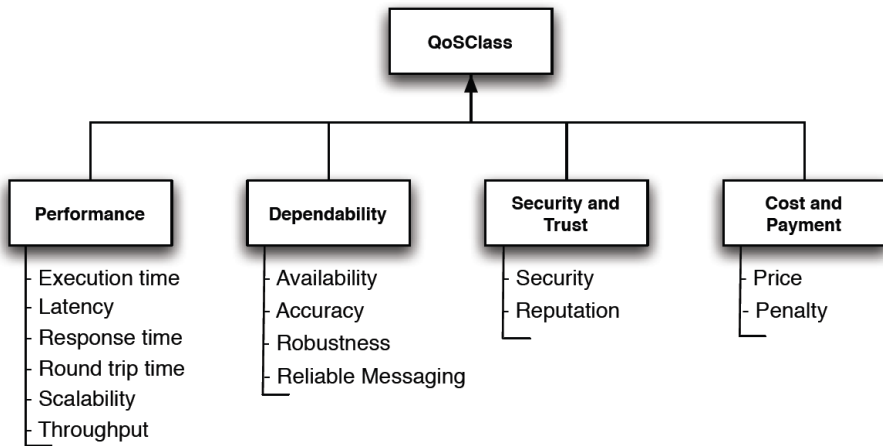


Figure: Taxonomy of QoS attributes [Rosenberg et al.,2006]

QoS Calculation

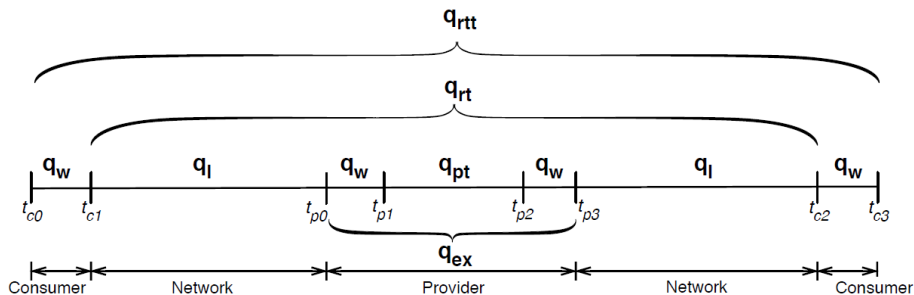


Figure: Timestamps on Web service invocation [Michlmayr et al.,2009]

QoS Calculation

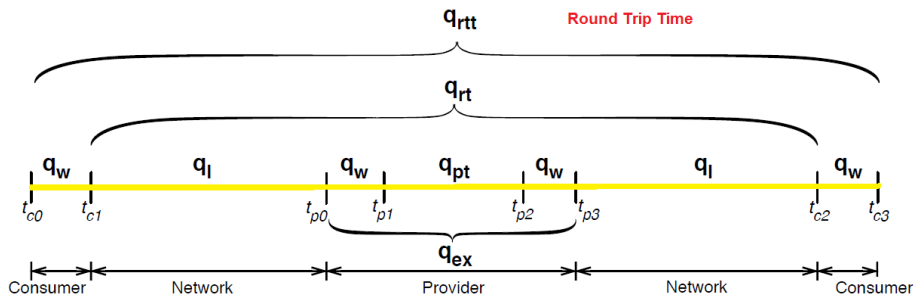


Figure: Timestamps on Web service invocation [Michlmayr et al.,2009]

- **Contract** = Formal agreement between two or more parts in order to define requirements and guarantees.

SLA

- **Contract** = Formal agreement between two or more parts in order to define requirements and guarantees.
- **SLA** = Service Level Agreement is a contract that involves requirements and guarantees of QoS.

- **Contract** = Formal agreement between two or more parts in order to define requirements and guarantees.
- **SLA** = Service Level Agreement is a contract that involves requirements and guarantees of QoS.
- **A SLA consist of:**
 - ▶ Parts
 - ▶ Service Operations:
 - ▶ Obligations:

- **Contract** = Formal agreement between two or more parts in order to define requirements and guarantees.
- **SLA** = Service Level Agreement is a contract that involves requirements and guarantees of QoS.
- **A SLA consist of:**
 - ▶ Parts
 - ▶ Service Operations:
 - ★ Operations
 - ▶ Obligations:

- **Contract** = Formal agreement between two or more parts in order to define requirements and guarantees.
- **SLA** = Service Level Agreement is a contract that involves requirements and guarantees of QoS.
- **A SLA consist of:**
 - ▶ Parts
 - ▶ Service Operations:
 - ★ Operations
 - ★ **SLA Parameters:** QoS metric definitions.
 - ▶ Obligations:

- **Contract** = Formal agreement between two or more parts in order to define requirements and guarantees.
- **SLA** = Service Level Agreement is a contract that involves requirements and guarantees of QoS.
- **A SLA consist of:**
 - ▶ Parts
 - ▶ Service Operations:
 - ★ Operations
 - ★ **SLA Parameters:** QoS metric definitions.
 - ▶ Obligations:
 - ★ **QoS Guaranties (objectives or constraints).**

- **Contract** = Formal agreement between two or more parts in order to define requirements and guarantees.
- **SLA** = Service Level Agreement is a contract that involves requirements and guarantees of QoS.
- **A SLA consist of:**
 - ▶ Parts
 - ▶ Service Operations:
 - ★ Operations
 - ★ **SLA Parameters:** QoS metric definitions.
 - ▶ Obligations:
 - ★ **QoS Guaranties (objectives or constraints).**
 - ★ Actions to take if violations of QoS guarantees are detected (**Reaction**).

A SLA example

SLA	
Parts	Client X Provider Y

Figure: A simple SLA example

A SLA example

SLA	
Parts	Client X Provider Y
Operations	Buy airline tickets
SLA Parameters	Response Time Availability
QoS Metrics	Response Time, <i>ms</i> , real number

Figure: A simple SLA example

A SLA example

SLA	
Parts	Client X Provider Y
Operations	Buy airline tickets
SLA Parameters	Response Time Availability
QoS Metrics	Response Time, <i>ms</i> , real number
Guaranties	Response Time < 10 ms Availability > 90%
Actions to take	Notification Renegotiation

Figure: A simple SLA example

QoS Aggregation

- Process to obtain cumulative value of composition QoS from QoS values of component services.
- There is not a general solution.
- Depends of QoS attributes and composition model.
- Approaches:
 - ▶ Addition, Maximum, Minimum, Average, etc.
 - ▶ Analytics: Petri Nets, Queue Networks, etc.
 - ▶ Heuristics: Genetic Algorithms.
 - ▶ **Simulations.**

QoS Aggregation example

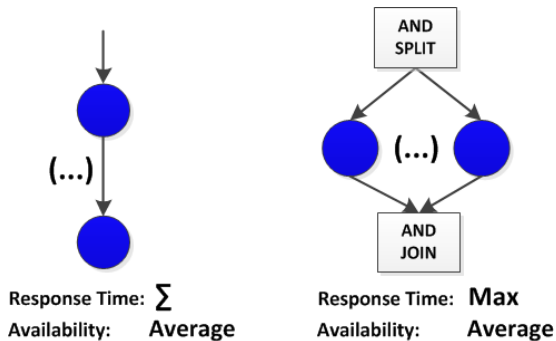


Figure: QoS Aggregation Example

Hard Contracts

- Most agreements are done based on **hard contracts**:

Hard Contracts

- Most agreements are done based on **hard contracts**:
 - ▶ Response time < 10 ms.

Hard Contracts

- Most agreements are done based on **hard contracts**:
 - ▶ Response time < 10 ms.
- Hard contracts don't reflect the dynamic behavior of QoS attributes of Web Services.

Dynamic behavior of QoS Attributes

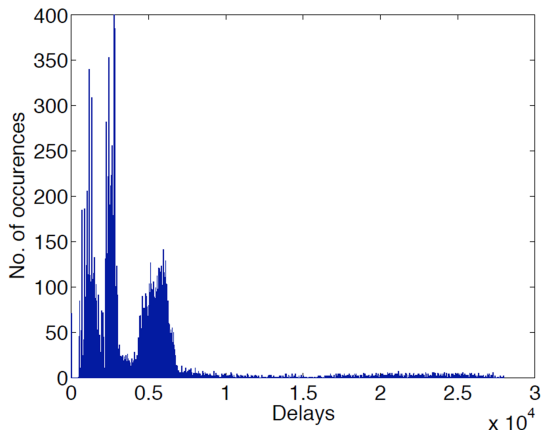


Figure: Measurement records for response times from a 20,000 Web service invocations [Rosario et al., 2008]

Soft Contracts

- Soft Contracts:

- ▶ **Time response < 10 ms, for 95% of the cases.**

Thus, it isn't possible to compose restrictions or contracts.

Soft Contracts

- Soft Contracts:

- ▶ **Time response < 10 ms, for 95% of the cases.**

Thus, it isn't possible to compose restrictions or contracts.

- **Solution:** Soft probabilistic contracts.

- ▶ **For each SLA parameter (response time). A probability distribution is offered as a guarantee of QoS.**

Soft Contracts

- Soft Contracts:

- ▶ **Time response < 10 ms, for 95% of the cases.**

Thus, it isn't possible to compose restrictions or contracts.

- **Solution:** Soft probabilistic contracts.

- ▶ **For each SLA parameter (response time). A probability distribution is offered as a guarantee of QoS.**

- As **probabilistic constraints** can be composed.

- ▶ There are some approaches for orchestration.
- ▶ But, there are no approaches for service Choreographies .
- ▶ Only response time(or time based) are considered .

Monitoring based on QoS

Responsibilities:

- To measure and calculate QoS metric values, that also includes **aggregation** of QoS attributes values.
- To check for violations of some QoS constraints.
- Choreography monitoring should be non-intrusive .

Monitoring Approaches

Intrusive Monitoring: **Instrumentation**

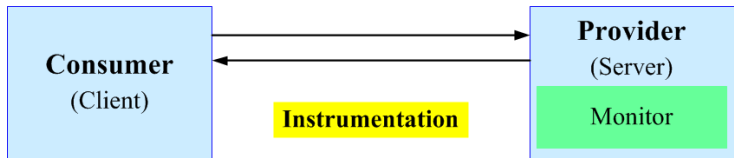


Figure: Intrusive Monitoring

Monitoring Approaches

Non-intrusive Monitoring: **Interception**



Figure: Monitoring by Interception

Monitoring Approaches

Non-intrusive Monitoring: **Probe-Request**

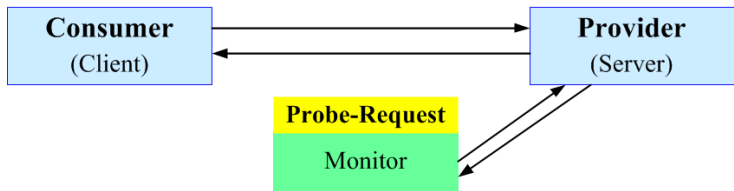


Figure: Monitoring by Probe-Request

Monitoring Approaches

Non-intrusive Monitoring: **Sniffing**

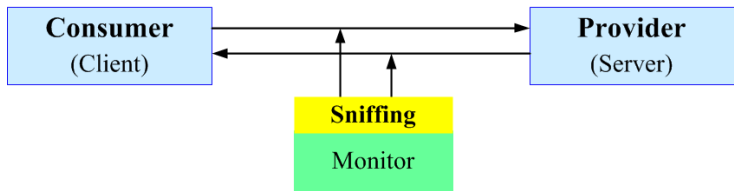


Figure: Monitoring by sniffing techniques

Monitoring Layers

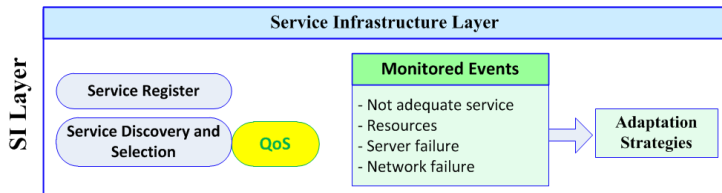


Figure: Monitoring layers

Monitoring Layers

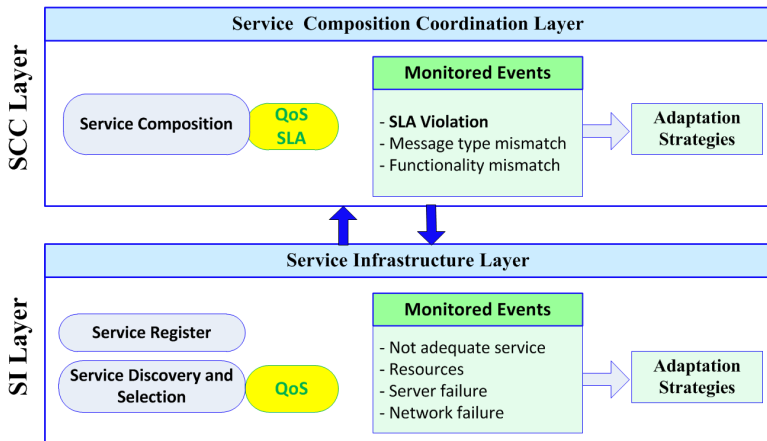
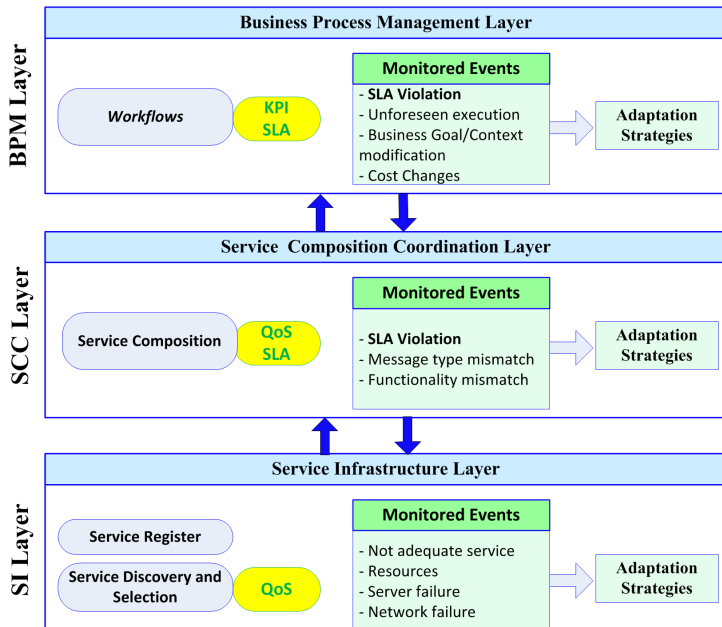
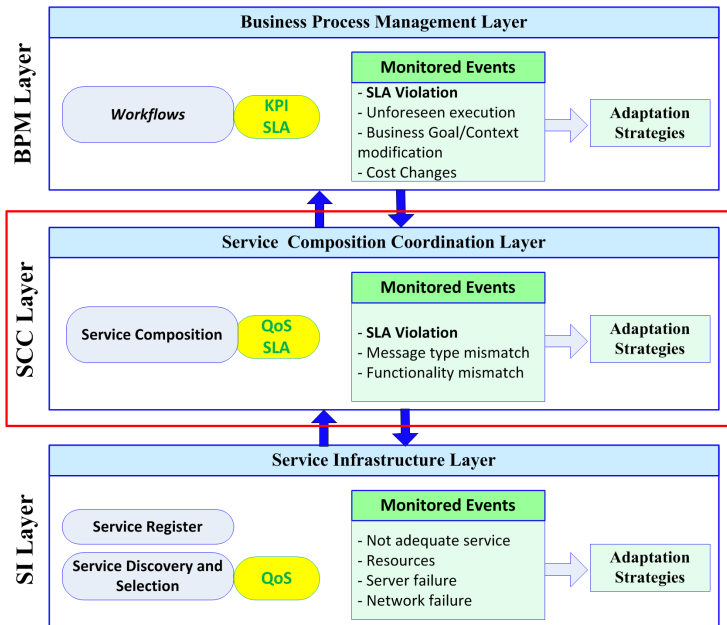


Figure: Monitoring layers

Monitoring Layers



Monitoring Layers



Multilayer QoS on Web service choreography

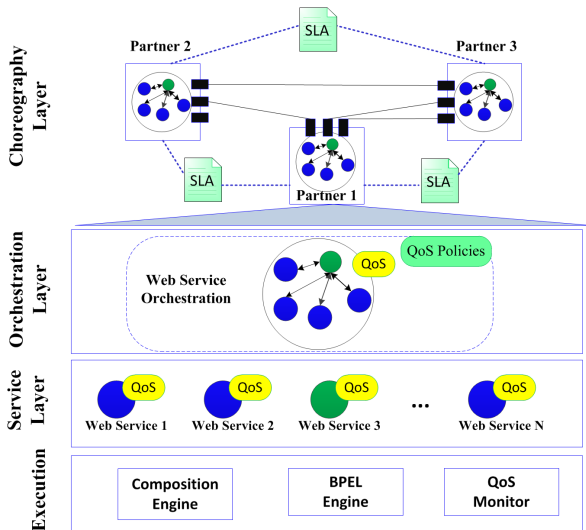


Figure: Integration of multilayer QoS and SLA in choreographies

- 1 Problem
- 2 Objectives
- 3 QoS and Monitoring on Web Service Choreography
- 4 Related Works**
- 5 Contributions
- 6 Purpose

Web Service Choreography Monitoring based on QoS

- (Xiangpeng et al.,2007), (Pandey and Chaudhary, 2008) and (Pandey, 2010)
:
 - ▶ Formal methods for specifying QoS in choreographies.
 - ▶ Focus on specification language .

Web Service Choreography Monitoring based on QoS

- (Xiangpeng et al.,2007), (Pandey and Chaudhary, 2008) and (Pandey, 2010) :
 - ▶ Formal methods for specifying QoS in choreographies.
 - ▶ Focus on specification language .
- (Wetzstein et al., 2010):
 - ▶ Cross-Organizational Process Monitoring.
 - ▶ Focus in KPIs and not in QoS.

Web Service Choreography Monitoring based on QoS

- (Xiangpeng et al., 2007), (Pandey and Chaudhary, 2008) and (Pandey, 2010) :
 - ▶ Formal methods for specifying QoS in choreographies.
 - ▶ **Focus on specification language .**
- (Wetzstein et al., 2010):
 - ▶ Cross-Organizational Process Monitoring.
 - ▶ **Focus in KPIs and not in QoS.**
- (Ul-Haq et al., 2010) :
 - ▶ Hierarchical aggregations of SLAs on service choreography.
 - ▶ Framework based on rules.
 - ▶ **Focus business process and KPIs.**

Web Service Choreography Monitoring based on QoS

- (Xiangpeng et al., 2007), (Pandey and Chaudhary, 2008) and (Pandey, 2010) :
 - ▶ Formal methods for specifying QoS in choreographies.
 - ▶ **Focus on specification language .**
- (Wetzstein et al., 2010):
 - ▶ Cross-Organizational Process Monitoring.
 - ▶ **Focus in KPIs and not in QoS.**
- (Ul-Haq et al., 2010) :
 - ▶ Hierarchical aggregations of SLAs on service choreography.
 - ▶ Framework based on rules.
 - ▶ **Focus business process and KPIs.**
- (Xia et al., 2009) :
 - ▶ Analytic QoS prediction for choreographies specified using **WSCl**.
 - ▶ **Focus in specification language.**

Web Service Choreography Monitoring based on QoS

- (Xiangpeng et al., 2007), (Pandey and Chaudhary, 2008) and (Pandey, 2010) :
 - ▶ Formal methods for specifying QoS in choreographies.
 - ▶ **Focus on specification language .**
- (Wetzstein et al., 2010):
 - ▶ Cross-Organizational Process Monitoring.
 - ▶ **Focus in KPIs and not in QoS.**
- (Ul-Haq et al., 2010) :
 - ▶ Hierarchical aggregations of SLAs on service choreography.
 - ▶ Framework based on rules.
 - ▶ **Focus business process and KPIs.**
- (Xia et al., 2009) :
 - ▶ Analytic QoS prediction for choreographies specified using **WSCl**.
 - ▶ **Focus in specification language.**
- (Rosenberg, 2009) :
 - ▶ Multilayer model of QoS for Web service choreography.
 - ▶ **Without techniques for SLA establishment .**
 - ▶ **Focus in hard constraints.**

Service Monitoring using probabilistic SLAs

- (Rosario et al., 2008) e (Rosario et al., 2009)
 - ▶ Monte-Carlo method in order to define SLAs on orchestrations.
 - ▶ Focus in response time.
 - ▶ Monitoring of probabilistic SLAs on service orchestrations using Orc.

Service Monitoring using probabilistic SLAs

- (Rosario et al., 2008) e (Rosario et al., 2009)
 - ▶ Monte-Carlo method in order to define SLAs on orchestrations.
 - ▶ Focus in response time.
 - ▶ Monitoring of probabilistic SLAs on service orchestrations using Orc.
- (Zheng et al., 2010):
 - ▶ Probability Density Function (PDF) of QoS.
 - ▶ Demonstrations by means of simulations.
 - ▶ Focus in response time and don't consider SLAs.

- 1 Problem
- 2 Objectives
- 3 QoS and Monitoring on Web Service Choreography
- 4 Related Works
- 5 Contributions**
- 6 Purpose

Contributions

- An architecture for **SLA violations detection** on Web Service Choreography.

Contributions

- An architecture for **SLA violations detection** on Web Service Choreography.
- **Establishment of probabilistic SLAs** for Web service choreography.

Contributions

- An architecture for **SLA violations detection** on Web Service Choreography.
- **Establishment of probabilistic SLAs** for Web service choreography.
 - ▶ Probabilistic QoS aggregation.

Contributions

- An architecture for **SLA violations detection** on Web Service Choreography.
- **Establishment of probabilistic SLAs** for Web service choreography.
 - ▶ Probabilistic QoS aggregation.
 - ▶ QoS constraints estimations using Monte-Carlo simulations.

Contributions

- An architecture for **SLA violations detection** on Web Service Choreography.
- **Establishment of probabilistic SLAs** for Web service choreography.
 - ▶ Probabilistic QoS aggregation.
 - ▶ QoS constraints estimations using Monte-Carlo simulations.
- **Monitoring** for Web service choreography based on QoS.

Contributions

- An architecture for **SLA violations detection** on Web Service Choreography.
- **Establishment of probabilistic SLAs** for Web service choreography.
 - ▶ Probabilistic QoS aggregation.
 - ▶ QoS constraints estimations using Monte-Carlo simulations.
- **Monitoring** for Web service choreography based on QoS.
 - ▶ QoS metric measures and probabilistic aggregation.

Contributions

- An architecture for **SLA violations detection** on Web Service Choreography.
- **Establishment of probabilistic SLAs** for Web service choreography.
 - ▶ Probabilistic QoS aggregation.
 - ▶ QoS constraints estimations using Monte-Carlo simulations.
- **Monitoring** for Web service choreography based on QoS.
 - ▶ QoS metric measures and probabilistic aggregation.
 - ▶ Comparison of distribution Quantiles established in SLA with estimated quantiles by monitor.

Contributions

- An architecture for **SLA violations detection** on Web Service Choreography.
- **Establishment of probabilistic SLAs** for Web service choreography.
 - ▶ Probabilistic QoS aggregation.
 - ▶ QoS constraints estimations using Monte-Carlo simulations.
- **Monitoring** for Web service choreography based on QoS.
 - ▶ QoS metric measures and probabilistic aggregation.
 - ▶ Comparison of distribution Quantiles established in SLA with estimated quantiles by monitor.
- **QoS attributes**: response time, network latency and bandwidth (passive monitoring).

- 1 Problem
- 2 Objectives
- 3 QoS and Monitoring on Web Service Choreography
- 4 Related Works
- 5 Contributions
- 6 Purpose**

Purpose

- **SLA establishment** in a probabilistic way for Web service choreography.

Purpose

- **SLA establishment** in a probabilistic way for Web service choreography.
- **Non-intrusive monitoring** based on events for Web service choreography..

Purpose

- **SLA establishment** in a probabilistic way for Web service choreography.
- **Non-intrusive monitoring** based on events for Web service choreography..
- **Violations detection of probabilistic SLAs.**

Purpose

- **SLA establishment** in a probabilistic way for Web service choreography.
- **Non-intrusive monitoring** based on events for Web service choreography..
- **Violations detection of probabilistic SLAs.**
- **QoS attributes:** response time, network latency and bandwidth.

System Architecture for SLA violations detection

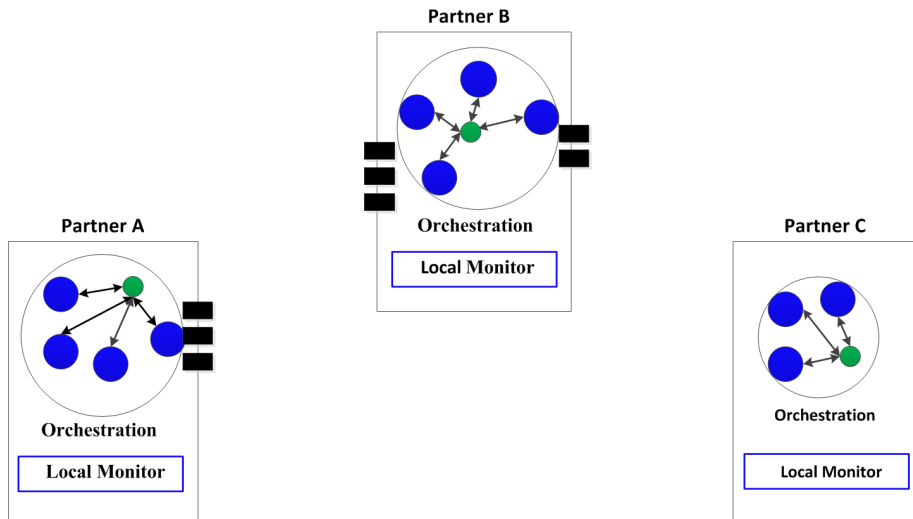


Figure: Overview of proposed monitoring

System Architecture for SLA violations detection

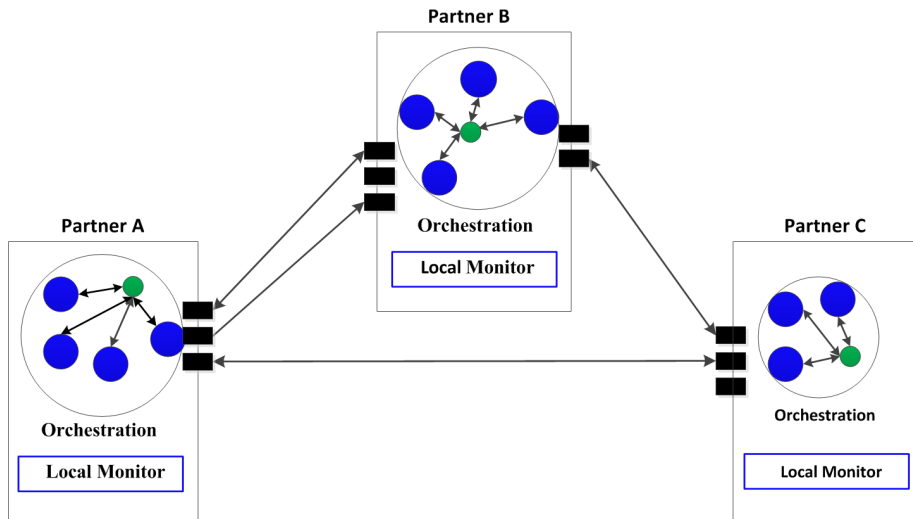


Figure: Overview of proposed monitoring

System Architecture for SLA violations detection

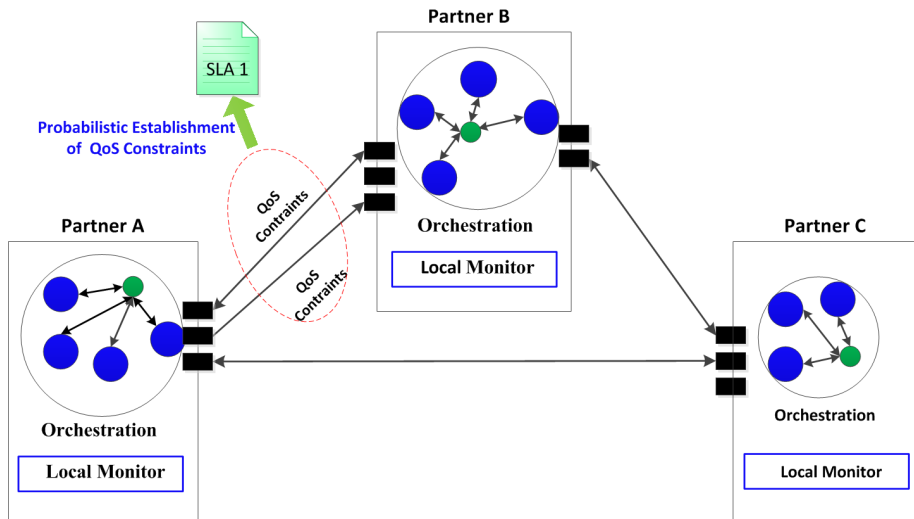


Figure: Overview of proposed monitoring

System Architecture for SLA violations detection

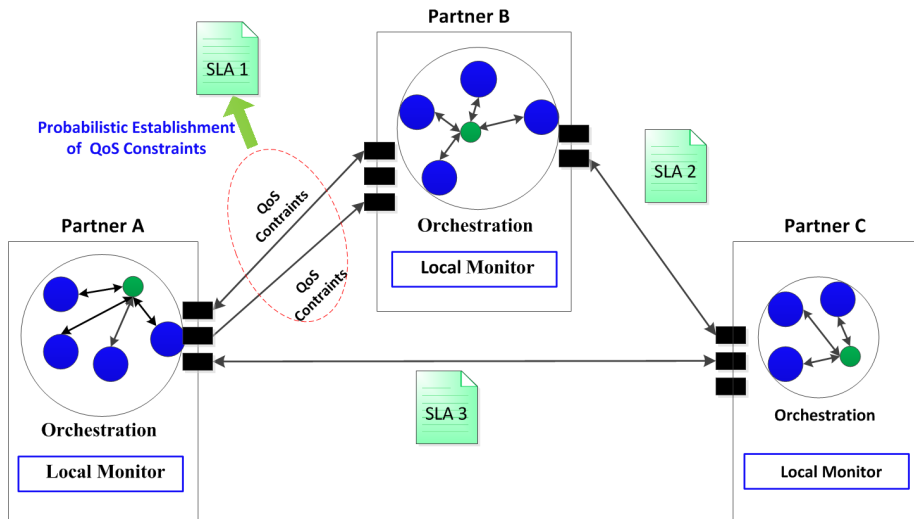
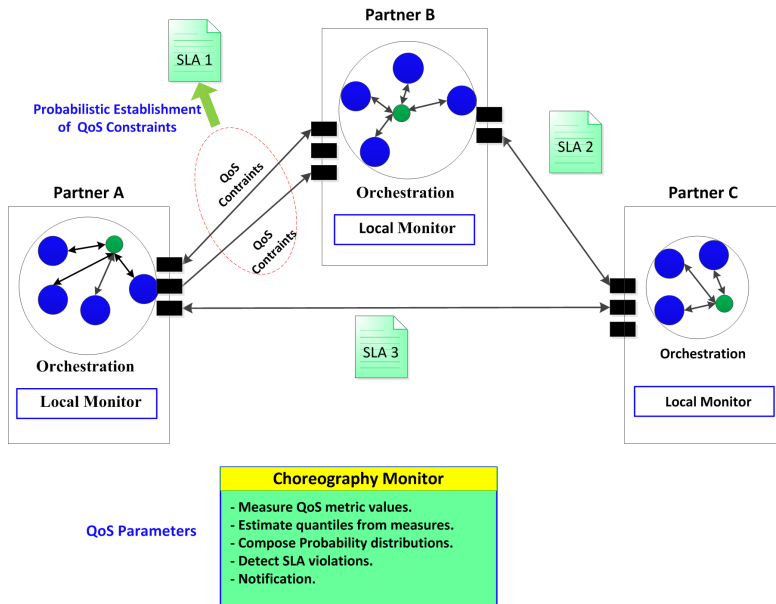
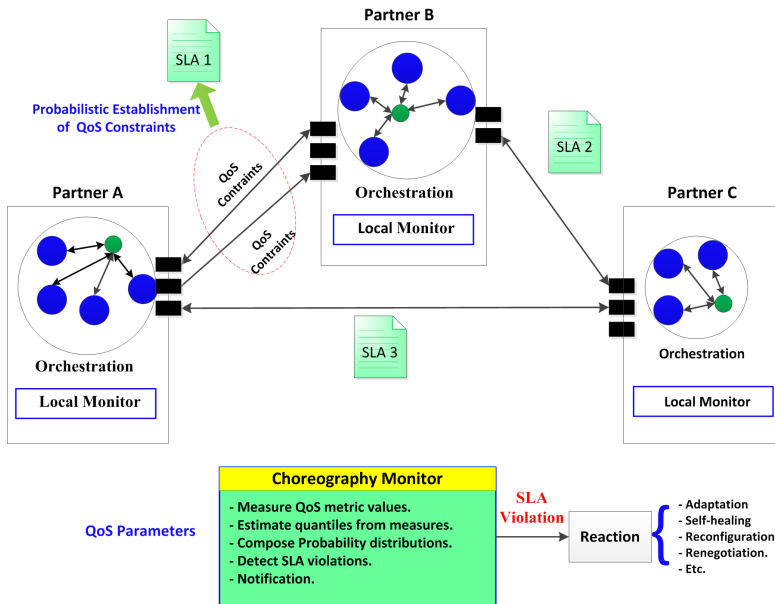


Figure: Overview of proposed monitoring

System Architecture for SLA violations detection



System Architecture for SLA violations detection



Establishment of probabilistic SLA

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

is a cumulative distribution function (cdf) of a QoS parameter δ of service S .

Establishment of probabilistic SLA

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

is a cumulative distribution function (cdf) of a QoS parameter δ of service S .

① Initial Conditions:

- ▶ Estimate the cumulative distribution function F_{S_i} for each service S_i .
- ▶ Define quantiles for each service F_{S_i} .

Establishment of probabilistic SLA

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

is a cumulative distribution function (cdf) of a QoS parameter δ of service S .

① Initial Conditions:

- ▶ Estimate the cumulative distribution function F_{S_i} for each service S_i .
- ▶ Define quantiles for each service F_{S_i} .

② Monte-Carlo simulations:

- ① For each service invocation of s_i at interaction between a provider partner A with a consumer partner B, draw a random value of QoS parameter q from $F_{S_i}(x)$.

Establishment of probabilistic SLA

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

is a cumulative distribution function (cdf) of a QoS parameter δ of service S .

① Initial Conditions:

- ▶ Estimate the cumulative distribution function F_{S_i} for each service S_i .
- ▶ Define quantiles for each service F_{S_i} .

② Monte-Carlo simulations:

- ① For each service invocation of s_i at interaction between a provider partner A with a consumer partner B, draw a random value of QoS parameter q from $F_{S_i}(x)$.
- ② **Aggregation:** Estimate the composition QoS (choreography) from values obtained in step a.

Establishment of probabilistic SLA

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

is a cumulative distribution function (cdf) of a QoS parameter δ of service S .

① Initial Conditions:

- ▶ Estimate the cumulative distribution function F_{S_i} for each service S_i .
- ▶ Define quantiles for each service F_{S_i} .

② Monte-Carlo simulations:

- ① For each service invocation of s_i at interaction between a provider partner A with a consumer partner B, draw a random value of QoS parameter q from $F_{S_i}(x)$.
- ② **Aggregation:** Estimate the composition QoS (choreography) from values obtained in step a.
- ③ Performing Monte-Carlo simulation by means of repeating step 2.1 and 2.2 many times in order to empirical estimate F_Q .

Establishment of probabilistic SLA

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

is a cumulative distribution function (cdf) of a QoS parameter δ of service S .

① Initial Conditions:

- ▶ Estimate the cumulative distribution function F_{S_i} for each service S_i .
- ▶ Define quantiles for each service F_{S_i} .

② Monte-Carlo simulations:

- ① For each service invocation of s_i at interaction between a provider partner A with a consumer partner B, draw a random value of QoS parameter q from $F_{S_i}(x)$.
- ② **Aggregation:** Estimate the composition QoS (choreography) from values obtained in step a.
- ③ Performing Monte-Carlo simulation by means of repeating step 2.1 and 2.2 many times in order to empirical estimate F_Q .

- ③ From F_Q it is possible to select adequate quantiles for establishing a QoS constraint (QoS guarantee) in a SLA.

Probabilistic Monitoring of Choreographies

- F_s : Agreed probability distribution in SLA.
- Δ : Finite set of samples from measured QoS parameters of service S .
- f'_s : Empirical distribution of a service.
- F'_s : Probability distribution after QoS aggregation.

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

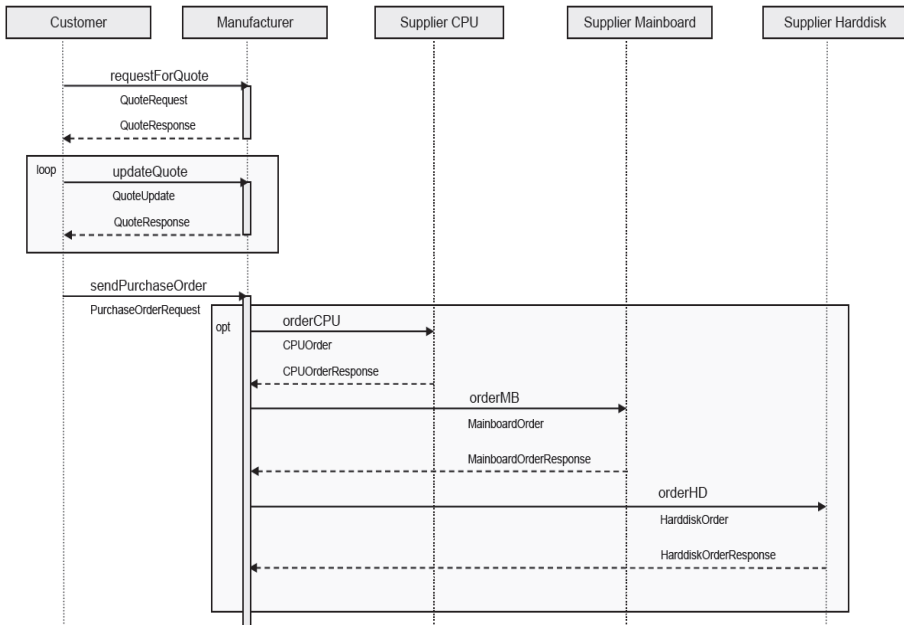
$$\exists x \in R^+ : F'_{s,\Delta}(x) < F_s(x) \quad (2)$$

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

Implementation

Aspect	Tools and Frameworks
Choreography	WS-CDL, WS-SLA
Orchestration	WS-BPEL, WS-Policy
Web service	WSDL, QoS metrics
BPEL engine	Petals ESB, Apache ODE
Monitoring foundation	Easier BSM
Event engine	WildCAT/ESPER of Galaxy

Case study



Development of service choreographies

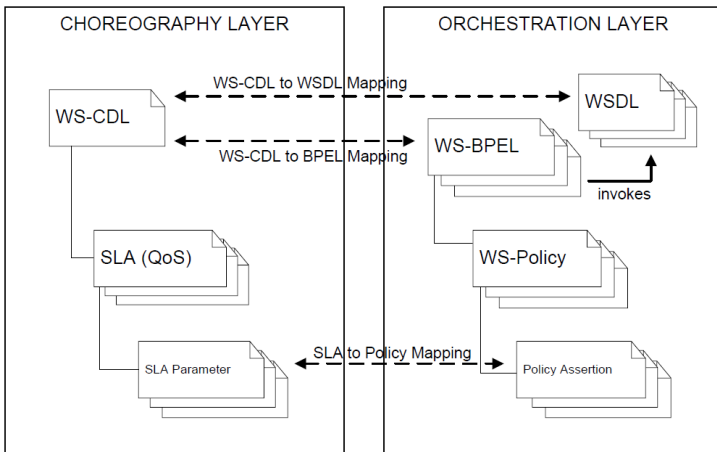


Figure: Development methodology of service choreographies [Rosenberg et al., 2007]

Involvement



Thanks!