FormSavvy

Architecture and data flow

This page describes how FormSavvy for Salesforce works: its components, where each runs, how data moves between them, and the control at every trust boundary. It is written for security reviewers and for customers' security and IT teams.

Managed package namespace `formsavvy` · Package version 1.0 · November 2026 · Online at www.getformsavvy.com/docs/architecture

Summary

FormSavvy is a form builder that is native to Salesforce. It is a managed package of Apex, Lightning Web Components, Visualforce pages, and static resources, installed into the customer's own org. Admins build forms in a Lightning app; anonymous visitors fill them in on the customer's Salesforce Site or Experience Cloud site; each submission becomes a record on the customer's own object.

No FormSavvy servers hold customer data

Form design, submissions, files, analytics, and logs are all stored in the customer's Salesforce org.

Three optional outside services

FormSavvy uses Cloudflare Turnstile for CAPTCHA, Anthropic (directly or through the FormSavvy AI gateway) for admin AI features, and the documentation site.

One server-side pipeline

Both public renderers submit through the same Apex validation and write path. Nothing the browser sends is trusted on its own.

Least privilege for the guest

The guest user gets Apex and page access only, plus the object and field access its forms need (published forms, and drafts, which an admin previews as the guest), granted by the admin.

Contents

1. How FormSavvy works
2. System architecture
3. Public form submission pipeline
4. File uploads
5. AI features: data flow
6. Licensing and AI activation
7. Data storage information
8. Authentication at each boundary
9. External endpoints
10. Security controls in brief

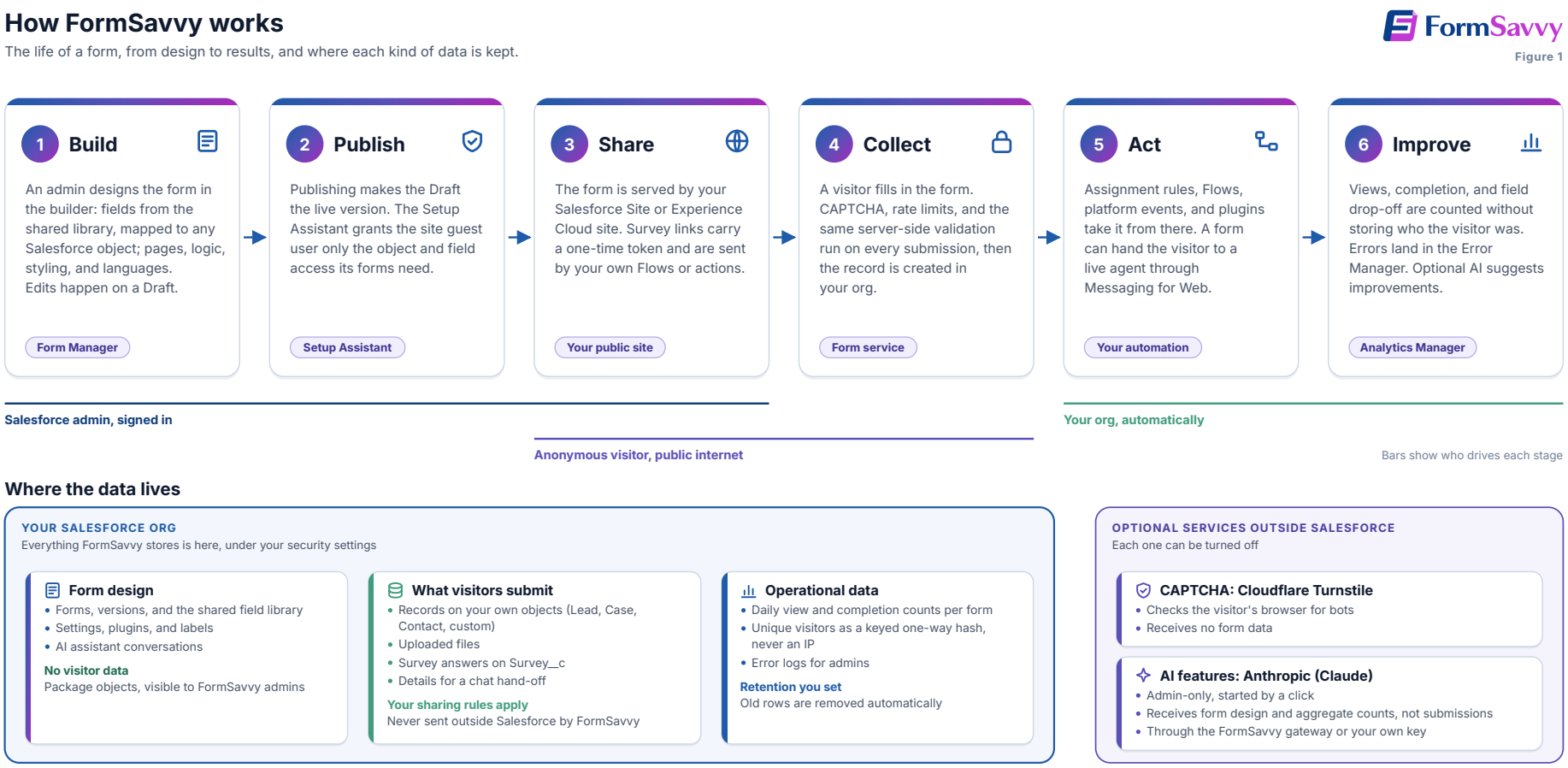


Figure 1. This is the life of a form, showing where each kind of data is kept.

- Admins build and publish; anonymous visitors only ever see a published form; the customer's own automation acts on the result.
- Edits happen on a Draft version. The live version changes only when an admin publishes.

System architecture

Every component, where it runs, and each connection that crosses a trust boundary. Numbers match the authentication table.

FormSavvy

Figure 2

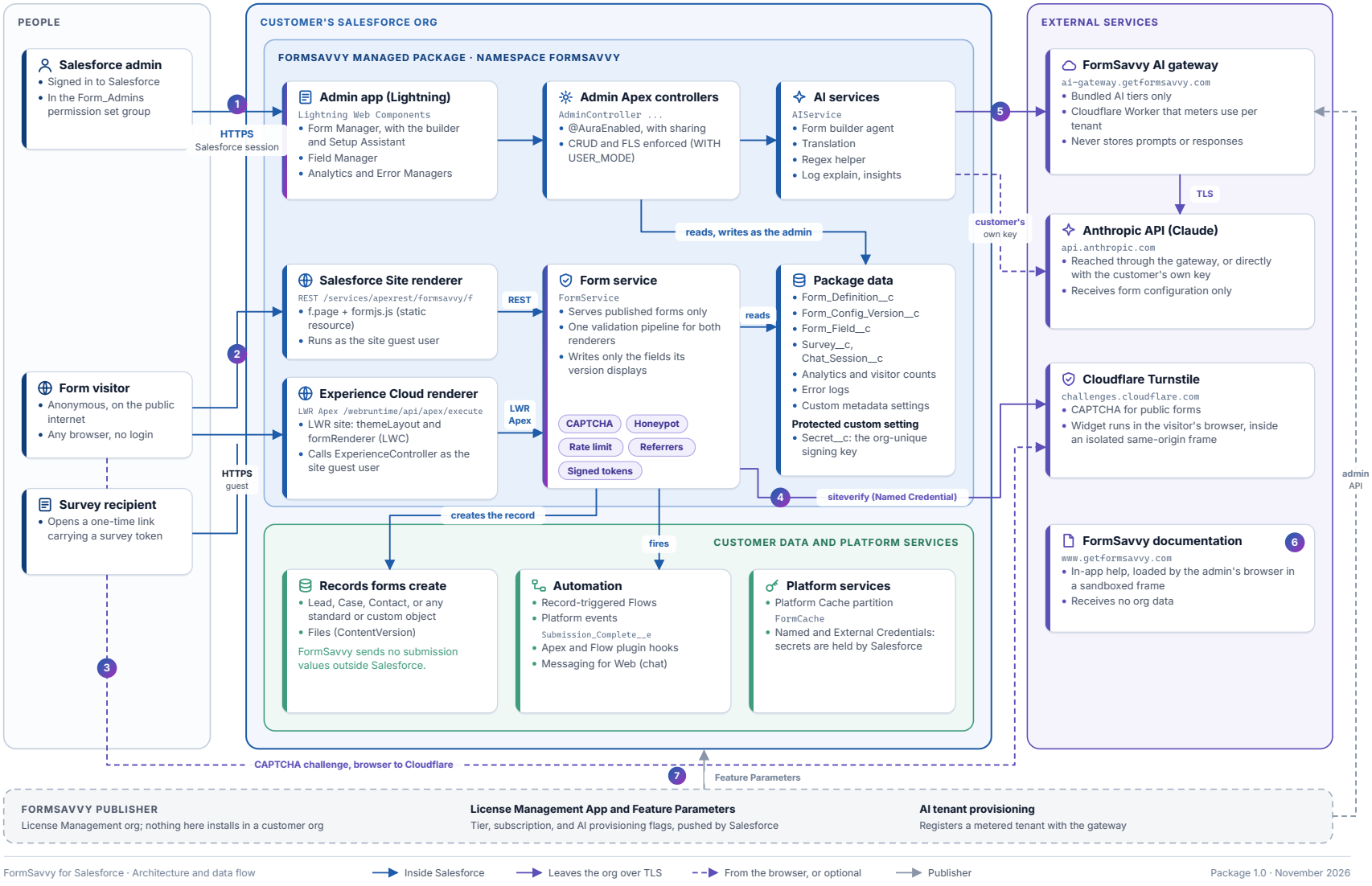


Figure 2. These are the components and their trust boundaries. Numbered connections are described in Authentication at each boundary.

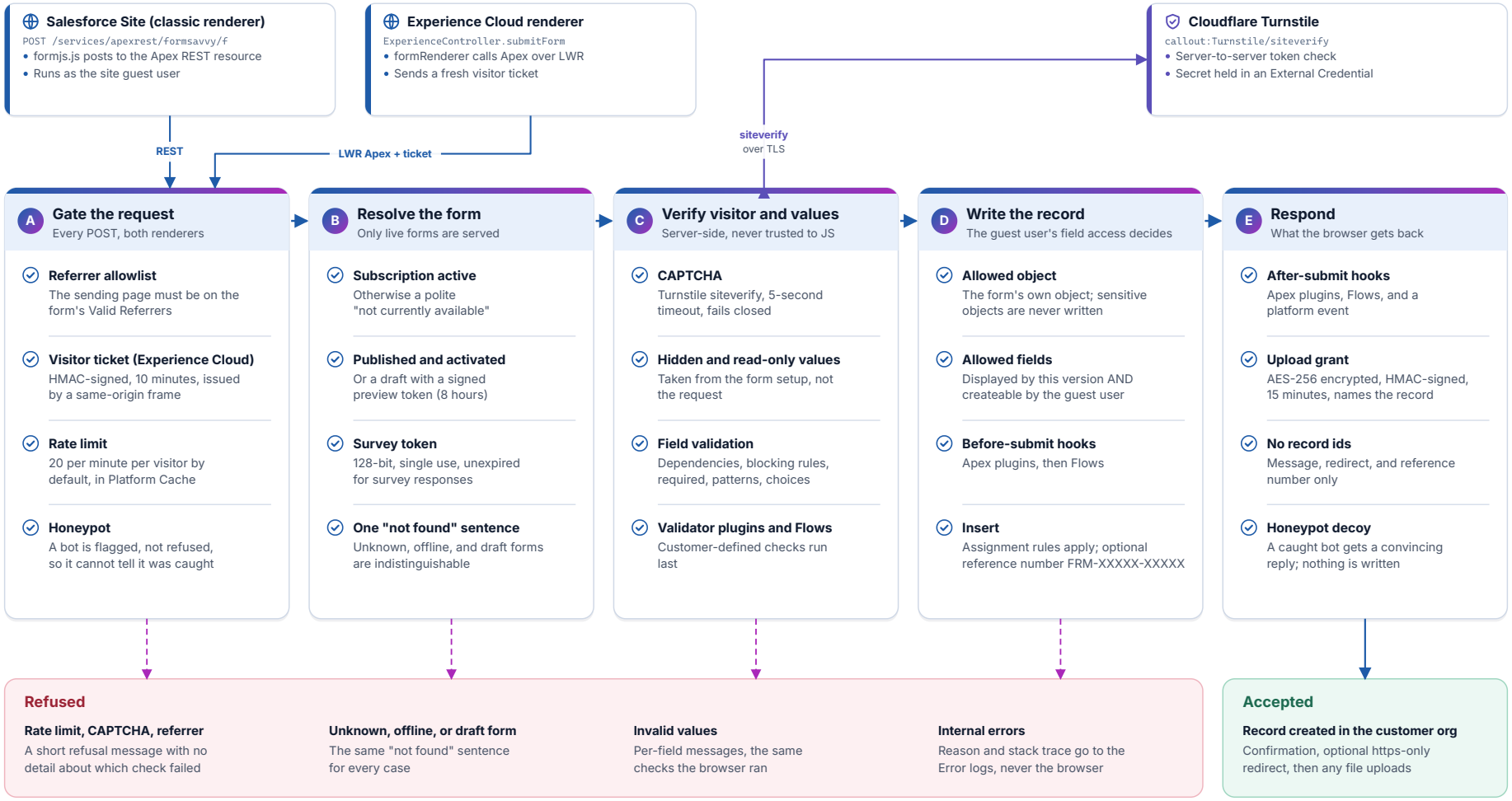
- The admin plane (top) runs as the signed-in admin, with sharing and with CRUD and FLS enforced.
- The public plane (middle) runs as the site guest user. `FormService` is the one class both renderers reach.
- Only connections 3 to 6 leave Salesforce, and each is optional or limited to admins.

Public form submission: the server-side pipeline

Both renderers reach one Apex pipeline. Each stage can stop the request; nothing the browser sends is trusted on its own.

FormSavvy

Figure 3



Loading a form (GET) applies the same rules as B: only a published, activated form, a valid preview token, or a valid survey token is served.

Figure 3. Checks run in this order for a submission from either renderer.

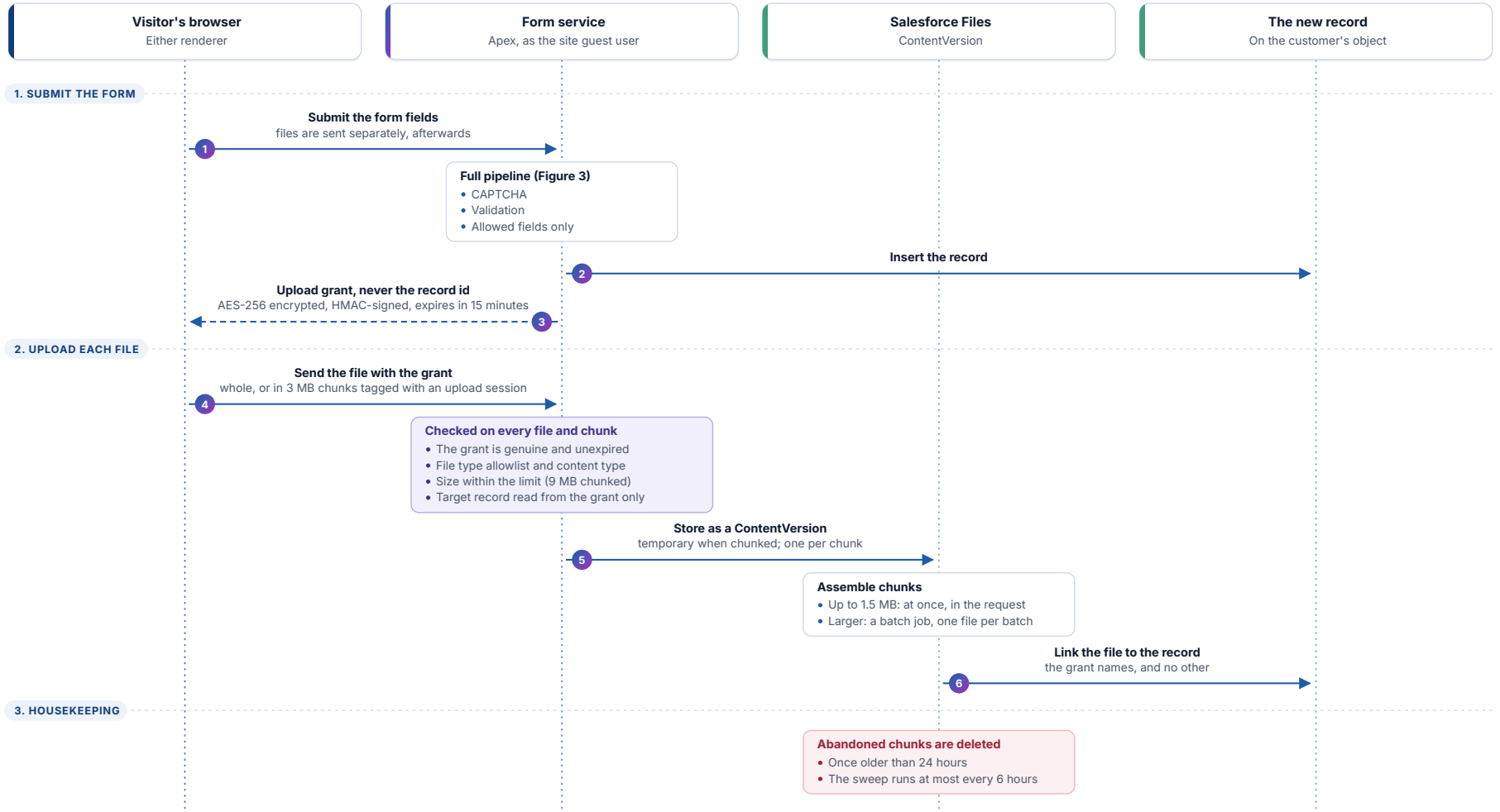
- Every refusal tells the visitor as little as possible. Unknown, offline, and draft forms all get the same “not found” sentence; internal reasons go to the Error logs only.
- A record is written only for fields the submitted version displays (the published one, or a draft opened with a preview token) AND the guest user can create, so the admin’s guest permissions always decide.
- The browser never receives a record id.

File uploads

A file reaches the right record through a short-lived encrypted grant. The browser never learns the record id.

FormSavvy

Figure 4



Experience Cloud can also use the standard Salesforce file upload component for large files (up to 2 GB);
FormSavvy links those by ContentVersion id only after checking the upload session tag, and never trusts a record id from the browser.

Figure 4. Files are tied to the new record by a short-lived encrypted grant, never by an id the browser supplies.

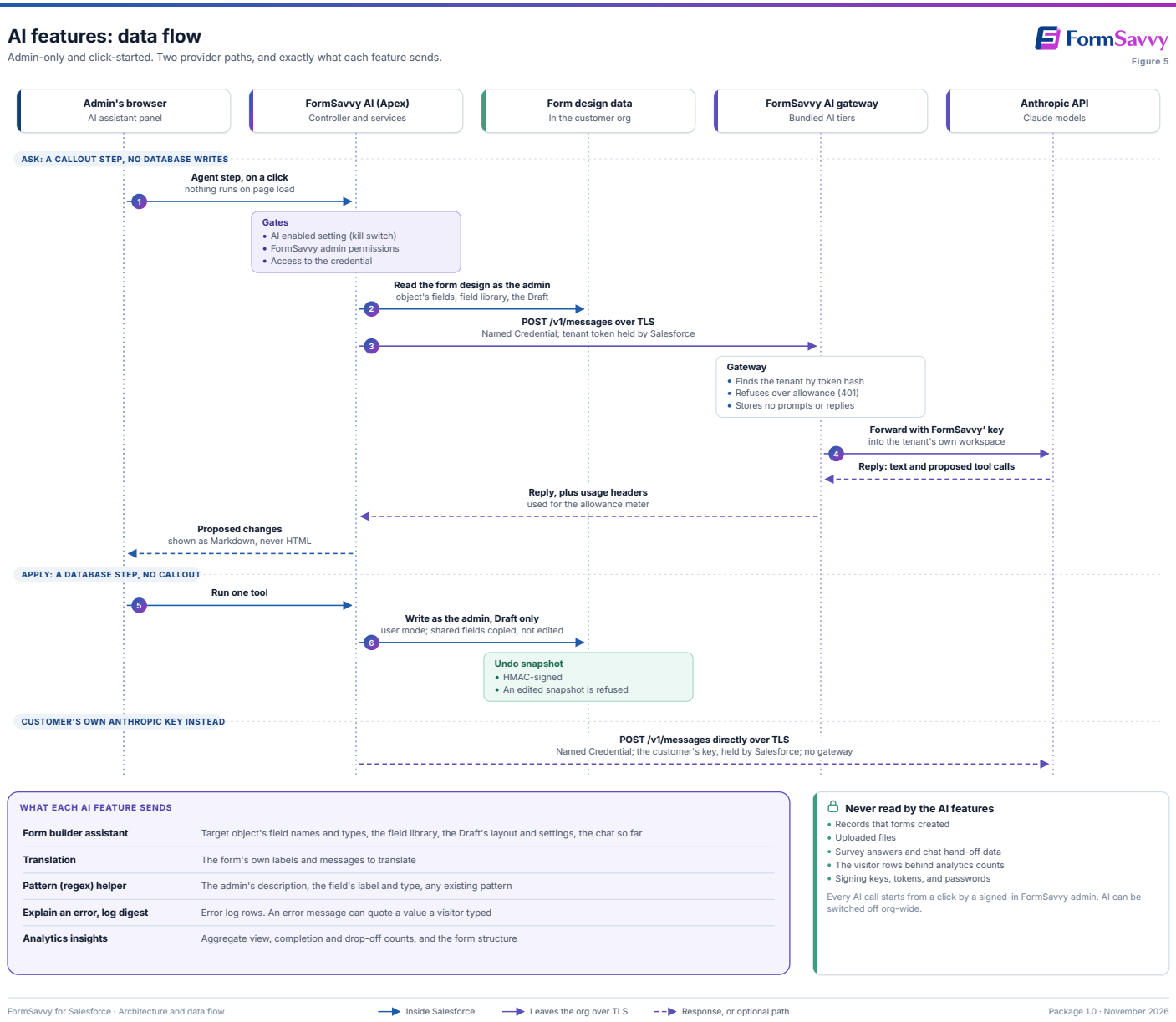


Figure 5. AI is admin-only, started by a click, and can be switched off for the whole org.

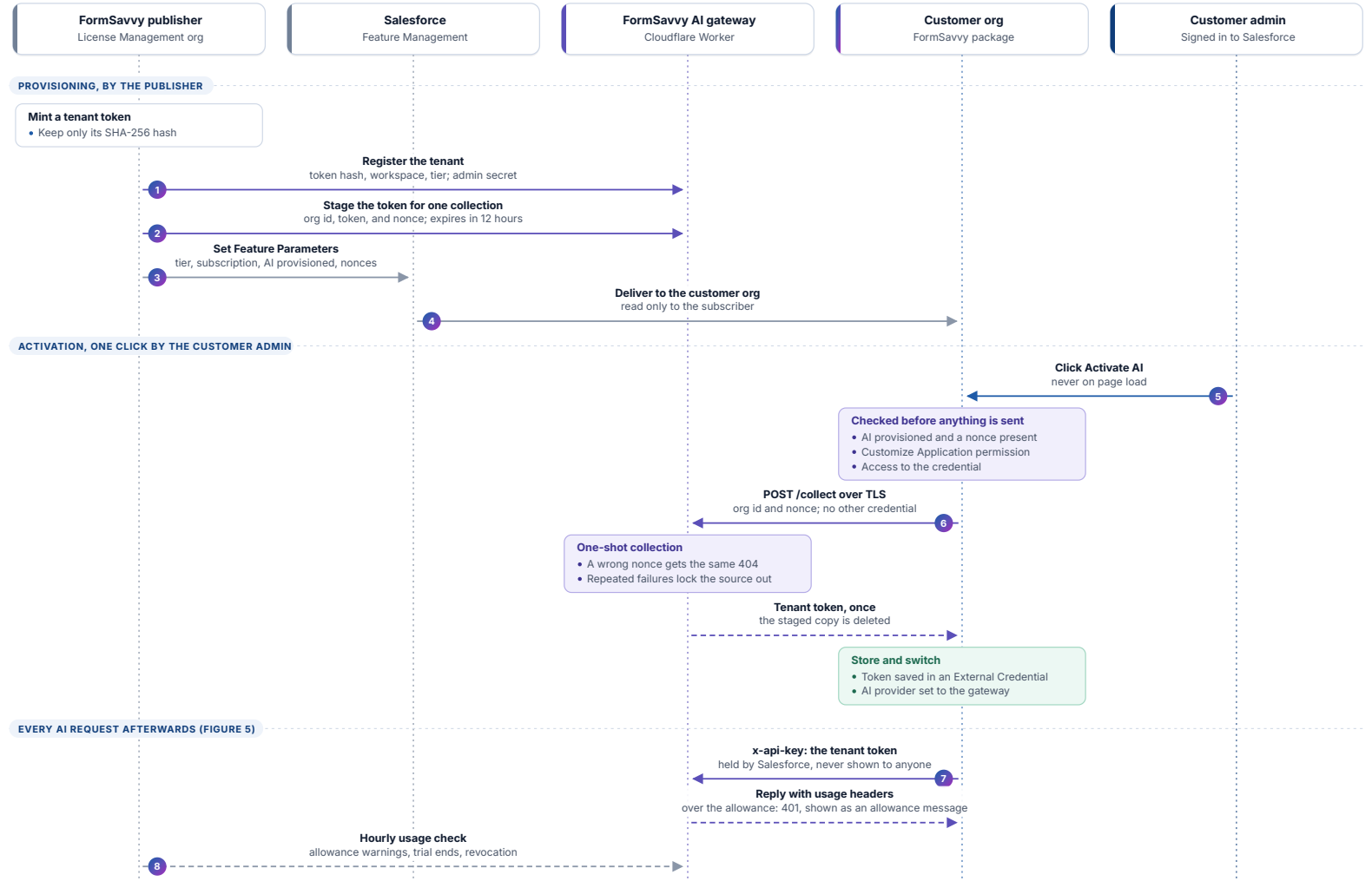
- The agent alternates a callout step with no database writes and a database step with no callout, so a model reply is never applied in the same transaction that fetched it.
- Changes are written as the admin (user mode), to the Draft only. A shared field is copied rather than edited, and Undo uses an HMAC-signed snapshot.
- The AI features do not read submitted records. An error log row can quote a value a visitor typed, which is why log explanation is listed separately.

Licensing and AI activation

How a customer org receives its license flags and, for bundled AI, a gateway token it collects exactly once.

FormSavvy

Figure 6



The publisher org and the gateway are operated by FormRuntime; neither installs anything in the customer org. Revoking a tenant at the gateway stops bundled AI at once; the rest of FormSavvy is unaffected. A customer using their own Anthropic key never contacts the gateway.

Figure 6. License flags arrive as Feature Parameters; the bundled AI token is collected once, on an admin’s click.

Data storage information

Every object FormSavvy writes, grouped by the kind of data it holds. All of it is in the customer org.

FormSavvy

Figure 7

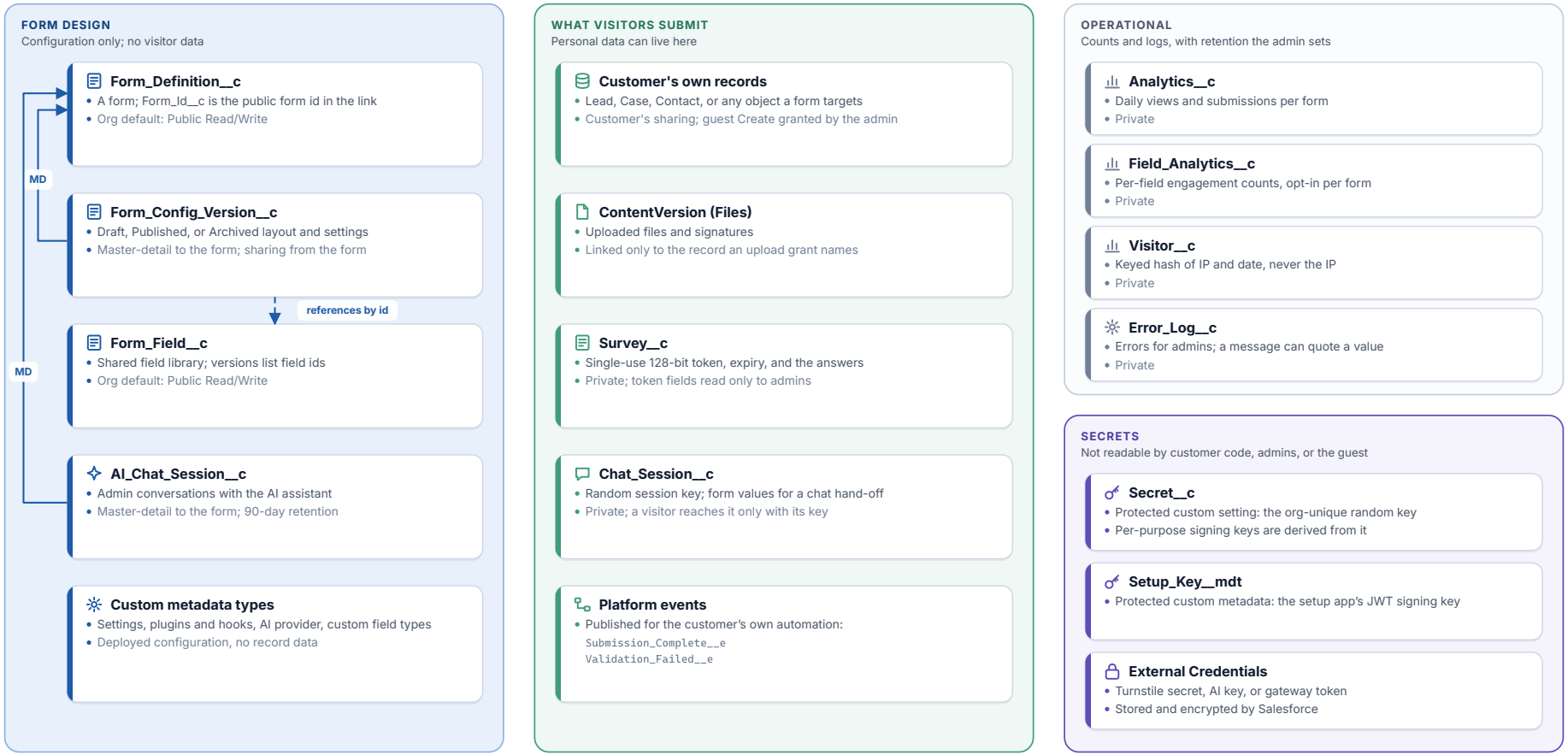


Figure 7. Every object FormSavvy writes is grouped by the kind of data it holds.

Authentication at each boundary

Numbers match the connections in Figure 2. Rows A and B are calls the org makes to its own Salesforce API.

#	Connection	Authentication and authorization	Further controls
1	Admin to Salesforce	Salesforce login, under the customer's own SSO and MFA policy. Access comes from the `Form_Admins` permission set group.	Admin Apex controllers run with sharing and enforce CRUD and FLS (`WITH USER_MODE`). Admin-authored HTML is sanitized with DOMPurify before it is rendered.
2	Visitor to the public site	Anonymous site guest user. `Form_Guest` grants Apex classes and pages only, with no object permissions. Object and field access comes from a separate permission set the admin syncs from published and draft forms, never including sensitive objects.	Valid Referrers allowlist, per-visitor rate limit, CAPTCHA, honeypot. Experience Cloud requests carry an HMAC-signed visitor ticket (10 minutes). Preview tokens are HMAC-signed (8 hours); survey tokens are 128-bit and single-use.
3	Visitor's browser to Cloudflare Turnstile	None needed: the public Turnstile site key identifies the widget.	The widget loads only inside the isolated same-origin `TurnstileFrame` frame, never in the form page. The frame talks to the form by `postMessage` restricted to the page's own origin.
4	Org to Turnstile siteverify	Named Credential `Turnstile` ; the secret key is in an External Credential.	TLS; 5-second timeout; a failed or unreachable check refuses the submission.
5	Org to the AI provider	Named Credential `AI_Gateway` with a per-tenant token, or `Anthropic` with the customer's own key. Both secrets are in External Credentials, held by Salesforce. The one-time token collection uses `AI_Gateway_Collect` with a single-use nonce.	Admin-only and click-started; org-wide kill switch. The gateway hashes the token to find the tenant, enforces the allowance, rebuilds headers, and stores no prompts or replies.
6	Admin's browser to the documentation site	None: public pages.	Loaded in a sandboxed iframe that cannot navigate or reach the Salesforce page. No org data is sent.
7	Publisher to the customer org	Salesforce Feature Management: Feature Parameters set by the License Management org and delivered by Salesforce.	Read only in the customer org. Nothing from the publisher org is installed in the customer org.

#	Connection	Authentication and authorization	Further controls
A	Setup automation to the org's own API	The packaged `FormSavvy Setup` External Client App: a JWT bearer exchange at the org's own token endpoint, as the signed-in admin, with OAuth scopes `Api` and `RefreshToken`. The signing key is protected custom metadata. The access token is never stored.	The admin must pre-authorize the app and hold Customize Application. Every action runs only after a confirmation. Changes to org settings FormSavvy does not own remain guided steps the admin performs.
B	Setup Assistant checks	Metadata API reads with the admin's session, a supported use under Salesforce's guidance on session IDs.	Read operations only (`listMetadata` , `readMetadata` , `checkDeployStatus`) to the org's own domain. The session id is never logged, stored, returned to a browser, or sent off the org.

External endpoints

This table lists every host outside the customer's Salesforce org that FormSavvy code or pages contact.

Host	Called from	Purpose	Data sent
`challenges.cloudflare.com`	Visitor's browser (CSP Trusted Site) and Apex (Named Credential)	Turnstile CAPTCHA widget and token verification	The browser challenge; the Turnstile token and secret. No form data.
`ai-gateway.getformsavvy.com`	Apex (Named Credentials)	Bundled AI tiers: metered proxy to Anthropic; one-time token collection	The AI request content listed in Figure 5; for collection, the org id and nonce only.
`api.anthropic.com`	Apex (Named Credential), only with the customer's own key	Claude models	The AI request content listed in Figure 5.
`www.getformsavvy.com`	Admin's browser (CSP Trusted Site), sandboxed iframe	Product documentation	No org data. The documentation pages count views with Cloudflare Web Analytics, which sets no cookies.

Chat hand-off loads Salesforce's own Messaging for Web script from the customer's Salesforce site host; FormSavvy allows only Salesforce site hosts or the page's own origin. Plugins the customer writes run the customer's own Apex, under the customer's Remote Site and Named Credential settings.

Security controls in brief

Signed and encrypted tokens

Preview (8 h), visitor ticket (10 min), upload grant (AES-256 and HMAC, 15 min), and AI undo snapshots all use keys derived per purpose from an org-unique random key in a protected custom setting.

Server-owned values

Hidden and read-only field values come from the form setup. A link can set one only when the admin marked that field as link-settable.

No ids, no internals

No record id reaches the browser or a JavaScript plugin. Error detail goes to the Error logs, never to a visitor.

Sanitized HTML

All admin-authored HTML passes through DOMPurify, one configuration shared by both renderers and the builder.

Injection-safe queries

Queries are static, with bind variables; dynamic object names come from a describe-based allowlist, never from request input.

https-only redirects

The post-submit redirect is checked where it is served and again in both renderers.

Bounded telemetry

Analytics beacons and browser error logs have their own per-visitor limit and never count against the submission limit.

Retention

Analytics, logs, and AI conversations are removed automatically after the period the admin sets.