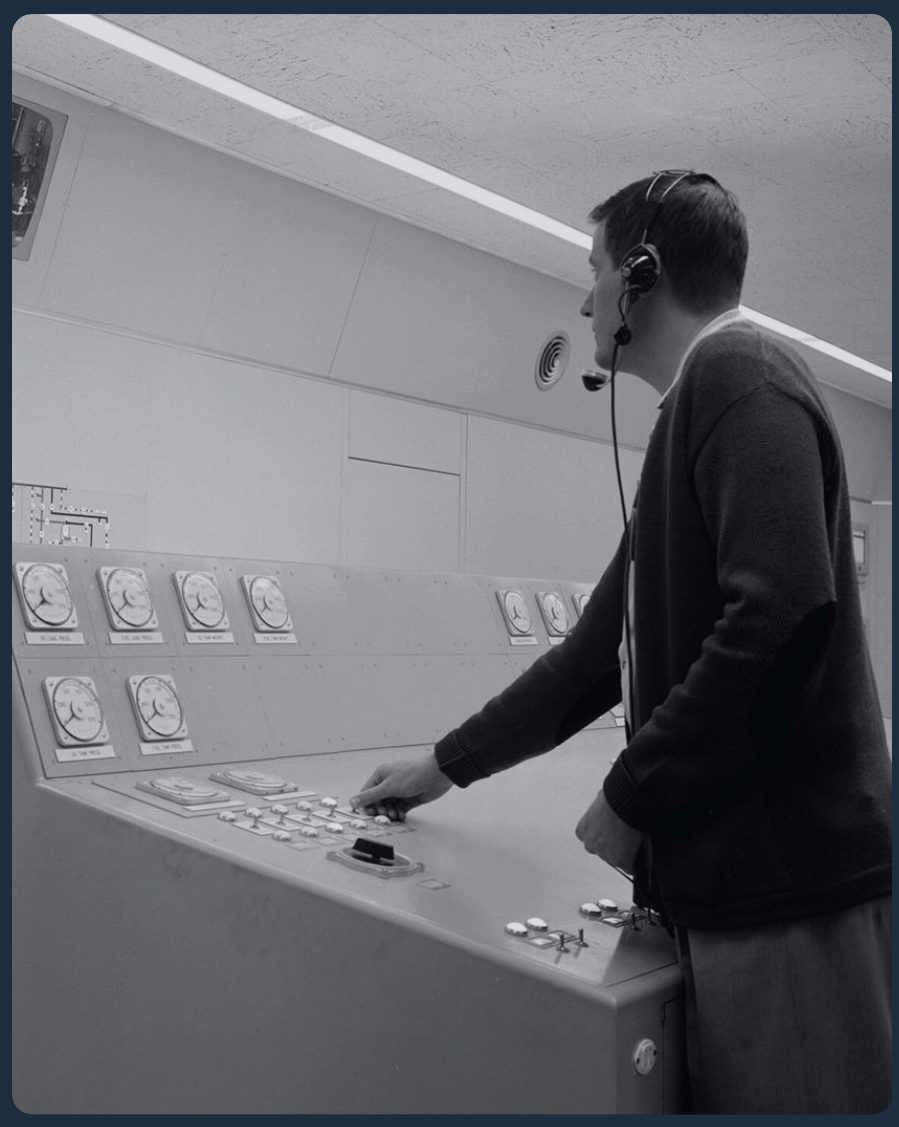
SOFSEK

ACCESS GOVERNANCE AND SEGREGATION OF DUTIES

Know who can do what, across every application you run.

SOFSEK pulls users, roles and entitlements from your connected systems and resolves them into one access model. It tests that model against your segregation of duties policies, tracks every conflict to closure and proves it to your auditor.



Someone has always needed to see who is touching what. NASA, public domain.

One access model
Normalised from every connected system

Policy engine
Severities, controls and governed exceptions

Closed loop
Every finding mitigated, accepted or resolved

Audit ready
Packs mapped to regulatory frameworks



A lock only works if you know who holds the key. Historic American Buildings Survey, public domain.

THE PROBLEM

Every system looks fine alone. The conflict only shows when you add them up.

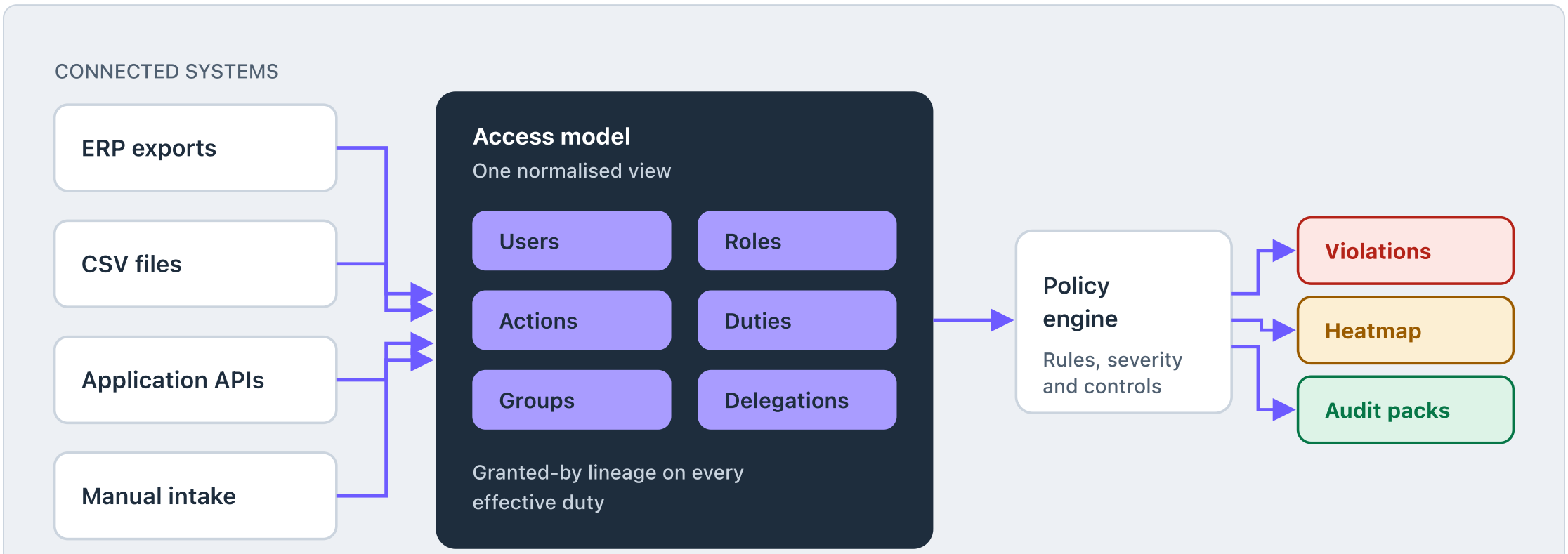
People change jobs, cover for colleagues and collect roles. One person ends up able to raise a payment and approve it, or create a supplier and pay them. No single application sees that.

- **Spreadsheets go stale**
  An access review done once a year describes a company that no longer exists.
- **Roles hide duties**
  A role name tells you little about the actions it really grants.
- **Auditors want evidence**
  Saying it is under control is not enough. You need the history to show it.

ONE CROSS-APPLICATION ACCESS MODEL

Many systems in. One picture out.

SOFSEK ingests users, roles and entitlements from each connected application and normalises them into a single model of users, roles, actions, duties and groups. Delegations and temporary cover belong in it too, so the picture matches how your organisation really works.



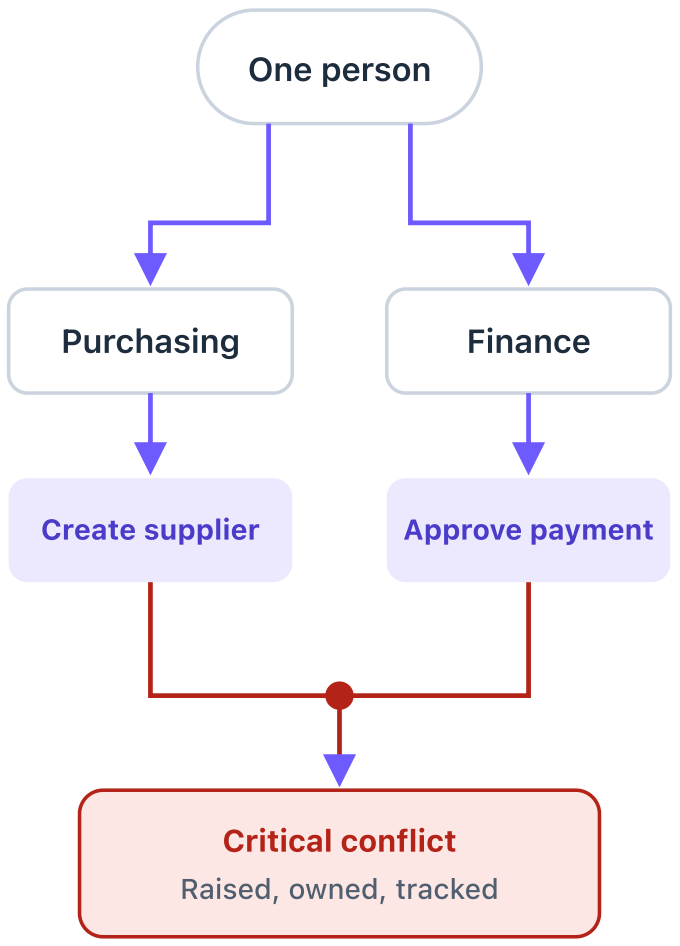
Sources vary. The model does not. Every downstream test runs against the same normalised picture.

POLICY ENGINE

Write the rule once. Test every person against it.

Define which duties must never sit together and how serious the conflict is. Each scan looks at a person's combined access across all their roles and raises a violation when two conflicting duties meet.

- **Cross-role detection**
  The conflict is found even when no single role is at fault.
- **Severity on every rule**
  Critical findings rise above the noise.
- **Mitigating controls**
  Record the control that makes a conflict tolerable, and who owns it.
- **Governed exceptions**
  An exception is never open ended. It has an owner and an expiry date.

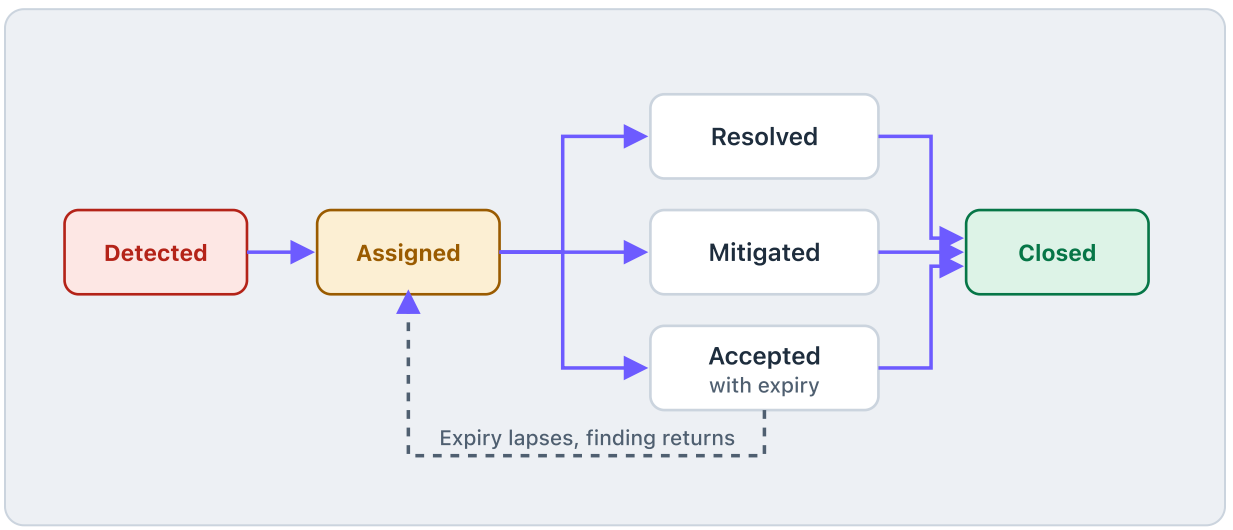


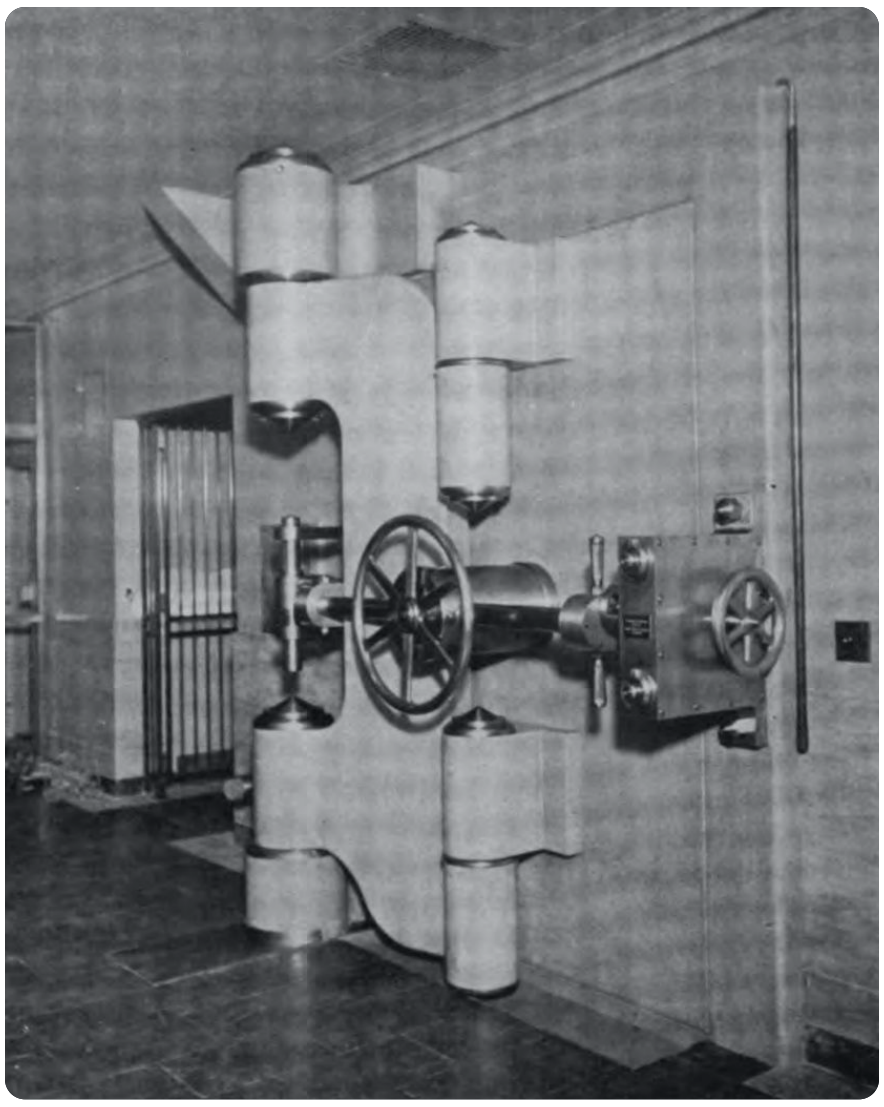
An illustrative rule. Your own rules and severities replace it.

VIOLATIONS TRACKER AND REMEDIATION WORKSPACE

Every finding has an owner and an end.

A violation has a life. It is detected, assigned, then resolved, mitigated or accepted as an exception. When an exception lapses, the finding returns to view on its own. The history stays with it for the auditor.



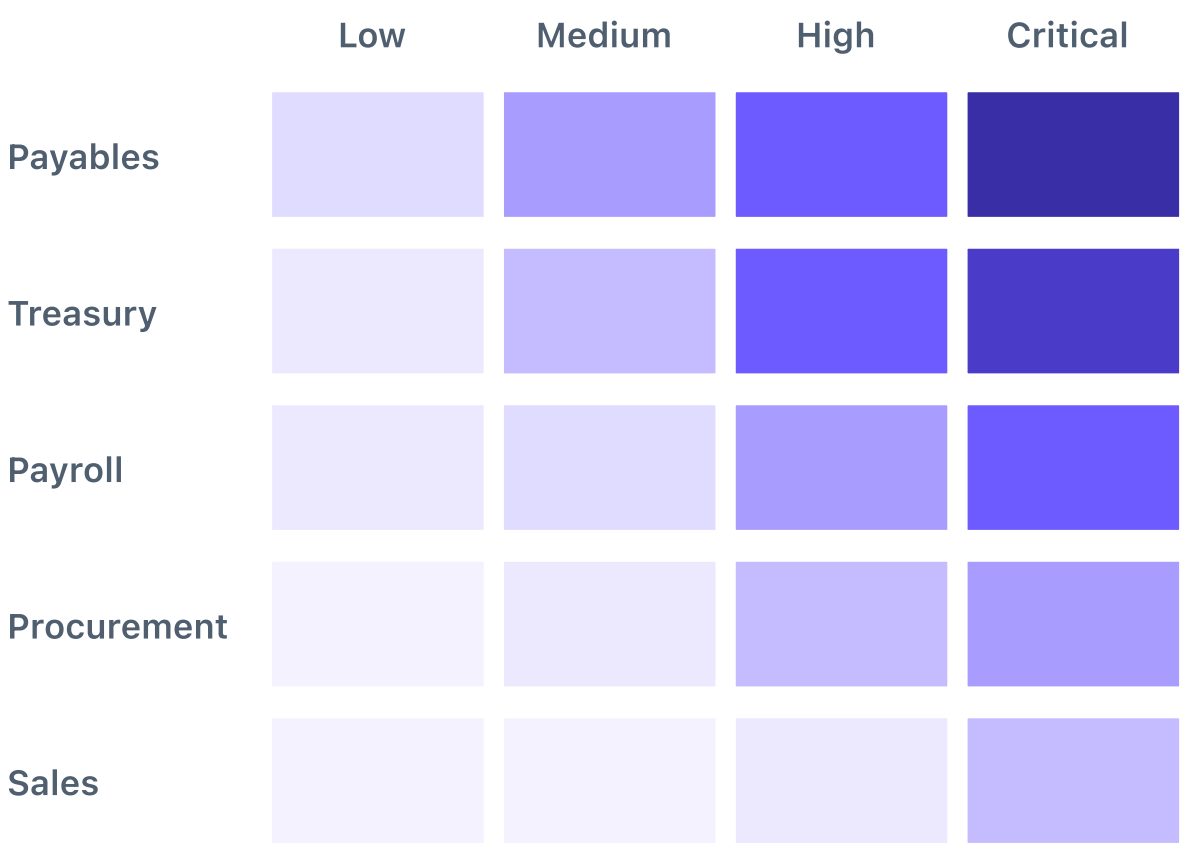


Two wheels, two hands. Dual control is an old idea. Bowery Savings Bank, 1923, public domain.

RISK HEATMAP

Weighted by the value that actually flows through the conflict.

A conflict on a duty that moves a little money is not the same as one that moves a lot. The heatmap puts the financial exposure on the map, so you fix the expensive problems first and can show why.



Illustrative layout with sample values. Your own processes and exposure fill the grid.

Fix in order of cost

Rank conflicts by exposure, not by how many there are.

Filter to the owner

Slice by company, role or person so each process owner sees their own part.

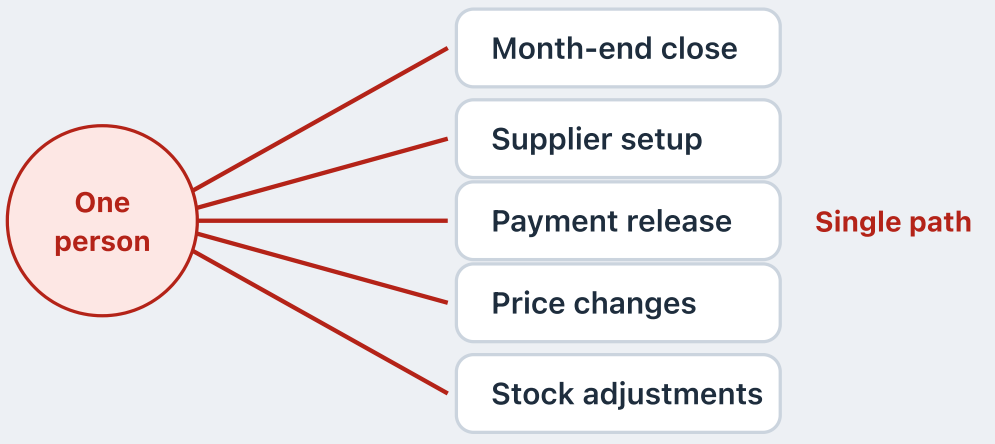


Every dial has someone who knows how to read it. NASA, public domain.

OPERATIONAL BOTTLENECK DASHBOARD

Find the places where one person is the only path.

Segregation of duties is not only about too much access. It is also about too little cover. SOFSEK shows where a single person holds a duty nobody else can perform, so a holiday or a resignation does not stop the business.

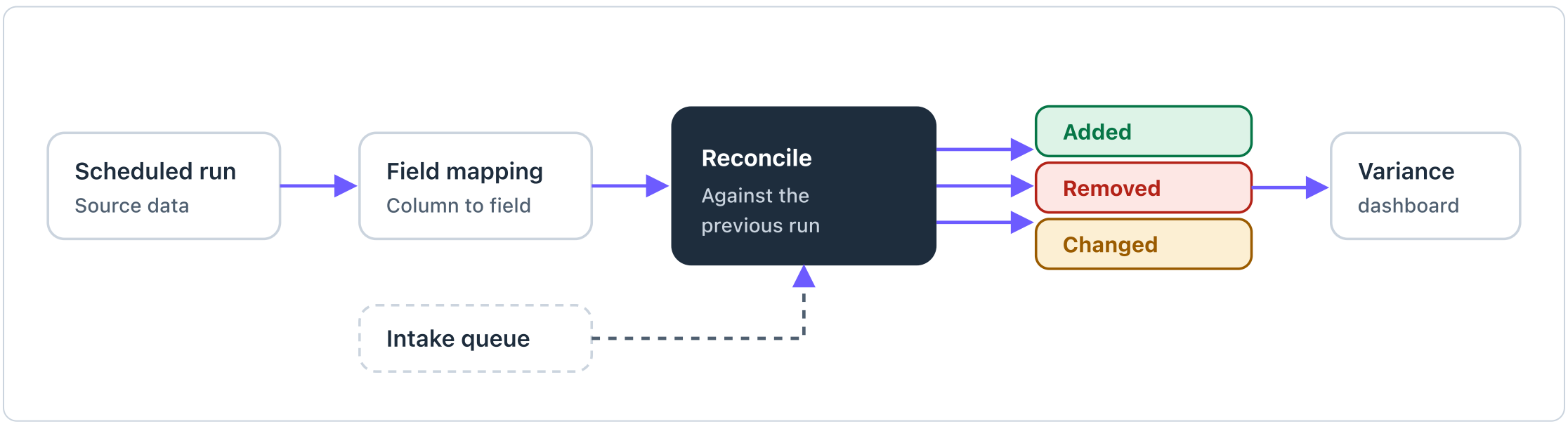


Illustrative. A finding can be excluded from the analysis when the dependency is intended.

SCHEDULED INGESTION AND RECONCILIATION

Fresh data, with the changes made visible.

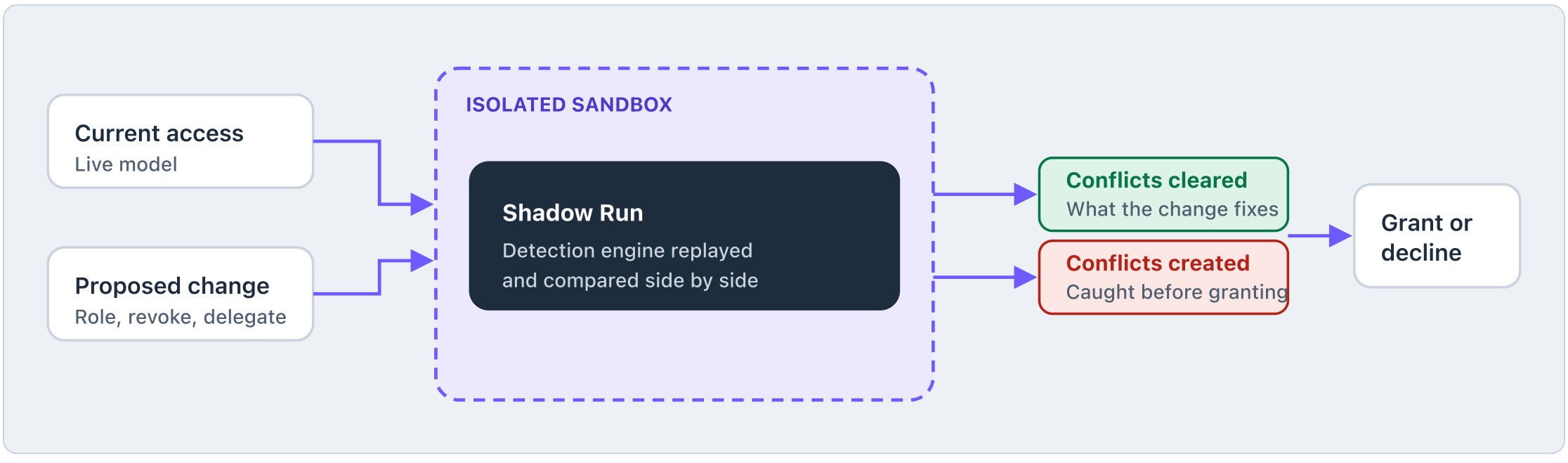
Load roles, users and assignments through a column-to-field mapping engine, with no scripting.
Each run is reconciled against the one before, so you see the variance and not just the snapshot.
Requests that arrive from people rather than systems wait in an intake queue.



SANDBOX SIMULATION

Model the change before you grant it.

Try a fix before you make it. Shadow Run replays the detection engine against a proposed change such as a new role, a revocation or a delegation. It shows what the change clears and what it creates. Nothing is written to your applications.



AUDIT AND PROVISIONING

Proof on demand, not in a scramble.

Audit packs

Generate a pack mapped to the regulatory frameworks you report against. Hand it to the auditor with the history already inside.

Provisioning gateway log

See each access request as it passed through the gateway, so approvals and grants can be traced from request to result.

Full lineage

For every effective duty, see which role or delegation brought it, and who granted it.

BUILT ON THE INTEGRYTAS PLATFORM

Security that is engineered in, not bolted on.

Multi-tenant

Each customer runs in its own tenant with its own data boundary and its own URL.

Signed licensing

Module entitlement is a cryptographically signed token verified at the gateway, not a flag in a database.

Row-level security

Every table is protected by policies enforced in the database itself.

Audit trail

Every change records who, what and when. An administrator acting on someone's behalf is recorded as themselves.

Role-based access

Fine-grained page and action permissions per role, with multi-level sign-off where a change needs a second pair of eyes.

Three locales

US English, UK English and Mexican Spanish out of the box, with a translation editor for more.

NEXT STEP

See your own conflicts, not a demo.

Start with a snapshot of your own access. SOFSEK takes it from there. Take the PDF overview to your risk, finance or audit lead and see what they make of it.

Photo credits. All photographs are in the public domain, sourced from Wikimedia Commons.

- Control room at the NACA Rocket Engine Test Facility, 1957. NASA Glenn Research Center, image GRC-1957-C-45021.
- Chain padlock and key, Jefferson County Jail and Sheriff's Office, Historic American Buildings Survey, 1971.
- Vault door, Bowery Savings Bank, from Architecture and Building, 1923.
- Operators in the Plum Brook Reactor Facility control room, 1970. NASA Glenn Research Center, image GRC-1970-C-00860.

Diagrams use illustrative data. SOFSEK is a product of IntegrytasGroup.