

ContextWorks Security Statement

Akoonu, Inc. · Updated September 2026

Curated, permission-correct Salesforce context for AI tools and agents.

Overview

This document describes the security architecture, access control model, AI handling, data handling, write controls, configuration governance and compliance posture of Akoonu ContextWorks. It is written for security, IT and data-governance reviewers evaluating whether to connect AI tools and agents to Salesforce data through ContextWorks.

ContextWorks is a native Salesforce application that curates what an AI tool or agent receives from CRM records: which fields, what they mean, and how to use them. Its governing principle is that Salesforce decides what a caller may see. ContextWorks does not implement a second permission model, does not copy, sync or index your data, and makes no calls to AI models of its own.

Akoonu has been a Salesforce ISV Partner since 2017, has passed Salesforce's mandatory AppExchange security review, maintains a 5.0 AppExchange rating, and tests against every Salesforce release.

Architecture

ContextWorks is built inside Salesforce, not connected to it. It installs from the Salesforce AppExchange as a managed package. All application code runs on the Salesforce platform (Apex, Lightning Web Components, SOQL). Configuration is stored as custom objects in your org.

Every payload an agent receives is assembled at call time from live records in your org, as the calling user. There is no ContextWorks-side database, cache of record data, external index or vector store. Nothing is copied out to build it.

ContextWorks does not call any AI model. It returns context (as Markdown or JSON) to the tool, agent or prompt that requested it. Which AI model reads that context, and where inference runs, is decided by the surface you connect and by your agreements with those vendors. This is described per surface under AI Handling.

Access Control

ContextWorks inherits your Salesforce org's access control model completely. Every record query runs in user mode, so Salesforce enforces access on the query itself.

Salesforce control	Effect on an agent's payload
--------------------	------------------------------

Object permissions	No read access to the object, no rows
Field-level security	A field the caller cannot see is never retrieved and never reaches the payload
Record sharing (rules, role hierarchy, territories)	Records the caller cannot see are absent, including inside related lists and query results

ContextWorks never fetches a value and then hides it. A field the caller cannot see is absent because it was never retrieved. Two users calling the same tool on the same record receive different payloads, correctly, with no per-user configuration.

Curation and security are separate. Removing a field from a ContextWorks entity is a relevance decision; it does not protect the field, which remains reachable through reports, the API, or another entity for a user with access. Field-level security is the access control; ContextWorks curation decides what is relevant.

Permission sets

Access to ContextWorks is granted through three standard Salesforce permission sets. None of them grants access to record data; an administrator still sees only the records their own permissions allow, in preview and everywhere else.

Permission set	Grants
ContextWorks Admin	Write access to the configuration objects
ContextWorks Domain Manager	Configuration rights over the entities homed in specific domains, when delegated administration is on
ContextWorks User	Access to the runtime actions and read on the domain object, so ContextWorks can resolve sharing. What an Agentforce or MCP caller needs, and nothing else

Domains scope discovery, not data access

A domain is a named set of entities. A user discovers only the domains shared to them, using standard Salesforce sharing on the domain object. Holding a domain never widens data access: two people holding the same domain still receive different records, each under their own permissions. ContextWorks reports whether domain scoping is in force and names any entity no domain carries.

Enrichment sources: the one deliberate exception

Enrichment sources are customer-authored code (Apex, an invocable action or a flow) that add computed values to a payload. For record results, ContextWorks keeps only the record IDs a source returns, discards every field value, and re-queries those records as the calling user, so a record or field the caller cannot see never appears regardless of what the source could see. Scalar values a source returns are trusted as returned; whoever writes a source owns what its scalars expose, and the documentation directs them to query as the calling user.

AI Handling

ContextWorks makes no AI model calls. It has no model credentials, no model configuration, and no inference of its own. Whether data leaves Salesforce infrastructure depends on the surface you connect:

Surface	Where the context goes
Agentforce	Stays within Salesforce. The agent's model calls pass through the Einstein Trust Layer under your Salesforce agreement
MCP (Claude, ChatGPT, other MCP clients)	Goes to the MCP client and model you chose, under your agreement with that vendor. Each user authenticates as themselves through an External Client App; no shared service account
Prompt Builder and Flow	Stays within Salesforce, governed by your Salesforce agreement and the model you configured
Apex and custom tools	Goes wherever your own code sends it

Because no data is used by ContextWorks for any model call, there is no ContextWorks training question. Training and retention terms are those of the model vendor behind the surface you connect, under your agreement with that vendor.

Data Handling

What ContextWorks stores in your Salesforce org

Configuration: entities, elements (metrics, formulas, filters, enrichments), briefs, skills, domains and their versions. Operational records: the write log, usage counts, quality and

issue records, and time-boxed payload logs when an administrator turns debug logging on. All of it lives in custom objects in your org, under your Salesforce security and backup posture.

What ContextWorks does not store

Your Salesforce record data. Opportunities, accounts, contacts, cases and every other standard or custom object remain where they are. ContextWorks reads them live at call time and stores no copy.

Reads are counted, not recorded

ContextWorks counts usage (which entity, which surface, which domain, how often) and does not record the content of reads. Payload content is stored only when an administrator turns on debug logging, which is time-boxed: it stops on its own at the end of a 15- or 60-minute window or at 500 stored rows, whichever comes first, and holds real record data shaped by the permissions of the caller who received it. A captured payload opens only for the person who ran the load, a user with View All Data, or someone who can read the record it captured. Debug logging is a troubleshooting tool, not a permanent record.

Configuration leaves as data

The full configuration, or a single domain, exports as ContextWorks JSON (complete and importable) or as Apache Ossia YAML (an export for tools that read that standard). Export is an administrator action. No record data is included.

Clean uninstall

Removing ContextWorks removes its custom objects and configuration. Your Salesforce data is untouched.

Write Controls

Agents cannot change records unless writes are deliberately enabled, and a write happens only if every level allows it:

- **Org:** an administrator-held switch that is off by default
- **Domain:** writes allowed for calls through that domain
- **Entity:** write access declared on the entity
- **Field:** each writable field declared individually; nothing is writable unless declared

Then Salesforce's own field-level security, sharing and validation rules apply. Every write runs as the calling user, so an agent cannot change anything that user could not change by

hand. Creating records is governed by its own separate chain.

By default a write returns a proposal and nothing commits until the caller confirms it. ContextWorks enforces the confirmation: the preview issues a change digest, and a commit that does not echo it, or whose record changed in the meantime, is refused. Unattended writes (committing on the first call) must be enabled per domain by that domain's manager or an administrator.

Every write attempt is logged durably: committed, refused or failed validation; confirmed or unattended; entity, domain and version; the surface that made the call; the calling user; and per-field from and to values as of commit time. Write configuration is captured in each entity version, and the activation screen lists it, because a rollback can change what an agent may write.

Configuration Governance

Every entity, brief and skill exists as a draft and an active version. Agents receive only the active version; nothing an author saves reaches an agent until it is activated, and activation requires a comment. Versions are numbered and never reused. Any prior version can be reactivated in one action, and the history records who activated what, when, and why.

Administration can be delegated by domain. A domain manager configures the entities homed in their domain and nothing else, without org-wide administrator rights. An access overview lists every administrator, every domain manager and their domains, and any grant that confers nothing yet.

Before an agent is connected, an administrator can preview exactly what a given user would receive for a given record, at each level of detail, in Markdown or JSON.

Operational Security

Because ContextWorks runs natively inside Salesforce, there is no integration to monitor, no sync to maintain, no webhook to configure and no ContextWorks-side credential to rotate. Connecting a surface is standard Salesforce configuration: an External Client App and an MCP server for MCP clients, agent actions for Agentforce, both documented step by step.

Observability is built in. Administrators can see usage by entity, surface and domain; quality measures including the share of designed fields a caller could actually see; issues such as fields stripped by field-level security or a failed enrichment source; and the write log. If an enrichment source fails, the engine drops that one element, delivers the rest, and never exposes the exception text to the agent.

Compliance

Because ContextWorks runs entirely within your Salesforce org and stores no data externally, your Salesforce org's compliance posture is your ContextWorks compliance posture. Salesforce maintains SOC 2 Type II, ISO 27001, GDPR, HIPAA and FedRAMP certifications; data read through ContextWorks is protected by the same platform controls.

At the application level, ContextWorks has passed the Salesforce AppExchange security review, which evaluates CRUD/FLS compliance, vulnerability testing and code quality. Because ContextWorks introduces no external data storage, it creates no new data residency requirement and no new geographic processing location. The model vendor behind any surface you connect is governed by your agreement with that vendor.

You retain full control through Salesforce's native security tools: profiles, permission sets, sharing rules, field-level security, setup audit trail and session management. ContextWorks inherits every control you have configured.

Salesforce ISV Partnership

Akoonu has been a Salesforce ISV Partner since 2017. This includes passing Salesforce's mandatory security review, maintaining compliance with partner program requirements, and testing compatibility in advance of each of Salesforce's three major releases per year. Akoonu was selected as an ISV partner showcase for Headless 360 at Dreamforce 2026 and is featured on Salesforce's AgentExchange ecosystem page.

Frequently Asked Questions

Where does my data go?

Nowhere outside your Salesforce org on ContextWorks' account. Payloads are assembled live in your org and returned to the surface that asked. Where an MCP client or agent sends it next is governed by your agreement with that vendor.

What exactly can a connected tool see?

The fields an administrator curated into the entity, minus anything the calling user's field-level security and sharing do not allow. The exposure surface is a list you can read, preview per user, and export.

Does ContextWorks respect our sharing rules?

Yes, on the query itself. Object permissions, field-level security and record sharing are applied in user mode. There is no code path where a value is fetched and then hidden.

Can an agent change our data?

Not unless writes are enabled at the org, domain, entity and field level, and then only what the calling user could change by hand, with a confirmation step enforced by default and every attempt logged durably.

Is there an audit trail of what an agent saw?

Writes are logged durably. Reads are counted, not recorded. Configuration is versioned, so you can see what definition was active at any time. Reproducing the exact payload a caller received on a past date is not a feature; time-boxed debug logging exists for troubleshooting a current problem.

Does ContextWorks send data to an LLM?

No. ContextWorks makes no model calls. The Agentforce path stays within Salesforce and the Einstein Trust Layer; an MCP call goes to the client you connected, under your agreement with that vendor.

Do we need a shared service account for MCP?

No, and you should not use one. Each user authenticates as themselves through an External Client App, so each receives only what their own permissions allow.

What if we want to remove ContextWorks?

Uninstall removes ContextWorks' custom objects and configuration. Your Salesforce data is untouched. Configuration can be exported first.

Do we need a separate security review?

Most reviews are shorter because there is no external infrastructure: platform questions resolve to Salesforce's certifications, and ContextWorks itself has passed the AppExchange security review. The questions specific to ContextWorks are the ones this document answers.

Contact

For security questions, vendor assessment support, or to schedule a call to discuss your organization's requirements:

Email: security@akoonu.com

Web: www.akoonu.com/security