**WORKFLOW-EXECUTION WITHIN WORKFLOW-INITIATOR MICRO-SERVICE**

**Order-Client-Application**

**It does two things**

[1] **Completes its local actions (validation and other business logics) with a received order**. After that it saves the order with a PENDING status in its local database.

[2] After saving it, **it calls a command interface method for executing a defined workflow for an order**.

**Order-Micro-Service**

**Workflow-Controller**

**Workflow-Command**

**It is doing nothing except calling workflow orchestrator**. Its objective is to call different workflow orchestrators if different workflow has been defined within this **workflow implementing workflow-initiator micro-service.**

**It creates a stub corresponding to a workflow and submits it to the Temporal-Server for its execution.** While creating a stub, it uses OrderFulfillmentWorkflowImpl class where the complete workflow definition using different activities have been made.

**Workflow-Orchestrator**

[1] **A Temporal-Server**, which has already registered four workers corresponding to four micro-services when the respective micro-service was started, receives a workflow execution-request through a workflow initiator ms-order micro-service.

{2] **As the Temporal-Server has a complete definition of workflow execution**, it starts calling activities defined within the workflow one by one in the sequence in which they have been defined. While calling different activities of workflow, it uses **gRPC (A remote method calling mechanism).**

[3] **When an activity of workflow is called,** the respective micro-service activity implementing method is called and the activity is completed accordingly.

**Temporal-Server**

**WORKFLOW-EXECUTION WITHIN AN ACTIVITY-EXECUTOR-MICRO-SERVICE**

A workflow initiator could request Temporal-Server to start a workflow execution either **in synchronous** or **in asynchronous mode**. In synchronous mode, a workflow corresponding to a received order is started and completed first before executing any new order in the pipeline or in a queue. But, in asynchronous mode, Tempral Server starts different workflows corresponding to different received orders parallelly.

**TEMPORAL-SERVER**

**Following activities have already been submitted by workflow-initiator [ms-order] micro-service to the Temporal-Server**

**[1] completePaymentActivity**

**[2] reserverInventoryAgainstAnOrder**

**[3] completeShipmentActivity**

**[4] completeOrderWorkflowActivity**

**Temporal-Server starts calling these activities of the workflow in the same sequence in which they’ve been defined above.**

Call-**Activity-1**. Step-1

Call-**Activity-4** Step-1

Call-**Activity-3** Step-1

Call-**Activity-2** Step-1

Acknowledge Step-4

Acknowledge Step-3

Acknowledge Step-4

Acknowledge Step-4

ms-payment

ms-order

ms-shipment

ms-inventory

Record-payment step-3

Record-shipment step-2

Update inventory step-2

Update Status step-2

Order- DB

Shipment- DB

Inventory- DB

Payment-DB

Call api for sending shipment detail step-3

Call api for payment step-2

Call api to send order confirmation. Step-3

Internal-Micro-Service

Ms-mail

Third-Party-Service

Shipment-Logistic-Service

Third-Party-Service

Payment-Gateway-Service