ATLASSIAN

The content described herein is intended to outline our general product direction for informational purposes only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described herein remain at the sole discretion of Atlassian and is subject to change.

DevOps best practices

to better manage your Jira instances

salto

Hello there!

Gil Hoffer
Co-Founder & CTO
Salto

Founded
2019

Offices
TLV, SF

Employees
70

Two funding rounds
$69M

Investors

Bessemer Venture Partners
Lightspeed
salesforce ventures
Accel

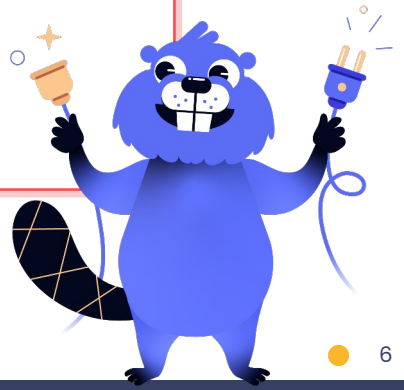
SaaS apps run the modern company

All these apps require deep customizations

salto

Challenges (1/2)

1. How do we really know what is implemented?
2. Why a change was made, by whom, and when?
3. What will be the impact of a planned change? Inside Jira? On other systems?
4. How do we develop changes reliably, without impacting production?



Challenges (2/2)

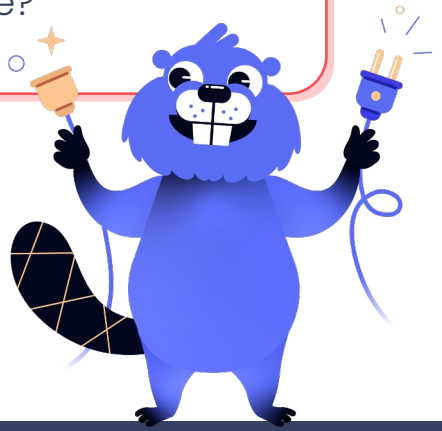
5 How do we collaboratively review those changes in a team and create collective code ownership?

6 How do we reliably promote those changes to production?

9 How do we know what is already broken in our implementation (broken automations, screens, workflows, etc.)

7 How do we roll back, if needed?

8 How do we keep a paper trail for compliance?

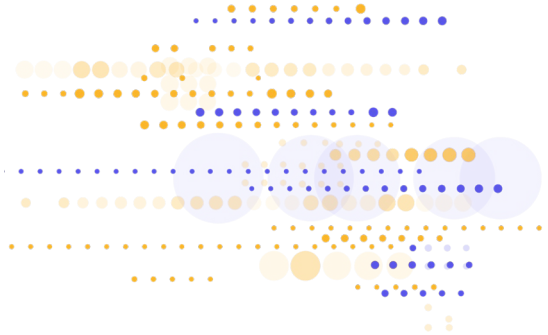


salto

The “as code” concept completely transformed IT

A fusion between development and operations helped IT take a big leap forward

- Manual error-prone processes
- Organizational silos
- Long delivery cycles
- Infrequent releases
- Reduced quality



Automating operational tasks with code

- Faster release cycles
- Shorter delivery sessions
- Increased quality
- Continuous delivery

Why does “as code” matter (subset)?

Version-controlled and immutable, makes auditing easier and rollbacks feasible

Linters and static analysis tools to enforce consistency

Maintainable, testable, and collaborative — reduces the bus factor

Modular, composable, and separate

Predictable, repeatable, and consistent

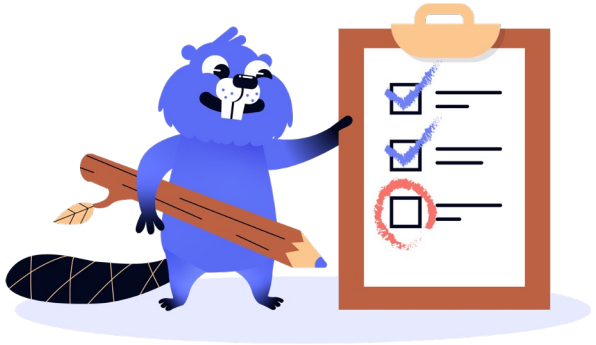
Business engineering

We should adapt best practices, methodologies, and tools from the Dev and DevOps world into how we manage our business applications.

salto

Business Engineering 101 (1/2)

- Having a clear and **complete textual representation of the implementation** (declarative and code) will enable visibility and easily answer “what is implemented?” (As easy as ⌘+F)
- **Version** that representation **in a source control system** (such as Git) and be able to answer questions on changes over time.
- Code analysis tools can **answer** questions on **dependencies** and **change impact.**
- Create **shared responsibility** within the **team** — everyone should be aware of all major changes.

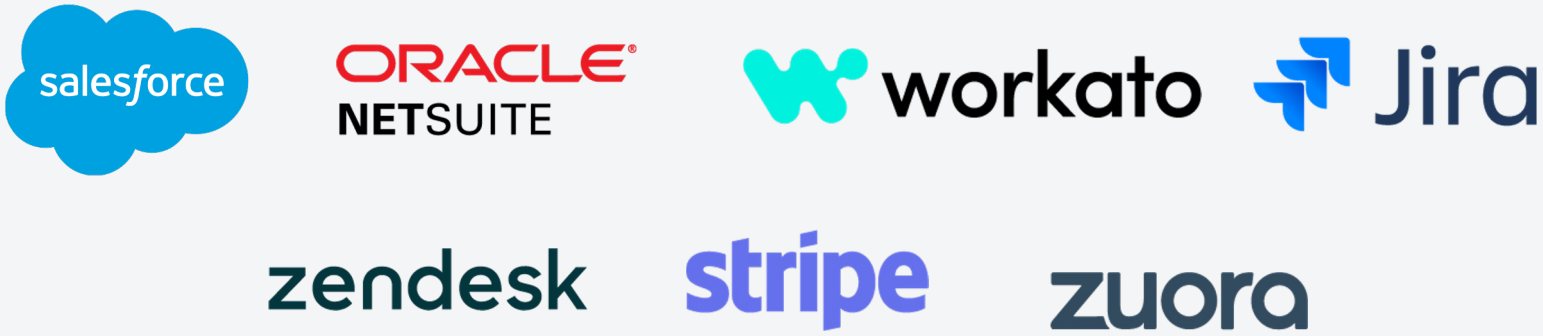


Business Engineering 101 (2/2)

- **Always** develop in high-fidelity **sandboxes.**
  - Not impacting production
  - Not testing on nonrepresentative setup
- A change proposal → **feature branch in Git and pull request.**
- **Promoting a change** from dev to UAT to prod → **git merge and deploy.**
  - The holy grail — **CI/CD**
  - The holier grail — trunk-based development (TBD)
- **Rollback** → Git revert and deploy.
- **Maintain** a proper **paper trail** for changes → tie change requests (for example, from Jira) to Git commits.

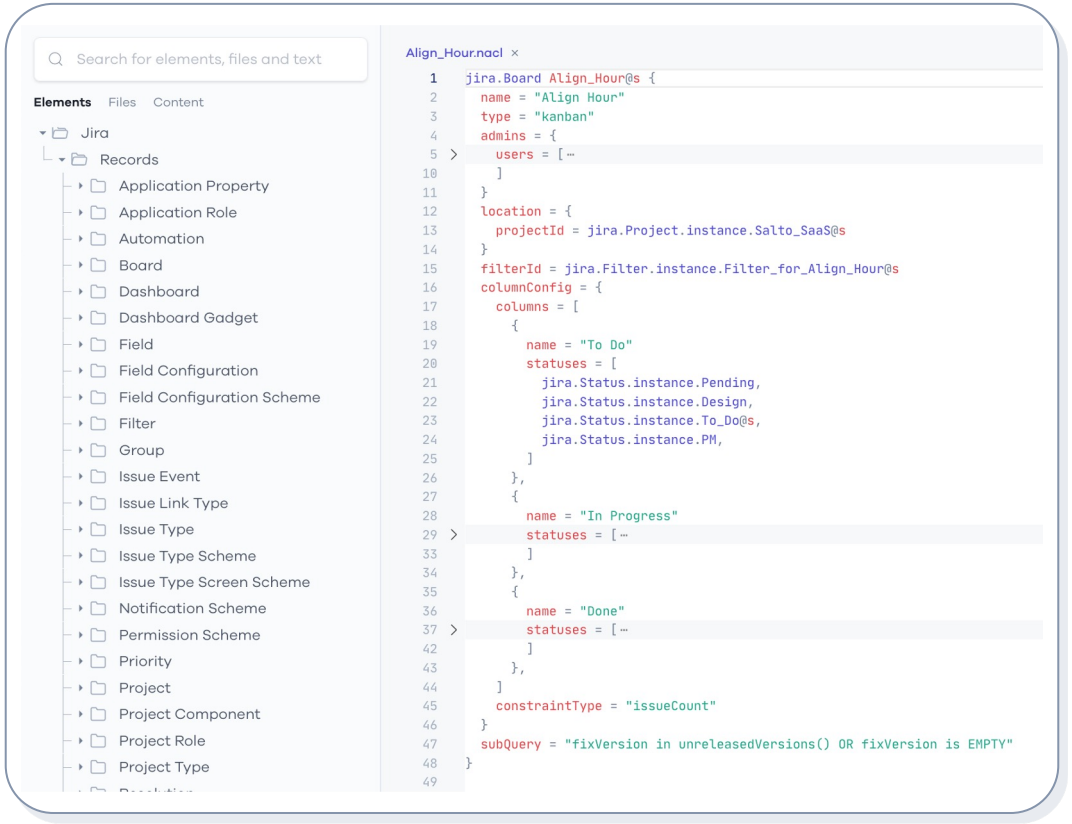
Cross-business application applicability

- Everything we discussed so far is **not** Jira-specific.
- As an industry, we should have a standard way to practice business engineering **across** our entire stack.

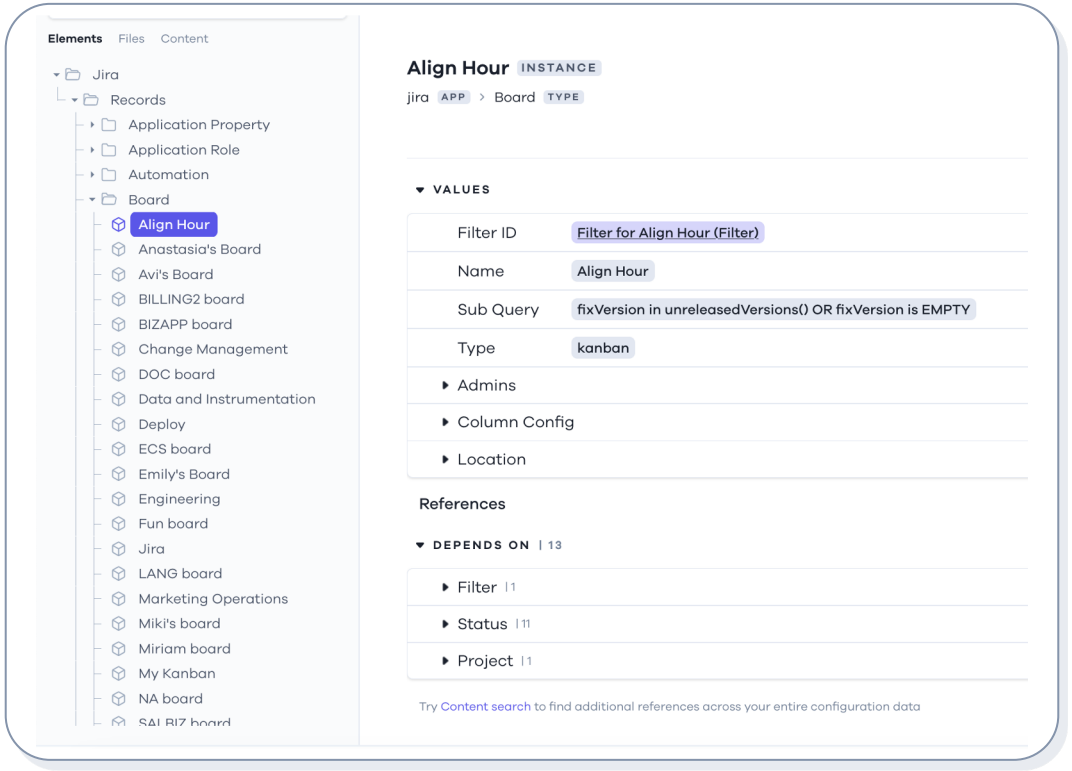


Examples

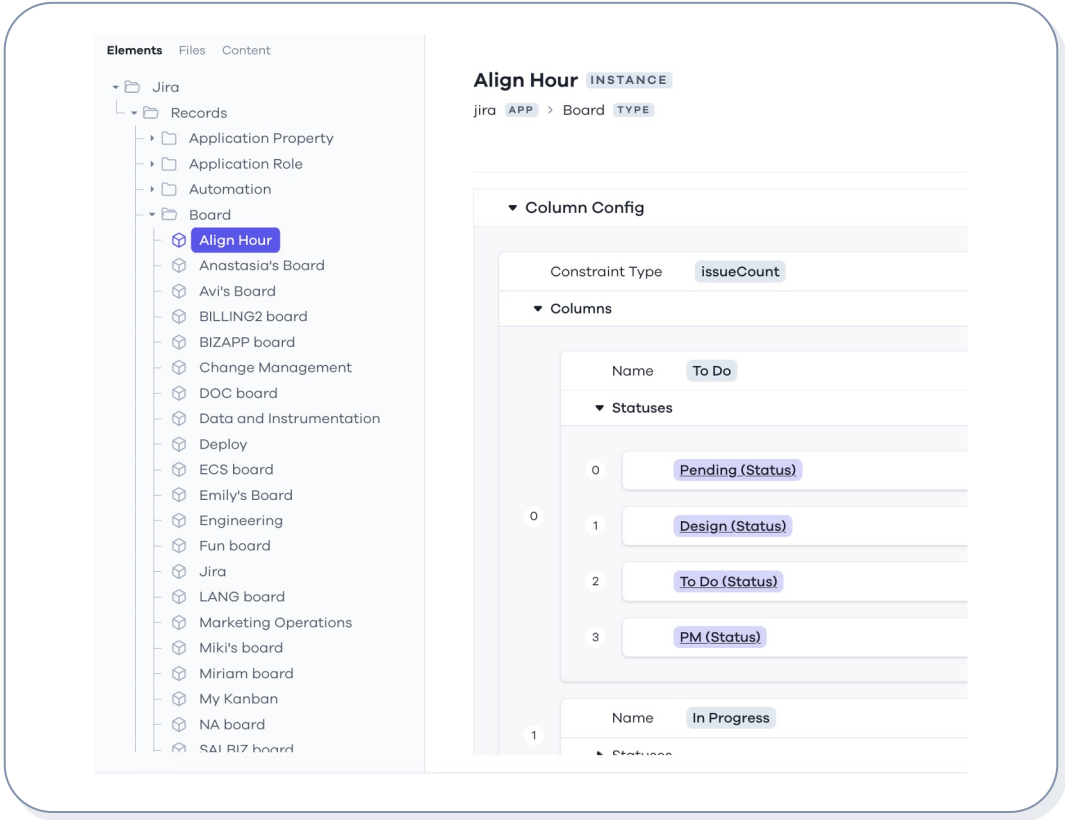
Represent everything “as code”



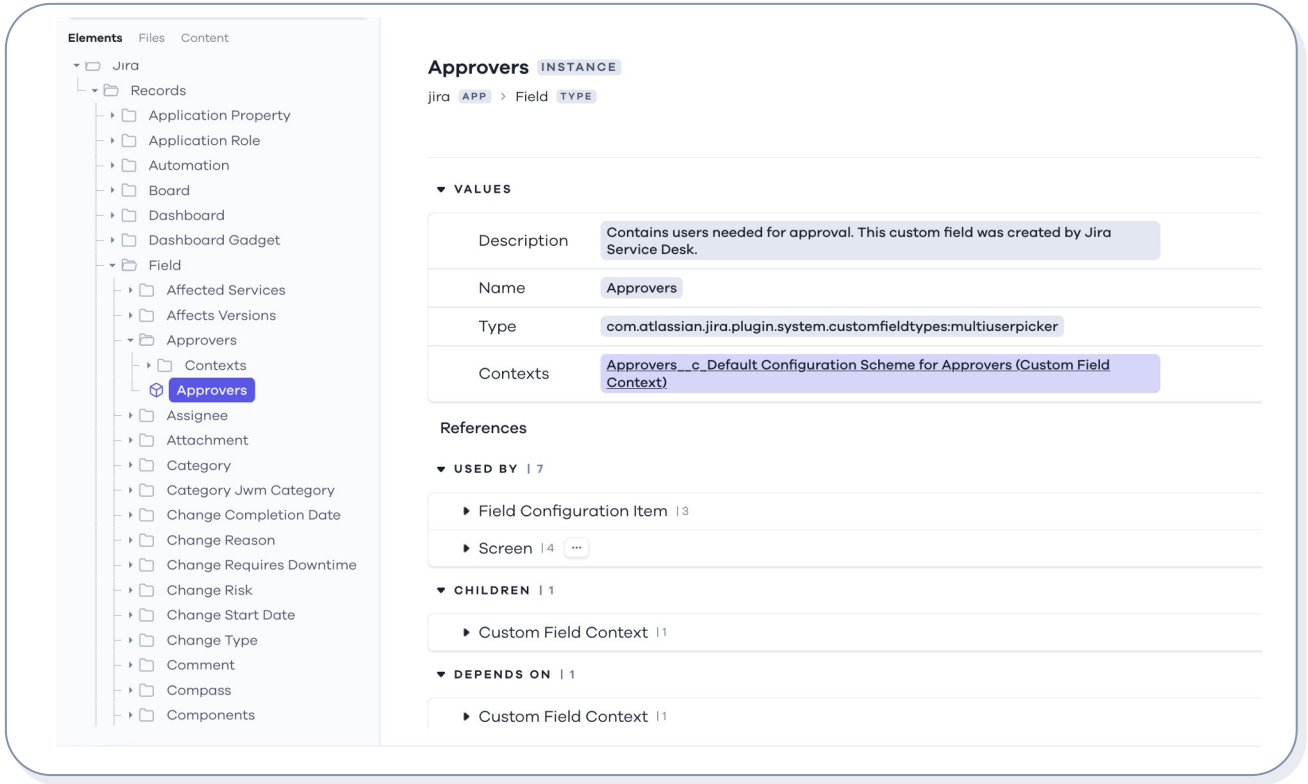
Render the metadata in a friendly way (1/2)



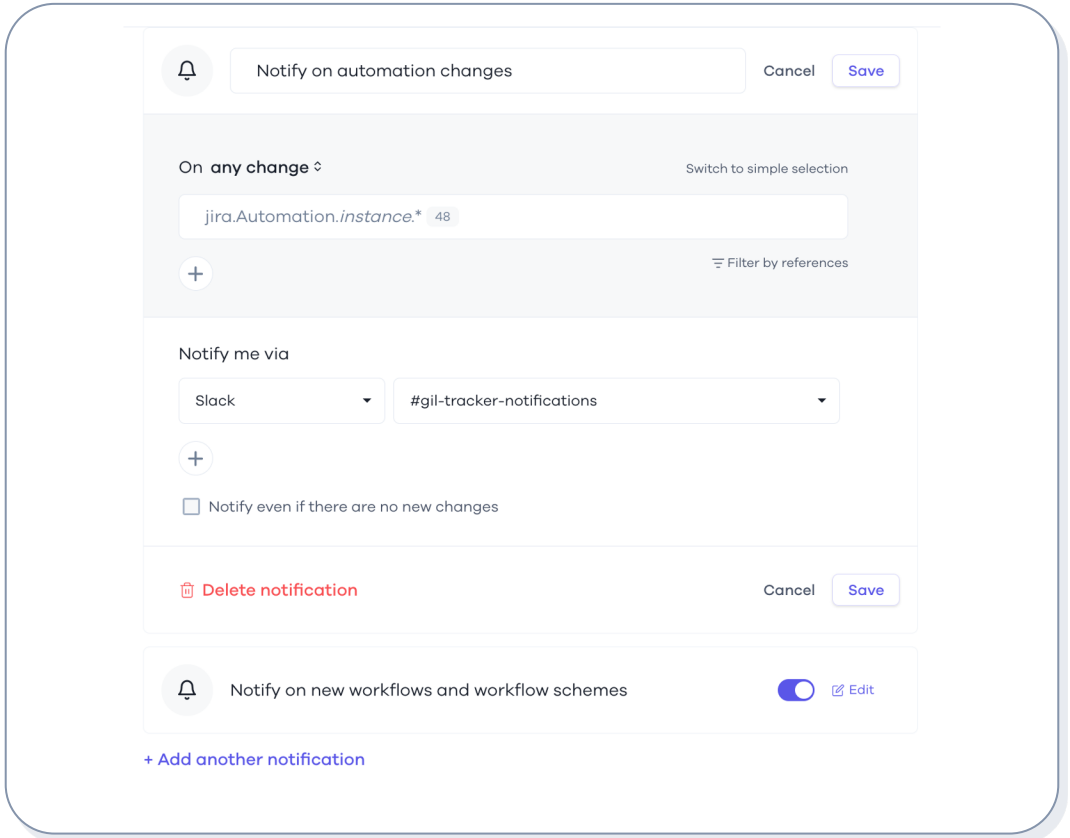
Render the metadata in a friendly way (2/2)



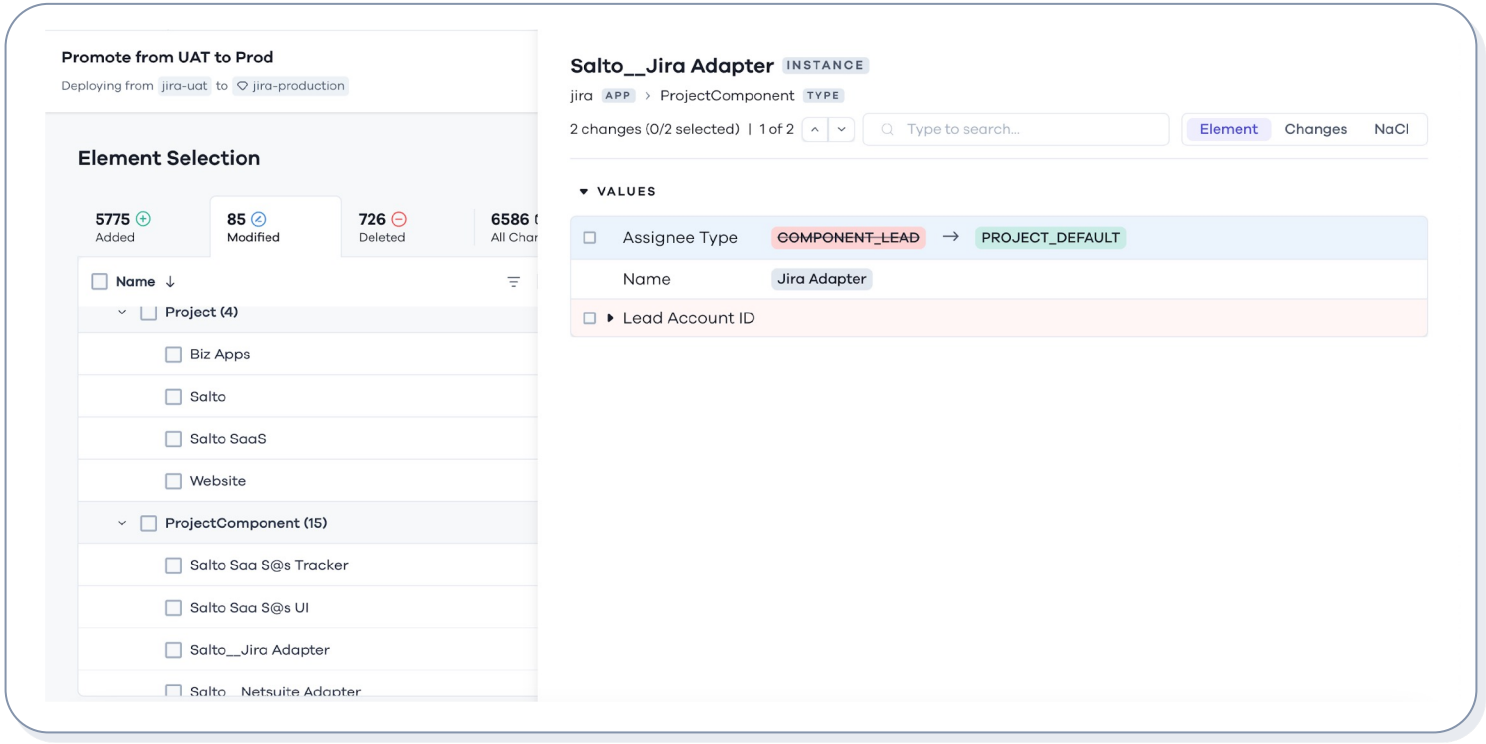
Understand dependencies



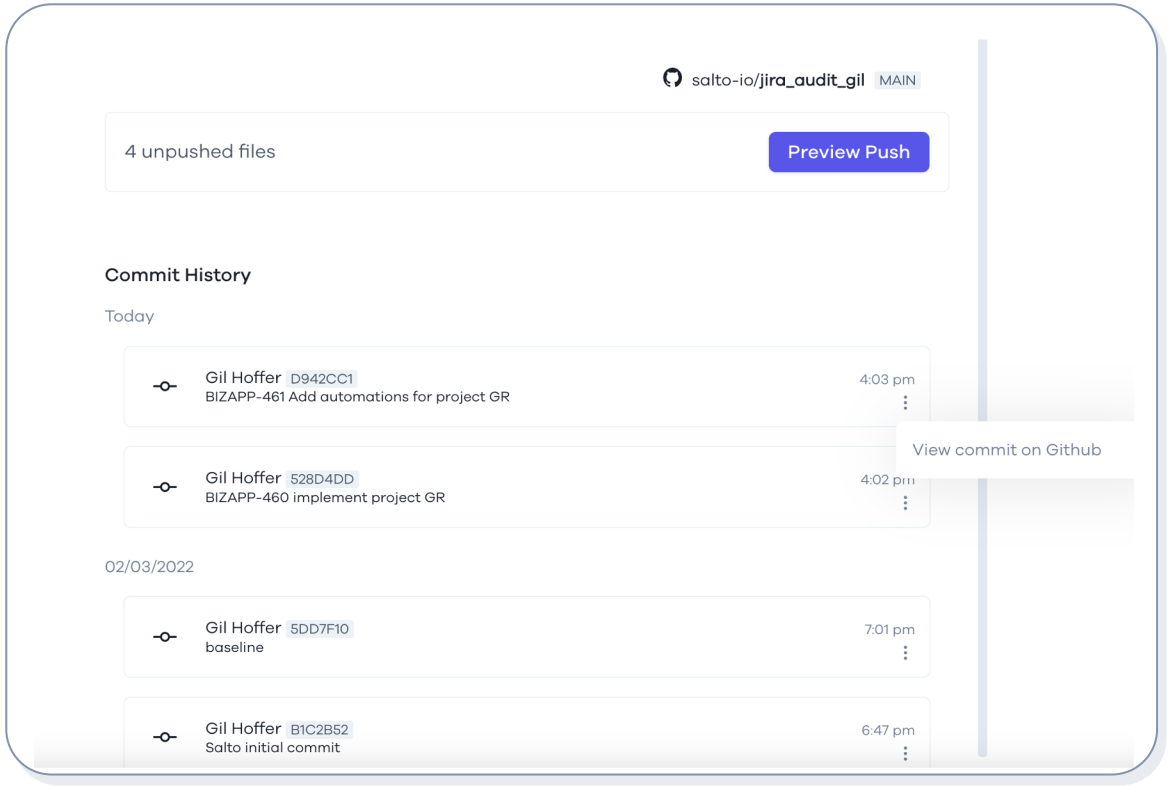
Monitor configuration changes



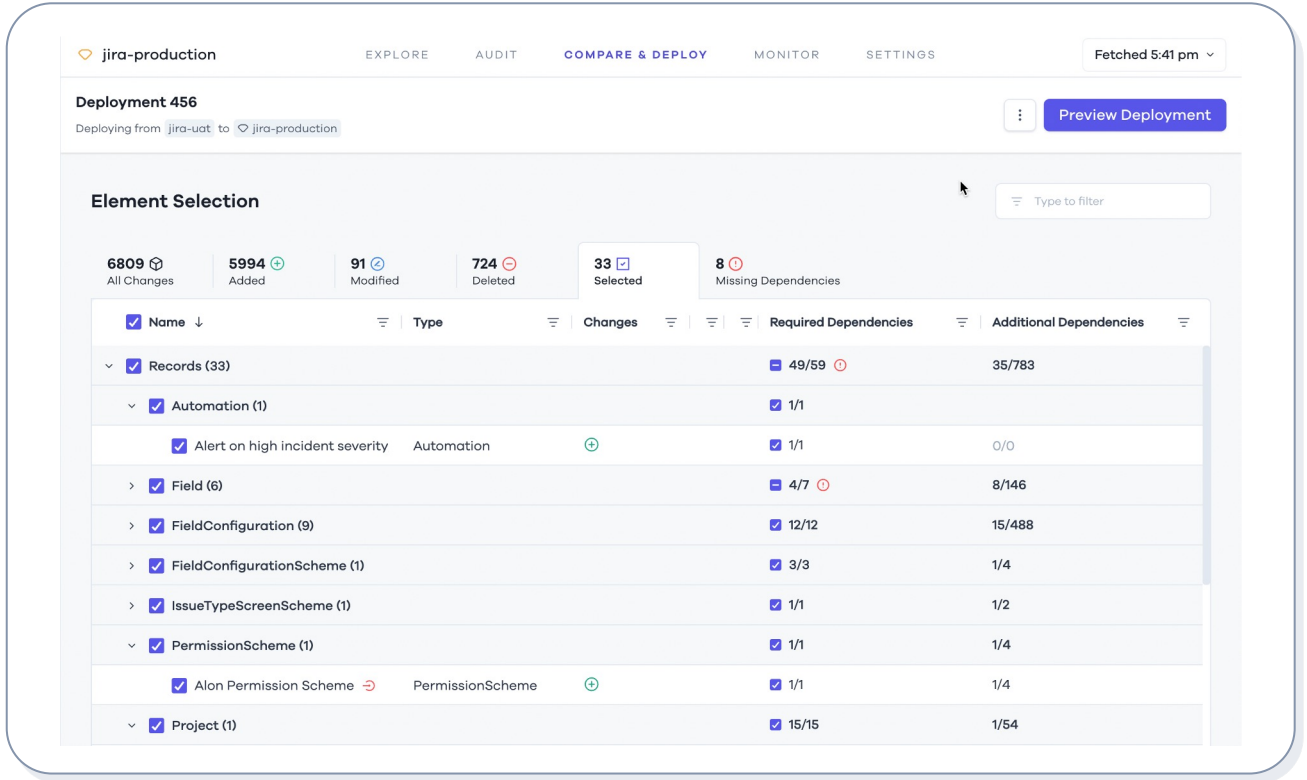
Compare environments



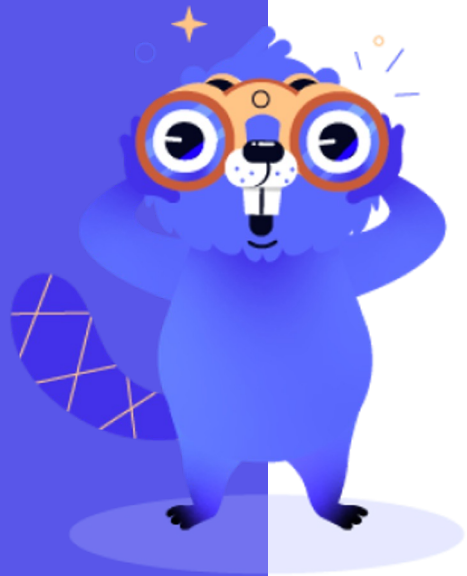
Version control



Deploy changes (from UAT ⊠ Prod)



Recap



Manage your customizations “as code.”
Apply DevOps best practices to the way you manage them.
You can start **today**, even by taking small steps.

salto

Stay in touch

salto.io

@salto-io

@salto_io

Free tier (free for life)

salto