

CASE STUDY 03 OF 07 | CONNECT APP

File Sharing& In-App Preview

Upload files, preview them instantly, and keep everything in context

#FileSharing #Preview #Storage #UX

Overview

Files and conversations belong together. When someone shares a design mockup, a contract draft, or a data report in a channel, the team should be able to open it without leaving the app, see it in context with the conversation around it, and not have to hunt for it later. This case study covers how file sharing and in-app file preview were built into Connect.

Project Snapshot

Project	Connect — Workplace Communication Platform
Feature Area	File Sharing and In-App Preview
Role	Senior Fullstack Engineer
File Types	Images, PDFs, documents, spreadsheets, videos, archives
Upload Method	Direct-to-storage via signed URLs, background upload
Outcome	Fast, reliable file sharing with instant in-app preview

The Problem

Sharing files through a messaging tool is often a clunky experience. The file goes somewhere, a link appears in the chat, the recipient clicks it, gets redirected to a download, opens a separate application, and by that point the conversation context is gone. For teams sharing files constantly, this friction compounds quickly.

The goal was to make file sharing feel like a first-class part of the conversation. Drop a file, see it previewed inline, discuss it in the thread, and find it again in the conversation's file history without having to remember where it was sent.

What Was Built

File Upload

Files are uploaded via drag-and-drop into any message input, through a file picker button, or by pasting directly from the clipboard. The upload begins immediately in the background while the user continues composing their message. A progress indicator appears in the message input area. When the upload completes, the file attachment is ready to send with the message.

Large files are chunked and uploaded in parallel segments, which significantly improves upload speed on fast connections and makes resumable uploads possible for files that are interrupted mid-upload.

Inline Preview by File Type

Once a file is shared in a conversation, it renders inline in the message feed with a preview appropriate to its type:

- Images render as full thumbnails directly in the message, expanding to full size on click.
- PDFs open in an embedded document viewer within the app, showing the full document with page navigation.
- Video files show a thumbnail and play inline with a built-in video player, without requiring download.
- Spreadsheets and office documents show a rendered preview of the first page or sheet.
- Archives and unknown file types show a file card with the file name, size, and type icon, with a download button.

File Gallery per Conversation

Every conversation has a files tab that shows all files ever shared in that conversation, sorted by date and filterable by file type. This makes it easy to find a specific file without scrolling through the entire message history. Clicking any file in the gallery opens the in-app preview.

Inline Annotations and Comments

For image and PDF previews, users can add comments tied to specific points in the file. This turns a shared design or document into a collaboration surface without needing a separate review tool.

Technical Architecture

Direct-to-Storage Upload Files are uploaded directly to object storage via pre-signed URLs generated by the server. The API never handles the file bytes, which removes the server as a bottleneck and keeps upload speeds consistent regardless of API load.	Chunked Multipart Upload Files over a threshold are split into chunks and uploaded in parallel. A manifest tracks which chunks have landed successfully. Failed chunks are retried independently, making the upload resumable without starting over.
Preview Generation Pipeline On upload completion, an async job generates preview assets: thumbnails for images, first-page renders for PDFs and documents, video poster frames. Previews are stored in a separate CDN-cached bucket for fast delivery.	CDN-Backed Delivery All file assets and previews are served via CDN edge nodes. Files are delivered from the closest edge location to the recipient, keeping load times fast regardless of where the file was originally uploaded from.
Access Control	File Indexing

File access is tied to conversation membership. A file shared in a private channel is only accessible to members of that channel. Pre-signed download URLs are scoped and time-limited, with access checked at generation time.

File metadata (name, type, uploader, timestamp, conversation) is indexed in a search service. Users can search for files by name across all conversations they are a member of.

Key Challenges

Preview Quality for Complex Document Types

Office documents (Word, Excel, PowerPoint) vary enormously in complexity and formatting. Generating a clean, accurate preview without a full Office installation on the server required integrating a document rendering service that handles the common formatting cases reliably. Complex documents that do not render well fall back to a clean download card rather than showing a broken preview.

Video Playback Without Full Download

Video files can be large, and waiting for a full download before playback starts is a poor experience. The video delivery pipeline segments uploaded videos into short chunks using HLS and serves them progressively. The in-app player loads the first segment immediately and buffers subsequent segments in the background, so playback begins in seconds regardless of file size.

Access Control Under Conversation Membership Changes

When a member is removed from a channel, they should no longer be able to access files that were shared in that channel. File access is re-validated at the pre-signed URL generation step rather than at file storage level, so revoking channel membership immediately prevents further file access without needing to move or re-encrypt any stored files.

Outcome

File sharing in Connect became a seamless part of the conversation rather than an interruption of it. Teams sharing design files, documents, and data reports could review everything inline without context-switching to another tool. The file gallery per conversation eliminated the common complaint of losing track of files sent days or weeks earlier. Upload speeds for large files improved significantly through parallel chunked uploads, and the CDN delivery kept preview load times fast for distributed teams.

Engineering Highlights

- Direct-to-storage uploads via pre-signed URLs keeping the API out of the file transfer path
- HLS video segmentation enabling instant playback without full download
- Async preview generation pipeline covering images, PDFs, documents, and video
- Conversation-scoped access control with time-limited pre-signed delivery URLs