

Congreso Internacional de **DESARROLLO DE SOFTWARE**

**Monitorización de calidad de servicios cloud
mediante modelos en tiempo de ejecución**

Ing. Priscila Cedillo O. PhD.
priscila.cedillo@ucuenca.edu.ec



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Contenidos



1. Introducción
2. Método Cloud MoS@RT
3. Infraestructura de Monitorización
4. Instanciación de la Solución
5. Conclusiones

Contenidos



1. **Introducción**
2. Método Cloud MoS@RT
3. Infraestructura de Monitorización
4. Instanciación de la Solución
5. Conclusiones

Cloud Computing

Introducción

Estado del Arte

Método Cloud MoS@RT

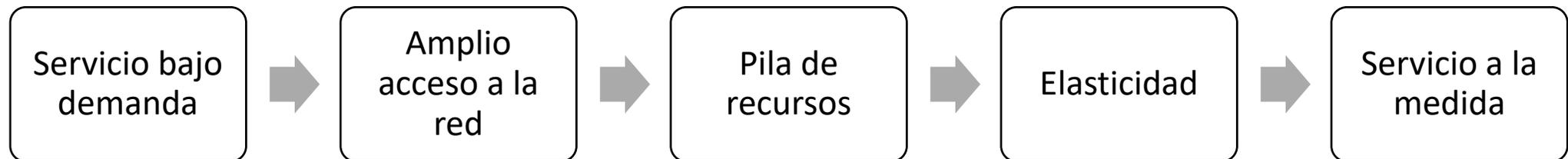
Infraestructura de Monitorización

Instanciación de la Solución

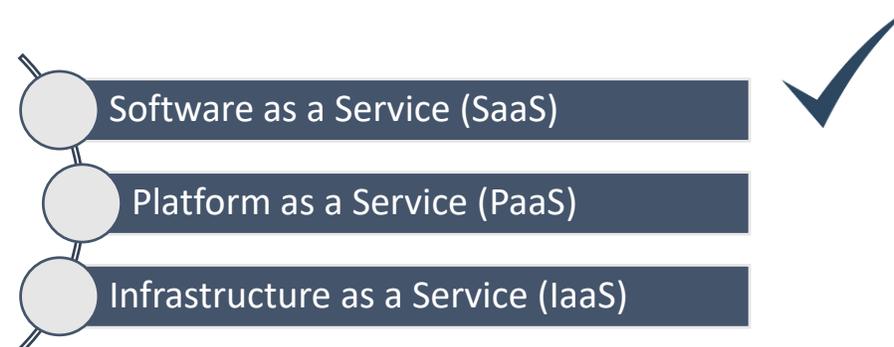
Conclusiones y Trabajo Futuro

Cloud computing es un modelo para habilitar el acceso ubicuo, conveniente, bajo demanda a la red con una pila de recursos configurables de cómputo (e.g., redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser **rápidamente provistos** y revocados **con un mínimo esfuerzo** por parte del proveedor (**NIST, 2011**)

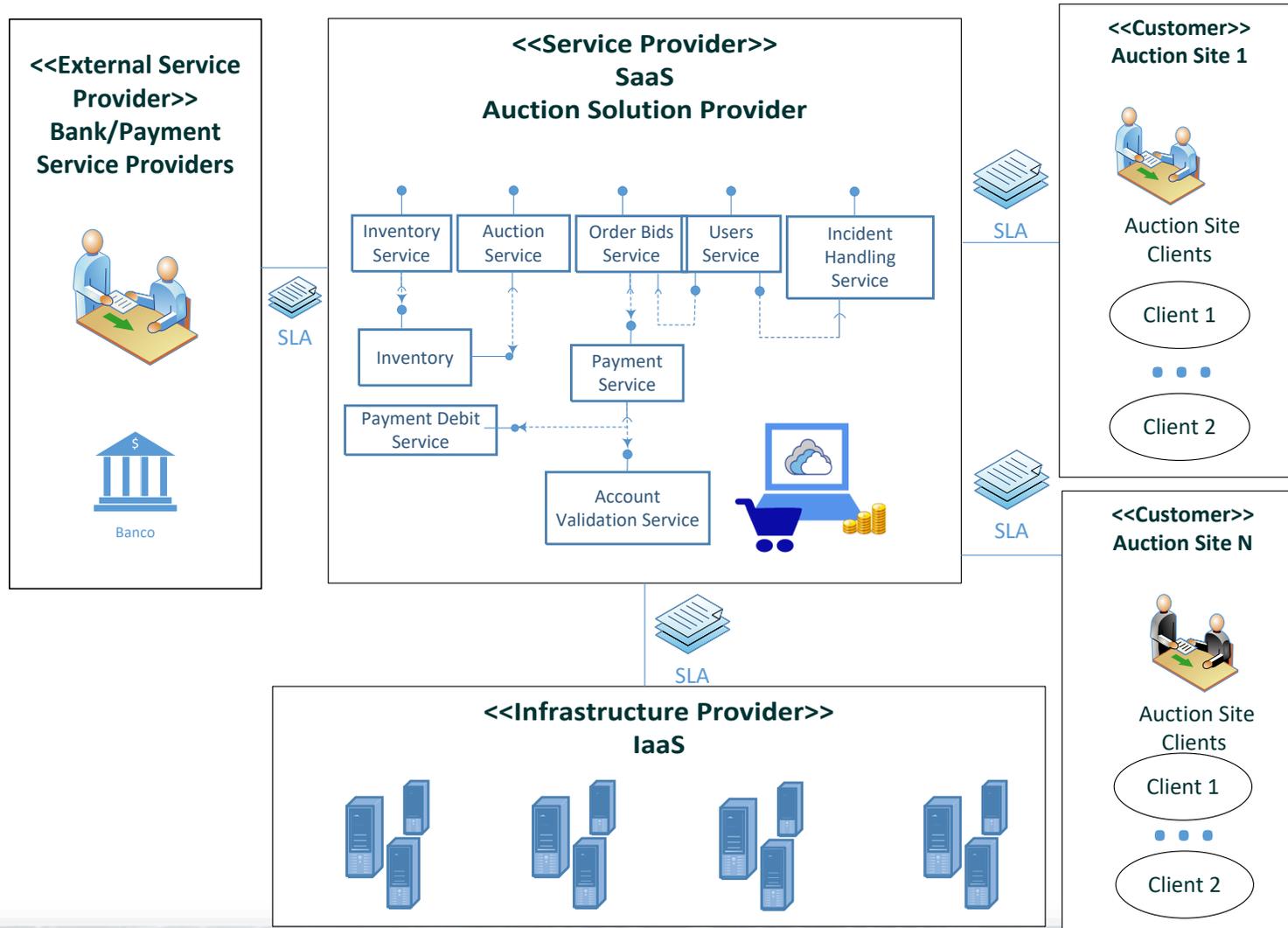
Características esenciales



Modelos de servicio



Software as a Service (SaaS), Monitoreo y QoS



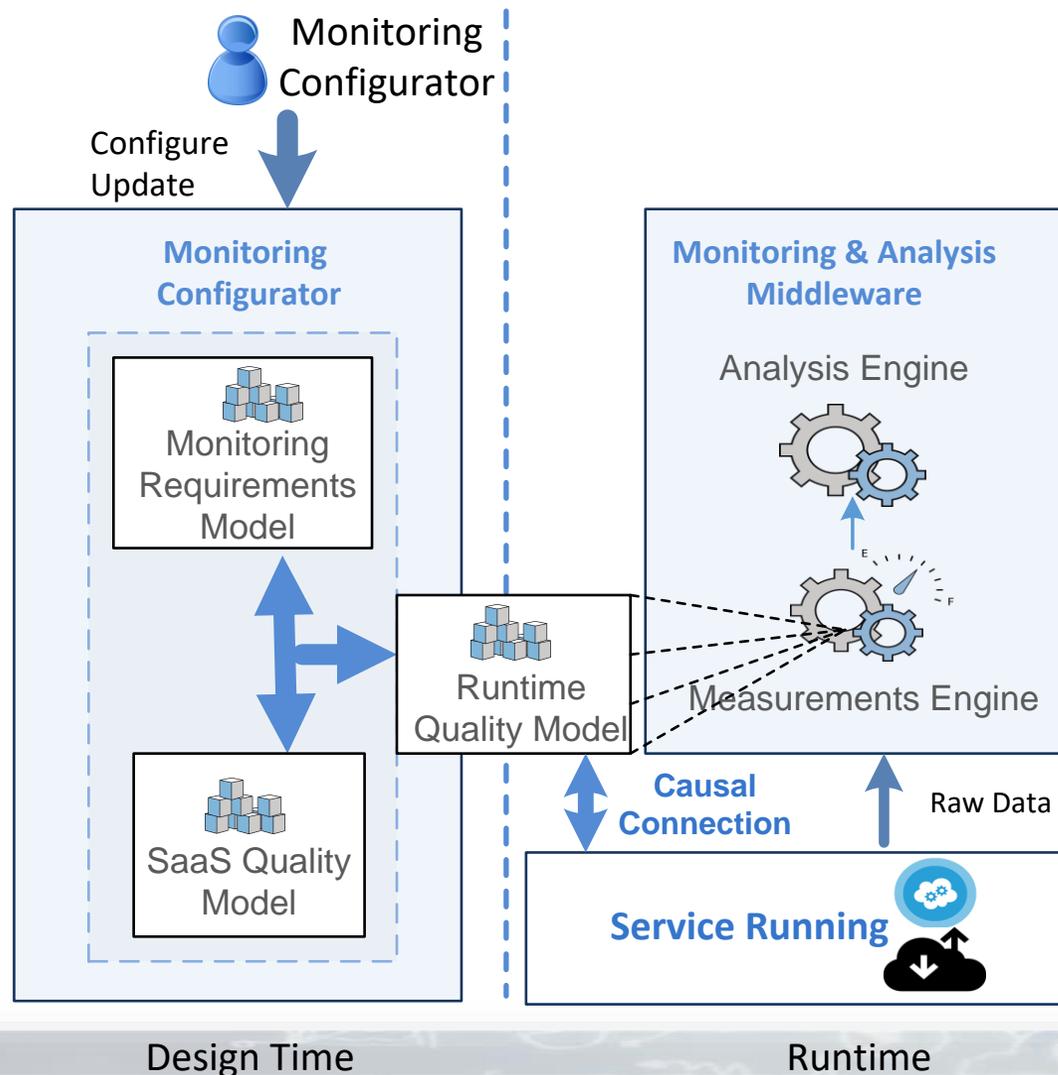
¿Problemas? ¿Limitaciones?

Limitaciones de los enfoques actuales

- Los métodos de monitorización están muy **acoplados** a la plataforma con una especificación dada del SLA y se centran mayoritariamente a la medición de atributos de calidad de muy bajo nivel.
- Las infraestructuras de monitorización presentan una **falta de flexibilidad** para cumplir los requisitos de monitoreo
- **SaaS** presenta **nuevos desafíos** (e.g., elasticidad, escalabilidad, multi-tenencia, etc.)
- Verificación del **SLA** son difíciles de **automatizar**.
- **Dependencia de la plataforma** herramientas de monitoreo.
- No se consideran **estándares** o lenguajes de dominio específico



Solución propuesta



La investigación en Ingeniería del Software debería centrarse en proveer soporte inteligente del software en tiempo de ejecución, **rompiendo** la barrera **rígida** entre el **tiempo de desarrollo** y el **tiempo de ejecución** (Baresi *et al.*, 2010)

Un modelo en tiempo de ejecución es una **abstracción** de un sistema ejecutándose que está siendo manipulado en **tiempo de ejecución** para un propósito específico.

En particular, el modelo refleja un **fenómeno en ejecución** en un ambiente dado (Bencomo *et al.*, 2013).

Objetivo General

Introduction

Estado del Arte

Método Cloud MoS@RT

Infraestructura de Monitorización

Instanciación de la Solución

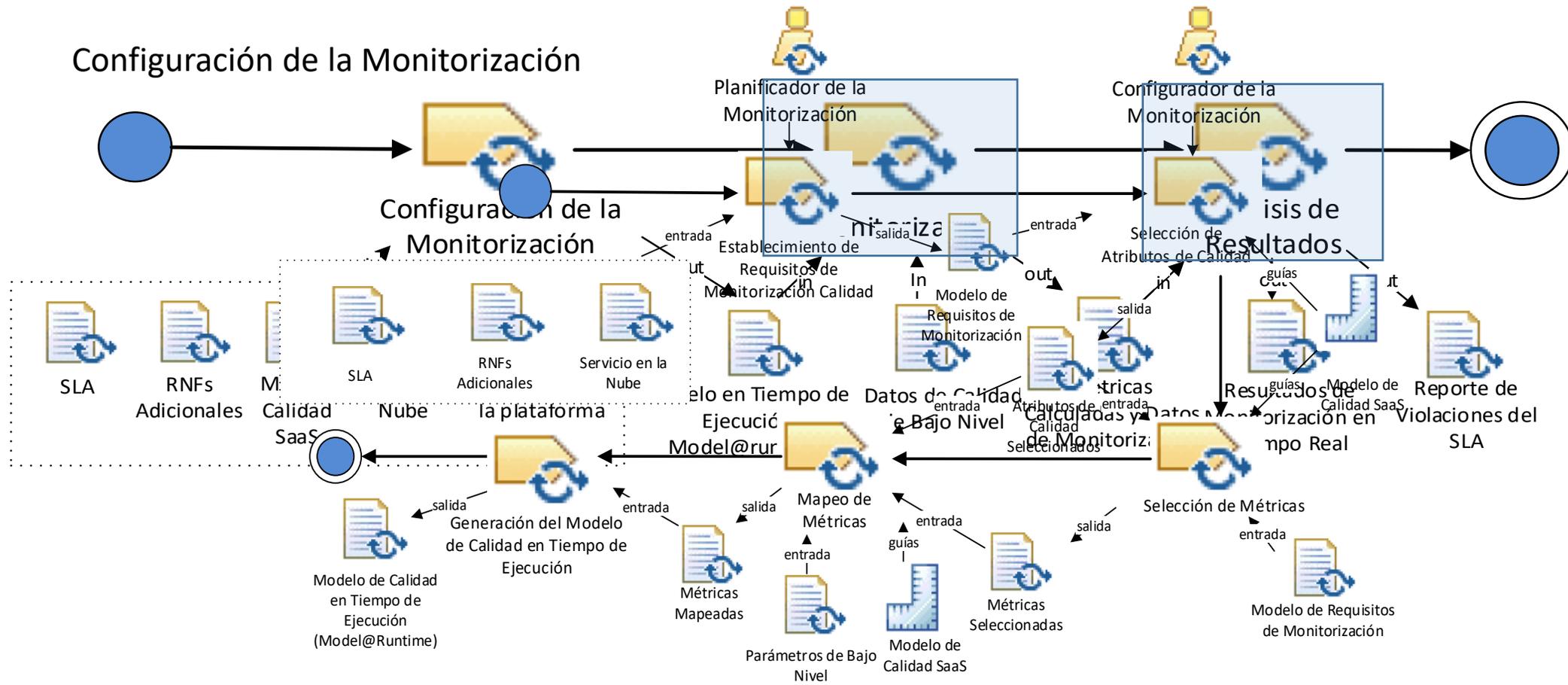
Conclusiones y Trabajo Futuro

- *El objetivo de este trabajo es la definición y evaluación empírica de un método con el cual se realizará el monitoreo de servicios desplegados en SaaS utilizando modelos en tiempo de ejecución*

Contenidos



1. Introduction
2. Método Cloud MoS@RT
3. Infraestructura de Monitorización
4. Instanciación de la Solución
5. Conclusiones y Trabajo Futuro



El método en definitiva ofrece:

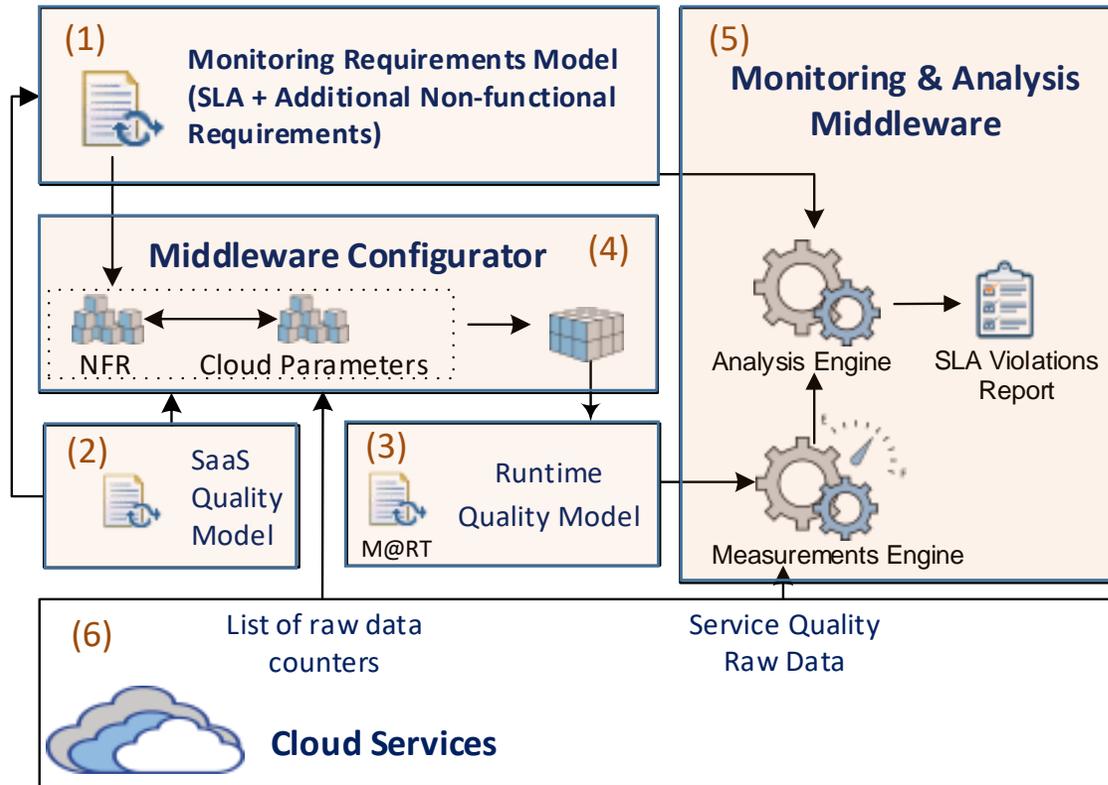
- Un proceso **sistemático** con un conjunto de actividades
- **Separación** entre las diferentes abstracciones (requisitos de calidad, atributos de calidad, parámetros específicos de la plataforma cloud, etc.)
- **Soporte** para la especificación de requisitos de monitorización
- **Flexibilidad** para agregar o modificar los parámetros de monitorización (requisitos de monitorización, función de medición, etc.)
- **Independencia** de la plataforma cloud

Contenidos



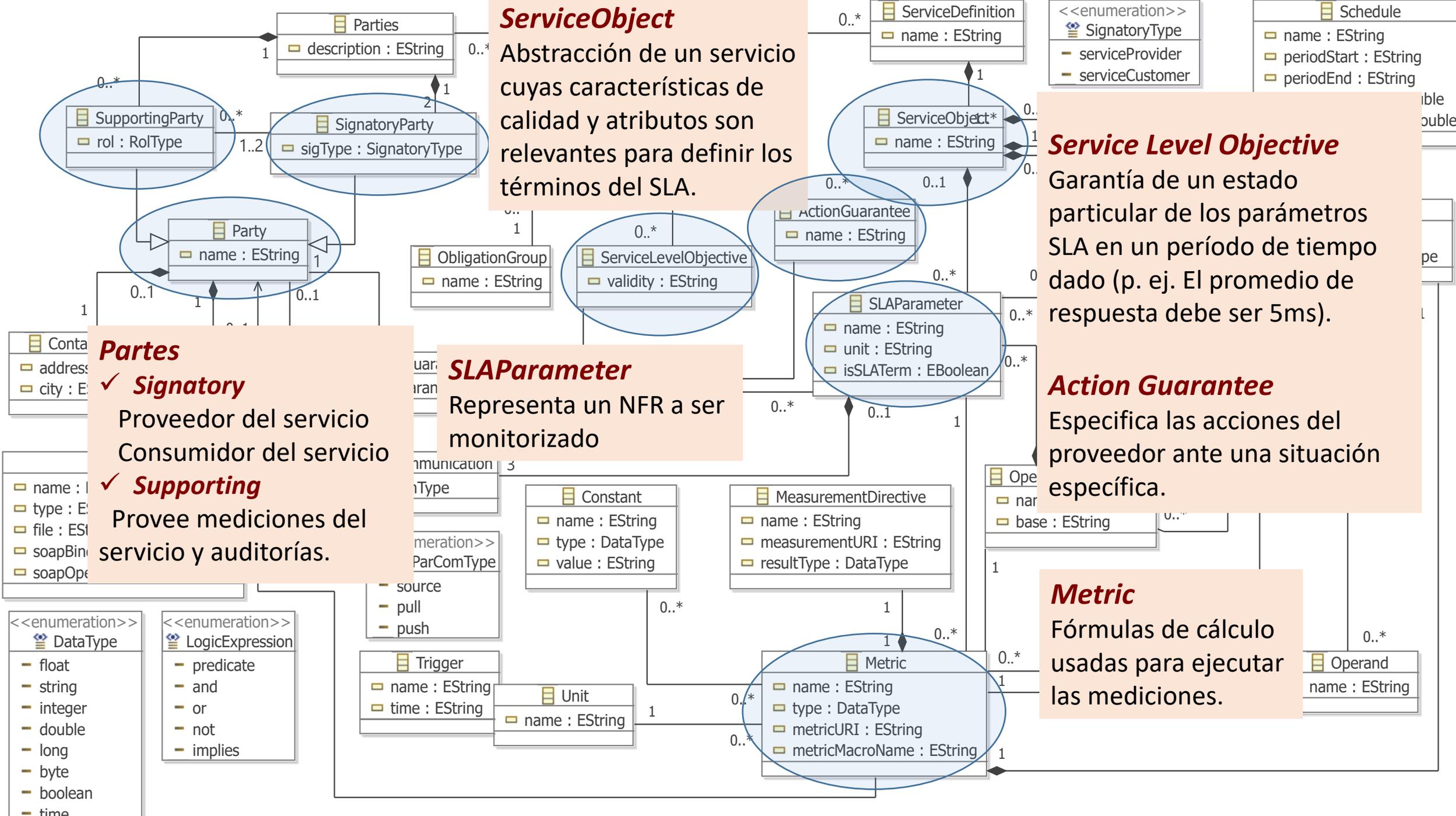
1. Introduction
2. Estado del Arte
3. Cloud MoS@RT
- 4. Infraestructura de Monitorización**
5. Instanciación de la Solución
6. Conclusiones y Trabajo Futuro

Arquitectura del Middleware Independiente de la Plataforma



- (2) Modelo de Calidad SaaS en tiempo real:** Utiliza tres modelos de calidad en tiempo real que se actualizan con los valores de los indicadores de calidad de los servicios de Monitorización de Recursos y Requisitos de Monitorización.
- **Modelo de Calidad SaaS** a los valores
 - **Alternativas de operación** de los
 - **Modelo de Calidad en Tiempo de Ejecución** de las fórmulas dependientes e independientes de la plataforma que son contenidas en el **Modelo de Calidad SaaS**.

Modelo de Requisitos de Monitorización



ServiceObject
 Abstracción de un servicio cuyas características de calidad y atributos son relevantes para definir los términos del SLA.

Service Level Objective
 Garantía de un estado particular de los parámetros SLA en un período de tiempo dado (p. ej. El promedio de respuesta debe ser 5ms).

Action Guarantee
 Especifica las acciones del proveedor ante una situación específica.

Metric
 Fórmulas de cálculo usadas para ejecutar las mediciones.

Partes
 ✓ **Signatory**
 Proveedor del servicio
 Consumidor del servicio
 ✓ **Supporting**
 Provee mediciones del servicio y auditorías.

SLAParameter
 Representa un NFR a ser monitorizado

Contato
 address
 city : EString

<<enumeration>>
 DataType
 - float
 - string
 - integer
 - double
 - long
 - byte
 - boolean
 - time

<<enumeration>>
 LogicExpression
 - predicate
 - and
 - or
 - not
 - implies

source
 pull
 push

Trigger
 name : EString
 time : EString

Unit
 name : EString

Constant
 name : EString
 type : DataType
 value : EString

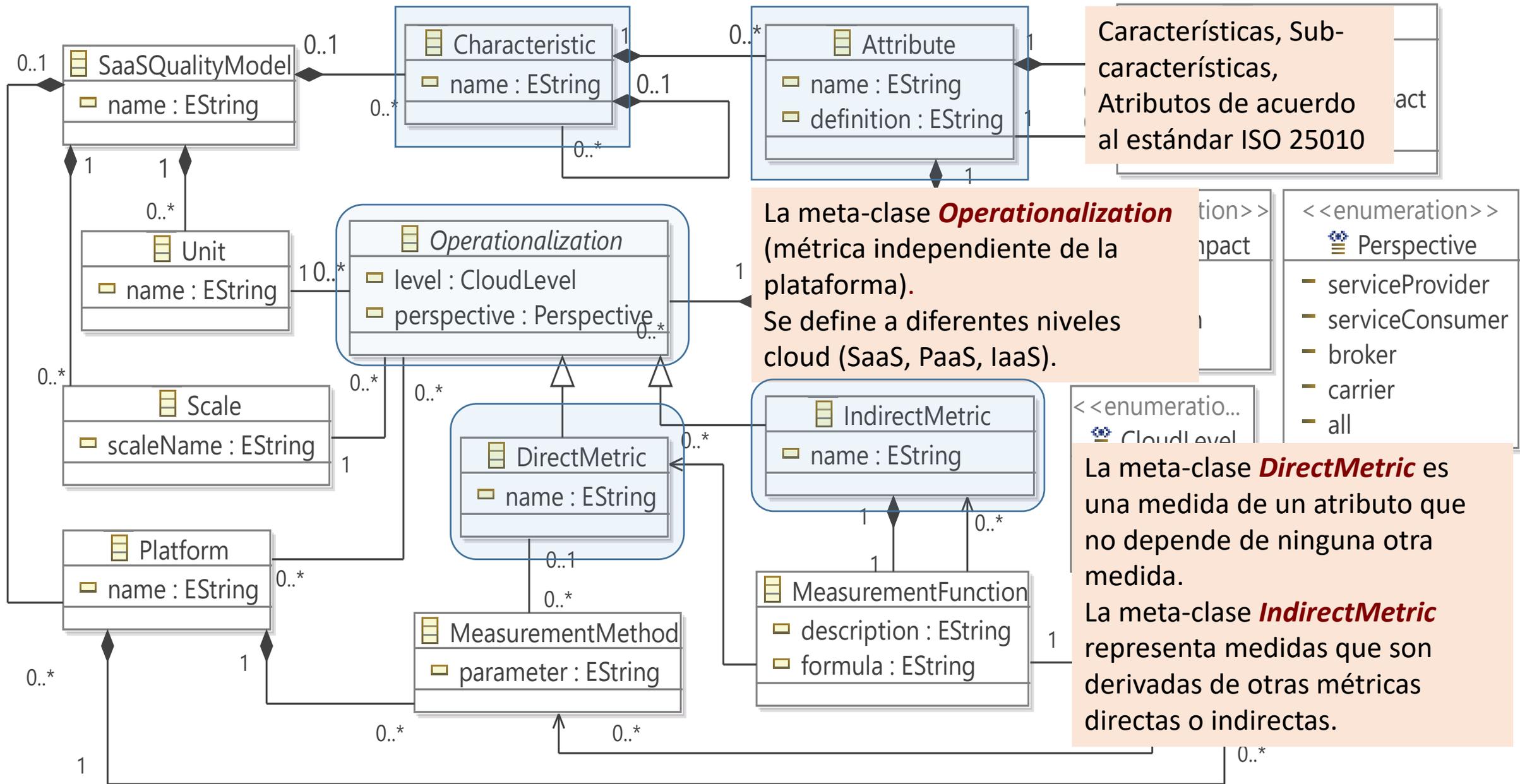
MeasurementDirective
 name : EString
 measurementURI : EString
 resultType : DataType

Metric
 name : EString
 type : DataType
 metricURI : EString
 metricMacroName : EString

Operand
 name : EString

Operand
 name : EString

Modelo de Calidad SaaS



Modelo de Calidad en Tiempo de Ejecución

Especifica los requisitos, métricas, operacionalizaciones y configuraciones usadas durante la monitorización

Lehmann puntualiza que un modelo en tiempo de ejecución debe estar conformado por:

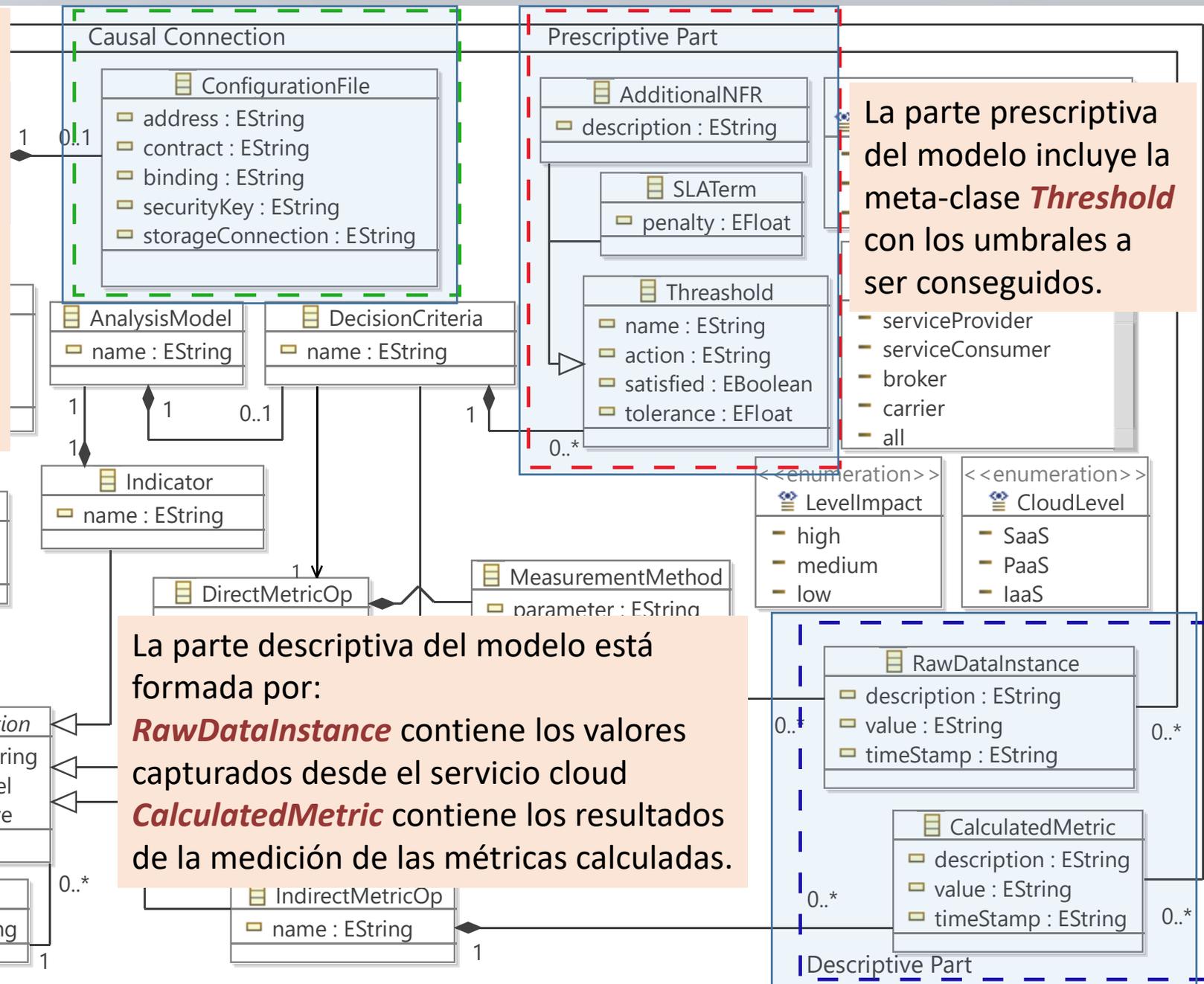
- Una parte **prescriptiva**
- Una parte **descriptiva**
- **Modificaciones** en tiempo de ejecución de la parte **descriptiva** del modelo
- **Modificaciones** en tiempo de ejecución de la parte **prescriptiva** del modelo
- **Conexión causal**

Beneficios:

- Flexibilidad
- Mantenibilidad
- Cualquier modificación en el modelo es fácilmente **reflejada** en la infraestructura.

CloudService describe el servicio a ser monitorizado.

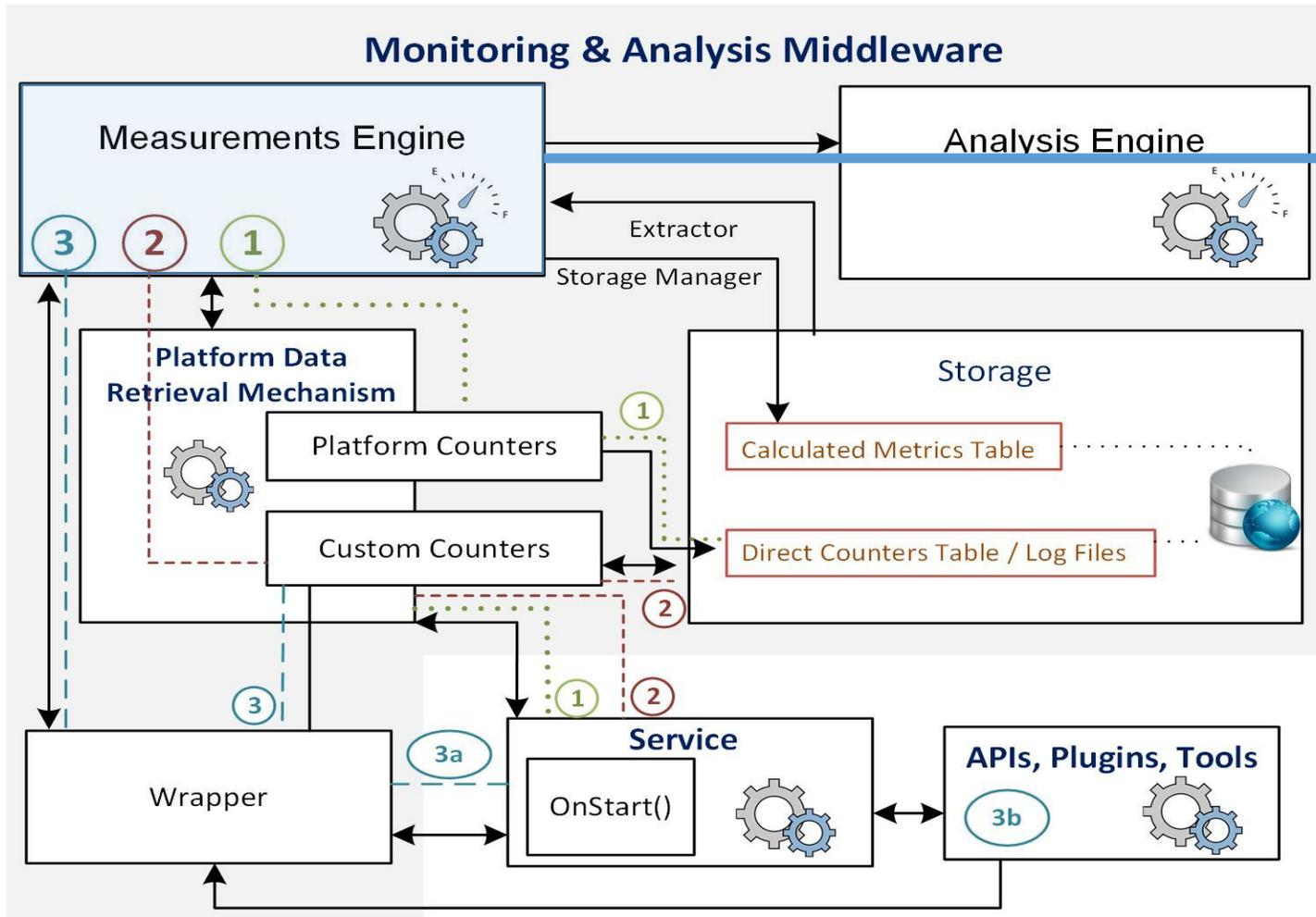
ConfigurationFile contiene información de cada plataforma. Permite la interacción entre la infraestructura de monitorización y el servicio cloud. Consigue la conexión causal entre la infraestructura de monitorización y el servicio cloud.



La parte prescriptiva del modelo incluye la meta-clase **Threshold** con los umbrales a ser conseguidos.

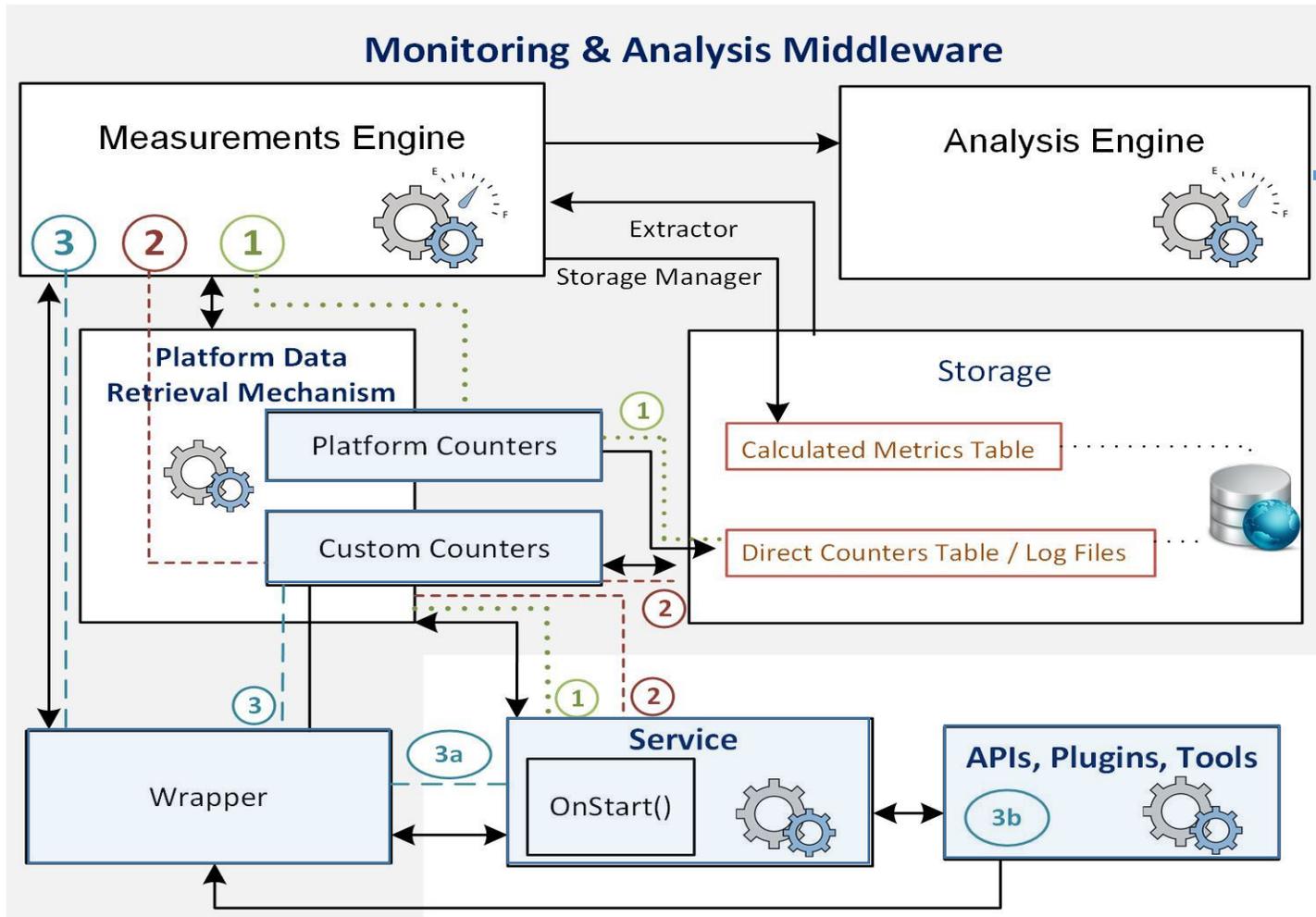
La parte descriptiva del modelo está formada por:
RawDataInstance contiene los valores capturados desde el servicio cloud
CalculatedMetric contiene los resultados de la medición de las métricas calculadas.

Middleware de Monitorización y Análisis Independiente de la Plataforma



1. Recibe el modelo en tiempo de ejecución y **configura el servicio** para su monitorización.
2. **Extrae los datos** de monitorización utilizando los diferentes escenarios de extracción.
3. **Realiza los cálculos** descritos por las operacionalizaciones de las métricas contenidas en el modelo en tiempo de ejecución.
4. **Almacena los resultados** de las métricas en un sistema de almacenamiento permanente para su posterior consulta y/o análisis.

Middleware de Monitorización y Análisis Independiente de la Plataforma



Toma los datos calculados por el motor de medición y **compara** los resultados obtenidos con los establecidos en el **modelo de requisitos de monitorización** (SLA+NFR adicionales).

Escenarios de recolección de datos:

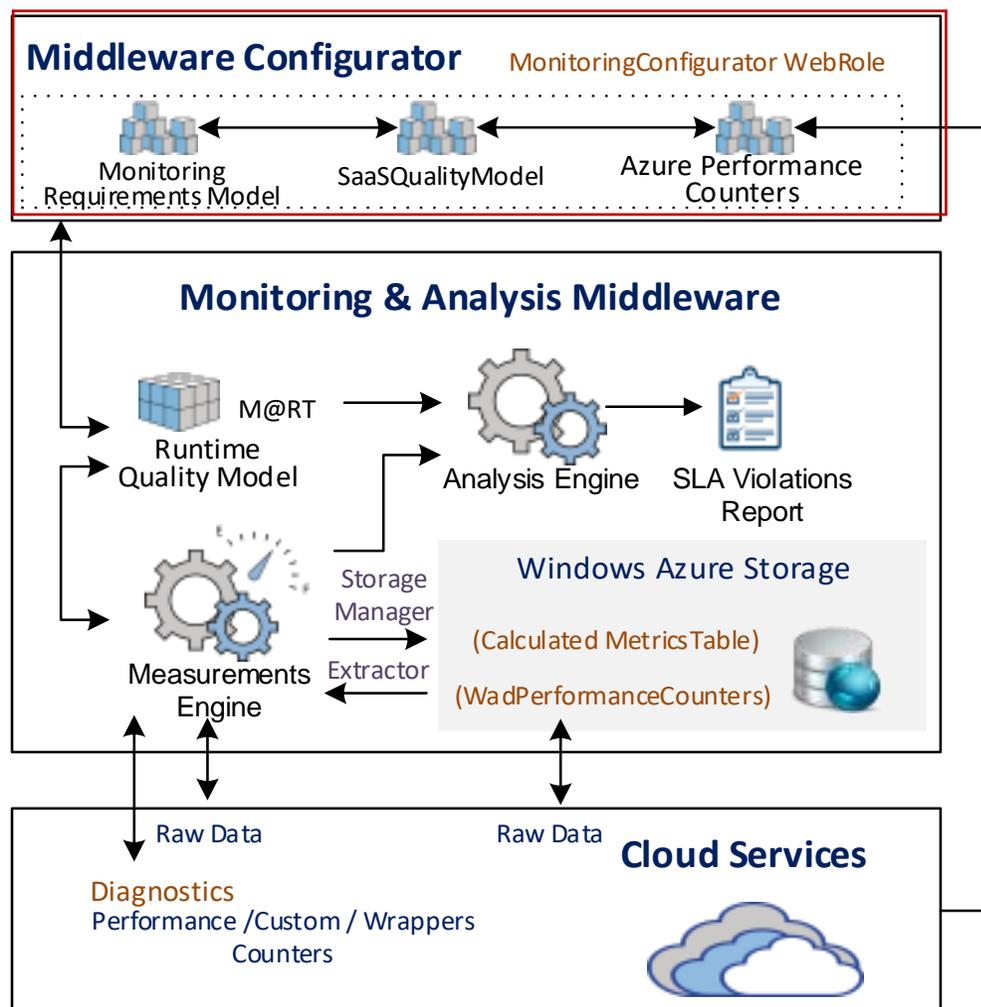
1. Métricas Directas – Contadores Plataforma
2. Métricas Indirectas – Contadores Personalizados
3. (a) Wrappers-envoltorios
(b) Soluciones provistas por terceros

Contenidos



1. Introducción
2. Estado del Arte
3. Cloud MoS@RT
4. Infraestructura de Monitorización
- 5. Instanciación de la Solución**
- 6.
7. Conclusiones y Trabajo Futuro

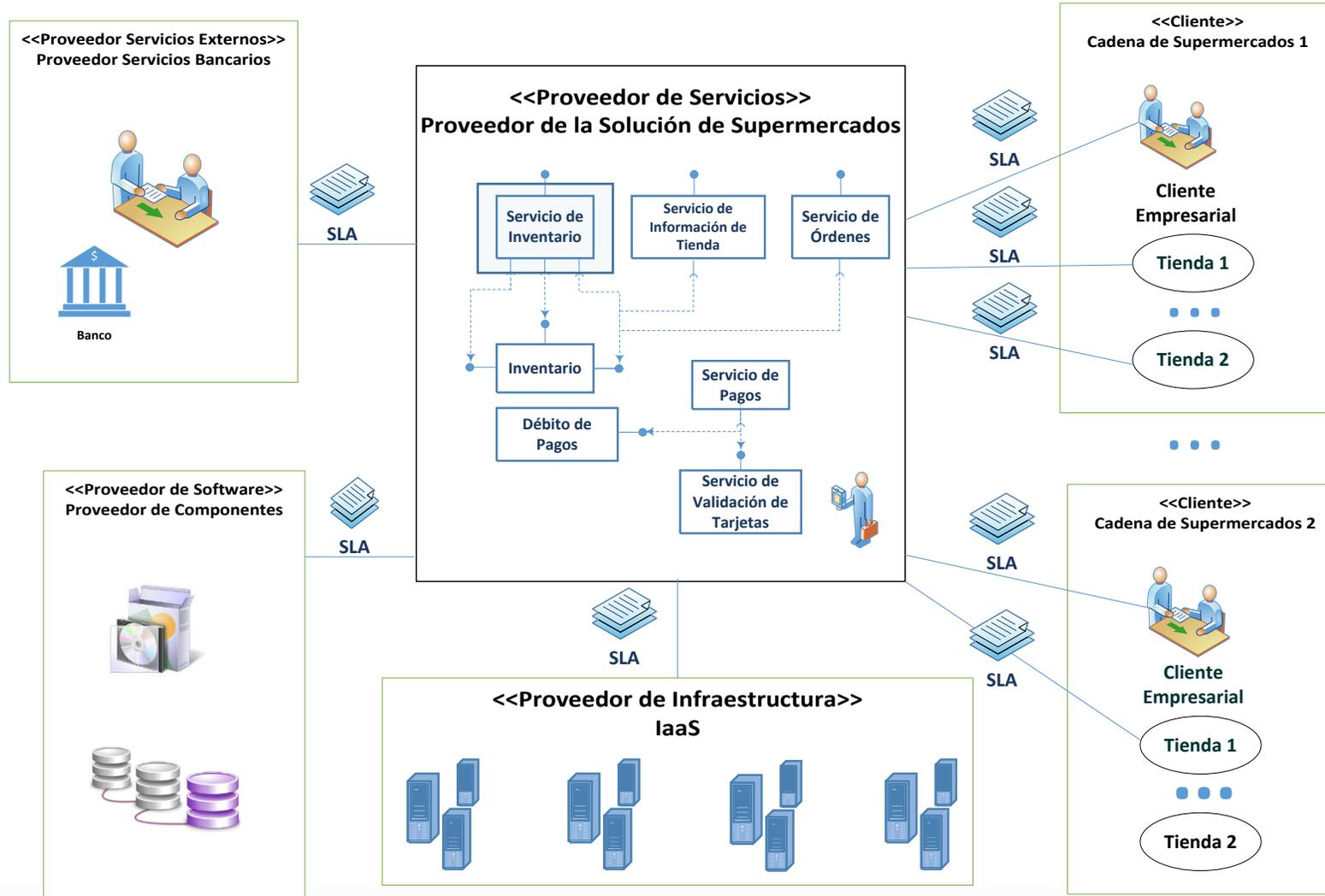
Instanciación de la Solución en Microsoft Azure



Configurador de la Monitorización:

- Configura los parámetros de monitorización
- Los RNF están contenidos en el **Modelo de Requisitos de Monitorización**.
- Los usuarios utilizan el configurador para **mapear los RNF con las fuentes de datos (Microsoft Azure Performance Counters)**.
- El configurador utiliza esta información para generar el **Modelo de Monitorización en Tiempo de Ejecución**, el cual es usado por el middleware de monitorización.
- El configurador del middleware ha sido implementado en **C#** como un **RoI Web** de Microsoft Azure utilizando **Microsoft Visual Studio**.

Caso de Estudio: “Open Case Reference Study- ORC”



Se ha utilizado para la instanciación el **Open Reference Case Study (ORC)**

Los servicios fueron desplegados en la plataforma **Microsoft Azure** bajo el modelo SaaS

Como ejemplo de la instanciación del método se tomará en consideración la Disponibilidad (**Availability**) del **Servicio de Inventario**

Especificación de los Requisitos de Monitorización

El término del SLA que establece el RNF a ser monitorizado dice:

Término del SLA:

El servicio de inventario en la solución de supermercados estará disponible el 99.50% o más del tiempo en un mes calendario.

Penalización

En caso de que el servicio falle al momento de cumplir este requisito, un crédito del 10% del total de la facturación será acreditado a la cuenta a favor del cliente.

Métrica de la Disponibilidad

$$\text{Disponibilidad} = \frac{\text{Tiempo de Servicio Acordado} - \text{Tiempo Total de Inactividad}}{\text{Tiempo de Servicio Acordado}}$$

Tiempo de Servicio Acordado (Agreed Service Time): el período durante el cual el servicio de inventario debería estar disponible.

Tiempo Total de Inactividad (Downtime): la suma, sobre un período dado, del tiempo en el cual el servicio no ha estado disponible.

Instanciación del método en Microsoft Azure

Configuración de la Monitorización

1. Establecimiento de los requisitos de monitorización

- El **Modelo de Requisitos de Monitorización** que contiene los RNF a ser monitorizados es la primera entrada del configurador de Monitorización.

MonitoringRequirements.xml MonitoringRequirements.xml

- platform:/resource/QualityModelSaaS/model/MonitoringRequirements.xml
 - SLA Document Monitoring Requirements
 - Parties
 - Signatory Party CoCoMeORCProvider
 - Signatory Party SuperMarketA
 - Supporting Party AuditorandServicesMeasurements
 - Supporting Party XYZAuditing
 - Service Definition
 - Service Object
 - Schedule FiveMinutesSchedule
 - Schedule hourlySchedule
 - Operation NewItemInventory
 - SLA Parameter Latency
 - SLA Parameter Reliability
 - SLA Parameter Availability
 - Metric DPM=(OperAttempted-OpeSuccess)/OperAttempted
 - Metric Latency=Avg(ReqTime-RespTime)
 - Metric Availability=(AgreedServiceTime-sum(Downtime))/ Agreed ServiceTime**

Instanciación del método en Microsoft Azure

Configuración de la Monitorización

2. Identificar las métricas para evaluar cada RNF

- El **Modelo de Calidad SaaS** contiene las métricas independientes de la plataforma (operacionalizaciones) junto a las métricas que pueden ser utilizadas para obtener los resultados de las mediciones.

Characteristics	Sub-Charact.	Attributes	Metrics	Operationalization
Reliability		Maturity		P. Ej. : el Modelo de Calidad SaaS contiene dos operacionalizaciones para disponibilidad:
		Availability	Robustness of a S (ROS)	
			Metric of Availability	
		Fault Tolerance	Defective Operations Per Million (DPM)	((Operations Attempted - Operations Successful)/Operations Attempted)*10^6
			Availability = $\frac{(\text{Agreed Service Time} - \text{Downtime})}{\text{Agreed Service Time}}$	
				$\frac{\text{Successful+Operations Failed}}{\text{Operations Attempted}} * 10^6$
	Service Availability	Coverage of Fault Tolerance (GFR)	Uptime	Number of Faults Without Become Failures / Total Faults Occurred
		Coverage of Failure Recovery (CFR)	Agreed Service Time	Number of Failures Remedied / Total Number of Failures

Instanciación del método en Microsoft Azure

Configuración de la Monitorización

3. Mapeo de fórmulas

- **Agreed Service Time:** Período acordado entre cliente y proveedor durante el cual el servicio deberá estar disponible.
- **Downtime:** El downtime puede ser obtenido en base a un contador de Microsoft Azure, con el cual se mapea la fórmula:

$$\text{Availability} = \frac{(\text{Agreed Service Time} - \Sigma \text{Downtime})}{\text{Agreed Service Time}}$$

Azure Performance Counter

“\Web Service(_Total)\ Service Uptime”

En caso de que el servicio cae el Uptime es negativo, de ahí el Downtime será calculado en base a la sumatoria de los valores negativos del Uptime entre dos marcas de tiempo dadas:

$$Uptime2.PreciseTimeStamp - Uptime1.PreciseTimeStamp$$

Instanciación del método en Microsoft Azure

Medición

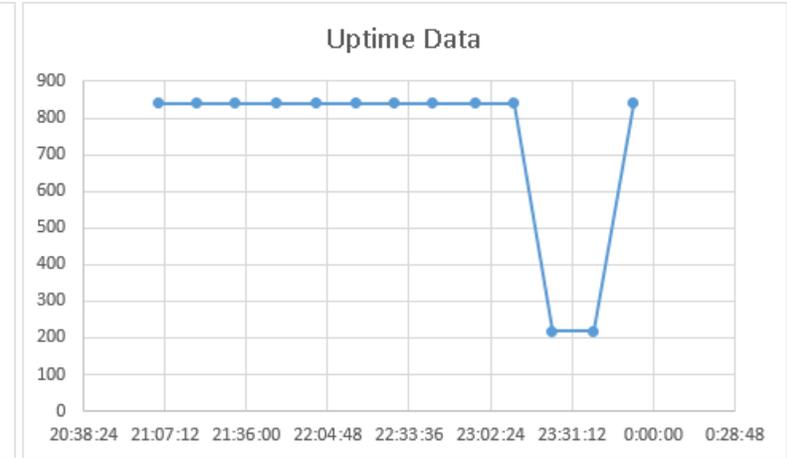
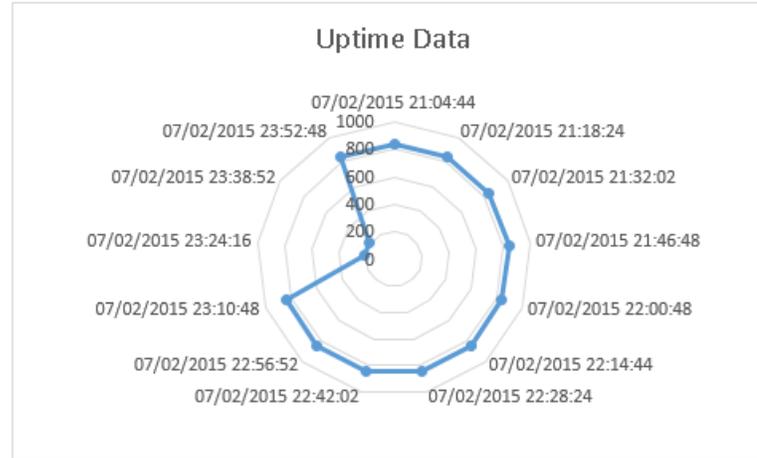
- Luego de la configuración se genera el Modelo en Tiempo de Ejecución.
- Usado para recolectar la información desde el servicio monitorizado.
- Usando los datos recolectados se podrá obtener el valor de la disponibilidad, aplicando la métrica establecida en el SLA.

PartitionKey	RowKey	Timestamp	Value	Name	Service
63558948382441...	63558948382441...	07/02/2015 21:04:44	840,2434256	Downtime	CoCoMe
63558948137973...	63558948137973...	07/02/2015 21:18:24	840,2434256	Downtime	CoCoMe
63558948017201...	63558948017201...	07/02/2015 21:32:02	840,2434256	Downtime	CoCoMe
63558948257089...	63558948257089...	07/02/2015 21:28:24	840,2434256	Downtime	CoCoMe
63558947898294...	63558947898294...	07/02/2015 21:46:48	840,2434256	Downtime	CoCoMe
63558948140439...	63558948140439...	07/02/2015 22:00:48	840,2434256	Downtime	CoCoMe
63558947684078...	63558947684078...	07/02/2015 22:14:44	840,2434256	Downtime	CoCoMe
63558948384601...	63558948384601...	07/02/2015 22:28:24	840,2434256	Downtime	CoCoMe
63558948498862...	63558948498862...	07/02/2015 22:42:02	840,2434256	Downtime	CoCoMe
63558948620832...	63558948620832...	07/02/2015 22:56:52	840,2434256	Downtime	CoCoMe
63558947901321...	63558947901321...	07/02/2015 23:10:48	840,2434256	Downtime	CoCoMe
63558948019070...	63558948019070...	07/02/2015 23:24:16	217,6923056	Downtime	CoCoMe
63558947900136...	63558947900136...	07/02/2015 19:18:21	796,5	Latency	CoCoMe
63558948018318...	63558948018318...	07/02/2015 19 20:19	796	Latency	CoCoMe
63558948139601...	63558948139601...	07/02/2015 19:22:20	796	Latency	CoCoMe
63558948258528...	63558948258528...	07/02/2015 19:24:20	147,416666666667	Latency	CoCoMe
63558948383763...	63558948383763...	07/02/2015 19:26:24	35	Latency	CoCoMe
63558947683038...	63558947683038...	07/02/2015 19:28:43	6923,92307692308	Latency	CoCoMe
63558948497632...	63558948497632...	07/02/2015 19:30:18	21,1818181818182	Latency	CoCoMe

Resultados de la Monitorización

Análisis de Resultados

- Según el término establecido en el SLA, se requiere una **disponibilidad de 99.5%** en un período de tiempo dado.
- El **Motor de Medición** usa los resultados para determinar que la disponibilidad actual del servicio ha sido del **99.444%**, de ahí se ha determinado una **violación del SLA**.



SLA Violations Report

Date: February 7, 2015 Time: 23:29:55
 From: January 6, 2015

NFR	Monitoring Value	SLA Expected Value	SLA Violation
Availability	0.994444444	0.995	YES

Date	Timestamp	Value
07/02/2015	21:04:44	840,243426
07/02/2015	21:18:24	840,243426
07/02/2015	21:32:02	840,243426
07/02/2015	21:46:48	840,243426
07/02/2015	22:00:48	840,243426
07/02/2015	22:14:44	840,243426
07/02/2015	22:28:24	840,243426
07/02/2015	22:42:02	840,243426
07/02/2015	22:56:52	840,243426
07/02/2015	23:10:48	840,243426
07/02/2015	23:24:16	217,692306
07/02/2015	23:38:52	217,692306
07/02/2015	23:52:48	840,243426

Contenidos



1. Introduction
2. Estado del Arte
3. Método Cloud MoS@RT
4. Infraestructura de Monitorización
5. Instanciación de la Solución
6. Conclusiones y Trabajo Futuro

- Este trabajo resuelve un problema práctico en el **dominio cloud**. Este evalúa la **calidad de los servicios** y la **monitorización** de los mismos en búsqueda del **cumplimiento del SLA**
- Hemos propuesto un método independiente de monitoreo independiente de la plataforma y su infraestructura que utiliza:
 - MDE (diseño y modelos en tiempo de ejecución)
 - Un proceso de monitorización para guiar a los usuarios de una manera sistemática.
 - **Resuelve limitaciones** (e.g., falta de flexibilidad, inhabilidad de monitorizar aspectos de alto nivel).
- La solución ha sido instanciada en plataformas del mundo real.

Método de monitorización

- Mejora del SaaS Quality Model
- **Refinamiento del método**
- Extensión a otras tecnologías (Fog Computing)

Infraestructura de monitorización

- Construcción de adaptadores y plugins
- Pruebas en otras plataformas (e.g., Amazon AWS)
- Creación de formas afinadas de visualización
- Uso en ambientes industriales.

Publicaciones Relacionadas

International Conferences and Workshops	ERA CORE Tier A	<ul style="list-style-type: none">• P. Cedillo, J. Jimenez-Gomez, S. Abrahão, and E. Insfran, “Towards a Monitoring Middleware for Cloud Services,” 9th IEEE International Conference on Services Computing (SCC 2015), June 27- July 2, 2015, New York, USA, pp. 451-458.• P. Cedillo, J. Gonzalez-Huerta, S. Abrahão, and E. Insfrán, “A Monitoring Infrastructure for the Quality Assessment of Cloud Services,” 24th International Conference on Information Systems Development (ISD 2015), August 25 - 27, 2015, Harbin, China, pp. 17-32.
	Workshops	<ul style="list-style-type: none">• P. Cedillo, J. Gonzalez-Huerta, E. Insfrán, and S. Abrahão, “Towards Monitoring Cloud Services Using Models@run time”. 9th International Workshop on Models at run.time (MRT 2014), co-located with the ACM/IEEE 17th International Conference on Model Driven Engineering Language and Systems (MODELS 2014), Valencia, Spain, pp. 31–40.• P. Cedillo, A. Fernandez, E. Insfran, and S. Abrahão, “Quality of Web Mashups: A Systematic Mapping Study,” 4th International Workshop on Quality in Web Engineering (QWE 2013), co-located with the 13th International Conference on Web Engineering (ICWE), Aalborg, Denmark, pp. 66–78

Publicaciones Relacionadas

Other and on-going publications	International Conferences	<ul style="list-style-type: none">E. Insfran, P. Cedillo, A. Fernandez, S. Abrahão, and M. Matera, “Evaluating the Usability of Mashups Applications,” 8th International Conference on the Quality of Information and Communications Technology (QUATIC 2012), Lisbon, Portugal, 3-6 September 2012, IEEE Computer Society, pp. 323–326
	National Conf.	<ul style="list-style-type: none">P. Cedillo, J. Jimenez-Gomez, S. Abrahão, and E. Insfrán, “Definición de Mecanismos Personalizados de Monitorización de Servicios Cloud,” XI Jornadas de Ciencia e Ingeniería de Servicios (JCIS 2015), Santander, 15-17 Septiembre 2015.
	Magazines & Journals	<ul style="list-style-type: none">M. A. Zúñiga-Prieto, P. Cedillo, J. González-Huerta, E. Insfrán, and S. Abrahão, “Monitoring Services Quality in the Cloud,” ERCIM News, no. 99, <i>Special Theme on Software Quality</i>, 2014, pp. 19–20.P. Cedillo, E. Insfrán, J. González, S. Abrahão, “Design and Evaluation of a Monitoring Infrastructure for Cloud Services”, <i>Information Sciences Journal</i>, Elsevier, Impact Factor 3.364 (JCR 2015), which was sent in September, 2016.

Pre-doctoral research stay for 3.5 months at the **National Institute of Informatics** (NII) in **Tokyo-Japan**, from December 10th, 2013 to March 27th, 2014.

The purpose of this stay was twofold:

- Self-adaptive systems, which was useful as a starting point for the definition of the strategy proposed in this thesis.
- Characteristics of quality involved with the energy preservation of mobile devices with the Android OS, which has contributed to one of the NII projects.



Contexto de la investigación

Introduction

Estado del Arte
Método Cloud MoS@RT
Infraestructura de Monitorización
Instanciación de la Solución
Conclusiones y Trabajo Futuro

Software Engineering and Information Systems (ISSI) Research Group - Universitat Politècnica de València:

Under a **four-year pre-doctoral** grant (2011-2015) funded by

- Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT-ECUADOR)
- Universidad de Cuenca – Ecuador



Research Project: **Value@Cloud** - “Desarrollo Incremental de Servicios Cloud Dirigido por Modelos y Orientado al Valor del Cliente”. Funded by the Ministerio de Economía y Competitividad (MINECO-Spain) (2014-2017).



Research Project: **TwinTIDE** - “Towards the Integration of Transectorial IT Design and Evaluation”. Funded by COST, European Union (2009-2013).



Microsoft Azure Research Award: “Model-Driven Incremental Development of Cloud Services”, Microsoft Research (2014-2016).





GRACIAS POR SU ATENCIÓN...



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

