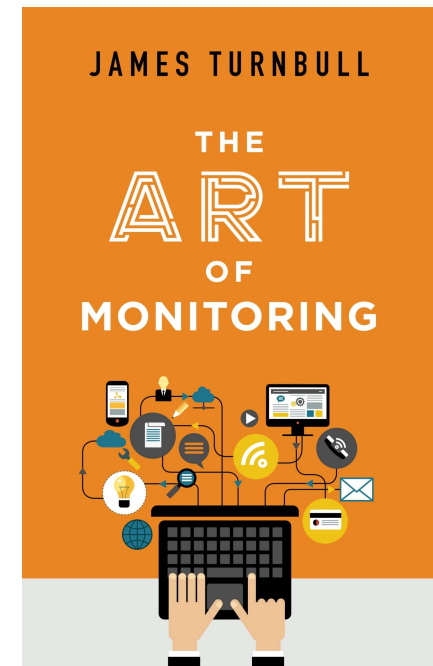


# JMX

Fundamentos

# JMX. La necesidad de supervisar.

- The Art of Monitoring
  - James Turnbull



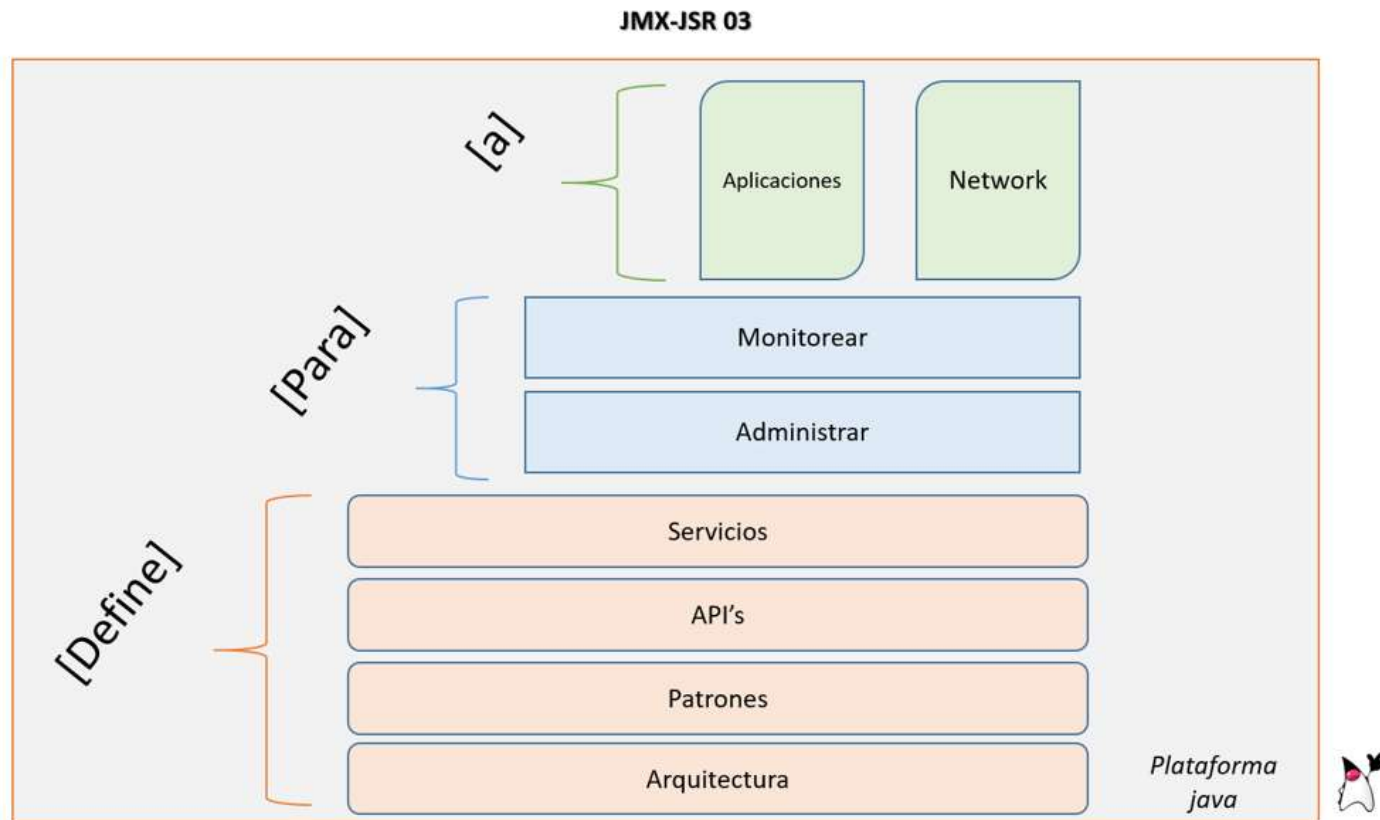
# JMX. La necesidad de supervisar.

- Management technologies
  - SNMP

# JMX. JSR-003

- The Java Management extensions (also called the JMX specification) define an architecture, the design patterns, the APIs, and the services for application and network management and monitoring in the Java programming language.
- JMX (Java Management extensions por sus siglas en inglés) define una arquitectura, patrones de diseño, API's y los servicios para monitorear y administrar aplicaciones y redes en el lenguaje de programación java.

# JMX. JSR-003



*Figura 1. El objetivo de la JSR 03 visto de manera gráfica.*

## JMX. JSR-003. ¿Por qué usarlo?

- Permite gestionar las aplicaciones Java sin invertir grandes esfuerzos.
- Proporciona una arquitectura de administración escalable.
- Se integra a soluciones de supervisión existentes.
- Aprovecha las tecnologías Java estándar ya existentes.
- Puede aprovechar a futuro conceptos de supervisión.
- Define sólo las interfaces necesarias para la supervisión.

JMX. JSR-003.

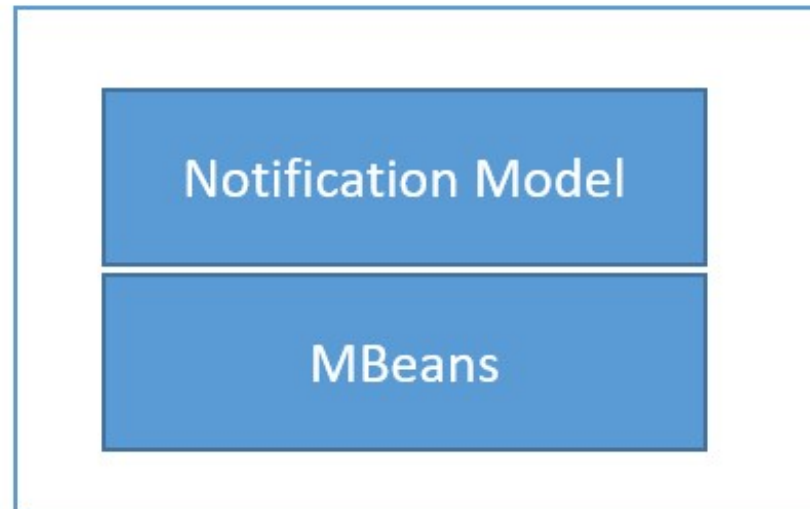
Nivel de Servicios Distribuidos

Nivel de Agentes

Nivel de instrumentación

# JMX. JSR-003. Nivel de instrumentación.

Instrumentation level.





# JMX. JSR-003. Nivel de instrumentación.

- Un <JMX manageable resource> puede ser:
  - Una aplicación.
  - Una implementación de un servicio.
  - Un dispositivo.
  - Un usuario.
  - Étcetera.

# JMX. JSR-003. Nivel de instrumentación.

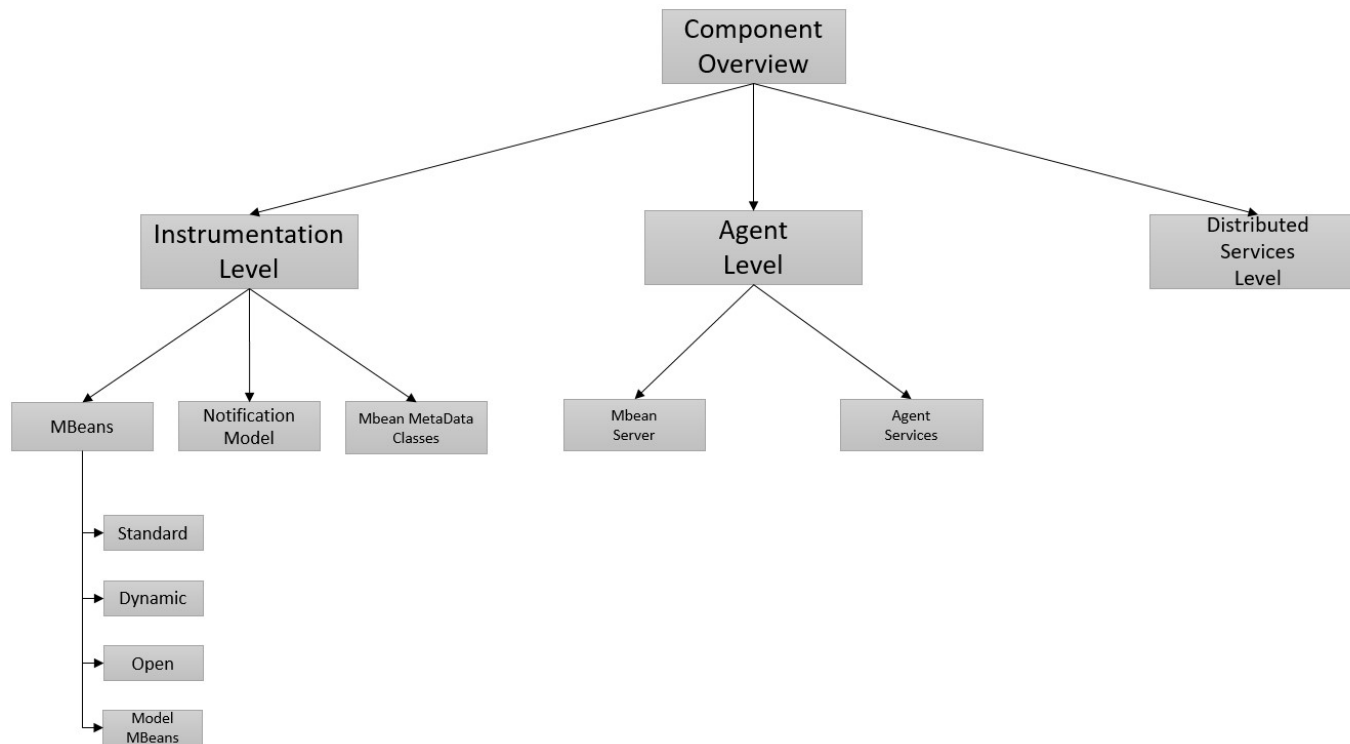
- MBean
  - Standard
  - Dinamico

## JMX. JSR-003. Nivel de agentes.

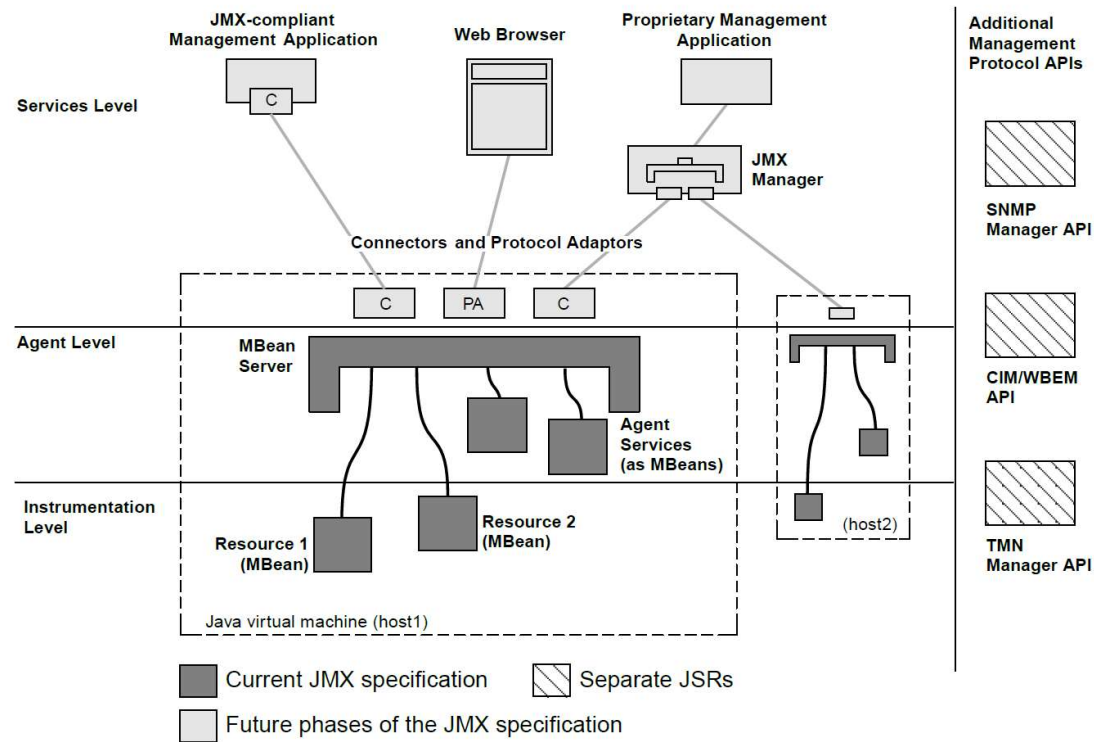
Agent level.



# JMX. JSR-003



# JMX. JSR-003



JMX. Hablemos de Mbean (*instrumentación*)



# JMX. Hablemos de Mbean (*instrumentación*)

- Managed Bean.
  - Operaciones que puedan ser invocadas.
  - Atributos que puedan ser accedidos.
  - Notificaciones que puedan ser emitidas
  - Constructores que puedan ser utilizados.

# JMX. Tipos de Mbean.

|                 |   |
|-----------------|---|
| Standard MBeans | Es el más sencillo, su interface define todos sus atributos y su implementación es fija, no funcionalidad no puede ir más allá que la definida en su interface. |
| Dynamic MBeans  | Debe implementar una interface proporcionada por la especificación. Proporciona flexibilidad de en tiempo de ejecución.   |
| Open MBeans     | Son Dynamic MBeans; se basan en tipos de datos básicos para hacer más genérica su utilización. Son auto-descriptivos.   |
| Model MBeans    | También son Dynamic MBeans, pero son completamente configurables y auto-descriptivos en tiempo de ejecución   |



JMX.

Manos a la obra.

JMX.

- Di su nombre.
  - <https://www.linkedin.com/pulse/d%C3%AD-su-nombre-isaac-ruiz-guerra>

## JMX. objectName

- El <Object Name> está compuesto de 2 partes:
  - Un nombre de dominio.
  - Un listado no ordenado de una o más propiedades.

# JMX. objectName

- Nombre de dominio.

El nombre de dominio es una cadena sensible a mayúsculas y minúsculas, puede contener cualquier carácter excepto:

|   |  |
|---|--|
| : | Es el token que indica el fin del nombre de dominio. |
| * | Caracter comodín.                                    |
| ? | Carater comodín.                                     |

# JMX. objectName

- Propiedades

Las propiedades sirven para complementar el nombre del dominio y darle así un nombre único.

Cada propiedad es una combinación de llave=valor, y no necesariamente se refieren a propiedades del MBean, sirven sólo para completar el nombrado y para efectos de organización de los MBeans.

El valor de la propiedad puede ser cualquier secuencia de caracteres exceptuando:

: " , = \* ?

# JMX. En todos lados.

- App Servers.
- Microservicios.
- IoT.
- Dónde exista una JVM, se puede generar un Mbean Server y exponer Mbeans.

# JMX.

- La guía perdida.
  - Parte 1.
    - <http://www.oracle.com/technetwork/es/articles/java/java-management-extensions-3403127-esa.html>

¿Preguntas?

@rugi



Para continuar.....

<https://github.com/javamexico/propuestas-meetup>

Introducción al Java VisualVM #12

<https://github.com/javamexico/propuestas-meetup/issues/12>