**Resource Control Operations**

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Register Composer Session |
| Actors | Composability Manager, Sunfish-Redfish Session Service, Event Service |
| Description | Enable the Composability Manager from Sunfish and start the Redfish Session Service |
| Input Data | IP address of the running Sunfish service |
| Pre Conditions | The Sunfish service is running and the RESTful interface is available, the Composability Manager has a running session |
| Post Conditions | A session token is created for the Composability Manager |
| Trigger | Composability Manager start-up |
| Normal Flow | * curl -k -X POST 'https:///redfish/v1/SessionService/Sessions/' \ -H 'X-Auth-Token: ' * Return success * Record created showing Composability Manager connected * Record created providing the new Token   + MessageID, event type=’Register Composer’, resource\_id, token, description=N/A, severity=N/A, proposed resolution=N/A, resource connections=N/A, path=N/A, time\_stamp |
| Alternate Flow 1 | * #> curl -k -X POST 'https:///redfish/v1/SessionService/Sessions/' \ -H 'X-Auth-Token: ' * Return failure * Record failed creating showing the Composability Manager connect failed   + MessageID, event type=’Register Composer’, resource\_id, token=N/A, description=N/A, severity=N/A, proposed resolution=N/A, resource connections=N/A, path=N/A, time\_stamp |
| Alternate Flow 2 | * #> curl -k -X POST * 'https:///redfish/v1/SessionService/Sessions/' \ -H 'X-Auth-Token: ' * Sunfish service not reached * Timeout * Failure * Record created showing the Composability Manager disconnect failed |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Disconnect Composer Session |
| Actors | Composability Manager, Sunfish-Redfish Session Service, Event Service |
| Description | Disconnect the Composability Manager from Sunfish and start the Redfish Session Service |
| Input Data | IP address of the running Sunfish service |
| Pre Conditions | The Sunfish service is running and the RESTful interface is available, the Composability Manager has a running session |
| Post Conditions | A session token is revoked for the Composability Manager |
| Trigger | Composability Manager shutdown/disconnection request |
| Normal Flow | * curl -k -X DELETE 'https:///redfish/v1/SessionService/Sessions/' \ -H 'X-Auth-Token: '' * Return success * Record created showing Composability Manager disconnected   + MessageID, event type=’Register Composer’, resource\_id, description=N/A, severity=N/A, proposed resolution=N/A, resource connections=N/A, path=N/A, time\_stamp |
| Alternate Flow 1 | * curl -k -X DELETE 'https:///redfish/v1/SessionService/Sessions/' \ -H 'X-Auth-Token: '' * Return failure * Record failed creating showing the Composability Manager disconnect failed   + MessageID, event type=’Register Composer’, resource\_id, token, description=N/A, severity=N/A, proposed resolution=N/A, resource connections=N/A, path=N/A, time\_stamp |
| Alternate Flow 2 | * curl -k -X DELETE 'https:///redfish/v1/SessionService/Sessions/' \ -H 'X-Auth-Token: '' * Sunfish service not reached * Timeout * Failure * Record created showing the Composability Manager disconnect failed |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Get Request from Client |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Get Request from Client |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Send Aggregated Resource Information |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Client Request Fulfilled |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—List Logged Events |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Query Inventory |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Create System |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Assemble System |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Delete System |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Show System |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Assemble System |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Modify System |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Reserve Inventory |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Resource Authorization Modification |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Add Resource to Database |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Delete Resource Namespace |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Create Aggregated Resource |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Request Resource QoS |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Modify Resource Data |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |
| Use-Case Description | Composability Manager—Gather Resource Security Information |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Get Resource Information Request from Composition Policies |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Send Resource Connections to Composition Policies |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Request Best Resource Composition from Composition Policies |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Request Resource Composition Decision |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Choose best Resource Failover Options from Composition Policies |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Assemble System |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Gather Resource Failover from Composition Policies |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Release Resource Failover Option |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Query Resource Features |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Query Resource  Connections |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Query Resource Path |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Build Resource Path |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Modify Resource Path |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Delete Resource Path |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Create Resource Namespace |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

**Events**

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Assemble\_System |
| Actors | Composability Manager, Sunfish, Database, Get\_Request\_from\_Client, Gather\_Resource\_Security\_Information, Query\_Resource\_Connections, Choose\_Best\_Resource\_Failover\_Options, Build\_Resource\_Path, Create\_Aggregated\_Resource, Modify\_System, Client\_Request\_Fulfilled |
| Description | Assemble an Aggregated Resource out of the Free Pool of Resources, build a resource to client path, update the System, and update Sunfish and the database |
| Input Data | Client requirements: CPU architecture, Memory Device Type, Storage Capacity, Storage Type, Accelerator Type, Network Interface Types, best choice for resource selection from Composition Decisions and Policies, available resources from the database |
| Pre Conditions | OFMF contains a Free pool of Resources, network Agents active |
| Post Conditions | Composed Turing Compatible System from Free Pool, Active Pool incremented by Composed Resources |
| Trigger | Client request for fully Composed Resources |
| Normal Flow | * Is this request for a dynamic expansion to a running allocated server? * Is this request for dynamic expansion to an unallocated server for batch job allocation? * Receive Client Requirements * g.V().has (‘<property>’ to provide appropriate resources from Janusgraph * GET current Free Pool resources from the Janusgraph database * Get best choice for resource selection from Decisions and Policies Block * Create a framework package of allocation requirements, using the Client Requirements, the available Free Pool Resources, and input from the Decisions and Policies Block   + Create JSON     - CPU       * Type of CPU(s)       * Quantity of CPU(s)     - Memory       * Type of Memory       * Amount of Memory     - Storage       * Type of Storage     - Resource endpoints       * What network links are available?       * What networks?       * Aggregated? * Associate Components with links into the Sunfish-Redfish/Swordfish Tree * POST Constrained Composable JSON to Sunfish   + Post of Free Resources to /redfish/v1/CompositionService/ResourceZones and /redfish/v1/CompositionService/ResourceBlocks * POST used resources to Sunfish-Active Resources * g,addE(‘property’).from vertex to vertex path * Update vertex and edge information in the Decisions and Policies Block   Return success |
| Alternate Flow 1 | * Receive Client Requirements * .V().has (‘<property>’ to provide appropriate resources from Janusgraph * Get current Free Pool resources from the Janusgraph database * Resources don’t exist to fulfill requirements   Return failure |
| Alternatie Flow 2 | * Receive Client Requirements * .V().has (‘<property>’ to provide appropriate resources from Janusgraph * Get current Free Pool resources from the Janusgraph database * Resources don’t exist to fulfill requirements, locally * Get best choice for resource selection from Decisions and Policies Block * Create a framework package of JSON Constrained requirements, using the Client Requirements and the available Free Pool Resources * Warning to the client that the requested resources are remote   + Create JSON     - CPU       * Type of CPU(s)       * Quantity of CPU(s)     - Memory       * Type of Memory       * Amount of Memory     - Storage       * Type of Storage     - Resource endpoints       * What network links are available?       * What networks?       * Aggregated? * Associate Components with links into the Redfish Tree * POST Constrained Composable JSON to Sunfish   + Post of Free Resources to /redfish/v1/CompositionService/ResourceZones and /redfish/v1/CompositionService/ResourceBlocks * POST used resources to Active Resources * Return success |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager---Report\_new\_resource\_and\_create\_new\_graph\_entry |
| Actors | Composability Manager, Janusgraph database, Execute\_Change |
| Description | Report new resource and enter the new resource into the graph database |
| Input Data | MessageID, event type, resource, description, severity, proposed resolution, resource connections, path, date, time |
| Pre Conditions | Composability Manager is running, Janusgraph database is running, Events are registered with Sunfish-Redfish service |
| Post Conditions | Report is sent to Composability Manager Event clients, graph database entry is updated |
| Trigger | Call is made from Execute\_Change |
| Normal Flow | * Report resource issue to clients * If issue is on the resource   + Call Search\_Graph for resource   + Call Modify\_Resource\_Detail * Return success |
| Alternate Flow 1 | * Report resource issue to clients * If issue is a connection   + Call Search\_Graph for edge   + Call Modify\_Resource\_Detail   + Call Delete\_Resouce\_Edge   + Call Modify Resource\_Path * Return success |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager----Report\_resource\_issue\_and\_modify\_graph\_entry |
| Actors | Composability Manager, Janusgraph database, Execute\_Change |
| Description | Report event issue or resolution, modify the graph database with new information |
| Input Data | MessageID, event type, resource, description, severity, proposed resolution, resource connections, path, date, time |
| Pre Conditions | Composability Manager is running, Janusgraph database is running, Events are registered with Sunfish-Redfish service |
| Post Conditions | Report is sent to Composability Manager Event clients, graph database entry is updated |
| Trigger | Call is made from Execute\_Change |
| Normal Flow | * Report resource issue to clients * If issue is on the resource   + Call Search\_Graph for resource   + Call Modify\_Resource\_Detail * Return success |
| Alternate Flow 1 | * Report resource issue to clients * If issue is a connection   + Call Search\_Graph for edge   + Call Modify\_Resource\_Detail   + Call Delete\_Resouce\_Edge   + Call Modify Resource\_Path * Return success |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager----Execute\_Change |
| Actors | Composability Manager, Janusgraph database, Log\_Event, Report\_new\_resource\_and\_create\_graph\_entry, Report\_deleted\_resource\_and\_delete\_graph\_entry, Report\_resource\_issue\_and modify\_graph\_entry |
| Description | Report event, resource, description, severity, resource connections, and path is logged time and date and execute appropriate graph database operation |
| Input Data | MessageID, event type, resource, description, severity, proposed resolution, resource connections, path, date, time |
| Pre Conditions | Composability Manager is running, Janusgraph database is running, Events are registered with Sunfish-Redfish service |
| Post Conditions | Report is sent to Composability Manager clients and graph database is updated with new information |
| Trigger | Call is made from Log\_Event block |
| Normal Flow | * Parse the Event type * If Event type is create, call Report\_deleted\_resource\_and\_create\_graph\_entry * Return success |
| Alternate Flow 1 | * Parse the Event type * If Event type is delete, call Report\_new\_resource\_and\_delete\_graph\_entry * Return success |
| Alternate Flow 2 | * Parse the Event type * If Event type is issue or solution, call * Report\_resource\_issue\_and modify\_graph\_entry |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager----Log Event |
| Actors | Composability Manager, Cassandra Database, Sunfish Event Service |
| Description | Log MessageID, event, resource, description, severity, resource connections, and path is logged time and date |
| Input Data | MessageID, event type, resource, description, severity, proposed resolution, resource connections, path, date, time |
| Pre Conditions | Composability Manager is running, Cassandra database is running, Cassandra database log table created, Events are registered with Sunfish-Redfish service |
| Post Conditions | MessageID, event type, resource, description, severity, proposed resolution, resource connections, and path is logged with date and time |
| Trigger | Sunfish signals an event has occurred |
| Normal Flow | * Check to see if the table is created * Record MessageID, event type, resource, description, severity, proposed resolution, resource connections, path, date, time * Execute\_Change with log information |
| Alternate Flow 1 | * Check to see if the MessageID, event type, resource, description, severity, proposed resolution, resource connections, and path are supplied * If inputs aren’t complete, retry * If retries fail, return error * Record MessageID, event type, resource, description, severity, proposed resolution, resource connections, path, date, time * Execute\_Change with log information |
| Alternate Flow 2 | * Check to see if table is created * If table isn’t complete, create table * Record MessageID, event type, resource, description, severity, proposed resolution, resource connections, path, date, time * Execute\_Change with log information |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Resource-Events-List-Available-Events |
| Actors | Composability Manager Resource Control Operations-List-Logged-Events |
| Description | Provide a current running list of Resource Events |
| Input Data | Beginning time period, ending time period |
| Pre Conditions | Composability Manager is running, events are registered with Sunfish service |
| Post Conditions | Logs are searched for and dumped to List-Log-Events block |
| Trigger | Client request for logs during a time period |
| Normal Flow | * Check to see if the event type, begin time, and end time are supplied * Call List\_Available\_Events with: event type (all, specific), begin time, end time * SELECT \* FROM clicks WHERE campaign\_id=? AND date\_time>=? AND date\_time <=? * Return logs to List-Logged-Events |
| Alternate Flow 1 | * Check to see if the event type, begin time, and end time are supplied * If inputs aren’t complete, return error |
| Alternate Flow 2 | * Check to see if the event type, begin time, and end time are supplied * Call List\_Available\_Events with: event type (all, specific), begin time, end time * SELECT \* FROM clicks WHERE campaign\_id=? AND date\_time>=? AND date\_time <=? * Return successful empty logs to client |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—List Logged Events |
| Actors | Composability Manager Resource Control Operations, Client, Resource Events |
| Description | Gather a current running list of Resource Events |
| Input Data | Beginning time period, ending time period |
| Pre Conditions | Composability Manager is running, events are registered with Sunfish service |
| Post Conditions | A session token is deleted for the Composability Manager |
| Trigger | Client request for logs during a time period |
| Normal Flow | * Check to see if the event type, begin time, and end time are supplied * Call List\_Available\_Events with: event type (all, specific), begin time, end time * Return logs to client |
| Alternate Flow 1 | * Check to see if the event type, begin time, and end time are supplied * If inputs aren’t complete, return error |
| Alternate Flow 2 | * Check to see if the event type, begin time, and end time are supplied * Call List\_Available\_Events with: event type (all, specific), begin time, end time * Return successful empty logs to client |
|  |  |

**Resource Graph Representation**

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager---Search Graph |
| Actors | Composability Manager, Janusgraph database, Resource Control Operations, Resource Events Framework, Composition Policies, Authorization Block |
| Description | Search the graph database for vertices and edge information |
| Input Data | ResourceID, ComposerID, MessageID, event type, resource, description, severity, proposed resolution, resource connections, path, tenancy, security value, security association |
| Pre Conditions | Composability Manager is running, Janusgraph database is running |
| Post Conditions | Query results are provided to requestor |
| Trigger | Call is made for a graph database query |
| Normal Flow | * Gather query request variable, variable type, resource or connection * If query is a resource   + g.V().has(variable type,request variable).out * Return success |
| Alternate Flow 1 | * Gather query request variable, variable type, resource or connection * If query is a connection   + g.E().has(variable type,request variable).out * Return success |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager---Create Resource Vertex |
| Actors | Composability Manager, Janusgraph database, Resource Control Operations, Resource Events Framework, Composition Policies, Authorization Block |
| Description | Add a resource to the graph database with Sunfish details, as a graph Vertex |
| Input Data | ResourceID, ComposerID, MessageID, resource, description, severity, proposed resolution, resource connections, path, tenancy, security value, security association |
| Pre Conditions | Composability Manager is running, Janusgraph database is running, Sunfish service is running |
| Post Conditions | A resource vertex is added to the graph database |
| Trigger | Call is made for a new graph database resource addition |
| Normal Flow | * Gather resource name and input data details * g.addV(‘resource’,).property((‘ComposerID’,0),   (‘MessageID’),0),(‘severity’,0),(‘proposed resolution’),‘proposed resolution’’),(’resource connections’,’resource connections’),(‘path’,’path’),(‘tenancy’,’tenancy’),  (‘security value’,’security value’),(‘security association’,’security association’))   * return success |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager---Create Resource Edge |
| Actors | Composability Manager, Janusgraph database, Resource Control Operations, Resource Events Framework, Composition Policies, Authorization Block |
| Description | Add a resource to the graph database with Sunfish details, as a graph Edge |
| Input Data | ResourceID, ComposerID, MessageID, resource, description, severity, proposed resolution, resource connections, tenancy, security value, security association |
| Pre Conditions | Composability Manager is running, Janusgraph database is running, Sunfish service is running |
| Post Conditions | A resource edge is added to the graph database |
| Trigger | Call is made for a new graph database resource addition |
| Normal Flow | * Gather resource name and input data details * g.addE(‘connection’,).property((‘ComposerID’,0),   (‘MessageID’),0),(‘severity’,0),(‘proposed resolution’),‘proposed resolution’’),(’resource connections’,’resource connections’),(‘path’,’path’),(‘tenancy’,’tenancy’),  (‘security value’,’security value’),(‘security association’,’security association’)).from(g.V(id1)).to(g.V(id(2))   * return success |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager---Delete Resource Edge |
| Actors | Composability Manager, Janusgraph database, Resource Control Operations, Resource Events Framework, Composition Policies, Authorization Block |
| Description | Delete a resource to the graph database with Sunfish details, as a graph Vertex |
| Input Data | ResourceID |
| Pre Conditions | Composability Manager is running, Janusgraph database is running, Sunfish service is running |
| Post Conditions | A resource edge is deleted from the graph database |
| Trigger | Call is made for a graph database resource deletion |
| Normal Flow | * Gather resource name and input data details * g.E(MessageID).drop * return success |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager---Delete Resource Vertex |
| Actors | Composability Manager, Janusgraph database, Resource Control Operations, Resource Events Framework, Composition Policies, Authorization Block |
| Description | Delete a resource to the graph database with Sunfish details, as a graph Vertex |
| Input Data | ResourceID |
| Pre Conditions | Composability Manager is running, Janusgraph database is running, Sunfish service is running |
| Post Conditions | A resource vertex is deleted from the graph database |
| Trigger | Call is made for a graph database resource deletion |
| Normal Flow | * Gather resource name and input data details * g.E(MessageID).drop * return success |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager---Delete Resource Edge |
| Actors | Composability Manager, Janusgraph database, Resource Control Operations, Resource Events Framework, Composition Policies, Authorization Block |
| Description | Delete an edge to the graph database with Sunfish details, as a graph Vertex |
| Input Data | ResourceID |
| Pre Conditions | Composability Manager is running, Janusgraph database is running, Sunfish service is running |
| Post Conditions | A resource edge is deleted from the graph database |
| Trigger | Call is made for a graph database path deletion |
| Normal Flow | * Gather resource name and input data details * g.E(MessageID).drop   gremlin> graph = TinkerFactory.createModern()  ==>tinkergraph[vertices:6 edges:6]  gremlin> g = graph.traversal()  ==>graphtraversalsource[tinkergraph[vertices:6 edges:6], standard]  gremlin> g.V(1).bothE()  ==>e[9][1-created->3]  ==>e[7][1-knows->2]  ==>e[8][1-knows->4]  For purpose of the example, let's say we want to drop edges between vertex 1 and vertex 2. We could find those with:  gremlin> g.V(1).bothE().where(otherV().hasId(2))  ==>e[7][1-knows->2]  and then remove it with:  gremlin> g.V(1).bothE().where(otherV().hasId(2)).drop()  gremlin> g.V(1).bothE()  ==>e[9][1-created->3]  ==>e[8][1-knows->4]  If you have the actual vertices, then you could just do:  gremlin> g.V(v1).bothE().where(otherV().is(v2)).drop()  gremlin> g.V(1).bothE()  ==>e[9][1-created->3]  ==>e[8][1-knows->4]   * return success |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager---List Resource Detail |
| Actors | Composability Manager, Janusgraph database, Resource Control Operations, Resource Events Framework, Composition Policies, Authorization Block |
| Description | List a resource in the graph database with Sunfish details |
| Input Data | ResourceID or ComposerID or MessageID or resource, or resource connections |
| Pre Conditions | Composability Manager is running, Janusgraph database is running, Sunfish service is running |
| Post Conditions | A resource detail is printed out |
| Trigger | Call is made for a graph database resource property set |
| Normal Flow | * g.V().hasLabel(‘ResourceID’,ResourceID).values(‘ComposerID’,   ’MessageID’, ‘resource’, ‘description’, ‘severity’, ‘proposed resolution’, ‘resource connections’, ‘tenancy’, ‘security value’, ‘security association’)   * return success |
| Alternate Flow 1 | * g.V().has(‘ComposerID’,ComposerID).values(‘ResourceID’,   ’MessageID’, ‘resource’, ‘description’, ‘severity’, ‘proposed resolution’, ‘resource connections’, ‘tenancy’, ‘security value’, ‘security association’)   * return success |
| Alternate Flow 2 | * g.V().has(‘MessageID’,MessageID).values(‘ResourceID’,   ‘ComposerID’, ‘resource’, ‘description’, ‘severity’, ‘proposed resolution’, ‘resource connections’, ‘tenancy’, ‘security value’, ‘security association’)   * return success |
| Alternate Flow 3 | * g.V().has(‘tenancy’,tenancy)..values(ResourceID, ComposerID, MessageID, resource, description, severity, proposed resolution, resource connections, security value, security association) |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager---Modify Resource Detail |
| Actors | Composability Manager, Janusgraph database, Resource Control Operations, Resource Events Framework, Composition Policies, Authorization Block |
| Description | Modify a resource in the graph database with new Sunfish details |
| Input Data | ResourceID, ComposerID, MessageID, resource, description, severity, proposed resolution, resource connections, tenancy, security value, security association |
| Pre Conditions | Composability Manager is running, Janusgraph database is running, Sunfish service is running |
| Post Conditions | A resource vertex is modified in the graph database |
| Trigger | Call is made for a graph database resource property modification |
| Normal Flow | * v = g.V().hasLabel('ResourceID').has('ResourceID', ResourceID).next() * g.V(v).property(ComposerID’,ComposerID).property(‘ResourceID’,   ’MessageID’, ‘resource’, ‘description’, ‘severity’, ‘proposed resolution’, ‘resource connections’, ‘tenancy’, ‘security value’, ‘security association’)   * return success |
| Alternate Flow 1 | * g.V().has(‘ComposerID’,ComposerID).property(‘ResourceID’,   ’MessageID’, ‘resource’, ‘description’, ‘severity’, ‘proposed resolution’, ‘resource connections’, ‘tenancy’, ‘security value’, ‘security association’)   * return success |
| Alternate Flow 2 | * g.V().has(‘ComposerID’,ComposerID).property(‘ResourceID’,   ’MessageID’, ‘resource’, ‘description’, ‘severity’, ‘proposed resolution’, ‘resource connections’, ‘tenancy’, ‘security value’, ‘security association’)   * return success |
| Alternate Flow 3 | * g.V().has(‘tenancy’,tenancy).property(ResourceID, ComposerID, MessageID, resource, description, severity, proposed resolution, resource connections, security value, security association) |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager---Modify Resource Path |
| Actors | Composability Manager, Janusgraph database, Resource Control Operations, Resource Events Framework, Composition Policies, Authorization Block |
| Description | Modify a resource path in the graph database with new Sunfish details |
| Input Data | ResourceID, ComposerID, MessageID, resource, description, severity, proposed resolution, resource connections, tenancy, security value, security association |
| Pre Conditions | Composability Manager is running, Janusgraph database is running, Sunfish service is running |
| Post Conditions | A resource path is modified in the graph database |
| Trigger | Call is made for a graph edge modification |
| Normal Flow | * v = g.V().hasLabel('ResourceID').has('ResourceID', ResourceID).next() * Edge e = graph.addEdge(v1, v2, "visits"); * return success |
| Alternate Flow 1 | * v = g.V().hasLabel('ResourceID').has('ResourceID', ResourceID).next() * ResourceID doesn’t exist * Return failure |
| Alternate Flow 2 |  |
| Alternate Flow 3 |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Modify Resource Group of Details |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |

|  |  |
| --- | --- |
| Use-Case Description | Composability Manager—Add Resource Edge |
| Actors |  |
| Description |  |
| Input Data |  |
| Pre Conditions |  |
| Post Conditions |  |
| Trigger |  |
| Normal Flow |  |
| Alternate Flow 1 |  |
| Alternate Flow 2 |  |
|  |  |