

Experience Cloud Portal Uploads to SharePoint for the Public Sector

Let applicants and partners upload into SharePoint from a Salesforce portal, then keep each submission read-only

What problem does this solve for public sector teams?

CloudFiles is Document Automation for Salesforce. This guide describes a pattern we see in the public sector: an agency runs an Experience Cloud portal where residents, applicants, grantees or partner organizations submit applications, and every submission comes with supporting documents. Those documents need to land in the agency's SharePoint, filed against the right Salesforce record, and they must not change once the application is submitted.

The problems are plain:

- External users have no account in the agency's Microsoft 365 tenant, and nobody wants to create one for every applicant.
- Files uploaded to the portal as Salesforce Files fill Salesforce storage and sit apart from the SharePoint libraries where caseworkers keep everything else.
- Staff copy files by hand from Salesforce into SharePoint and paste SharePoint URLs onto records, and those stored URLs break when folders move.
- Reviewers need a fixed record of exactly what was submitted.

CloudFiles Document Management handles Salesforce portal document upload from start to finish. External users upload without a SharePoint account, the files go into SharePoint folders linked to the Salesforce record, and a Flow keeps each submission read-only after submission. The same app connects Salesforce to SharePoint, AWS S3, Google Drive and Azure Blob, so an agency that archives in AWS S3 or Azure Blob, or runs some programs in Google Drive, uses one portal pattern for all of them.

Which Salesforce objects are involved?

A typical build uses:

- **Account** or **Contact** for the applicant or organization.
- A custom **Application** object (or Grant, Permit, License Request) holding the submission and its Status field.
- **Case** for follow-up questions after submission.

- The Experience Cloud site and its community users.

CloudFiles works on any standard or custom object, so the Application object gets its own folder the same way an Account does.

Where are the portal files stored?

The files are stored in the agency's own SharePoint, in a library the admin chooses, with one record-linked folder per Application. There is one copy, in SharePoint: CloudFiles works on the live files through Microsoft's own API. Files added from the portal appear in SharePoint, and files a caseworker adds in SharePoint appear on the Salesforce record once the folder is linked. Folder links survive record renames, so renaming an Application breaks nothing.

How does Experience Cloud file upload to SharePoint work, step by step?

1. **Connect SharePoint with a service principal.** CloudFiles authenticates as the CloudFiles app in the agency's Microsoft tenant. It uses no license, has no password or MFA to break, and shows an "Active for automations" badge. The Microsoft 365 admin can limit CloudFiles to only the SharePoint sites the agency chooses, granting access site by site.
2. **Create the folder when the application is created.** A record-triggered Flow on Application create runs **Create Folder** in the program's library, named by application number. About a third of CloudFiles actions are asynchronous, so when the async version is used, a platform-event Flow handles the completion event.
3. **Share the folder with the applicant.** The same Flow runs **Share with Community User**, so the portal widget shows that folder to that applicant.
4. **The applicant uploads from the portal.** The CloudFiles widget on the Experience Cloud page lets the applicant upload straight into the SharePoint folder. External users upload without a SharePoint account.
5. **Stamp metadata. Update Drive Columns** writes Program, Application Number and Status into SharePoint columns, so caseworkers can filter the library in SharePoint itself.
6. **Lock on submission.** A record-triggered Flow on Status changing to Submitted runs **Update Permissions** to revoke write access on the submission folder, and **Update Drive Columns** sets the Status column to Submitted. The files are read-only after submission.
7. **Remove access at close.** When the application closes, **Remove from Community** takes the folder out of the applicant's portal view. **Get Shared Resources** lists what a community user can still see, which is useful for audit.

Trigger	Flow action	Result
Application created	Create Folder, Share with Community User	Folder in SharePoint, visible to that applicant
Documents uploaded	Update Drive Columns	Program and status in SharePoint columns
Status set to Submitted	Update Permissions (revoke write)	Read-only after submission
Application closed	Remove from Community	Applicant no longer sees the folder

How do permissions and sharing work for external users?

External users never need SharePoint access. They see only what a Flow has shared with them through the portal widget.

Staff access follows Salesforce. Flow automation runs as the service principal; the widget and Content Library run as each user through their own storage connection. Flows keep SharePoint folder and site-group access in step with Salesforce owners, teams and status, using **Update Permissions, Create SharePoint Site Group** and **Update SharePoint Site Group Members**. For example, when an application is reassigned to a new caseworker, a Flow adds that caseworker to the program's site group. Users need no SharePoint rights for the widget experience when an admin designs it that way.

When a reviewer needs to send files outside the portal, for example to an outside assessor, **Create Link** issues a share link with an email wall, password, email verification code and consent step. Links can expire and can carry a view limit, every view is tracked with location, and every download is tracked per file.

Can the same portal pattern use AWS S3, Google Drive or Azure Blob?

Yes. External users can upload into SharePoint, Google Drive, AWS S3 or Azure Blob without having an account in that storage. The same folder Flows and Experience Cloud sharing actions work on each storage. The read-only lock described above uses SharePoint permissions through Update Permissions.

What are the security and compliance facts?

- ISO/IEC 27001:2022 certified, with ISO/IEC 27017:2015 and ISO/IEC 27018:2019.
- Independently audited for SOC 2 Type II.
- Compliant with HIPAA (BAA available). Compliant with GDPR.
- Hosted on AWS in four regions selected per customer account: United States, EU, United Kingdom and Australia.
- TLS 1.2 minimum in transit; AES-256 at rest on production data stores.
- Your files stay in your own storage. CloudFiles transfers them over encrypted connections and does not re-host them.

Security documents (ISO certificates, SOC 2 Type II report, penetration test summary) are available under NDA from the CloudFiles Trust Center.

Get started

Install CloudFiles Document Management from AgentExchange and start a free 14-day trial, or book a demo at cloudfiles.io.

CloudFiles, Document Automation for Salesforce · cloudfiles.io · Install CloudFiles Document Management from AgentExchange