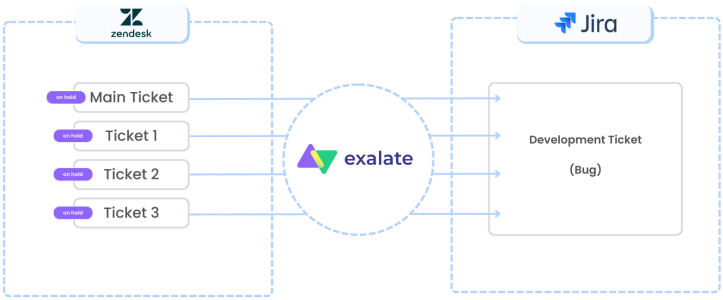
# Zendesk Jira OnPrem: Multiple Zendesk tickets linked to a single Jira issue

This is a very common use case encountered by Support and Development teams. If the support team receive a ticket and decide to report it to Development as a bug, then all subsequent reports of this same issue (by different customers probably) should continue to be linked to this same Development bug.

The use case will cover the following:

1. Support receives the first report of the issue and Exalates it to development via a macro.
2. Development starts working the bug
3. Any subsequent support tickets received for similar bug are Exalated to the same development bug (instead of creating multiple issues)
4. All support tickets are automatically set to On-hold once they are Exalated (assuming that support does not need to work on it for now)
5. Any comments added by developers will be reflected as internal notes in Zendesk
6. Once the development ticket is finalized and marked as Done,
   all related ZD tickets will get an update informing them of this and all Zendesk ticket status will go back to Open

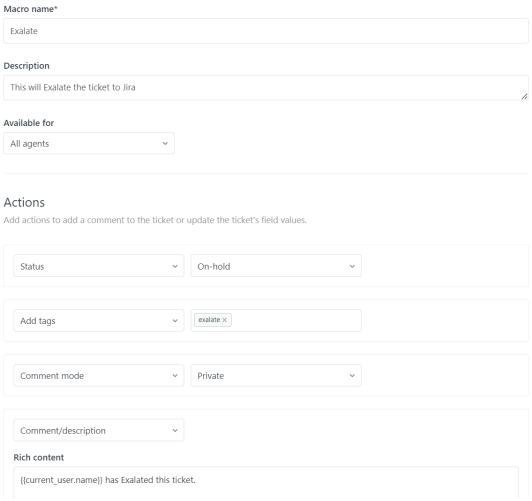
The following depicts this:




Configuration required for this use case:

- Create a Macro in Zendesk that take the following actions:
  - Adds the tag "exalate" to the ticket
  - Changes the status of the ticket to on-hold

- Add an internal note to the ticket




- On the Zendesk use a custom field called Key that will be populated by the support agent before Exalating the ticket.
  This will be populated with the Jira ticket number of the bug that this Zendesk ticket is to be linked with.
  In the outgoing sync script on Zendesk, include the value of this field in the replica:

```
replica.customFields."Key" = issue.customFields."Key"
```

- On the Jira Incoming script, we read the value of this field and employ the ComponentAccessor to search for the local issue corresponding to this key:

```
import com.atlassian.jira.component.ComponentAccessor
if(firstSync){
   issue.projectKey   = "SYEDR"
   issue.typeName     = "Bug"
   def remoteIssueUrn = replica.customFields."Key"?.value
   if (remoteIssueUrn){
        def issueManager = ComponentAccessor.issueManager
        def mainIssue = issueManager.getIssueByCurrentKey(remoteIssueUrn)
        issue.id = mainIssue.id
    }
    else{
        issue.summary = replica.summary
        issue.description  = replica.description
        syncHelper.syncBackAfterProcessing()
    }
}
```

The full Jira Incoming script can be downloaded here.

Following is a brief presentation of this use case in action: