



FormulaShare Trust & Security Whitepaper

First Published: 14 July 2025

Last Updated: 14 July 2025

Contact: formulashare@cloudsundial.com

1. Introduction

FormulaShare is a powerful rule-based sharing solution for Salesforce, designed to extend the core platform's access controls by supporting a high degree of flexibility and automation in sharing logic. This document outlines how the FormulaShare app operates to enable secure data sharing without data leaving your org, the architectural safeguards in place, and recommended practices for maintaining a secure and compliant environment.

2. Product Overview

FormulaShare allows you to set up rules in your Salesforce org that define record visibility logic based on field contents and records not usually accessible using traditional Salesforce sharing rules.

Unlike standard sharing rules where a rule is required for each user, role or group requiring access, FormulaShare rules can describe dynamic conditional sharing, so a single sharing rule can define the sharing for many different entities.

Record sharing implemented through FormulaShare can be updated on a scheduled basis or in near real-time using Apex triggers or record-triggered flows.

Three tiers of the FormulaShare app are available:

- **Lite** – Provides a limited feature set and support for a small number of rules and shared records
- **Enterprise** – Unlocks all features and provides a larger allowance of rules and shared records
- **Unlimited** – Includes all features and removes all limitations around numbers of rules and numbers of shared records

A single managed package is used to provide each of these product tiers, with features of paid tiers unlocked remotely by Cloud Sundial (the developer and vendor of FormulaShare). This approach means there's never a need to reinstall or upgrade when moving between tiers, and all rules and configuration are retained through this process.



For up-to-date details of pricing and features included in each of these product tiers, check <https://cloudsundial.com/formulashare-versions>.

3. Security Considerations

- FormulaShare has passed the rigorous Salesforce security review, which is indicated in the app's [official listing](#) on the Salesforce AppExchange. This review is carried out on a periodic basis, and involves a combination of automated and manual penetration testing to ensure that the app operates securely within the orgs in which it is installed and provides adequate protection against typical threat vectors. For full details on what is checked as part of this process check Salesforce's [Security Review](#) overview
- FormulaShare was first published in 2019, and is actively maintained through annual major releases, as well as maintenance releases when necessary. As a relatively mature app, more recent releases have been limited in scope with low risk to subscriber orgs. Our [Upgrade Process](#) article includes more information on how we manage and roll out upgrades.
- FormulaShare is distributed through a first-generation managed package, built and published from our Cloud Sundial FormulaShare packaging org, with deployments to this org managed through build automation from our CI/CD environment. Our packaging org permits no integrations other than secure deployments from our CI/CD environment. Multi-factor authentication is a requirement of both platforms, and user access is restricted to users directly responsible for build and deployment.
- The FormulaShare team do not have access to your org and your record data unless this access is explicitly granted within your org. Where this access has been provided, our team access through our Cloud Sundial Partner Business Org. This org is accessed only by our support users, is secured with multi-factor authentication, and no external systems are integrated with this org. Management of your data and appropriate confidentiality considerations are outlined in full in our Data Security Policy (available to existing customers on request).
- FormulaShare is installed and trusted in over 100 production customer orgs. These customers span various sectors including government, healthcare, manufacturing, financial services, education and nonprofit, and include several Global Fortune 500 companies.
- No credentials or external systems are used by the app. FormulaShare runs entirely on Salesforce's platform and no record data leaves the org in which FormulaShare is installed.



4. Compliance Alignment

FormulaShare helps customers meet data access compliance needs such as:

- Role-based data minimisation (e.g. GDPR and HIPAA access segregation)
- Providing evidence of record access logic through clear rules accessible to admin users.
- Support for internal or external security audits by demonstrating sharing logic is managed through our Salesforce security-reviewed application.
- Full logging and reporting to enable auditing of changes to sharing carried out by FormulaShare.

5. Technical Architecture

FormulaShare is a native Salesforce application. This means all of the processing logic and solution elements exist and operate entirely within the Salesforce org in which the app is installed - there are no API integrations with external systems, and other than very high-level telemetry data on usage of the app itself, no data is transferred from your org to other platforms.

This approach to architecture reduces threat vectors significantly - with the app built and operating entirely within Salesforce, only officially Salesforce supported development methodologies and frameworks are used, and your org's existing security ensures you retain complete control of visibility of your org's data.

As there is no reliance on services outside of your org, the availability of the app matches your Salesforce org's availability.

The app is updated to comply with current Salesforce security standards and best practices. Some of the Salesforce technologies we utilise as part of FormulaShare include:

- Lightning Web Components
- Trigger-handled platform events for asynchronous processing
- Scheduled Apex and batch Apex for large volume processing
- Lightning Flow Templates
- Custom metadata for storing app and rule configuration
- Permission sets providing access controls to the app itself



6. Data Shared with Cloud Sundial

Your data is stored entirely within your org and without explicit access being granted, this data is not accessible to Cloud Sundial or any other entity associated with FormulaShare.

As an approved Salesforce Independent Software Vendor, from time to time Cloud Sundial may offer to access your org to help to troubleshoot an issue. In these circumstances you can choose to grant Cloud Sundial temporary access to your Salesforce instance using Salesforce's standard functionality designed for this purpose. When you provide this access in this way, Cloud Sundial will operate in line with the Cloud Sundial Data Security Policy.

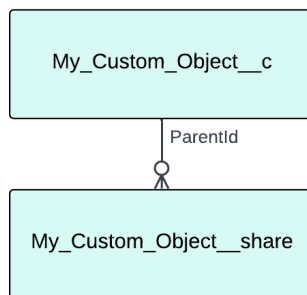
FormulaShare uses [Salesforce Feature Parameters](#), which provide a means for Cloud Sundial to activate and monitor use of FormulaShare features in your org from Cloud Sundial's own partner Salesforce org. We use feature parameters in the following ways:

- Activating FormulaShare Lite, Enterprise or Unlimited versions in your org, or increasing or reducing FormulaShare limits applied in the org.
- Receiving high level telemetry data from your org pertaining to use of the app. This data helps us identify when errors are encountered during FormulaShare processing, understand how close your org is to its FormulaShare usage limits, and monitor which FormulaShare features are being used in your rules and configuration. This **does not** include any data relating to the configuration of your rules, your data model or the records being shared.

The telemetry data we collect is outlined in full in our [Version Activation and Telemetry Data](#) article.

7. Design Paradigm

FormulaShare works by interacting with the core Salesforce feature known as Apex Managed Sharing. Whilst full details are available in the [Salesforce documentation](#), a key concept is that each object which can be shared (i.e. objects with an internal or external org-wide default sharing setting of private or public read-only) has an underlying share object. The presence of share object record indicates an access grant to the specified user, role or group:



FormulaShare works by assessing which share records should be created or removed to provide sharing which is consistent with the definition of the rules created through the app and the current state of the data relating to the record to be shared.

The actual implementation of visibility controls (i.e. showing and hiding records in the UI, reports, API access and other areas) is handled entirely through standard Salesforce functionality.

In this way, FormulaShare helps to control rather than bypass the Salesforce sharing model, ensuring core platform security is maintained in all contexts.

8. Application Architecture

FormulaShare utilises the following to process share logic within your org:

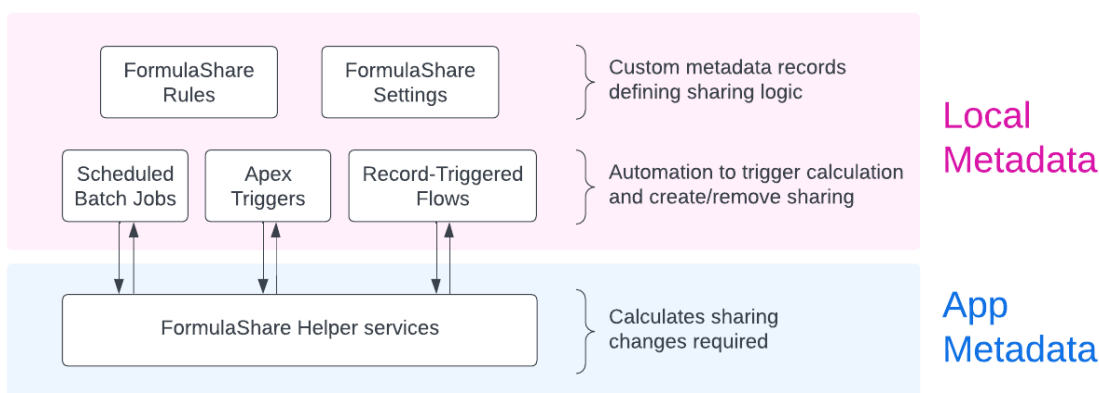
- **FormulaShare Rules**
Rules are stored as custom metadata records, and are created and updated via the [FormulaShare Rules custom interface](#) provided as part of the app. Rules specify the field in your org containing the target user, role or group to share with, as well as access level and other configuration relating to the rule.
- **FormulaShare Settings**
FormulaShare provides [object-specific](#) and [app-wide](#) configuration settings. These settings are stored as custom metadata records and updated through the [FormulaShare Setup](#) interface.
- **Scheduled Batch Jobs**
FormulaShare provides options for regular scheduled recalculation of records. If the [full recalculation batch](#) is used, the packaged Apex class FormulaShareProcessSchedulable will be scheduled in line with the frequency options selected. If [Targeted Calculation Jobs](#) are used, the class FormulaShareTargetedJobsSchedulable will be scheduled.



- **Optional Apex Triggers or Record-Triggered Flows**

Sharing changes can be applied in real-time on record creation and update by calling packaged helper methods from either Apex triggers or flows. Full information on these options is provided in the [Sharing in Real Time](#) section of our help documentation.

These elements are split between **local metadata** (components which are created directly in the org during configuration of FormulaShare) and **app metadata** (components within the FormulaShare managed package which are upgraded and maintained through package upgrades). The diagram below provides an overview of how these elements work together:



It's important to note that in line with Salesforce security best practices, all calculations and sharing changes are carried out in the context of the running user. This ensures the app is not able to carry out operations which are beyond the permissions of the running user.

For this reason, it's important to ensure that batch jobs are scheduled by a user with suitable permissions, typically modify all access to shared objects, and field and object access to all objects and fields involved in rules. When FormulaShare is called by an Apex trigger, the local code to insert or delete sharing should generally be carried out without sharing unless you're confident all users initiating the automation will have full access to the shared records (provided through ownership or the Modify All object permission).

9. Monitoring & Auditing

- FormulaShare tracks job runs and sharing operations carried out through triggers, flows or batches in [FormulaShare Logs](#).



- Logs are fully reportable or queryable through SOQL, UI inspection or through the [dashboard or reports](#) provided as part of the app.
- Any errors or warnings which are encountered during processing are detailed within the log records, and alerts can be easily configured to allow admins to take action when errors are encountered.
- No shared record data is held in the logs themselves, and standard permissions can be used to provide or restrict access to log records for different user groups.

10. Best Practices

Whilst not exhaustive, the following guidelines should be considered to ensure FormulaShare is used in a secure manner:

- Use a combination of real time sharing where possible and scheduled calculations for each object to ensure sharing always matches the current configuration of rules.
- When sharing custom objects, use a single Apex Sharing Reason per rule, and do not create Apex sharing using these sharing reasons in any other way.
- Avoid full ownership mode on standard objects unless you're confident no other sharing logic is in use.
- For record-triggered flows, utilise the flow template provided to ensure that the application of sharing is in line with FormulaShare guidance.
- For Apex code, ensure tests are written which cover the execution of FormulaShare code, and adhere to company standards and general development best practices.
- After deactivating one or more rules, carry out a full recalculation on this object to ensure that any sharing previously applied for this rule is removed.

Appendix: References and Related Documents

Company and Product Policies

Cloud Sundial Privacy Policy: <https://cloudsundial.com/privacy-policy>

Cloud Sundial Data Security Policy: Available to existing customers on request



Cloud Sundial Support Policy: <https://cloudsundial.com/Cloud-Sundial-Subscription-Support-Policy>

FormulaShare App Details and Documentation

FormulaShare AppExchange listing:

<https://appexchange.salesforce.com/appxListingDetail?listingId=a0N3A00000FR5TCUA1>

FormulaShare Help & Configuration guide: <https://cloudsundial.com/formulashare-configuration>