ORACLE
Transaction Manager for Microservices

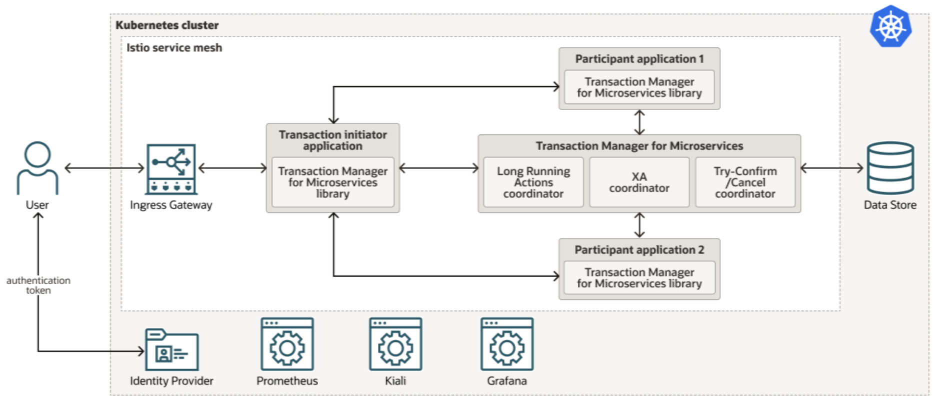
Oracle Transaction Manager for Microservices

Oracle Transaction Manager for Microservices (MicroTx), an enterprise product, enables distributed transactions and data consistency for microservices based and cloud native applications. Developers can choose from several transaction protocols to enable desired consistency for their applications and reduce development complexity and costs. MicroTx helps establish a simplified path for microservices adoption and expedites the digital transformation journey for your organization. MicroTx offers a comprehensive solution for distributed transaction management that is secure, highly-performant, highly available, scalable and can be deployed on-premises, cloud, or in hybrid environments.

Data Consistency with Microservices Application Development

Microservices based applications are inherently distributed and more often use/update multiple data sources in one business transaction. Ensuring data consistency across all the resources in a business transaction is a real challenge for microservices based architecture, and the problem becomes more visible if microservices are written in different programming languages using various development frameworks. MicroTx helps solve this challenge by providing a spectrum of data consistency solution for many development environments used to develop and deploy microservices based applications.

Transaction Manager for Microservices



Kubernetes cluster
Istio service mesh
User
authentication token
Ingress Gateway
Transaction initiator application
Transaction Manager for Microservices library
Participant application 1
Transaction Manager for Microservices library
Transaction Manager for Microservices
Long Running Actions coordinator
XA coordinator
Try-Confirm /Cancel coordinator
Participant application 2
Transaction Manager for Microservices library
Data Store
Identity Provider
Prometheus
Kiali
Grafana

Figure 1: Transaction Manager for Microservices Architecture

The main component of Oracle Transaction Manager for Microservices is a transaction coordinator. Applications initiating a transaction will interact via REST APIs with the transaction coordinator using one of the supported transaction protocols to enlist in the transaction as well as to listen for any calls back from the coordinator. The coordinator itself is a microservice and deployed as such. MicroTx also provides various libraries that are used in the applications participating in a transaction. As the MicroTx coordinator is a microservice, it can be well integrated into target deployment environments, such as Kubernetes, Docker Swarm, etc.

Key Business Benefits

- Increases developer productivity – easy to adapt existing applications to use transactions
- Leverages existing assets, development frameworks – continue to use tools as you were
- Based on industry standards
- Enables data consistency in multi-language apps
- Accelerates path to broader cloud adoption
- Quicker deployment and faster time to value

Key Features

- Support for multiple distributed transaction protocols - XA protocol for strong consistency; Long Running Actions or LRA for eventual consistency; Try-Confirm/Cancel (TCC) for reservation model
- Works with microservices applications running in on-premise, cloud, and hybrid environments
- Works with popular programming languages, such as TypeScript and Java
- Works with popular development frameworks, such as Spring Boot, Helidon, WebLogic Server, Express.js and so on
- XA transactions can include Oracle Tuxedo services and Oracle Database ORDS/APEX apps along with microservices
- Integrates well with Kubernetes ecosystem, such as Jaeger, Kiali, Prometheus, and Grafana

Spectrum of Transaction Protocols

You can select a suitable transaction protocol for your application based on your business requirements and the level of data consistency required. For example, financial applications that move funds require strong global consistency. The XA transaction protocol is a good fit for such applications. XA offers strong data consistency with low development complexity. XA transaction protocol eliminates the need for application specific business logic to ensure consistency.

On the other hand, making a travel reservation typically doesn't require this level of data consistency and the business transaction could be relatively long. MicroTx supports Sagas based on Eclipse MicroProfile Long Running Actions (LRA) for such use cases. LRA transactions provide eventual consistency and users may see inconsistent updates across resources for the duration of the distributed transaction. Development with LRAs can be more complex as users are required to write and test compensate business logic.

The third transaction protocol supported by MicroTx is Try-Confirm/Cancel or TCC. The TCC protocol is good for use cases that are based on a reservation model and can provide strong consistency. A reservation is made on a resource, which is either confirmed or cancelled depending upon the outcome of transaction. Travel reservation use case mentioned above might be a good fit for TCC as well, as will be a movie ticket booking use case.

Microservices Application Development

Once the transaction protocol to be used has been decided, developers build their application with MicroTx libraries. MicroTx provides libraries for Java, TypeScript and PL/SQL programming languages as of now. These libraries work in conjunction with underlying application development frameworks, for example Helidon or Spring Boot for Java and Express for TypeScript. These libraries make application development extremely simple and limit code changes required to an existing application by providing REST API filters and interceptors and by taking care of communication with the MicroTx transaction coordinator behind the scenes.

Deployment Flexibility

Microservices based applications generally leverage container architecture and are deployed in Kubernetes or similar platforms. MicroTx transaction coordinator is itself a microservice, just like any other application microservice and can be deployed using similar tools and methodology in a Kubernetes cluster. In addition, MicroTx integrates well with various tools in Kubernetes ecosystem, such as Prometheus, Grafana, Jaeger and Kiali and includes out-of-the-box configuration and/or instructions to leverage such tools. MicroTx also includes Helm charts, sample configurations, and detailed step-by-step instructions to make it easier to get started. MicroTx can also be used with Docker Swarm and includes the required artefacts to get started with Docker Swarm as well.

Interoperability with Blockchain and many other existing applications

While it is great to develop new applications as microservices, more often there is a need to interoperate with existing applications that are running in different infrastructures. One can include many such applications in an XA transactions being coordinated by MicroTx. MicroTx supports such interoperability with Oracle Database ORDS/APEX, Oracle Tuxedo, and Oracle Blockchain Platform applications. Any or all of these applications can participate in an XA transaction along with microservices written in Java or TypeScript programming languages.

- Support for various resource managers, such as Oracle Database, MySQL, Postgres, and MongoDB
- Supports multiple Identity providers, such as Microsoft Active Directory, Identity Domains and Keycloak
- Unique Resource Manager proxy approach to reduce number of database connections
- Use of filters /interceptors to make application development simple
- Supports transaction coordination with Blockchain networks
- Flexibility to deploy in containers like Kubernetes cluster or Docker Swarm

ORACLE
Transaction Manager for Microservices

Contact Us

For more information, visit www.oracle.com
or call +1.800.ORACLE1 to speak to an Oracle representative.

blogs.oracle.com/oracle facebook.com/oracle twitter.com/oracle

Oracle is committed to developing practices and products that help protect the environment