Legitly Backend Flow Architecture

Complete Backend Architecture & Technical Documentation

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Executive Summary

The Legitly backend is built on Google Apps Script with a sophisticated multi-layered architecture designed for enterprise-grade content verification. The system processes text, images, and video content through advanced AI analysis using Google's Gemini API, with intelligent caching, cyber crime detection, and multi-language support.

# System Overview

Legitly operates as a serverless web application that provides real-time content verification services. The backend handles three primary content types:  
• Text Analysis - News claims, articles, and textual content  
• Image Analysis - Photo authenticity and manipulation detection   
• Video Analysis - URL validation and deepfake detection (in development)  
  
The system is designed with security, performance, and scalability as core principles.

# Architecture Components

## 1. Web App Entry Point

Function: doGet()  
Purpose: Serves the HTML interface when users visit the deployed URL  
Security: Frontend runs in browser, backend executes on Google's secure servers  
API Protection: Gemini API key remains server-side, never exposed to clients

function doGet() {  
 return HtmlService.createHtmlOutputFromFile('index').setTitle('Legitly');  
}

## 2. Main Verification Pipeline

Function: performVerification(requestData)  
  
Request Flow:  
1. Frontend initiates call via google.script.run.performVerification()  
2. System generates unique request ID for tracking  
3. Performance monitoring begins (processing time measurement)  
4. Multi-step processing: Validation → Caching → Analysis → Response  
  
The pipeline ensures every request is properly tracked, validated, and processed through appropriate analysis channels.

## 3. Advanced Request Validation

Function: validateRequest(requestData)  
  
Comprehensive Validation Checks:  
• Content Type: Validates against supported types (text, news, image, deepfake, video)  
• Data Presence: Ensures required data field exists and is properly formatted  
• Language Support: Validates against 25+ supported languages  
• Content Limits: Text (50,000 characters max), URLs (2,000 characters max)  
• Security: Prevents malformed or malicious input injection  
  
The validation layer acts as the first line of defense against invalid or malicious requests.

## 4. Intelligent Caching System

Functions: getCachedResult(), setCachedResult()  
  
Performance Optimization Features:  
• SHA-256 Hashing: Generates unique cache keys from content + language + type  
• 1-Hour Cache Duration: Reduces API calls and improves response times  
• Smart Cache Miss Handling: Falls through to fresh analysis when cache expires  
• Memory Management: Uses Google Apps Script's CacheService for efficiency  
  
This caching system significantly reduces API costs while maintaining fresh analysis for new content.

## 5. Content Analysis Routing Engine

The system intelligently routes different content types to specialized analysis functions:  
  
• Text/News Content → analyzeTextContent()  
• Image Content → analyzeImageContent()   
• Video/Deepfake Content → analyzeVideoContent()  
  
Each routing path is optimized for the specific content type's analysis requirements.

if (requestData.type === 'text' || requestData.type === 'news') {  
 result = analyzeTextContent(requestData, requestId);  
} else if (requestData.type === 'image') {  
 result = analyzeImageContent(requestData, requestId);  
} else if (requestData.type === 'deepfake' || requestData.type === 'video') {  
 result = analyzeVideoContent(requestData, requestId);  
}

## 6. Gemini AI Integration Architecture

Function: callGeminiAPI(prompt, requestId, type, mediaData)  
  
AI Processing Pipeline:  
• Model: Uses gemini-1.5-flash-latest (Google's latest production model)  
• Multimodal Support: Handles text-only and text+image requests  
• Advanced Prompting: Custom-engineered prompts for maximum accuracy  
• Retry Logic: 3 attempts with exponential backoff for reliability  
• Response Processing: Regex-free JSON extraction prevents parsing errors  
  
The AI integration is designed for maximum reliability and accuracy in content analysis.

## 7. Specialized Analysis Functions

### Text Analysis Engine

Function: analyzeTextContent(requestData, requestId)  
  
Analysis Framework:  
• Comprehensive Assessment: Analyzes credibility, misinformation, context, and risk  
• Cyber Crime Detection: Identifies scams, fraud, and malicious content  
• Multi-language Support: Returns analysis in user's preferred language  
• Structured Prompting: Ensures consistent JSON response format  
  
The text analysis engine uses advanced prompt engineering to guide the AI through a systematic evaluation process.

### Image Analysis Engine

Function: analyzeImageContent(requestData, requestId)  
  
Analysis Capabilities:  
• Visual Authenticity: Detects digital manipulation and editing signs  
• OCR Integration: Extracts and analyzes text content within images  
• Fraud Detection: Identifies fake certificates, documents, and schemes  
• Multimodal Processing: Combines image and text analysis for comprehensive results  
  
The image analysis engine leverages Google's advanced computer vision capabilities.

### Video Analysis Engine

Function: analyzeVideoContent(requestData, requestId)  
  
Current Capabilities:  
• URL Validation: Analyzes video links for suspicious patterns  
• Platform Recognition: Identifies legitimate vs unknown platforms  
• Demo Mode: Professional "under development" messaging for advanced features  
• Deepfake Preparation: Architecture ready for future deepfake detection models  
  
The video analysis system provides immediate URL-level analysis while preparing for advanced deepfake detection.

## 8. Response Processing & Sanitization

Function: sanitizeAnalysisResult(result)  
  
Security & Reliability Features:  
• Input Sanitization: Prevents XSS and injection attacks  
• Data Type Validation: Ensures all fields match expected types  
• Length Limits: Truncates responses to prevent UI overflow  
• Fallback Values: Provides defaults for missing AI response fields  
  
This layer ensures all responses are safe and properly formatted for frontend consumption.

## 9. Cyber Crime Integration

The system includes advanced cyber crime detection and reporting capabilities:  
  
• Automatic Detection: AI identifies potential cyber crimes during analysis  
• Government Integration: Direct links to India's National Cyber Crime Portal  
• Multi-language Alerts: Cyber crime warnings in user's preferred language  
• Helpline Integration: Provides 1930 helpline number for immediate assistance  
  
Portal Integration: https://cybercrime.gov.in/Accept.aspx

## 10. Error Handling & Resilience

Comprehensive Error Management:  
• API Failures: Graceful degradation with retry logic  
• Network Issues: Timeout handling and connection recovery  
• Malformed Responses: JSON parsing error recovery  
• Rate Limiting: Handles API quota and rate limit responses  
• User-Friendly Messages: Technical errors converted to actionable guidance  
  
The system is designed to handle failures gracefully and provide meaningful feedback to users.

## 11. Performance Optimizations

Enterprise-Grade Performance Features:  
• Minimal API Calls: Intelligent caching reduces costs and latency  
• Efficient Memory Usage: Optimized object creation and garbage collection  
• Fast Response Times: Sub-2-second analysis for most content  
• Concurrent Processing: Handles multiple simultaneous requests  
• Resource Management: Proper cleanup prevents memory leaks  
  
Performance metrics are continuously monitored and optimized.

## 12. Configuration Management

Centralized configuration system manages all system parameters:  
  
• API Configuration: Model selection, retry policies, timeout settings  
• Cache Configuration: Duration, prefixes, and storage policies  
• Validation Rules: Content limits, supported languages, security policies  
• Cyber Crime Settings: Portal URLs, helpline numbers, alert messages  
  
All configuration is centralized for easy maintenance and updates.

# Technical Architecture Benefits

• Serverless Deployment: No infrastructure management required  
• Auto-scaling: Google Apps Script handles traffic spikes automatically  
• Cost-Effective: Pay-per-use model with generous free tiers  
• Secure by Default: Google's enterprise security model  
• Global CDN: Fast response times worldwide  
• Version Control: Built-in deployment and rollback capabilities  
  
This architecture provides enterprise-grade reliability, security, and performance while maintaining simplicity for rapid development and deployment.

# Data Flow Summary

Request Flow:  
1. User submits content via web interface  
2. Frontend calls backend via google.script.run  
3. Backend validates request and checks cache  
4. If cache miss, routes to appropriate analysis function  
5. Analysis function calls Gemini AI with specialized prompts  
6. AI response is processed, sanitized, and enhanced  
7. Result is cached and returned to frontend  
8. Frontend displays formatted results with cyber crime alerts if applicable  
  
This flow ensures every request is processed efficiently while maintaining security and reliability.

Generated on September 20, 2025 | Legitly Content Verification System