GoPhishFree

Architecture Document

Course: EECS 582 - Capstone Project

Team Number: 24

Team Members: Ty Farrington, Brett Suhr, Andrew Reyes, Nicholas Holmes, Kaleb Howard

Project Name: GoPhishFree

Date: February 2026

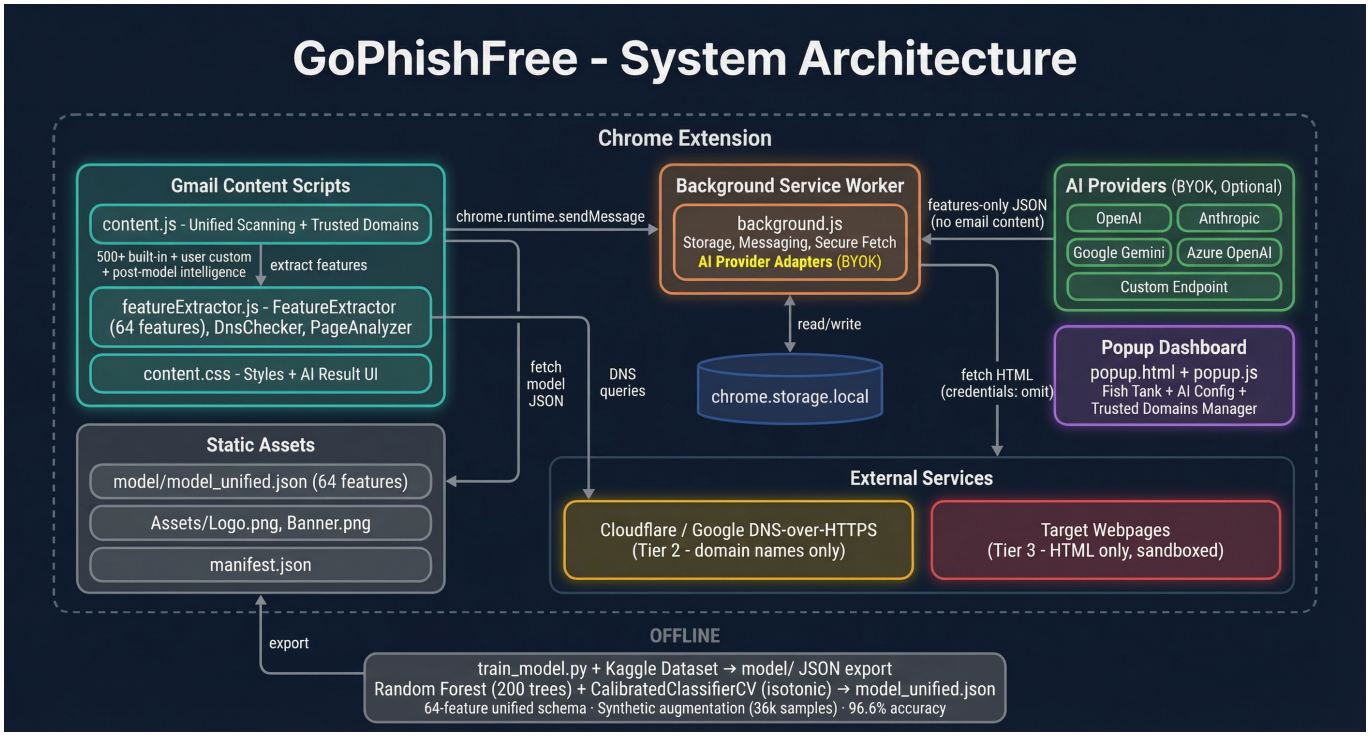
Project Synopsis

A privacy-first Chrome extension that detects phishing emails in Gmail using a unified 64-feature calibrated Random Forest model with optional cloud AI enhancement (BYOK). All core ML processing runs locally; AI receives only extracted signal features - never email body, subject, or sender address.

1. System Overview

GoPhishFree is a Chrome Manifest V3 extension that scans emails in real time as users read them in Gmail. The entire detection pipeline - feature extraction, machine-learning inference, and risk scoring - executes locally inside the browser, ensuring that no email content ever leaves the user's device. The system is composed of five major components: a Gmail content script that orchestrates unified scanning and renders the UI, a feature extraction module that derives 64 numerical signals across 7 groups (URL, custom rules, DNS, page structure, BEC/linkless, attachment, and context flags), a background service worker that manages storage, proxies network requests, and hosts AI provider adapters, a popup dashboard that presents a gamified "fish tank" collection and AI configuration, and an optional cloud AI enhancement layer that provides a second opinion using features-only payloads via the user's own API key (BYOK).

Figure 1 - System Architecture: High-level component map showing content scripts, service worker, popup, AI providers, and external services



2. Unified Detection Pipeline

Detection is organized into three progressive tiers, each adding deeper analysis. Unlike the previous approach with separate models and post-hoc rule adjustments, the current architecture uses a single unified Random Forest model that accepts all 64 features in one pass. Features from unavailable tiers (e.g., DNS not run, Deep Scan not triggered) are default-filled with 0 and signaled via context flags (dns_ran, deep_scan_ran).

Tier 1 - Email Analysis (always active)

When a user opens an email, the content script extracts the sender address, display name, body text, attachment metadata, and every hyperlink. The FeatureExtractor class computes 25 URL lexical features, 9 custom rule features (urgency, credential requests, secrecy language, suspicious TLD, header mismatch), 5 BEC/linkless features (financial request scoring, authority impersonation, phone callback patterns, reply-to mismatch, linkless detection), and 5 attachment features (risky extensions, double extensions, filename entropy). All features are assembled into a 64-element vector via buildUnifiedVector().

Tier 2 - DNS Validation (enabled by default)

If Enhanced Scanning is toggled on (the default), GoPhishFree queries Cloudflare's DNS-over-HTTPS resolver (with Google DNS as a fallback) for the sender's domain and every linked domain. Five DNS features are derived: whether the domain resolves, MX record count, A record count, whether the domain label is a random string based on Shannon entropy analysis, and whether MX records are present. Results are cached with a 10-minute TTL. The dns_ran context flag is set to 1 when DNS features are populated.

Tier 3 - Deep Scan (user-initiated)

Users may optionally trigger a Deep Scan, which fetches the raw HTML of up to 10 linked pages through the background service worker. The fetch is sandboxed: credentials are omitted, responses are capped at 2 MB, content-type is validated, redirects are checked, and no JavaScript is executed. The PageAnalyzer class extracts 13 page-structure features. The deep_scan_ran context flag is set to 1 and the unified model re-scores with the expanded vector.

3. Machine Learning Model

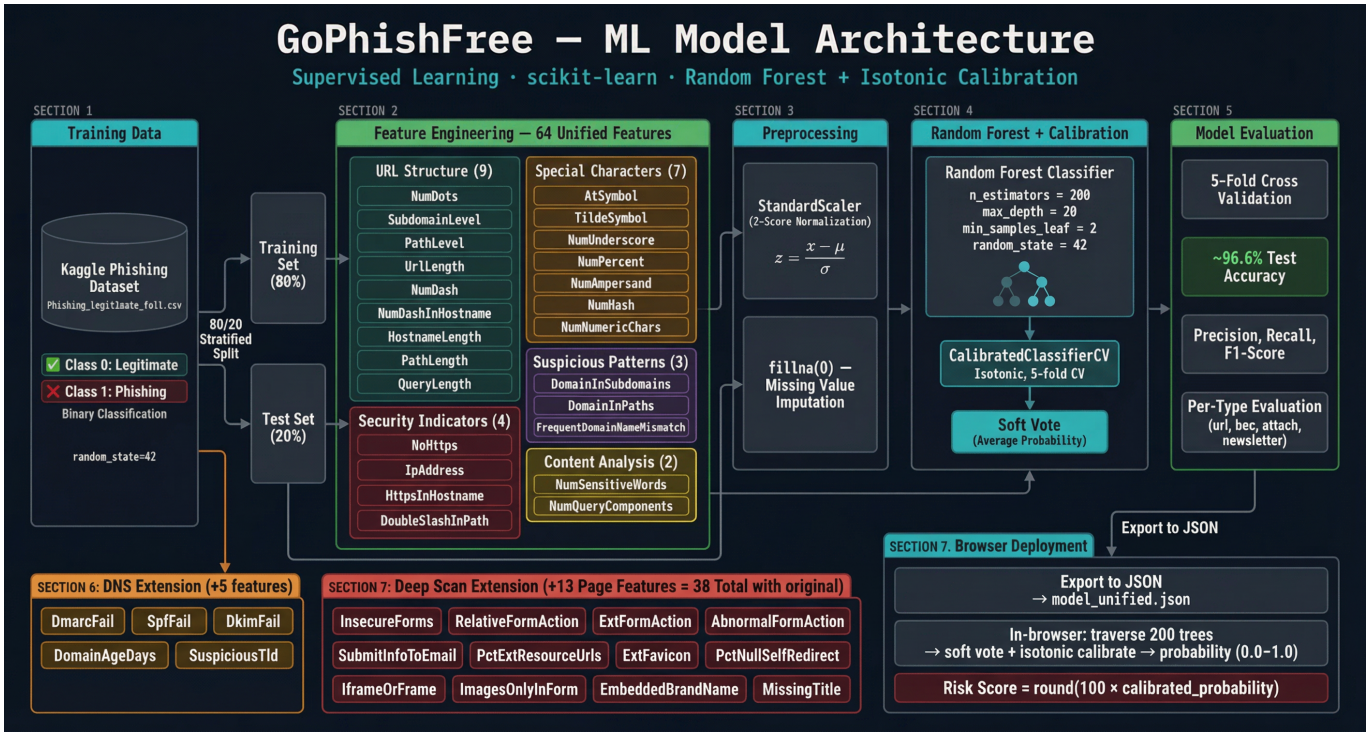
GoPhishFree uses supervised learning with scikit-learn's RandomForestClassifier wrapped in CalibratedClassifierCV for binary classification (legitimate vs. phishing). The training dataset is augmented from approximately 10,000 Kaggle samples to ~36,000 by generating scan-scenario variants (base, DNS-augmented, full) as well as synthetic BEC phishing, attachment phishing, legitimate newsletter, and transactional email samples to teach the model to handle all phishing types and missing feature groups gracefully.

Preprocessing consists of selecting the 64 unified feature columns, imputing missing values with zero, and applying StandardScaler Z-score normalization. The classifier is configured with 200 estimators, a maximum tree depth of 20, a minimum of 2 samples per leaf, and parallel training across all CPU cores. CalibratedClassifierCV applies isotonic regression with 5-fold cross-validation to ensure well-calibrated probabilities.

Model evaluation uses classification reports, confusion matrix, feature importance analysis, and per-phishing-type evaluation (URL-credential, BEC-linkless, attachment-led, deep scan impersonation). The unified model achieves approximately 96.6% test accuracy with 100%% detection on synthetic BEC/linkless and attachment-led phishing samples.

After training, the full Random Forest structure - every tree's split features, thresholds, child pointers, and leaf probabilities - is exported to JSON along with the scaler parameters and the isotonic calibration lookup table (x_values, y_values). At runtime the content script performs inference by traversing all 200 trees, averaging leaf probabilities (soft voting), and interpolating through the calibration lookup to produce a calibrated probability.

Figure 3 - ML Model Architecture: End-to-end pipeline from training data through 64-feature engineering, preprocessing, Random Forest + iso



4. Risk Score Composition & Post-Model Intelligence

The base risk score is computed as: riskScore = round(100 x calibrated_probability). Features that were formerly separate rules (urgency score, credential request score, suspicious TLD, header mismatch) are now direct model inputs, allowing the Random Forest to learn optimal weighting from data.

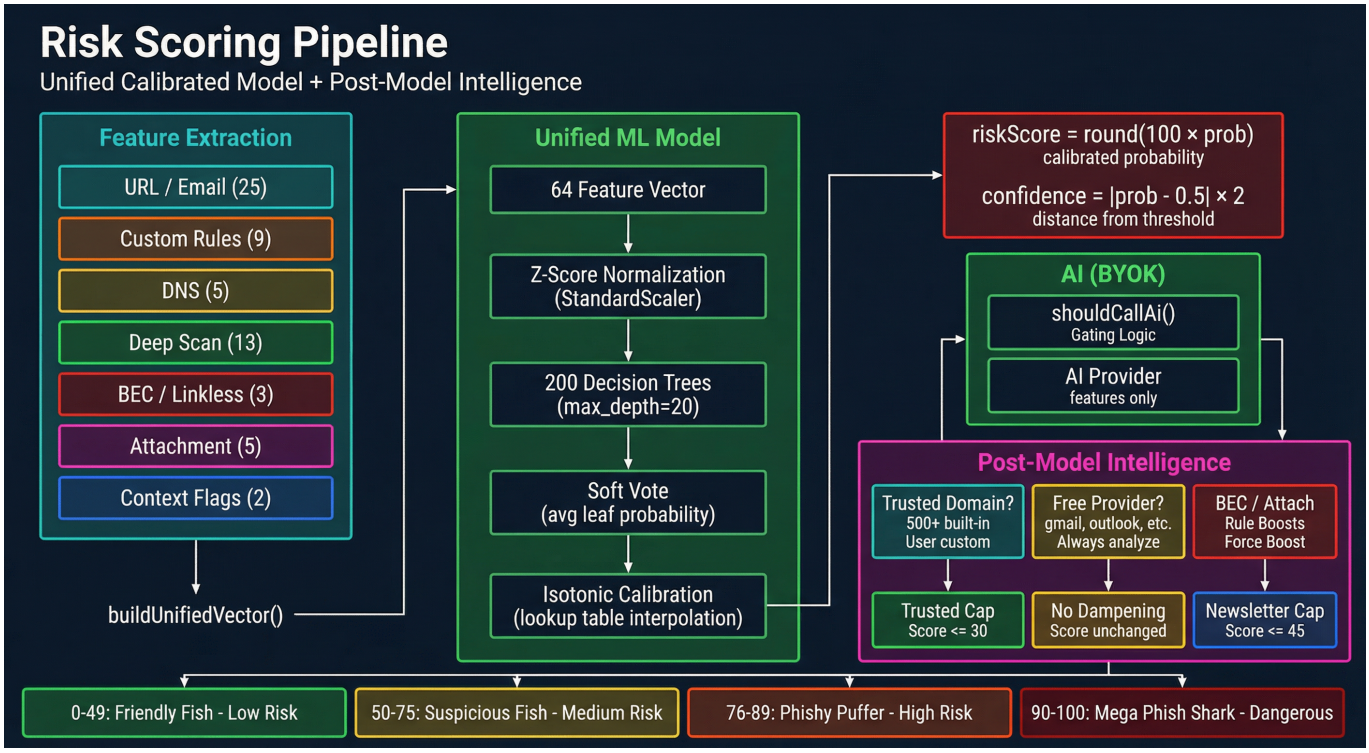
After the ML model produces its calibrated score, a post-model intelligence layer applies targeted adjustments. BEC rule boosts raise the floor to 70-80 for emails exhibiting strong financial request, authority impersonation, or phone callback signals. Trusted domain dampening caps the score at 30 for emails from the 500+ built-in trusted corporate domains (or user-added custom domains) when no BEC or attachment signals are present. Newsletter detection caps scores at 45 for emails exhibiting unsubscribe links, "view in browser" patterns, and newsletter footer text from non-trusted domains.

Critically, free/public email providers (gmail.com, outlook.com, icloud.com, yahoo.com, protonmail.com, hotmail.com, etc.) are explicitly excluded from trusted domain dampening. Anyone can register on these services, so a phishing email from scammer@gmail.com is scored on its own merits without any dampening. This distinction is maintained by a separate FREE_EMAIL_PROVIDERS set.

Users can manage trusted domains through the popup settings UI: adding custom trusted domains (which will receive dampening) or blocking built-in trusted domains (overriding the whitelist). Changes propagate instantly to the content script via chrome.storage change listeners. The priority order is: user blocked > user trusted > built-in list.

The isotonic calibration ensures that the output probability is well-calibrated: a score of 70 means approximately 70%% likelihood of phishing according to the model. A confidence metric is derived as |probability - 0.5| x 2, ranging from 0 (maximum uncertainty) to 1 (maximum confidence). The resulting score maps to four risk levels displayed as themed fish: Friendly Fish (0-49, Low), Suspicious Fish (50-75, Medium), Phishy Pufferfish (76-89, High), and Mega Phish Shark (90-100, Dangerous).

Figure 2 - Risk Scoring Pipeline: Unified calibrated pipeline from 64 features through buildUnifiedVector, Random Forest, isotonic calibration, t



5. AI Enhancement (Cloud BYOK)

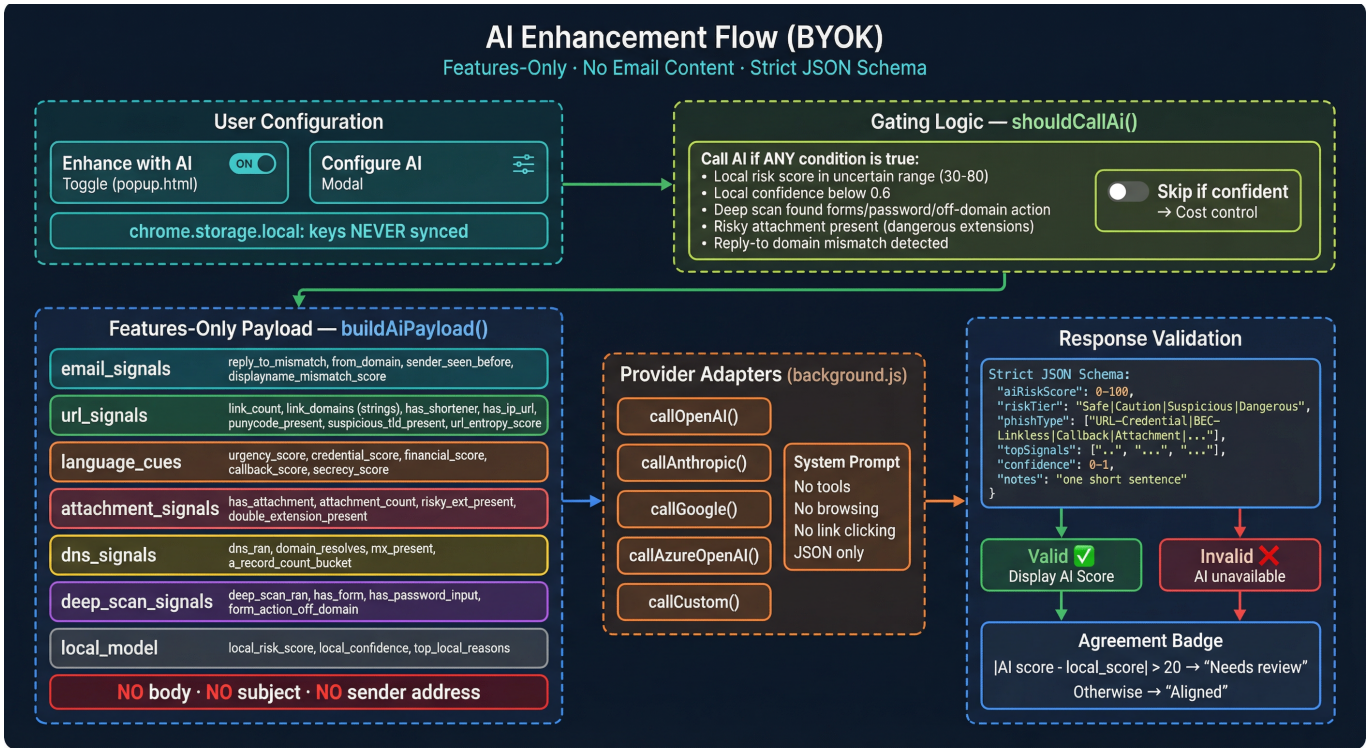
GoPhishFree includes an optional cloud AI enhancement that provides a second opinion on email risk. Users bring their own API key (BYOK) for one of five supported providers: OpenAI, Anthropic, Google Gemini, Azure OpenAI, or any custom OpenAI-compatible endpoint. Keys are stored exclusively in chrome.storage.local and are never synced.

When the "Enhance with AI" toggle is enabled, AI analysis runs automatically after each local scan, subject to gating logic. The shouldCallAi() function checks whether AI is useful: local risk score in the uncertain range (30-80), local confidence below 0.6, deep scan found suspicious forms/password inputs/off-domain actions, risky attachment present, or reply-to domain mismatch detected. If none of these conditions apply, AI is skipped with a "high confidence" message.

The AI payload is constructed by buildAiPayload() and contains only extracted signal features: email identity signals, URL/link signals, language cues (as scores, not raw text), attachment metadata, DNS signals, deep scan signals, and the local model's risk score and confidence. NO email body, subject line, or sender address is ever transmitted.

The system prompt enforces strict rules: no tools, no browsing, no link visiting, analyze only the provided JSON signals, and output must be strict JSON matching the required schema (aiRiskScore, riskTier, phishType, topSignals, confidence, notes). Invalid responses are rejected and the UI shows "AI unavailable". An agreement badge indicates whether the AI and local scores are "Aligned" (within 20 points) or "Needs review" (differ by more than 20).

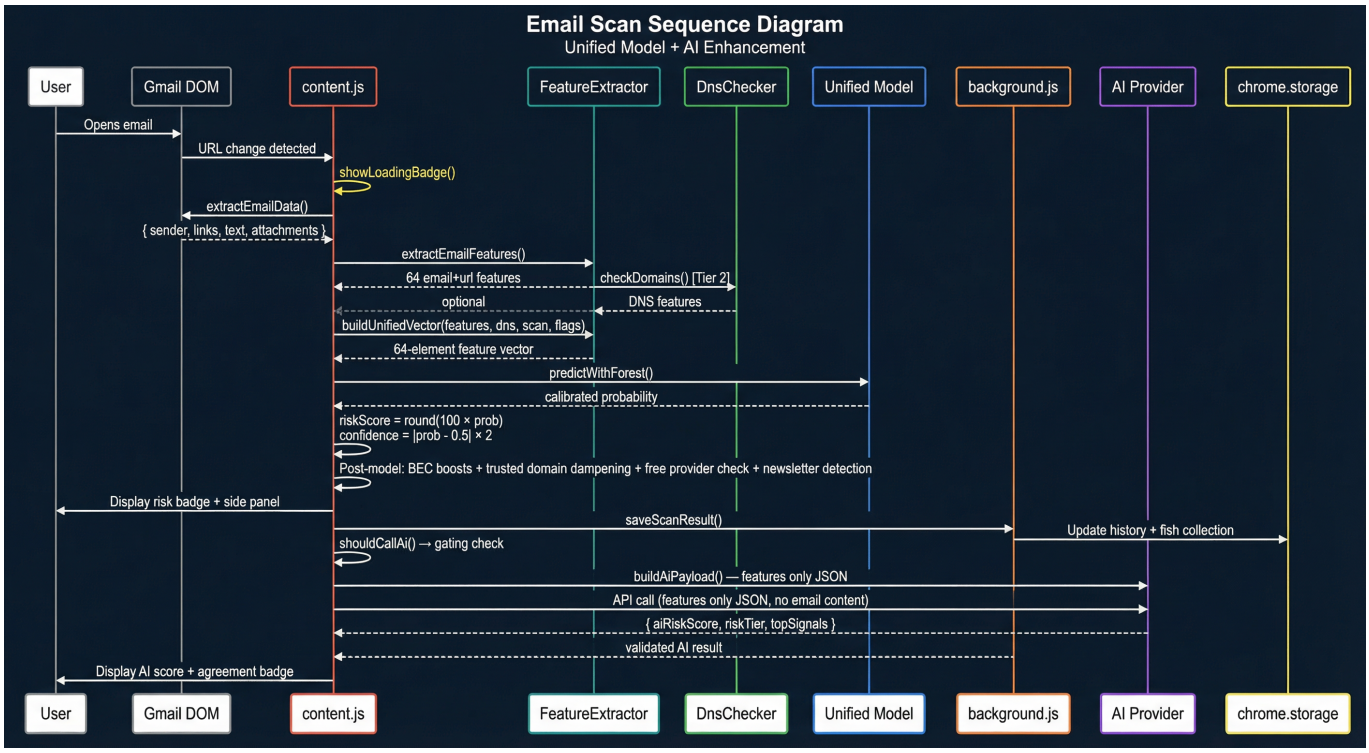
Figure 5 - AI Enhancement Flow: BYOK provider routing, features-only payload, and strict JSON schema validation.



6. Runtime Scan Flow

The diagram below illustrates the full sequence from the moment a user opens an email to the final risk badge display and optional AI enhancement. A MutationObserver in the content script detects Gmail navigation events, triggers feature extraction across all available tiers, assembles the 64-element unified vector, runs calibrated Random Forest inference, derives informational reasons (without modifying the score), renders the badge and side panel, persists the result via the background service worker, and optionally triggers AI analysis if the gating conditions are met.

Figure 4 - Email Scan Sequence: User opens email -> DOM detection -> 64-feature extraction -> calibrated inference -> risk badge -> AI enha



7. Security & Privacy

GoPhishFree enforces strict privacy guarantees. All ML inference and feature extraction execute entirely within the browser - no email content, URLs, or metadata are transmitted to any external server during core scanning. DNS-over-HTTPS queries send only domain names (never email bodies or user data) to Cloudflare or Google public resolvers. Deep Scan fetches omit all credentials and cookies, cap response sizes at 2 MB, validate content types, enforce an 8-second timeout, and parse HTML via DOMParser without executing any scripts.

The optional AI Enhancement sends only extracted signal features to the configured AI provider - never the email body, subject line, sender address, or any raw text content. API keys are stored exclusively in chrome.storage.local (never synced to Chrome Sync or any external service). The AI system prompt enforces strict rules preventing tool use, browsing, or link visiting, and requires responses in a strict JSON schema. Invalid responses are silently rejected.

The extension requires no backend server and stores all scan data locally via the Chrome Storage API. The user maintains full control over whether AI enhancement is enabled and which provider receives the features-only payload.

The trusted domain whitelist (500+ domains) distinguishes between corporate/organizational domains that only employees can send from (e.g., google.com, microsoft.com) and free/public email providers where anyone can register (e.g., gmail.com, outlook.com, icloud.com). Free email providers never receive trusted dampening, preventing attackers from exploiting the whitelist by sending phishing from commonly trusted mail services. Users can further customize the trust list by adding or blocking domains through the popup settings.