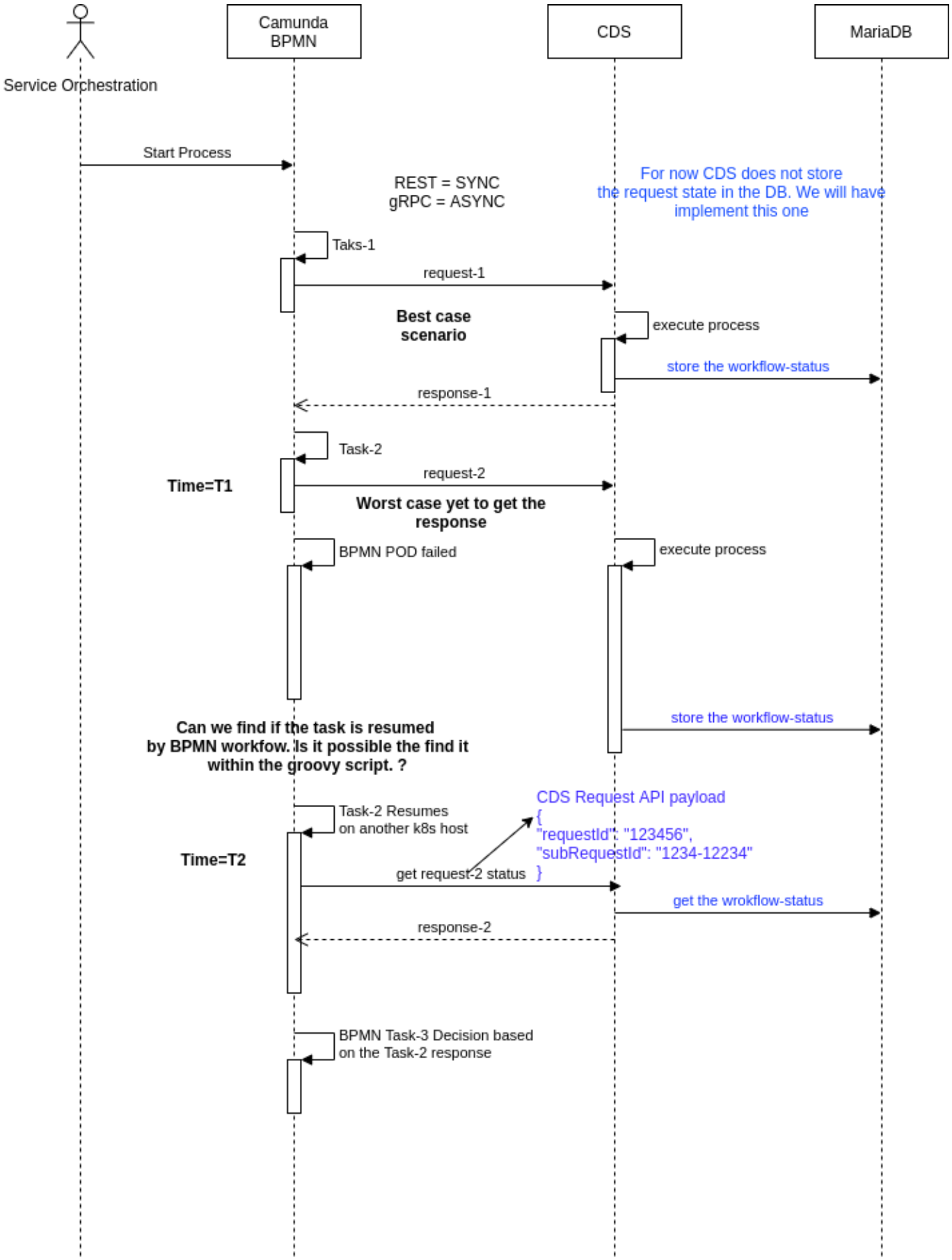
CDS Workflow status Persistence Design

CDS does not persist the workflow execution status now. This page describes the Camunda BPMN end to end service orchestration workflow and the related issues when CDS executes long running workflows.

We brought up these issues with the CDS community and they wants us to come up with a design to solve this problem.

The idea is that CDS should maintain the state of every single workflow execution using the **request-id** and **sub-request-id.** These two attributes are required NB fields for a CBA workflow execution over REST OR gRPC. So, having the workflow status persisted in CDS DB will enable clients to retrieve the workflow status in future. It will also helps us to meet the audit logging requirements for every single CBA workflow execution as well.

Problem : SO Camunda BPMN CDS Integration Issues:


Service Orchestration
Camunda BPMN
CDS
MariaDB
Start Process
REST = SYNC
gRPC = ASYNC
For now CDS does not store the request state in the DB. We will have implement this one
Taks-1
request-1
Best case scenario
execute process
store the workflow-status
response-1
Task-2
Time=T1
request-2
Worst case yet to get the response
BPMN POD failed
execute process
store the workflow-status
Can we find if the task is resumed by BPMN workfow. Is it possible the find it within the groovy script. ?
CDS Request API payload
{
"requestId": "123456",
"subRequestId": "1234-12234"
}
Task-2 Resumes on another k8s host
Time=T2
get request-2 status
get the wrokflow-status
response-2
BPMN Task-3 Decision based on the Task-2 response

BLUEPRINT_WORKFLOW_AUDIT_STATUS

This is new CDS table . Following attributes will be stored in this new CDS table

1. CDS NB request payload attributes - originator_Id ,request_Id, subRequest_Id, workflow_name , status, blueprint_version, blueprint_name, request_mode
2. Audit attributes - start_time, end_time, updated_date, updated_by
3. Response attributes : status, response output,

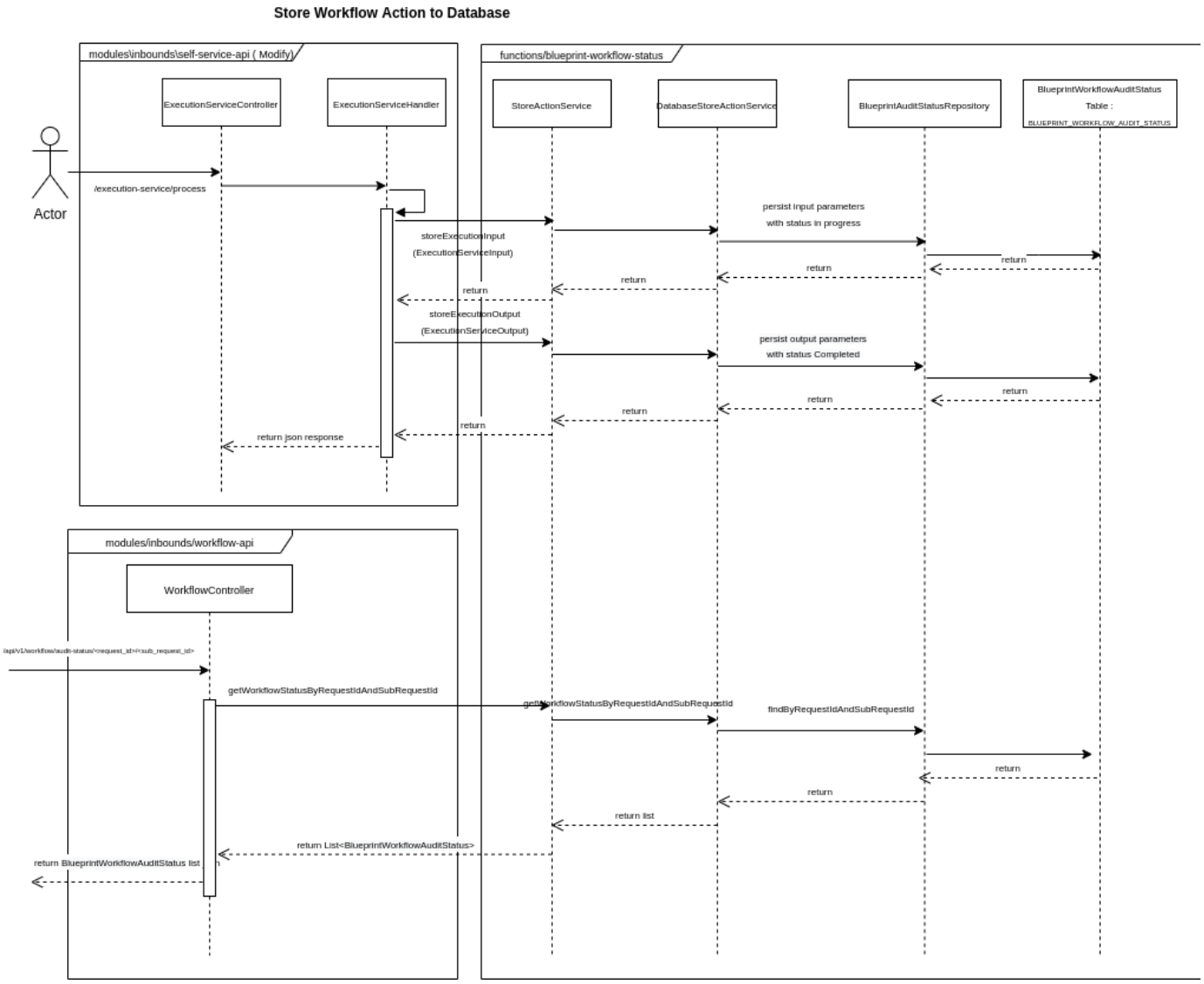
Field	Type	Null	Key	Default
workflow_audit_id	varchar(255)	NO	PRI	NULL
workflow_task_content	longtext	NO		NULL
originator_Id	varchar(255)	NO		NULL
request_Id	varchar(255)	NO		NULL
subRequest_Id	varchar(255)	NO		NULL
workflow_name	varchar(255)	NO		NULL
status	varchar(255)	YES		NULL
start_time	datetime	YES		NULL
end_time	datetime	YES		NULL
updated_date	datetime	YES		NULL
updated_by	varchar(255)	YES		NULL
blueprint_version	varchar(255)	YES		NULL
blueprint_name	varchar(255)	YES		NULL
request_mode	varchar(255)	YES		NULL
workflow_response_content	longtext	YES		NULL
blueprint_uuid	varchar(255)	YES		NULL

```
CREATE TABLE BLUEPRINT_WORKFLOW_AUDIT_STATUS (workflow_audit_id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE PRIMARY KEY,
workflow_task_content longtext NOT NULL,
originator_Id varchar(255) NOT NULL,
request_Id varchar(255) NOT NULL,
subRequest_Id varchar(255) NOT NULL,
workflow_name varchar(255) NOT NULL,
status varchar(255) NULL,
start_time datetime NULL,
end_time datetime NULL,
updated_date datetime NULL,
updated_by varchar(255) NULL,
blueprint_version varchar(255) NOT NULL,
blueprint_name varchar(255) NOT NULL,
request_mode varchar(255) NULL,
workflow_response_content longtext NULL,
blueprint_uuid varchar(255) NULL) AUTO_INCREMENT = 1000;
```

Set the starting value as 1000 for auto increment

```
ALTER TABLE BLUEPRINT_WORKFLOW_AUDIT_STATUS AUTO_INCREMENT = 1000;
```

Blueprint Action Workflow sequence diagram:



Rest APIs:

1. Retrieve records based on Request and Subrequest ID

		Return
GET	/api/v1/workflow/health-check	success or failure
GET	/api/v1/workflow/audit-status/{requestId}/{subRequestID}	List `<BlueprintWorkflowAuditStatus>`

Helm Chart Changes:

Add following in oom/kubernetes/cds/components/cds-blueprints-processor/resources/config/application.properties

#Workflow store configuration
#workflow Audit request
blueprintsprocessor.workflow.self-service-api.audit.storeEnable=false

TODO:

1. Request Payload - Handle sensitive data persistence
2. Retrieve and persist UUID while storing records into `BLUEPRINT_WORKFLOW_AUDIT_STATUS` table