

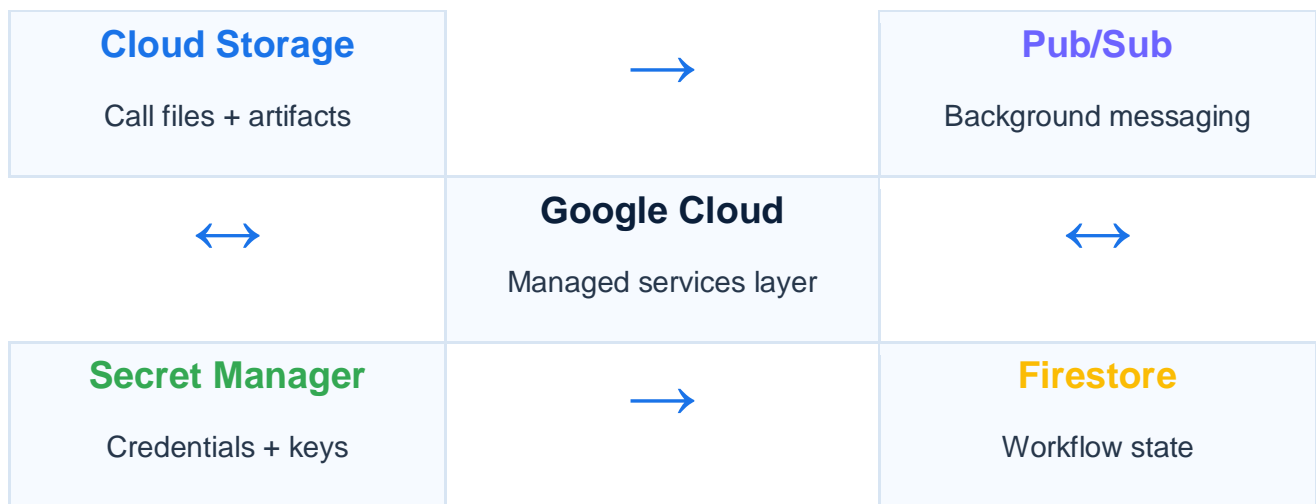
USE CASE DOCUMENT

# How ZipZap Uses Google Cloud to Power Call Migration and Processing

A publish-ready case study on how Google Cloud supports secure file handling, credential protection, asynchronous processing, workflow tracking, and multi-service delivery for a call migration platform.

## Google Cloud Architecture Backbone

Storage, secrets, messaging, state, and delivery working together for long-running call data operations.



## EXECUTIVE SUMMARY

# Google Cloud enables secure, scalable call migration operations

ZipZap uses Google Cloud to securely manage call migration and processing with Cloud Storage, Secret Manager, Pub/Sub, and Firestore. These services support reliable file handling, credential protection, background workflows, and operational visibility.

Use Case Area	Google Cloud Capability
Call file storage and generated artifacts	Google Cloud Storage
Secure handling of platform credentials and keys	Google Secret Manager
Background job communication across services	Google Pub/Sub
Job, call, and workflow state tracking	Google Firestore
Container build and delivery workflow	Cloud Build and Artifact Registry
User-friendly authentication support	Google sign-in through the frontend

## THE CHALLENGE

# Call migration platforms need more than a frontend and API

Large files, sensitive credentials, long-running jobs, and workflow state must be handled without creating operational bottlenecks or security risks.

<ul style="list-style-type: none"><li>● <b>Reliable file storage</b></li></ul> <p>Audio, transcripts, and generated records can be large, binary, and shared across services.</p>	<ul style="list-style-type: none"><li>● <b>Credential protection</b></li></ul> <p>External integrations require safe handling of tokens, service keys, and customer configuration.</p>
<ul style="list-style-type: none"><li>● <b>Asynchronous work</b></li></ul> <p>Migration, validation, indexing, and audit tasks should not block user-facing requests.</p>	<ul style="list-style-type: none"><li>● <b>Workflow progress</b></li></ul> <p>The platform must track running jobs, processed calls, failures, and retry needs.</p>
<ul style="list-style-type: none"><li>● <b>Consistent delivery</b></li></ul> <p>A multi-service backend needs repeatable container builds and central image storage.</p>	

## ZipZap approach

The platform architecture includes API, job, indexer, audit, and watchdog components. This points to a service-oriented model designed to manage integrations, process call-related data, and monitor progress across asynchronous tasks.

## Managed services mapped to clear product responsibilities

Each Google Cloud capability plays a focused role in the overall ZipZap architecture.

<p><b>■ Cloud Storage</b></p> <p><b>File and artifact storage</b></p> <p>Durable object storage for call files, generated artifacts, metadata exports, and sample audio used across processing services.</p>	<p><b>■ Secret Manager</b></p> <p><b>Credential and key protection</b></p> <p>Secure storage and retrieval for integration credentials, service keys, infrastructure secrets, and versioned secret access.</p>
<p><b>■ Pub/Sub</b></p> <p><b>Asynchronous communication</b></p> <p>A message-driven backbone that lets API, job, indexer, audit, and watchdog services work without tight coupling.</p>	<p><b>■ Firestore</b></p> <p><b>Workflow state tracking</b></p> <p>A document-oriented state layer for users, jobs, calls, watchdog entries, progress, failures, and retries.</p>
<p><b>■ Cloud Build + Artifact Registry</b></p> <p><b>Build and delivery pipeline</b></p> <p>Repeatable container builds and centralized image storage for packaging and promoting backend services consistently.</p>	<p><b>■ Google sign-in</b></p> <p><b>Frontend authentication support</b></p> <p>A familiar login flow that supports a SaaS-style user experience for business users.</p>

### ARCHITECTURE FLOW

## Connected workflow from user request to tracked processing state

The architecture separates user-facing requests from background processing, while keeping files, secrets, and workflow status managed through Google Cloud services.



Request → API Layer → Pub/Sub → Worker Services → Firestore State

## Supporting cloud services

<p><b>● Cloud Storage</b></p> <p>Call files and generated artifacts</p>	<p><b>● Secret Manager</b></p> <p>Credentials and keys</p>	<p><b>● Cloud Build / Artifact Registry</b></p> <p>Container build and delivery</p>
-------------------------------------------------------------------------	------------------------------------------------------------	-------------------------------------------------------------------------------------

## BUSINESS + TECHNICAL BENEFITS

# A cleaner, more reliable foundation for call migration workflows

### ● Scalable file handling

Call-related assets can be stored and accessed by multiple services without overloading databases or application servers.

### ● Stronger security posture

Sensitive integration credentials are separated from application code and ordinary workflow data.

### ● Reliable background processing

Long-running migration, indexing, verification, and audit work can run asynchronously with worker-level retry and scaling patterns.

### ● Improved workflow visibility

Firestore provides a structured place to track job progress, call status, failures, and watchdog data.

### ● Cleaner multi-service delivery

Cloud Build and Artifact Registry support repeatable packaging for separate backend services.

### ● Better product experience

Google sign-in supports a simpler authentication flow for users.

## Positioning

ZipZap demonstrates how a focused cloud architecture can support a complex operational product. Instead of relying on one large backend service, the platform uses managed Google Cloud services to separate responsibilities across storage, secrets, messaging, state, and delivery. This makes the system easier to scale, operate, and evolve as call migration and processing needs grow.

**Summary: ZipZap uses Google Cloud as the backbone for secure integration storage, file handling, asynchronous processing, and workflow tracking in a call migration platform.**