

10

WEB-RI

10.1 Purpose

The WEB-RI proposes itself as reference for the implementation of a user interface to allow human users to use and interact with the functionalities provided by i3-MARKET. The WEB-RI has three main objectives, which are:

- As a management tool, to allow i3-MARKET developers to test their functionalities in the context of a user usage.
- As a reference implementation, providing functional examples of how the i3-MARKET SDKs can be used to implement/integrate i3-MARKET functionalities into a data marketplace. As a reference implementation, WEB-RI is also a useful tool to help i3-MARKET pilots on the implementation of their use-case scenarios and on testing of backplane technologies by providing specifications and code that can be used.

10.2 Architecture

In Figure 10.1, the architecture of WEB-RI is represented.

A consumer or a provider can access WEB-RI¹ via internet browser and proceed with the authentication for which the wallet² must be installed and running on his personal computer. The authentication process is executed on WEB-RI frontend by calling the OIDC service which will call the wallet to perform the authentication itself.

¹<https://gitlab.com/i3-market-v2-public-repository/i3-market-web-ri>

²<https://gitlab.com/i3-market-v2-public-repository/sp3-scgbssw-i3mwalletmonorepo>

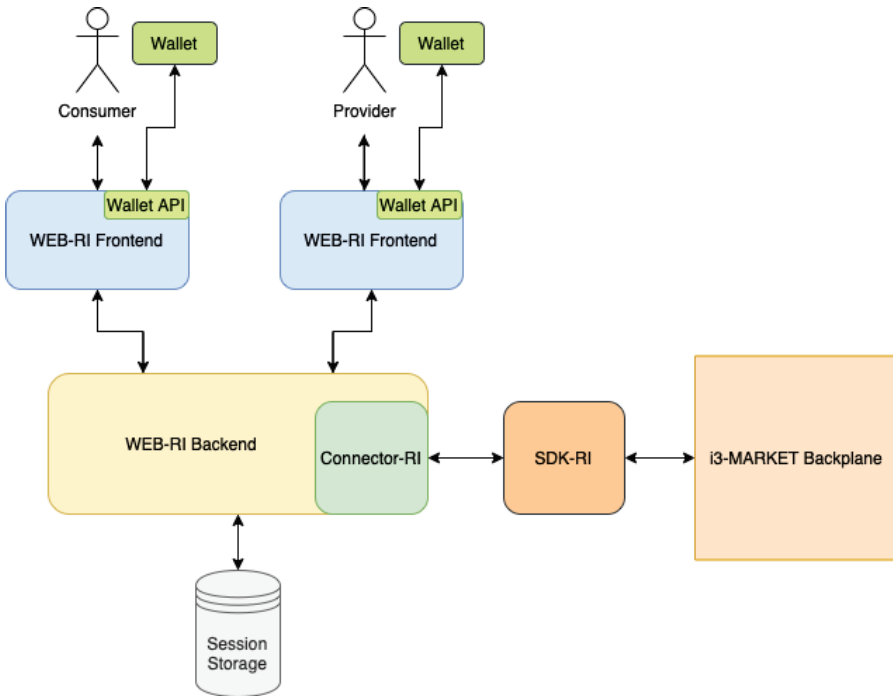


Figure 10.1 WEB-RI architecture.

The WEB-RI frontend is connected to a backend, which has two main functions: manage user sessions and have a way to interact with the functionalities provided by i3-MARKET.

To manage the user sessions, the WEB-RI backend saves the user session in a session storage called connect-mongo³.

To interact with the functionalities provided by i3-MARKET, a library was implemented, called Connector-RI⁴. This connector has all the methods needed to call the respective APIs from the SDK-RI, which have the functionalities to interact with the i3-MARKET Backplane. This allows to have a clean and simple WEB-RI backend where it is only needed to call the respective methods from the connector.

³<https://github.com/jdesboeufs/connect-mongo>

⁴<https://gitlab.com/i3-market-v2-public-repository/i3-market-connector-ri>

10.3 Sitemap

In Figure 10.2, the sitemap of WEB-RI is represented.

WEB-RI is composed of several pages, which are Authentication, Homepage, Offerings, Search, and Notifications.

In the Authentication page, the user has the possibility to register a new provider or consumer and login with some existing user registered in WEB-RI.

The Homepage is the main page of WEB-RI, which has a navigation bar that allows the user to navigate to the other available pages. Also, there are statistics related with the number of offerings and providers.

The Offerings page is only visible to a provider, where he can manage the offerings registered by him and register new ones.



Figure 10.2 WEB-RI sitemap.

The Search page is visible either to a provider or a consumer. The only difference is that a consumer has the possibility to create a purchase request for the offering he searched.

In the Notifications page, a provider can receive a purchase request for some of its offerings and he can accept (and create the agreement) or reject it. A consumer can sign the agreement if it was accepted before by the provider.

10.4 Run WEB-RI in Docker

The WEB-RI can be reused and customized, in order to do so run the WEB-RI docker, to get the code, use git clone command, the web-ri code available at (<https://github.com/i3-Market-V3-Public-Repository/WRR-WebRI>), first you must define the following environment variables in docker-compose.yml file:

```
environment:  
SDK_RI_ENDPOINT: sdk-ri endpoint  
MONGO_URL: mongodb url  
OIDC_URL: oidc provider  
VC_URL: verifiable credential service  
MARKET_NAME: market of notification service  
  
MONGO_INITDB_ROOT_USERNAME:  
mongodb username  
MONGO_INITDB_ROOT_PASSWORD:  
mongodb password
```

Then,

```
docker-compose up
```