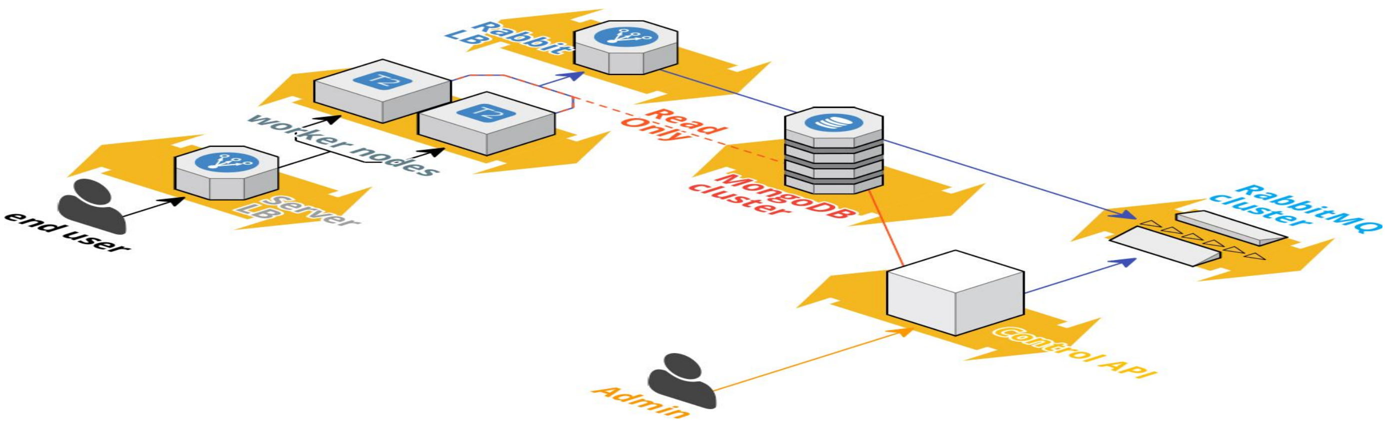
FLOW ORCHESTRATOR

Deployment Guide



INTRODUCTION

Purpose

This document provides an overview and the required details to understand how to build, execute the Flow Orchestrator system and also how to take the project code and import into eclipse. Along with the same this also details the steps to deploy any new Functional Components.

Intended Audience

The intended audience is any user who would want to build, deploy, test the system and also who would want to look into the source code of the system along with developing any new functional components if required.

Definitions, Acronyms & Abbreviations

PU:	Power User
FO:	Flow Orchestrator
FOE:	Flow of execution

Build & Deploy - Flow Orchestrator

Getting the project from GitHub

Please use the below link to get to the GitHub repository and download all the files from the repository. You will need to clock the "Clone or Download" button then click "Download ZIP" option.

https://github.com/vimalrv/gFlowOfExecution.git

This will download the project contents into your desktop/laptop. The repository consists of 2 eclipse projects and a folder for documentation like below:

FlowComponent (eclipse project)

FlowOrchestrator (eclipse project, but is dependent on FlowComponent)

Docs (documentation)

Binary (contains a runnable jar and config csv files)

Please use the below link to get to the GitHub repository and download all the files from the repository. You will need to clock the “Clone or Download” button then click “Download ZIP” option.

You will see a file named “gFlowOfExecution-master.zip” will get downloaded, once this is downloaded to any directory of your choice in your desktop/laptop, extract the contents into the same folder either by using unzip command in a command terminal or by any other means suitable for you.

Build and execute with maven

After the above step, you should have a folder named “gFlowOfExecution-master”, use a terminal/command window and go into this folder, then go into “FlowComponent” folder first and execute the below command in your terminal:

```
mvn clean compile test-compile jar:jar install:install
```

This command should compile, package and install the components related jar file into your local maven repository. This is essential to be done before we compile “FlowOrchestrator” as there is dependency involved here. Post this step, come out of the “FlowComponent” folder, now go into the folder “FlowOrchestrator” and execute the below command in your terminal:

```
mvn clean compile test-compile jar:jar install:install
```

This command should install the FlowOrchestrator jar file into your local maven repository. Now execute the below command:

```
mvn exec:exec
```

This command will execute the “FlowCaller” test class which would call the “FlowOrchestator” first as a synchronous call and then as an Async call with some dummy data to depict a simple flow of the prototype system that has been developed.

Now, you can go to “FlowOrchestrator/logs” folder and open the log to look into the debug statements and understand how the entire flow had occurred.

You can also create a new flow of execution following the instruction in the PU_UserGuide document to test any new flow of interest, however for the prototype, the “flow-config-create-employee.csv” file contains the config that will be run by default.

Execute the jar file directly

If you do not wish to build/execute via maven, go directly into the “binary” folder. Unzip the file “FlowOrchestrator.zip” into any local folder in your desktop/laptop. Now go into that unzipped folder you will a runnable jar file and a folder named ”config” this is essential for the prototype to run as this contains a sample flow of execution config and the sample components related metadata files in a csv format.

Once you are into this unzipped folder, execute the following command in your terminal/command window:

```
java -jar FlowOrchestrator.jar
```

You will see that the sample “FlowCaller” test executes and calls the “FlowOrchestrator” in both synchronous and async manner. Now, you can go to “./logs” folder and open the log to look into the debug statements and understand how the entire flow had occurred.

You can also create a new flow of execution following the instruction in the PU_UserGuide document to test any new flow of interest, however for the prototype, the “flow-config-create-employee.csv” file contains the config that will be run by default.

Import the project into Eclipse

Pre-requisite: you need to have maven installed in your desktop/laptop, as this is required to resolve all the library/jar dependencies of the projects. As such only the source code has been checked-in into GitHub, the dependant jar files are to be downloaded via maven. Maven pom.xml is already provided in the project, so you dont have to do anything, just that maven installation needs to be present for eclipse to load the pom.xml file and manage dependencies.

Once, you have unzipped the zip file from GitHub repository, you can go into eclipse and follow the below sequence to import the project into eclipse.

Open eclipse → File → Import → maven → Existing maven projects → select the “FlowComponent” folder

Follow the same steps above and select the “FlowOrchestrator” folder to import that project.

Steps to Develop & Deploy new components

To develop and make new functional components available for Power User follow the below steps:

Development

1. Create an Akka Actor for the component that would like to make available
2. In the “OnReceive” lifecycle method, code for any input parameter manipulation. You can assume that the input parameters will be passed in as a JSON string into the OnReceive method.
3. Then, prepare the input to the component in a way that the component would accept.
4. Now, call the component and pass in the input parameters
5. Get the response from the component and use getSender and send the response, just in case if the flow is constructed to be a synchronous flow then a response will be expected, if no response is provided for a synchronous flow it will become a timeout exception.

Deployment

1. Package the actor and any related dependent files into a jar file
2. Get into the “FlowOrchestrator” project and then into the config folder
3. Open the “deployed-components.csv” file to register this new component
4. Provide values for the below metadata columns:
    a. ComponentID - this is the id using which the power user will refer to this component while constructing a flow using this component.
    b. CompClassName - provide the fully qualified name of the component actor that you just created, this will be used to dynamically instantiate the actor by the “FlowOrchestrator” system
    c. ExpectedInputParameters - provide the field names that are expected as input parameters for this component call in this format. Format - fieldName=dataType#fieldName=dataType... n, Example. - passportNumber=string#firstname=string
    d. ResponseFields - like wise provide the response field information as well
5. Save the config file
6. Move the component actor jar file as a dependency in the “FlowOrchestrator” project, either through pom.xml or physical movement of the jar.
7. This completes the deployment and this functional component can now be accessed/used by the Power User while constructing a new flow of execution.