

BlueHats Semester of Code: Joram

Le projet Joram est un composant critique de nombreuses applications. En open-source depuis 1999 il est le seul MOM souverain. Il implante de nombreux standards (JMS, JCA, AMQP, MQTT, etc) et à ce titre doit respecter les spécifications correspondantes. L'objectif principal de ce projet est d'étendre la couverture de test du projet avec les suites de certification Eclipse Jakarta.

Définitions

JMS - Java Message Service

L'API JMS permet d'émettre et de recevoir des messages de manière asynchrone entre applications Java. La communication s'effectue via un broker de messages (MOM). JMS supporte deux modèles d'interaction : publish/subscribe et point-à-point.

MOM - Message Oriented Middleware

Les MOM permettent des interactions par envoi de messages entre composants applicatifs dans un cadre faiblement couplé, asynchrone et fiable. Les principales normes sont JMS, AMQP et MQTT.

JEE ou Jakarta EE

JakartaEE (J2EE ou Java EE), est une spécification pour la plate-forme Java, destinée aux applications d'entreprise.

JakartaEE définit les éléments suivants :

- une plate-forme pour héberger et exécuter les applications, incluant outre Java SE des bibliothèques logicielles additionnelles ;
- une suite de tests (Java EE Compatibility Test Suite) pour vérifier la compatibilité ;
- une réalisation de référence ;

JCA - Java Connector Architecture

Dans une architecture JEE, JCA est la spécification permettant de résoudre le problème d'intégration entre le monde JEE et le système d'information d'entreprise.

Le composant JMS, Joram, est intégré dans la plateforme JEE au travers d'un connecteur JCA.

MQTT - Message Queuing Telemetry Transport

MQTT est un protocole de messagerie publish/subscribe basé sur le protocole TCP/IP. Il est l'un des protocoles le plus utilisé dans le domaine de l'IoT.

Depuis la version 3.1.1, MQTT est un standard OASIS, la version 5 de la spécification est maintenant publiée depuis mars 2019.

OSGi

OSGi est une plate-forme de services fondée sur le langage Java qui peut être gérée de manière distante. Le cœur de cette spécification est un framework qui définit un modèle de gestion de cycle de vie d'une application, un répertoire (registry) de services, un environnement d'exécution et des modules.

Présentation du projet Joram

Cette section fait une présentation courte et synthétique du projet Joram, de nombreuses autres informations sont accessibles depuis le site du projet : <https://joram.ow2.io>.

Historique

Joram est une implémentation open source de la spécification JMS. Le projet a démarré en 1999 à l'Inria, il est publié sous licence LGPL depuis mai 2000 sous l'égide du consortium ObjectWeb et maintenant OW2 (<https://ow2.org>).

Joram est le fruit de très nombreuses collaborations, de partenaires industriels ou académiques (<https://joram.ow2.io/partners.html>), mais aussi de projets de recherches.

Depuis son origine, Joram fournit un MOM distribué prêt pour la production, et utilisé dans de nombreuses applications opérationnelles critiques (Télécom, spatial, industrie).

Initialement centré sur JMS, Joram a suivi les différentes initiatives de standardisation dans le domaine du messaging, et est devenu un MOM multi-protocole : JMS, AMQP, MQTT, etc.

Joram est un produit stable et vivant, il fait l'objet d'une ou deux releases par an. La criticité de ses usages nécessite que chacune de ces versions soit testée de manière approfondie.

Présentation technique

Joram est presque exclusivement développé en Java. Le projet utilise une plateforme OSGi pour la gestion de ces composants de code, ce qui le rend extrêmement configurable.

Hormis la fiabilité, une de ses principales qualités est la scalabilité.

Les outils utilisés pour son développement sont Git, Maven et Java.

Usages

Joram a également été utilisé, et l'est peut-être encore, par diverses administrations et entreprises nationales comme : l'ACOSS, Atos (Ministère de l'intérieur), CS (armée de l'air), Digital Liance (Téléphonie), EADS (Défense), eFluid, Etat de Genève, Orange, Thales (Spatial), et JoramMQ (ScalAgent). Ses principales utilisations actuelles sont dans le domaine de la gestion de l'énergie avec eFluid (Enedis, et de très nombreux fournisseurs et distributeurs d'énergie), et dans le domaine de l'IoT avec JoramMQ (Thales, ContiTech, eFluid, etc.).

Evolutions récentes

Avec le don par Oracle de JavaEE à la fondation Eclipse, et la sortie de la version 9 de JEE les spécifications ont évolué de manière importante (JMS v3.0 et JCA v2.0), et les plateformes de vérification de celles-ci ont été libérées.

Pour continuer à s'intégrer dans les plateformes applicatives et satisfaire ses utilisateurs, Joram a dû implanter ces évolutions. Il est maintenant nécessaire de vérifier la correction de ces implantations en mettant en œuvre les suites de tests correspondantes.

Présentation du projet étudiant

L'objectif principal du projet est la mise en place d'un environnement de test automatisé (Docker, Kubernetes) permettant la mise en œuvre des suites de tests JakartaEE qui sont maintenant accessibles librement.

Cet environnement devrait permettre d'automatiser le passage des tests de compatibilité sur les versions successives de Joram et ainsi de garantir le respect des spécifications aux utilisateurs du logiciel.

Le projet devrait comporter plusieurs phases :

- Une phase de découverte des technologies JEE, JCA et JMS.
- Une mise en œuvre manuelle des suites de tests JEE avec les implantations de référence, puis avec Joram.
- La spécification d'un environnement automatisé (Docker, Kubernetes) permettant l'exécution de ces suites de tests, et éventuellement leur extension à des environnements autres (Spring, etc.).
- La réalisation itérative de ces environnements dans un esprit agile.

Références

OW2/Joram :

- <https://joram.ow2.io>
- <https://gitlab.ow2.org/joram/joram>
- <https://www.openhub.net/p/joram>

JakartaEE :

- <https://jakarta.ee/specifications/platform/9>
- <https://jakarta.ee/specifications/messaging/3.0/jakarta-messaging-spec-3.0.htm>