Asset Studio

Live tables, charts and cards built on Jira Assets — on Jira dashboards for operations, and on Confluence pages for the documentation that explains them.

Model once. Present everywhere.

JIRA DASHBOARDS	CONFLUENCE	ONE MODEL
Operational reporting and shared team views	Live data inside runbooks and knowledge pages	Same Assets source, same filters, same formats

01 THE VISIBILITY GAP

Assets holds the truth. Few people ever see it.

Devices, services, licences, owners, dependencies — modelled properly in Jira Assets. But people need that data in a dashboard, a service review or a runbook, not in the Assets administration interface.

WHAT TEAMS DO INSTEAD

01 Export to a spreadsheet — accurate for a day, then a second source of truth.

02 Leave the dashboard or the page to go look an object up.

03 Rebuild the same CMDB view three times, in three formats.

04 Let the runbook drift away from the estate it describes.

WHAT ASSET STUDIO PUTS IN ITS PLACE

FILTERING	Query by schema, object type and attribute — including IQL.
ATTRIBUTES	Show only the fields this audience needs to decide.
TABLES	Vertical or horizontal, resizable, with border settings.
CHARTS	Pie and bar for grouped populations at a glance.
CARDS	Compact summaries where a table is too much.
REFERENCES	Referenced-object attributes inline, with detail popups.
CROSS-SITE	Assets from other Atlassian Cloud sites can be shown.

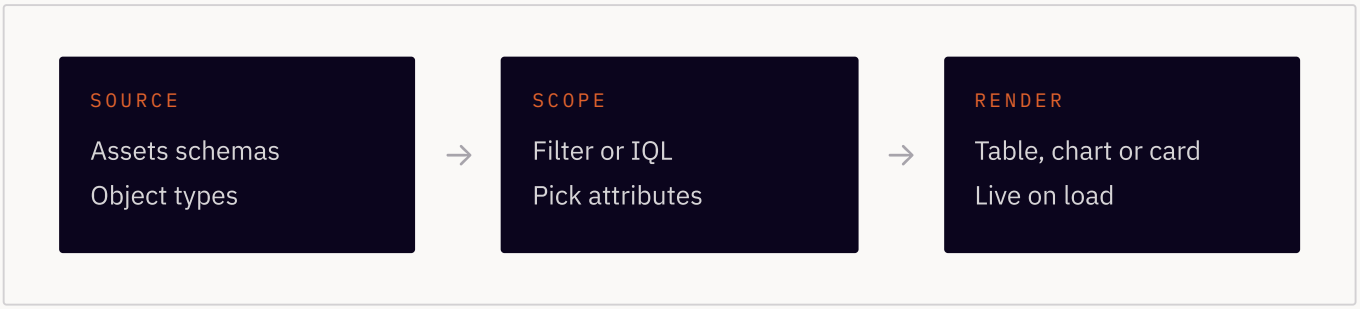
THE DESIGN PRINCIPLE

Jira Assets stays the authoritative model. Jira and Confluence become consumption surfaces — live views, not copies that quietly go stale.

NO DUPLICATE DATA | WIDER AUDIENCE | ONE VISUAL LANGUAGE

Same model. Different room.

The Jira gadget and the Confluence macro ship separately, but they build the same view from the same schema. Dashboards answer *what is happening*; pages explain *what it means and how we work with it*.



Jira dashboard gadget

OPERATE

Shared, at-a-glance visibility on the dashboards teams already open every morning.

Hardware estate — devices by owner, location, lifecycle

Licences — product, allocation, renewal status

Service portfolio — criticality, owner, supporting system

CMDB reviews — a data-quality view for the meeting

Management reporting — charts and cards, no attribute dump

Confluence macro

EXPLAIN

The same objects embedded in the long-lived pages that describe the process around them.

Runbooks — the servers and owners the procedure names

Architecture pages — live services behind the diagram

Inventories — published without a duplicate table

Service docs — description plus current ownership

Audit pages — evidence views, current at the moment of reading

SHARED FORMATS

`Vertical table` `Horizontal table` `Pie` `Bar` `Data card` `API (Confluence)`

Start with the view you maintain by hand.

The fastest first win is the spreadsheet or wiki table someone already updates every month. Replace that one, then decide where the next view belongs.

01 Pick a manual view

One or two reports or pages currently kept current by hand.

02 Reuse the schema

Build on existing object types and relationships — no reporting-only structures.

03 Choose the room

Dashboard if it drives a decision today; Confluence if it needs explaining.

04 Revisit as schemas change

Filters and attribute choices age with the model — review them.

FOUR RULES FOR A GOOD VIEW

AUDIENCE	MINIMUM	SCOPE	PERMISSIONS
Design for the reader, not the administrator	Only the attributes the decision needs	A focused query beats a full object listing	Access stays governed by Atlassian administration

THE TAKEAWAY

One Assets model, two Atlassian surfaces: live operational visibility in Jira, contextual documentation in Confluence — and no second reporting model to maintain.

Show us the spreadsheet you keep updating by hand.

We will map it to your Assets schema and show it as a live dashboard gadget and a Confluence macro.

rixter

RIXTER.SE/APPS/DASHBOARDGADGET
RIXTER.SE/APPS/MACROFORASSETS
SUPPORT@RIXTER.SE

Asset Studio for Jira dashboards and Asset Studio for Confluence are delivered as separate Marketplace apps; capabilities can vary by deployment model and app version. Live views make source-data quality visible — object ownership and lifecycle hygiene remain your own. Reviewed September 2026.