

Security & data handling summary

October 2026 · For IT and security reviewers · Contact: charley@buildflows.ai

How Build Flows builds and operates integrations, reporting, and AI agents: where your data lives, which identity has access, what an agent can change, and what you own at the end. The specific controls for your project are written into an access plan your IT team reviews before any build starts.

Where data lives and what crosses a boundary

A typical deployment runs in your tenant or cloud: secrets store, data platform, pipelines, agent gateway, and audit logs. Four boundaries are crossed:

- 1 Source systems → your environment.** Records the agreed workflow needs. Read-only connections wherever the use case allows, using an identity your admin approves. Credentials sit in your secret store.
- 2 Your environment → AI model provider.** Only for AI workflows: the prompt and the context retrieved for that task. You choose and approve the provider and region. No-training and zero or limited retention options are configured where the provider offers them.
- 3 Build Flows → your environment.** Named accounts you issue, scoped to the work, non-production first. Changes go through your version control. You revoke the access at handover.
- 4 Agent writes → source systems.** Off by default. Enabled one named workflow at a time with owner sign-off. Writes are previewed and approved by a person, deletes are blocked, and every call is logged.

How we build and operate

01 Deployment in your environment

- We build in your tenant or cloud wherever possible: your workspace, your subscription, your repositories, under your credentials.
- That way the systems run under your platform's own access controls, identity, and policies.
- If a component has to run somewhere else, the proposal names it before work starts.

02 Least-privilege, read-first access

- Source connections are read-only wherever the use case allows.
- Write scopes are added one named workflow at a time, with owner sign-off.
- We build against sandbox or non-production tenants when the platform provides them, and use offline exports (such as P6 XER files) to explore without touching live systems.
- We don't need passwords or admin access to scope the work. Access is requested later, through whoever approves it on your side, and only for what the agreed work needs.

03 Credentials

- Secrets live in your server-side secret store (Azure Key Vault for production builds), owned by your account, with your renewal process.
- Never in website forms, shared screenshots, spreadsheets, desktop configs, or model prompts.

- Agents reach systems through a scoped gateway key, so they never hold upstream credentials. Each key carries its own tools, upstream auth, and rate limits.
- Production and development keys are separate. Exchanged user tokens are short-lived and scoped per operation.

04 Agent permissions, approvals, and audit

- The tool list is the permission boundary. An agent can't be talked into a tool it doesn't have, so we route large APIs down to the tools a task needs.
- Writes are proposals: previewed, then approved by a person. Deletes are blocked and bulk actions are capped.
- Agents act as a scoped service identity or as the signed-in user via on-behalf-of exchange, so the source system's audit trail shows the real person.
- Every tool call is logged with who, which key and server, which tool, outcome, and latency. Secrets are redacted when the log is written.
- Changes to an agent itself (prompts, tools, models) go through review, and are tested against agreed examples before a pilot widens.

05 AI data handling

- Reporting and integration work that doesn't use an agent sends no data to an AI model provider.
- For AI workflows, you choose and approve the model provider and region. Its data terms apply to what it receives.
- We configure providers with no-training and zero or limited retention options where the provider offers them.
- Only the prompt and the context retrieved for the task are sent. Retrieved content is treated as data, never as instructions.

06 Code and IP ownership

- Code is delivered into your repositories under version control: readable, runnable, and extendable by your team. No black boxes.
- Ownership, intellectual property, licenses, and confidentiality are set out in the written agreement and documented again at handover.
- Assessment documents are yours whether or not you choose us for the build. Public open-source components keep their own licenses.

07 Environments, testing, and rollback

- Pipelines, models, reports, quality rules, and infrastructure deploy from version control by script: reviewable, repeatable, reversible.
- Outputs are reconciled against your own figures within written tolerances before acceptance.
- Nothing reaches production without a named owner, a runbook, and a tested rollback. For write workflows: authorization, duplicate prevention, and a tested recovery plan.

08 Monitoring

- Health checks, refresh status, and data age are shown where the scope requires them.
- A failed refresh stops publishing and keeps the last good data, rather than serving wrong answers.
- Missing, duplicate, rejected, and unmapped records are visible. A gap never passes for a good result.

09 Handover and offboarding

- Your named operating owner receives runbooks, training, and documented dependencies, licenses, asset ownership, and support responsibilities.

- You decide who holds credentials and licenses, and you remove our access when the work ends.
- Ongoing care is optional and defined in a separate written scope: named assets, support hours, severity definitions, and exclusions.

Third parties

The services in your own environment, the source system vendors you already use, and the model provider you approve for AI work. The proposal lists components, licenses, and hosting so you can review each one.

Certifications and agreements

We don't claim certifications or independent audits. Formal certification claims are made only where separately verified. Any engagement is governed by a separate written agreement covering scope, fees, acceptance, intellectual property, confidentiality, and support. Items this summary doesn't cover, such as insurance, incident notification, retention periods for engagement data, and project-specific subprocessors, are confirmed during scoping.

Information sent through our website

Inquiries are handled under the privacy policy at buildflows.ai/privacy. Website service providers: Vercel (hosting), Resend (email delivery), Google Analytics (usage analytics), Cloudflare Turnstile (spam protection, when enabled), and Calendly (call booking). Please don't send credentials or client documents through the website form.

Build Flows (Buildflows.ai), New York · charley@buildflows.ai · Book a call: calendly.com/buildflows/30min · Full page: www.buildflows.ai/security