CASE STUDY 05 OF 07 | CONNECT APP

Call Recording

Capturing meetings so nothing important is ever lost

#Recording #Video #Storage #Privacy

Overview

Not everyone can attend every meeting. And even for people who do attend, specific details from a call can be difficult to recall accurately afterward. Call recording in Connect gives teams a reliable way to capture meetings, review them later, and share them with colleagues who were not present. This case study covers how call recording was designed and built, including the technical implementation and the privacy considerations that shaped every decision.

Project Snapshot

Project	Connect — Workplace Communication Platform
Feature Area	Call Recording
Role	Senior Fullstack Engineer
Recording Type	Server-side composite recording with selectable layouts
Output Format	MP4 with H.264 video and AAC audio
Outcome	Reliable, consent-aware recordings stored and accessible per conversation

The Problem

Teams working across timezones or with members who have schedule conflicts have no good way to share the content of a meeting after the fact. Notes help, but they miss tone, context, and the back-and-forth that often contains the most important information. Screen recordings made with external tools are clunky, lack audio/video quality, and are usually shared as large file attachments in email rather than in the context of the conversation.

Call recording needed to be native to Connect: one button to start, one place to find the recording afterward, and a consent system that informed all participants before capturing anything.

What Was Built

Recording Initiation and Consent

Any participant in a call can start recording by clicking the record button in the call controls. The moment recording begins, all other participants receive a visible, unmissable notification that the call is being recorded. This is not a passive banner: it requires acknowledgment before the participant can continue. Participants who do not consent to being recorded can leave the call without penalty.

The recording state is persistent: if the person who started the recording leaves the call, the recording continues until someone explicitly stops it or the call ends.

Server-Side Composite Recording

Recording happens on the media server, not on any participant's client. The server composites all participant video streams, shared screens, and audio into a single MP4 file. This approach produces a clean, consistent output regardless of individual participants' device capabilities or network conditions. It also means the recording continues even if the initiator's device loses connectivity.

Recording Storage and Access

Completed recordings are stored in object storage and linked to the conversation where the call took place. They appear in the conversation's files section and in a dedicated recordings tab. Access is governed by the conversation's membership rules: a recording from a private channel is only accessible to members of that channel.

Playback and Sharing

Recordings play back in the in-app video player, which supports scrubbing, speed adjustment, and volume control. The call chat messages that were sent during the recording are displayed alongside the video playback, synchronized to the time they were sent during the call. Authorized users can generate a shareable link for a recording that gives access to a specific person for a limited time.

Technical Architecture

Composite Recording on SFU The same SFU media server that handles live call routing also performs composite recording. A recording session subscribes to all participant streams and feeds them into a compositor that combines video and audio into a single output file in real time.	**MP4 Finalization Pipeline** When recording stops or the call ends, the compositor flushes its buffer and finalizes the MP4 container. A post-processing job normalizes audio levels, verifies file integrity, and moves the file to the delivery bucket.
Progressive Upload During Recording For long calls, waiting until the call ends to upload the recording is risky: if the server restarts, the recording could be lost. The compositor writes segments progressively to object storage during the call, with the final finalization step assembling them into the complete file.	**Consent Tracking** Consent acknowledgments are stored per recording per participant as a timestamped record. This creates an audit trail showing who was notified and when, which is important for compliance in organizations subject to recording disclosure requirements.
HLS Delivery for Playback Completed recordings are transcoded into HLS format for adaptive playback. The player fetches	**Retention Policies** Organization administrators can set recording retention policies that automatically delete

the appropriate quality segment based on the viewer's available bandwidth, so playback starts quickly and adjusts quality dynamically.	recordings after a configured number of days. Deletion runs via a scheduled job with soft-delete first and hard-delete after a grace period to allow recovery from accidental policy misconfiguration.

Key Challenges

Recording Long Calls Reliably

A call that runs for two hours generates a significant amount of media data. Holding all of that in memory before writing to storage is not viable. The progressive segment upload approach solves this by writing completed segments to object storage continuously, so memory usage stays bounded regardless of call length. If anything goes wrong mid-call, all segments up to that point are already safely stored.

Composite Layout for Variable Participant Counts

The composite layout needs to look reasonable whether there are two participants or fifteen. The compositor uses an adaptive grid layout that tiles participants' video streams based on the current participant count. Screen shares always take priority and are displayed as the main panel with participant thumbnails alongside. Layout changes (participants joining or leaving mid-recording) are applied smoothly without interrupting the recording stream.

Compliance Across Different Regulatory Environments

Recording notification requirements vary by jurisdiction. In some regions, all-party consent is required; in others, one-party notification is sufficient. The consent system is configurable at the organization level, allowing administrators to set the required disclosure behavior to match their regulatory environment. The consent acknowledgment audit trail gives compliance teams the documentation they need.

Outcome

Call recording became one of the most used features for distributed teams in the organization. Teams with members in different timezones stopped scheduling redundant catch-up calls for people who missed a meeting. Important decisions discussed on calls were reviewable rather than relying on someone's notes. The consent system built trust with users who were initially concerned about recording, and the audit trail satisfied the compliance requirements of regulated-industry organizations using the platform.

Engineering Highlights

- Server-side composite recording independent of client device or network quality
- Progressive segment upload keeping recordings safe during long calls
- Synchronized call chat replay alongside video playback
- Configurable consent and retention policies for compliance across regulatory environments