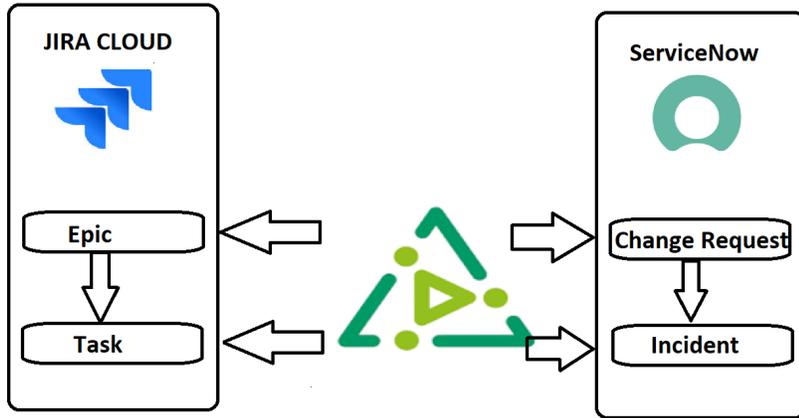


Jira Cloud ServiceNow: Maintaining Issue Hierarchy

The main requirement for this use case is to maintain the issue hierarchy while Exalating issues between Jira and ServiceNow (bi-directionally).

The Epic Task relationship must be reflected as Change Request Incident relationship on ServiceNow, and vice versa. The following depicts what we are trying to achieve here:



Let us first consider the Jira to ServiceNow side:

- The only extra bit of information that Jira needs to send out is the Epic Link, and it can be sent by adding the following line in Outgoing Scripts on the Jira side:

```
replica.customFields."Epic Link" = issue.customFields."Epic Link"
```

- On the ServiceNow side, we need to firstly map the issue types. This can be easily done using the following code segment on the Incoming Script on the ServiceNow side:

```
if(firstSync){
  if (replica.type.name == "Epic")
    entity.tableName = "change_request"
  else if (replica.type.name == "Task")
    entity.tableName = "incident"
}
```

- The next step on the ServiceNow side would be populate the rfc field of the Incident to ensure that the hirarchy is correctly established. This can be achived by employing the [getLocalIssueKeyFromRemoteId](#) method to look for the corresponding local parent:

```
entity.rfc = syncHelper.getLocalIssueKeyFromRemoteId(replica.parentId).idStr
```

Let us now consider the ServiceNow to Jira direction:

- The key to maintaining the hierarchy is again the rfc field of the incident. If this field is populated (i.e. there is a relationship we need), we employ the [getTableByLink](#) method to extract the sys_id of the parent:

```
if (entity.rfc)
  replica.parentid = nodeHelper.getTableByLink(entity?.rfc?.link)?.sys_id
```

- Next we need to ensure that the correct mappings are present on the Jira Incoming scripts:

```
if(firstSync){
    issue.projectKey = "CM"
    if (replica.snow_type == "incident")
        issue.typeName = "Task"
    else if (replica.snow_type == "changerequest"){
        issue.typeName = "Epic"
        issue.customFields."Epic Name".value = replica.summary
    }
}
```

- And to close this off, we will again employ `getLocalIssueKeyFromRemoteId` to fetch the local parent and maintain the hierarchy:

```
if(replica.parentid){
    def localParent = syncHelper.getLocalIssueKeyFromRemoteId(replica.parentid, 'change_request').id
    if(localParent)
        issue.parentId = localParent
    else
        throw new com.exalate.api.exception.IssueTrackerException("Cant field parent with id " + replica.
parentid)
}
```

The entire code from the Jira side is [here](#), and for ServiceNow is [here](#).

Please review the following video to see the use case in action:



Jira ServiceNow -...directionally.mp4