

[Xray] Federated Environments and Jira Federation

- [Overview](#)
 - [Use cases](#)
 - [Features](#)
 - [Limitations](#)
- [Before creating a federated environment](#)
- [Setting up a federated environment](#)
- [Using Xray in a federated environment](#)
 - [Supported and unsupported features](#)
 - [Notes and Recommendations](#)
 - [Possible use cases](#)
 - [Unfeasible use cases](#)
- [Copy information between instances](#)
 - [One-time, on-demand operations](#)
 - ["Live" Synchronization](#)
- [Notes](#)
- [Resources](#)

Overview

A federated environment is one where several autonomous instances of Atlassian products are connected to each other, normally sharing the same users. Interconnectable products are not restricted to Jira and may include Confluence, Bitbucket, Crowd, etc.

[blocked URL](#)

A federated Jira environment is **not** similar to having one big, consolidated Jira instance; that would be better achieved with a data center deployment.

Use cases

- public (e.g. for customers) and private Jira instance (e.g. for development)
- geolocated instances, for data sovereignty or configuration reasons
- manage instances with totally different levels of risk/concerns/tuning/installed apps
- constraint performance impacts (e.g. due to the number of issues or different and high-demanding usage patterns)
- more [here](#)

Features

- the ability to link issues between Jira instances
- unified activity stream (if users are "shared" across instances)
- possibility of easily switching between instances with unified access

Limitations

- searching (e.g. issues using JQL) is limited to the current instance; a workaround is to use a client tool
- creating/editing issues is limited to the current instance, although users can quickly jump to the remote issue screen
- unable to seeing remote projects and issues in the current instance
- configurations (except user management if you want) are still managed locally, at each instance

Before creating a federated environment

As depicted in [this thread](#) and on the answer provided by Mattias Gaiser, there are some points you should evaluate before creating a federated environment:

- What applications do you want to have in the federated environment?
- Do users want to access issues from other instances? Across all projects? Only of specific projects?
- Do they want to link to issues of other projects?
- Do these users need only read access to the issues? Or also write access?
- Do you use the same user base for all Jira instances? Or is this different per instance?
- In sum, what you want to see and "do" on each Jira instance? What is the ultimate purpose of each Jira instance?

Setting up a federated environment

This is the high-level steps

- configure Crowd to manage users access in a centralized point
- [set up application links](#) between the applications

Using Xray in a federated environment

Xray may be used in a federated environment with some restrictions.

Please see ahead of the supported and unsupported features along with the possible and unfeasible use cases.



Please note!

This is not a common scenario and thus has **not** been validated explicitly by the Xray team nor is part of the advertised product features. We advise you to validate your usage scenario before a global rollout.

Supported and unsupported features

Feature	Expect Xray behavior	Notes
central user management and authentication	OK	transparent to Xray
linking remote issues	OK	transparent to Xray; note that
link remote defects from a Test Run	OK	the execution screen allows you to add/link existing remote defects
create remote defects from a Test Run	N/A	creating remoting issues is not possible in a federated Jira environment; issues have to be created first on the destination instance and then they can be linked from the source location (e.g. instance performing the runs)
covering/linking a "requirement" with a Test in an external Jira (i.e linking a "requirement" to a Test issue stored in a different Jira instance)	Unsupported	"requirement" issues (e.g. stories) must be co-located with Test issues. All these entities must be co-located in the same Jira instance: Tests, Pre-Conditions, Test Sets, (Sub)Test Executions, Test Plans, Test Runs and all issues configured to be handled as "requirements" (i.e. coverable issues).
Synchronize the "Requirement Status", "TestRunStatus", "Test Set Status", "Test Execution Defects" or other calculated custom fields to another Jira instance	Unsupported	As these fields are calculated, they depend on historical execution data (i.e. Test Runs) which is not easily synchronizable
scenarios where testing related entities, including "requirements", are stored in different Jira instances	Unsupported	All these entities must be co-located in the same Jira instance: Tests, Pre-Conditions, Test Sets, (Sub)Test Executions, Test Plans, Test Runs and all issues configured to be handled as "requirements" (i.e. coverable issues).
searching across linked instances (e.g. whenever using JQL, for building saved filters, in any issue picked dialog)	N/A	Jira itself does not support searching in a federated Jira environment

Notes and Recommendations

- Use the "all-in-one" approach for organizing your projects (i.e. the issue types somehow related to your project); **keep all the previous entities related to a given project in the same Jira instance**
- With some external app (e.g. [Backbone Issue Synch](#)) or with some custom development, you may synchronize issues between instances; however, this is non-trivial (especially if you decide to implement it by yourself).
 - Xray data is stored in dedicated issue types (e.g. Test, Pre-Condition, Test Set, Test Execution, Test Plan), on custom fields and internal entities managed internally by Xray; thus, copying the issues to the destination Jira instance, or issues and their respective custom fields, may not be enough
 - Copying Test Runs (historical execution related information) is not possible

Possible use cases

- Manage some projects in instance A and manage some other projects in instance B
- Manage the development and testing in instance A and link remote defects created in instance B; in other words, you may have one instance with the external bugs that can be linked to the internal Jira instance where the development project is managed
- Have one instance A where improvements are created; synchronize them to instance B where development is managed and tests are created. Tests can be synched back to instance A just for coverage overview and insights about the testing specification

- Have one instance A where improvements are created; synchronize them to instance B where development is managed and tests are created and executed. Track testing in detail in instance 1; in instance 2, use "remote gadgets" for showing, for example, overall coverage and information about ongoing Test Plans and Test Executions



Learn more

It's possible to have a Jira dashboard with information from connected Jira instances but requires some initial configurations

- You need to import the gadgets from one instance to the other one:
 - a) you may use subscriptions
 - b) or you may manually import each gadget
 - in instance 1: Add gadget => Show XML link => copy to clipboard
 - in instance 2 => Add gadget> Manage gadgets > Gadgets => Add a gadget => paste previously copied URL

- Have one instance A where improvements are created; synchronize them to instance B where development is managed and tests are created and executed. Have visibility of coverage information in instance A by using a custom field such as "Requirement Status"
 - the "Requirement Status" field cannot be copied as such but a text-based representation of it may (please check with the team that implements the synchronization, e.g. Backbone Issue Synch team for example)

Unfeasible use cases

- perform testing in instance A and have visibility about progress, calculated coverage or Test Runs on another instance
- have some testing entities in one instance (e.g. Test Sets) and the related entities (e.g. the Tests that are part of the Test Set) in a different instance
- federated search
- out-of-the-box unified dashboards (dashboards by themselves cannot be shared across Jira instances; a given gadget configuration has only access to the local projects/issues/saved filters); it's possible to include remote gadgets that have access to the remote Jira instance information though
- unified REST API endpoints (each Jira or Xray REST API endpoints deal only with the data of their respective instance)

Copy information between instances

Whenever thinking about copying or synchronizing some sort of information between instances you have to clarify:

- what kind of information you want to copy/synchronize (e.g. issues, issues+custom fields, whole projects, settings)
- the direction of copy/synchronization and how to deal with possible conflicts in bi-directional scenarios
- the amount of information
- delay of synchronization

Please do have look at Atlassian [documentation](#) and work with an Atlassian partner, if possible, to help you with this as it may not be something trivial.

Nevertheless, below you may see a brief sum-up of different levels of "synchronization" along with some possible solutions. We advise you to try solutions (these or other) before going live.

One-time, on-demand operations

- **Project level:**
 - built-in project backup: Xray extends Jira's built-in project backup/restore capabilities, so you can use it to move projects between instances.
- **Issue level:**
 - [Jira Issue Copy](#)

"Live" Synchronization

- **Project level:** unavailable for the project as a whole; however, please see the option(s) for issue level as you may synchronize all the issues belong to the project
- **Issue level:**
 - [Backbone Issue Sync](#) (supports bi-directional synchronization); please check with Backbone Issue Synch the supported capabilities (some info is available [here](#) but should be confirmed).

Notes

- Atlassian recommends Data Center deployments as a way to scale, have redundancy and deal with higher levels of concurrency

Resources

- <https://confluence.atlassian.com/adminJiraserver/federating-Jira-managing-multiple-instances-961257360.html>
- <https://help.k15t.com/backbone-issue-sync/latest/server/backbone-xray-connector-app-165938132.html>
- <https://confluence.atlassian.com/adminJiraserver/using-applinks-to-link-to-other-applications-938846918.html>