Protected Fields

Field-level privacy and encryption for Jira Cloud — so teams keep collaborating in Jira while the values that matter stay visible to a much smaller audience.

Broad collaboration. Narrow data surface.

ENCRYPT	CONTROL	MINIMISE
AES-256 field values, BYOK optional	Separate view and edit rights per field	Cover APIs, exports and automation too

Not every field belongs to everyone.

Jira becomes a system of operational record, and the whole delivery team needs the issue. One or two fields on it may belong only to HR, Security, Legal, Finance or an incident manager. Project permissions are too coarse to say that.

WHERE THE GAP SHOWS UP

01 Over-broad visibility — a user needs the issue context, not every attribute on it.

02 Secondary paths — UI-only protection leaks through APIs, exports and automation.

03 No privilege separation — the right to read is not the right to change.

04 Workarounds — the value gets copied into a second tool, or into a comment.

WHO ACCESS CAN FOLLOW

ASSIGNEE	The current work owner.
REPORTER	Access follows the originator.
PROJECT LEAD	Managerial oversight.
PROJECT ROLES	Project-specific responsibility groups.
GROUPS	Centrally managed cohorts across projects.
PARTICIPANTS	Controlled access in JSM requests.

THE SECURITY OUTCOME

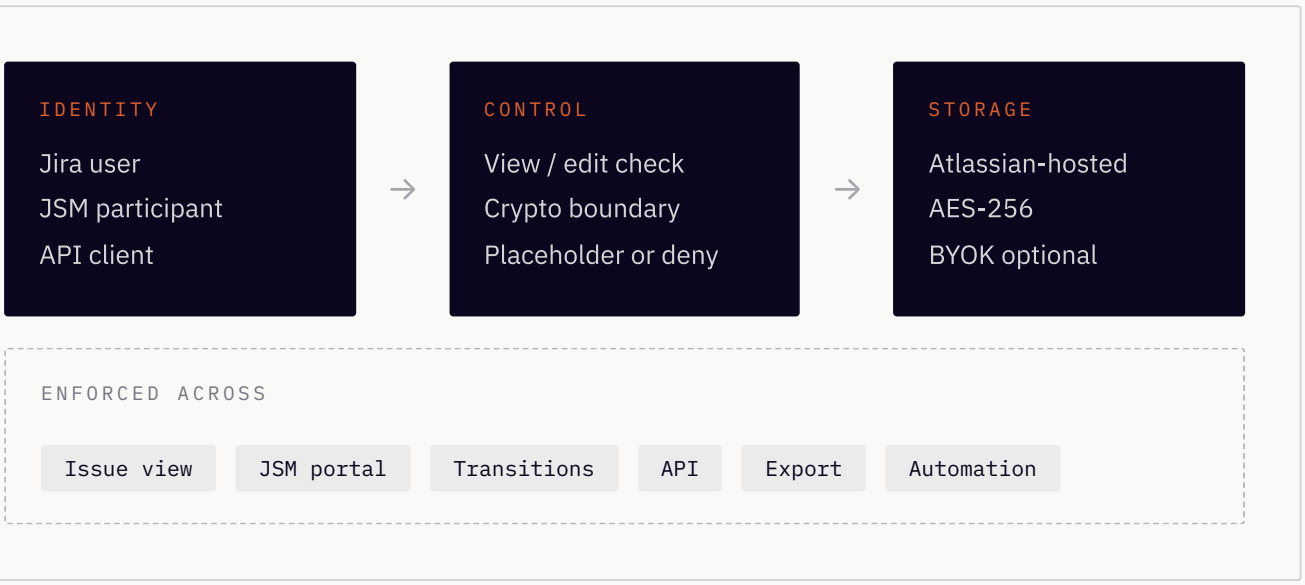
Keep the collaboration surface broad and the data surface narrow. Everyone keeps working in the same issue — only authorised identities can see or change the protected value.

LEAST PRIVILEGE | DEFENCE IN DEPTH | DATA MINIMISATION

02 SECURITY ARCHITECTURE

Protection at the field, not the project.

Protected Fields runs on Atlassian Forge. Control is applied where the value is created, displayed and changed — the surrounding issue stays an ordinary Jira issue.



NO VENDOR ACCESS

App data is stored and backed up inside your Atlassian Cloud instance. Rixter has no access to protected values.

FIELD TYPES

Text, number, date, user / group, select lists, checkboxes and radio buttons — several per project.

CHOOSING THE FIELD MODE

CONSIDERATION	ENCRYPTED	UNENCRYPTED
Primary goal	Maximum confidentiality	Permission control with integration access
API and automation	Designed to block value exposure	Readable by trusted integrations
Encryption	AES-256, BYOK optional	None at field level
Typical fit	Regulated or highly sensitive data	Sensitive workflow data an integration needs

Classify first. Then configure.

A rollout starts with deciding which values genuinely need stronger protection — not with creating fields. That keeps the permission model small enough to stay understandable.

01 Classify

Identify the sensitive values, and confirm they belong in Jira at all.

02 Choose the mode

Encrypted for confidentiality; unencrypted where an integration must read it.

03 Set view and edit

Define both rules using roles, groups or issue relationships — separately.

04 Test every path

Issue view, portal, transitions, API, export and automation — authorised and not.

05 Migrate and clean up

Move existing values in, then delete the obsolete copies left elsewhere.

06 Review periodically

Reassess groups, roles, integrations and whether the field is still needed.

WHERE TEAMS USE IT

SECURITY	HR	LEGAL	FINANCE	SERVICE
Indicators and investigation notes	Employee attributes in a normal ticket	Assessments and case classifications	Commercial amounts and approvals	Agents and participants see different levels

THE TAKEAWAY

The question is not whether Jira should be open or locked down. It is which data needs which level of protection.

Protect the fields that matter. Keep the rest of Jira open.

Tell us which field you cannot safely put in Jira today, and we will show you how it would be protected.

rixter

RIXTER.SE/APPS/PROTECTEDFIELDS
ATLASSIAN MARKETPLACE
SUPPORT@RIXTER.SE

Protected Fields is one control within your wider policy, identity and audit environment; using it does not by itself establish GDPR, ISO 27001 or any other compliance status. Capabilities referenced were verified against Rixter and Atlassian public product material, September 2026.