ChatGPT

RuleIQ Feature-Addition Plan (Release Candidate Scope)

Planning Overview

This plan targets a **UK-first MVP release** by Dec 1, focusing on closing all mandatory compliance gaps and weaving AI assistance into core workflows. The strategy is to **deliver must-have features** (ISO 27001/SOC2 controls, evidence automation, UK GDPR/PECR modules) with an AI-first approach that reduces manual effort. High-impact AI integrations (policy generation, evidence classification, risk analysis) are prioritized to cut audit prep time ~50% and give RuleIQ a competitive edge in the UK market. The intent is to reach a **release-ready state** that meets baseline certifications and delights early customers with intelligent automation.

Feature Backlog Table

Feature	Category	User Benefit	AI Component	Dependencies	Effort	Priority
Framework Library (ISO27001, SOC2, GDPR, CyberEssentials)	Compliance Content	Pre-loaded UK-specific controls & mappings for fast onboarding	No – curated standards content	Compliance SME input for control data	M	P0
Policy Templates & Editor	Policy Management	Jump-start via editable templates; version tracking of policies	**Yes** – GPT suggests phrasing & ensures coverage of requirements	Text editor component; template library	S	P0
Evidence Storage & Mapping	Evidence Core	Central repository to attach proof to controls (audit trail)	**Yes** – LLM auto-tags evidence to relevant controls, stores metadata [1]	File storage (S3), DB schema ready	S	P0
Automated Evidence Integrations (GitHub, AWS Cloud, G-Workspace)	Integration	Auto-collect evidence (code changes, cloud configs, logs) saving hours	**Yes** – ML parses raw logs and flags anomalies; LLM summarizes events [2]	OAuth/App creds; AWS IAM role; integration SDKs	L	P0

Feature	Category	User Benefit	AI Component	Dependencies	Effort	Priority
Task Workflow Engine (assignments, due dates)	Workflow	Tracks remediation tasks & deadlines for compliance gaps	**Yes** – AI prioritizes tasks based on risk (SmartCollector) and suggests fixes	Notification service; user roles	M	P0
Unified Audit Trail Logging	Platform Security	Immutable record of all actions for auditors (exportable)	**Yes** – Anomaly detection in logs (alert on unusual access patterns)	DB audit log table (extend EvidenceAuditLog) [3]	S	P0
RBAC & Auditor Role (read-only mode)	Platform Security	Principle of least privilege; external auditors can view only	No – rule-based access control (no AI needed)	User roles schema; permission checks	S	P0
Risk Register & Scoring	Risk Management	Capture risks with impact/ probability to prioritize mitigation	**Yes** – GPT evaluates described risks for severity & suggests mitigations	Risk scoring model; link to controls	M	P0
GDPR Workflows (DPIA wizard, Breach log)	Privacy Compliance	Simplifies mandatory GDPR processes (impact assessments, incident logging)	**Yes** – LLM guides DPIA questionnaire and auto-drafts report; NLP categorizes incidents by severity	PDF generator; incident DB model	M	P0
Consent & Cookie Manager (PECR)	Privacy Compliance	Lawful cookie consent capture for UK users (avoid fines)	**Yes** – AI optimizes banner text for clarity and higher opt-in rates	UI banner component; localStorage logic	M	P0
Compliance Dashboard & Report Export	Reporting UX	One-click view of status; export evidence & controls for auditors	**Yes** – GPT auto-generates an auditor-ready summary narrative of compliance status	Charting lib; doc export (CSV/PDF)	S	P0

Feature	Category	User Benefit	AI Component	Dependencies	Effort	Priority
UK Gov Filing Reminders (Companies House, HMRC MTD)	UK Differentiator	Prevent missed filings (annual returns, VAT) via automatic reminders	No – date retrieval is rule-based (no AI)	Companies House API; HMRC API or schedule config	S	P0
Baseline Security & Data Residency (MFA, KMS Encryption, eu-west-2)	Infra/ Security	Trust and legal compliance (UK data stays in UK, strong security controls)	No – configuration best-practices (no AI)	AWS KMS setup; user MFA integration	S	P0
Public API for Evidence (push/ pull)	Extensibility	Integrations with dev workflows (CI/CD, SIEM) via API	No – explicit API calls (no AI)	API auth, rate limiting, documentation	M	P1
AI Control Mapping Assistant	AI Differentiator	Recommends which controls a new evidence item satisfies (speeds mapping)	**Yes** – LLM suggests control IDs given evidence text (RAG)	Vector index of control texts; OpenAI Function for mapping	M	P1

(Effort S ≤4w, M=5–8w, L ≥9w; Priority P0 Must-have for GA, P1 next-up, P2 future)

AI Integration Blueprint

Policy Generation Assistant: Uses a large language model (Google PaLM 2 or Gemini) to help admins **draft and refine policy documents**. When a user creates a new policy from a template, the assistant can be invoked to fill in specifics (e.g. company name, roles) and ensure coverage of required clauses. **Model & Provider:** Google's latest text-generation model (prefer UK/EU region) for strong compliance language capabilities. **Data Sources:** The prompt includes the selected template text and company context (industry, size) from the profile. **Retrieval/Orchestration:** A retrieval-augmented approach will pull in relevant control statements or legal references from an internal library (e.g. ISO clauses) via a vector store lookup before generation. We'll use a **function-calling API** (tools) for factual inserts (e.g. company policies links) [4]. **Privacy:** No personal data is sent; template content is generic and any sensitive identifiers are masked or passed via an EU-compliant endpoint. **Latency/Cost:** Expect ~2–3 seconds per policy section with PaLM2 (cost ~$0.002 per section). We target sub-5s end-to-end for a full policy AI suggestion. **Fallback:** If the model fails or is unavailable, the system falls back to the base template text and highlights sections for manual input (ensuring the user isn't blocked).

Evidence Auto-Classifier: An AI service that **classifies and tags incoming evidence** (from integrations or uploads) to relevant compliance controls. It parses unstructured data (logs, configs) and suggests mappings to framework requirements (e.g. flags a CloudTrail log as evidence for ISO27001 A.12.4.1) [2]. **Model:** A lightweight GPT-4 model (or fine-tuned Llama) for classification tasks to keep latency low (<800ms per item). **Data Sources:** The evidence content (or its metadata) and a knowledge base of control definitions (embedded in a vector DB) are used. The model performs a similarity search to match evidence to likely controls and outputs a control tag and confidence. **Pattern:** Follows a **RAG** pattern – retrieve top candidate controls from the vector store, then have the model assign the best fit. **Privacy:** Any sensitive log data fields (usernames, IPs) are masked before sending to the model; alternatively, this can run on a local model if data residency is a concern. **Latency/Cost:** Under 1 second per evidence item using GPT-4-32k for classification (~$0.05/1k tokens). This is done asynchronously (e.g. in a worker) so as not to block UI. **Fallback:** If AI mapping fails or is uncertain, mark evidence as "ⓘ Needs review" and allow the compliance officer to map manually, with the AI's top suggestions shown as hints.

Compliance Insights Engine (Dashboard AI): A backend service generates **personalized compliance insights** for the admin's dashboard (e.g. tips, risk alerts, optimizations). It analyzes the org's data (controls not implemented, upcoming deadlines, repeated audit findings) and uses rules plus AI to output 2–3 key insights daily [5] [6]. **Model:** A smaller local model or prompt-engineered GPT-4 for summarization and recommendation generation. **Data Sources:** It ingests the compliance score trends, pending tasks, and recent audit log events from the DB. **Retrieval Pattern:** Uses a **rule-based trigger** (if GDPR priority high, suggest data mapping; if many tasks overdue, flag risk) combined with an LLM to **wordsmith the insight** for tone and clarity. No external knowledge needed, just organization data. **Privacy:** All processing happens server-side; no external API calls with proprietary data. **Latency/Cost:** Insights are generated offline (e.g. via a nightly job), so latency isn't user-facing; cost negligible as it's run infrequently and can batch multiple insights in one API call. **Fallback:** If the AI fails, default to pre-written tips (the component already has baseline tips/recommendations logic [7] that can show instead of an empty state).

Risk Analysis & Mitigation Advisor: When a new entry is added to the Risk Register, an AI agent **analyzes the description** and suggests a likelihood level, impact level, and potential mitigation steps. **Model:** GPT-4 (knowledgeable in security best practices) accessed via OpenAI API, because it can provide rich suggestions. **Data Sources:** The prompt includes the risk description and context (asset type, related control if any). Optionally, a library of past risks and mitigations is provided for retrieval to avoid repetition. **Pattern:** Direct LLM Q&A – ask the model to classify risk severity (perhaps using a custom function call for structured output). **Privacy:** No customer personal data is in these prompts – only internal risk info; still, to be safe, we might use Azure OpenAI with a UK data center. **Latency/Cost:** <2s per risk item; cost ~$0.03 per analysis. **Fallback:** If the AI doesn't return or is uncertain, mark the risk for manual review. The register UI will allow the user to select severity manually in any case, so AI is assistive, not mandatory.

DPIA & Incident Assistant: The GDPR module will incorporate AI to **guide users through complex forms**. For DPIAs, as the admin answers each section, an AI agent can provide examples or explain the relevance (acting like a smart helper). Once the DPIA is filled, the AI can generate a draft report summary highlighting high residual risks, which is then saved as part of the DPIA record. **Model:** GPT-4 (or specialized compliance model) due to the nuance of GDPR legal language. **Data Sources:** It uses static DPIA question templates and any existing data classification info from the system. For breach incidents, if the user inputs an incident description, an AI could auto-classify it (e.g. "likely notifiable to ICO" vs "minor incident") based on GDPR criteria. **Pattern:** This uses **tools/agents** – for example, a "GDPR Reg reader" tool that the LLM can invoke to quote Articles (to ensure accuracy in advice). **Privacy:** DPIA content can include sensitive processing info – we ensure either using an on-prem model

or that the data is sufficiently abstracted (e.g. using labels for systems instead of real personal data in prompts). **Latency/Cost:** Interactive Q&A in the wizard tolerates ~1–2s per response. Each full report generation ~5s (several hundred tokens). We budget ~$0.10 per complete DPIA in API costs. **Fallback:** If AI is unavailable, the DPIA wizard falls back to static help text and blank report template which the user can fill manually.

(All AI features will include a "verify" or "edit" step to keep a human in the loop – ensuring compliance officers can adjust AI outputs. AI suggestions will be logged along with final decisions for audit transparency.)

Technical Design Notes

- **Repo & Module Impacts:** We will leverage the existing modular structure – e.g. the `integrations` **module** for evidence collection and the `services/ai` **module** for AI logic. New code will be added in a forward-compatible way (no breaking changes to base).
- **Compliance Frameworks:** Load UK frameworks (ISO 27001, SOC2, GDPR, Cyber Essentials) into the `compliance_frameworks` table or config. Possibly extend the `ComplianceFramework` model (in `database/compliance_framework.py`) to include region or version info. No major code changes – mostly seed data and ensuring UI can render new frameworks.
- **Policy Editor:** Use the existing policy model and CRUD API. The `Policy` model exists [8], and the frontend likely has a policy editor component. We will integrate an **AI assist** button that calls a new endpoint (e.g. `/ai/policy_suggest`) feeding the current draft and receiving suggestions. A new backend function in `services/ai/assistant.py` will handle `policy_generation` content type [9] by invoking the ComplianceAssistant with the appropriate template.
- **Evidence Model & AI Metadata:** The `Evidence` ORM model already has an `ai_metadata` JSONB field [1] to store AI analysis (e.g. suggested control tags, quality scores). We will use this to record classifier outputs. For instance, when evidence is auto-collected, a post-processing step (in `workers/evidence_tasks.py` or `services/automation/evidence_processor.py`) will call the AI classifier and save results into `Evidence.ai_metadata`. The UI can then display a "AI suggested mapping: XYZ" for each evidence item.
- **Integrations (GitHub, AWS, G-Workspace):** The integration framework is partially built – there's a base Integration class and evidence collection workflow [10] [11]. Google Workspace integration is implemented as a reference (pulls login/admin logs) [12]. We need to **implement AWS** similarly: create an `AWSIntegration` subclass to fetch AWS Config or CloudTrail data via boto3. The provider enums in code already include `"aws"` [13]. We'll add an OAuth or IAM key storage in `Integration.encrypted_credentials`. For GitHub, likely use their API to fetch commit history (mapping to change management controls). If time is short, at minimum implement **GitHub commit fetch** (for evidence of code reviews) and one cloud evidence (AWS or GCP); the architecture supports adding others later.
- **Task & Workflow:** Introduce a `Task` model linking to controls or evidence (for remediation tasks). We can repurpose the existing `ImplementationPlan.phases` JSON [14] or create a simpler `tasks` table. The **PendingTasksWidget** in the frontend is already expecting `DashboardTask` objects [15] – we will populate these via an API endpoint (`/tasks`) that collates open tasks from various sources (failed controls, risk treatments, etc.). The **SmartEvidenceCollector** logic will be used to auto-generate some tasks (it defines prioritization and task data structures in code) [16] [17]. We'll integrate that by calling `SmartEvidenceCollector.create_collection_plan()` after initial framework assessment to get a list of tasks and then persist them.
- **Unified Audit Log:** We have an `EvidenceAuditLog` model for evidence/integration actions [3]. To meet the requirement "all CRUD actions logged," we will generalize usage of this or

create a parallel `AuditLog` for other modules (policies, tasks, user access). A likely approach: add a **logging decorator** to service methods and Flask/FastAPI endpoints to record any create/update/delete action. The log entries will include actor, timestamp, resource, and be stored in this audit table (expanding resource_type beyond evidence to include "policy", "user", etc.). Ensure **immutability** by not exposing any delete or edit on log records (could even write to an append-only store or use DB insert-only rules).

- **RBAC & Roles:** The codebase currently lacks an explicit roles field on `User` (none in `User` model [18]). We will add a `role` column (enum: Admin, Contributor, Auditor) or a separate table for user roles if many-to-many. Enforcement: define a simple decorator or middleware to check `user.role` for each endpoint (e.g. auditor cannot call POST/PUT, only GET on certain resources). Implementing **permission export** means providing an endpoint to list all users and roles, and possibly their last login (for access review – could reuse audit log data).
- **Risk Register:** Define a new model `Risk` with fields (description, impact, likelihood, owner, mitigation, etc.). Add a frontend form to create risks and list to view them. The risk scoring formula (impact × likelihood) can be computed, and a color-coded heat level stored. Integrate AI by calling the advisor on creation (non-blocking) to fill `impact` / `likelihood` if user left them blank, or to suggest a mitigation (store in a suggestions field or directly populate a draft).
- **DPIA & Breach Modules:** Create models `DPIA` (with fields: context, results, PDF path, etc.) and `Incident` for breaches (fields: date, summary, severity, reportable Y/N). The DPIA wizard can be a multi-step form in Next.js, and on final step, call an API to generate the PDF. We'll use a library like ReportLab or markdown->PDF to format the DPIA report, including any AI-generated text. The breach log is simpler: just a CRUD list of incidents. Possibly incorporate a **breach assessment function** – e.g. when an incident is saved, run a check (if personal data affected and risk high, mark "consider reporting to ICO"). This logic can be rules-based initially, with AI used to assist description analysis if possible.
- **Consent Manager:** This will mostly be front-end. We'll add a React context or component at the app root to show a cookie banner for new visitors. User choices (which categories allowed) get stored in a new `ConsentLog` model or even just in browser for MVP. But requirement is to record opt-in categories with timestamp – so yes, a backend hit to `/consent` logging user (if known) or an anonymous ID, categories accepted, time. Ensure this log is exportable in case of audit. We might integrate a **cookie scanner** later; for now, we assume admin manually configures what categories are present. AI involvement: optionally use GPT to generate a privacy/cookie policy page content based on the categories selected (could reuse Policy AI assistant).
- **Compliance Dashboard & Reports:** The dashboard UI already shows compliance score, tasks, insights, etc. We will ensure all new data (e.g. newly added frameworks, tasks, risks) feed into it. The **exportable report** will be implemented as an **"Export Audit Report"** button, which compiles key info: list of controls with status, evidence attached, open risks, etc., into a PDF/Docx. We will include an AI-generated executive summary at top (via the insights engine or a dedicated prompt). For implementation, generate a markdown or HTML report server-side and run it through a PDF converter. Also allow CSV export of the raw evidence list and audit log (the audit log export was a Non-functional req: generate CSV under 30s).
- **UK Integrations (Companies House, HMRC):** Implement a scheduled job (Celery beat or cron) that **calls Companies House API** nightly to get upcoming filing dates for the company (requires storing company number in the BusinessProfile). Companies House API provides next confirmation statement and accounts due dates; we'll parse and if a date is within say 30 days, create a Task or send an alert email. For HMRC MTD (VAT returns), direct API access requires OAuth with HMRC – likely too heavy for MVP. Instead, we allow the admin to input their VAT quarter schedule in settings, and then the system can generate recurring tasks for those deadlines. No AI here – straightforward date calculations and reminders.

- **Security & Infrastructure:** Enable **AWS KMS encryption** for RDS and S3 buckets (Terraform or AWS Console setting). This is largely ops configuration; from code perspective, use ORM-level encryption for sensitive fields if not already (the `Integration.encrypted_credentials` is already handled via encryption service [19]). Enforce MFA in the app by integrating with our auth (if using Cognito or custom, require TOTP setup for Admin accounts at least). For **UK data residency**, ensure all cloud resources (DB, storage) are in `eu-west-2` region and update any hard-coded region endpoints in config. The CI/CD and infrastructure-as-code templates will be updated accordingly. We will run **OWASP Zap** or similar on the staging app to catch top-10 issues (and have already addressed obvious ones like parameterized queries, etc.). A `scripts/security_audit.py` exists which might be a placeholder for checks – we'll extend that or incorporate into testing pipeline.
- **Public API:** Likely REST endpoints (or GraphQL if decided) to push evidence. We will expose secure endpoints like `POST /api/v1/evidence` (authenticated with API key or OAuth token) so external tools can send evidence artifacts (for example, a CI pipeline can push test results as evidence for control "Vulnerability Scans"). Also `GET /api/v1/evidence?control=XYZ` for pulling evidence status. We'll generate API docs (OpenAPI spec). This will require adding token auth and rate limiting middleware. Not critical for GA, so it's flagged as P1 – can launch as Beta after core features stable.
- **Code References:** Throughout development, we will update or utilize existing code where available. For instance, the AI orchestration engine (`ComplianceAssistant`) is already structured to handle various **task types with model selection and tool use** [20] – we will add specific tools (like a KnowledgeBase tool for regs) and ensure the circuit breaker & safety nets are tuned for compliance use-cases (less stringent on certain content like regulatory terms [21]). All new logic will have unit tests (many `tests/` exist for AI and API) to maintain quality.

Incremental Milestones & Timeline

We will execute in **2-week sprints** (except initial design phase), aligning with the needed GA date:

- **Phase 0 – Discovery & Design (Aug 4 – Aug 29, 2025)**: *Milestone:* Product requirements clarified, architecture finalized. **Exit Criteria:** User flows wireframed; data model updated for new entities (Risk, DPIA, Incident, Consent); security architecture reviewed (GDPR & ISO compliance); backlog refined with estimates. *Target Date:* Aug 29, 2025.
- **Sprint 1 – Frameworks & Policy Core (Aug 30 – Sep 12)**: **Exit Criteria:** Compliance frameworks loaded (≥4); Policy template library (20 UK-specific templates) visible in UI; Policy editing and versioning (POL-1) working in app; basic permission model in place (only Admin can edit policies). AI integration: Policy Generation Assistant endpoint delivering draft text suggestions. *Target Date:* Sep 12, 2025.
- **Sprint 2 – Evidence MVP & GitHub Integration (Sep 13 – Sep 26)**: **Exit Criteria:** Evidence model fully functional with upload UI; GitHub integration (EVI-1) fetches commit history nightly and creates evidence items mapped to change-management control (e.g., SOC2 CC8.1) [22]; Evidence list view shows status and AI-proposed control tags; audit logging captures evidence add/remove (AUD-1). AI integration: Evidence auto-classifier tagging at least 50% of new evidence with suggestions (stored in `ai_metadata`). *Target Date:* Sep 26, 2025.
- **Sprint 3 – Workflows, Tasks & Basic Insights (Sep 27 – Oct 10)**: **Exit Criteria:** Task assignment UI live (TAS-1) – Admin can create tasks linked to controls or evidence with due dates and assign to Contributors; Contributors see their tasks in dashboard widget with status update capability. Basic workflow automation: failed controls automatically generate "remediation" tasks. AI integration: **Smart prioritization** – system generates an initial Implementation Plan (list of tasks with priorities) for a selected framework using the SmartEvidenceCollector logic. Dashboard

shows count of open tasks and basic AI Insights (at least 2 insight types, e.g. “Policy X is outdated” tip) [6] . *Target Date:* Oct 10, 2025.

- **Sprint 4 – UK Compliance Features (Oct 11 – Oct 24)**: **Exit Criteria:** GDPR module (DPIA wizard and Breach log) implemented – Admin can complete a DPIA form and download a report (GDPR-1), and log incidents with time stamp; Consent banner (PECR-1) live on app with choices stored in backend; Companies House integration (COH-1) – system pulls next filing dates and shows a reminder task X days before due. **UK data residency** verified (all data stores in London region). AI integration: DPIA assistant provides help text or auto-filled examples in the form; Risk Register available with AI risk scoring suggestions. *Target Date:* Oct 24, 2025.
- **Sprint 5 – Security Hardening & Reporting (Oct 25 – Nov 7)**: **Exit Criteria:** Full RBAC in effect – Auditor role implemented (RBAC-1) with read-only access verified (e.g., test that an Auditor user cannot upload evidence [3]); End-to-end encryption enabled (DB columns encrypted, TLS enforced); MFA optional for all users and required for Admins; Comprehensive audit log export working (able to generate CSV of last 30 days). Compliance Dashboard finalized with framework scores and trend charts; “Export Audit Report” button generates a consolidated report PDF. AI integration: final tuning of AI components – e.g. verify auditor report summary generation is accurate and safe. All P0 features complete and passing tests. *Target Date:* Nov 7, 2025.
- **Sprint 6 – Penetration Test & Freeze (Nov 10 – Nov 28)**: **Exit Criteria:** External CREST-certified pentest completed; all critical/high findings resolved or have compensating controls. Performance testing done – system handles 200 req/s with <500ms p95 latency on key APIs (evidence fetch, task update). Compliance review – internal audit using RuleIQ on RuleIQ (meta!) to ensure we meet ISO/GDPR requirements for our own product (e.g. run a DPIA, security policies in place). *Target Date:* Nov 28, 2025.
- **GA Release – “UK Launch” (Dec 1, 2025)**: System deployed to production (AWS London). **Exit Criteria:** UAT sign-off from 2 pilot customers in UK (feedback incorporated), documentation published (user guide and API docs), support & on-call processes ready. Go/No-Go meeting held and go-ahead given. *Launch Date:* Dec 1, 2025.
- **Phase 2 – Extended AI & Integrations (Dec 2, 2025 – Feb 20, 2026)**: *Scope:* P1 backlog features and enhancements deferred from MVP. Sprints 7–12 will cover things like Azure/Atlassian connectors, AI-driven control mapping (if not in MVP), advanced analytics, and additional frameworks (ISO 27701, etc.). *Target:* complete by Feb 20, 2026 for a broader GA or early Q2’26 updates.

(Timeline buffer: We have built-in 1–2 weeks of float before Dec 1, used for bugfix or slippage recovery. The team will triage any P0 feature delays by reducing scope or fast-follow in a patch release if absolutely necessary.)*

Resource & Skill Matrix

Milestone / Sprint	Backend Eng	Frontend Eng	ML Engineer	DevOps	Compliance SME	QA Engineer
Phase 0: Discovery/ Design	1 (Architect)	0 (UX only)	0	1	1	0
Sprint 1: Framework & Policy	1	1	0	0.5	0.5 (template review)	1

Milestone / Sprint	Backend Eng	Frontend Eng	ML Engineer	DevOps	Compliance SME	QA Engineer
Sprint 2: Evidence & GitHub	1	1	1 (AI classifier)	0.5	0	1
Sprint 3: Tasks & Insights	1	1	0.5 (SmartColl. tuning)	0	0	1
Sprint 4: UK GDPR/PECR	1	1	0.5 (DPIA assist)	0.5	1 (legal review)	1
Sprint 5: Security & Reporting	1	1	0.5 (Report summary AI)	1	0.5 (compliance check)	1
Sprint 6: Pen-test & Hardening	1	0	0	1	0	1 (security test)
Total Team (FTE)	**~3 BE**	**~2 FE**	**1 ML**	**1 DevOps**	**1 Compliance**	**2 QA**

- *Backend Engineers:* Responsible for API development, database schema changes, integration implementations, and AI service integration on server-side.
- *Frontend Engineers:* Build UI components (policy editor, tasks dashboard, cookie banner, etc.) and integrate front-end with back-end APIs/AI suggestions.
- *ML Engineer:* Focused on AI features – fine-tuning prompts, evaluating model outputs, optimizing calls, integrating any on-prem models or vector DB. May also handle data engineering for AI (embedding frameworks, etc.).
- *DevOps:* Sets up infrastructure (AWS environment, CI/CD), configures monitoring, ensures security configurations (KMS, VPC) are in place. Will also manage the production deployment and any scaling concerns from load test.
- *Compliance SME:* Part-time involvement to validate content (frameworks completeness, correct mapping of controls), review AI outputs for accuracy (especially GDPR guidance, policy templates), and ensure the product meets regulatory interpretations.
- *QA Engineer:* Creates test plans for each feature, does exploratory testing especially around workflow and security edge cases, and verifies compliance features (e.g. does consent log record correctly, can an auditor truly only read). Also will run automated test suites and coordinate the pen-test fix verification.

(Team members may wear multiple hats; e.g. a backend engineer with ML experience might double as ML Eng for some sprints. Compliance SME involvement is heavier in design and final validation phases.)

Risk Register & Mitigations

Risk	**Impact** (1-5)	**Likelihood** (1-5)	**Mitigation Strategy**	**Owner**
AI errors or hallucinations in compliance advice – LLM might produce incorrect control mappings or policy text that could mislead users.	5 – Could lead to compliance gaps or legal mistakes if unchecked.	2 – Medium-Low (with human review in place).	*Mitigation:* Keep human approval step for all AI outputs. Use retrieval to ground answers in official text [23]. Extensive testing with compliance team on AI suggestions; tune prompts and use guardrail functions to validate outputs (e.g. ensure control IDs returned actually exist).	ML Engineer (QA cross-check by Compliance SME)
Integration delays (GitHub/AWS not ready) – Automated evidence connectors take longer than expected, leaving evidence collection semi-manual.	4 – Lacking automation breaks the promise of 70% auto-evidence, could slip launch if core to value prop.	3 – Medium. GitHub likely straightforward (APIs exist), AWS more complex (permissions).	*Mitigation:* Stub out integrations with minimal viable data (e.g. fetch a simple artifact) to have something working, then iterate. If AWS full integration not ready, document manual steps or use a third-party SDK temporarily. Parallelize connector work between engineers. Leverage existing libraries where possible.	Backend Lead

Risk	**Impact** (1-5)	**Likelihood** (1-5)	**Mitigation Strategy**	**Owner**
PECR Consent misimplementation – e.g. banner not truly blocking non-essential cookies or incorrect record of consent, leading to regulatory exposure.	3 – Could cause user complaints or regulatory fine (moderate impact).	2 – Low. This is a known pattern to implement (many examples).	*Mitigation:* Follow ICO guidance on cookie consent wording. Have legal counsel or SME review the banner text and functionality. Use open-source consent manager libraries as reference for compliance. Do a pilot on our own site and get user feedback.	Frontend Lead (review by Compliance)
Scope creep for UK extras (Companies House, etc.) – These could consume too much dev time, risking core MVP stability.	4 – Core features might be undercooked if team diverts to peripheral ones.	4 – High. Stakeholders may keep adding "just one more" UK feature.	*Mitigation:* Time-box the development of each UK-specific feature (e.g. "reminders only, no full filing integration"). Clearly document out-of-scope items for Phase 2 (e.g., "no automated filing submissions in MVP"). Product manager to enforce MoSCoW priorities – P0 items only [24] .	Product Manager

Risk	**Impact** (1-5)	**Likelihood** (1-5)	**Mitigation Strategy**	**Owner**
Data residency misconfiguration – Data inadvertently stored outside UK (e.g. a backup in US region) violating customer expectations.	5 – High impact (could lose customer trust, violate promises).	2 – Low. We control our AWS setup, but human error is possible.	*Mitigation:* Infrastructure as Code (Terraform) with explicit region tags for all resources to avoid drift. Implement an automated check: e.g. AWS Config rule to flag any resource not in eu-west-2. Perform a compliance review before launch verifying S3, RDS, etc. are all UK. Document data flow in our GDPR DPIA.	DevOps Lead
Audit log performance hit – If every action is logged synchronously, could slow down high-traffic operations, or the log DB table grows huge (affecting queries).	3 – Medium impact on user experience (slower UI) and storage costs.	3 – Medium. Likely to manifest when many evidence items or tasks are created.	*Mitigation:* Make logging asynchronous (write to queue) so user actions aren't blocked. Archive old logs to a separate table or S3 after 90 days to keep the active table small (the model supports archiving). Add DB indexes on key fields (already planned [25]) to optimize queries. Test with 10k+ log entries for any slowdown.	Backend Lead

Risk	**Impact** (1-5)	**Likelihood** (1-5)	**Mitigation Strategy**	**Owner**
Security regression (new features introduce vulns) – New code (esp. AI and integrations) might open XSS, injection, or privilege escalation paths.	5 – High. A security breach would be catastrophic to a compliance product's credibility.	3 – Medium. Developers might overlook something under time pressure.	*Mitigation:* Integrate security reviews in each sprint (definition of done includes basic OWASP check). Use the **SecurityAudit script** and SAST tools in CI. Penetration test in Sprint 6 will catch issues – have resources allocated in Nov to fix any findings promptly. Also ensure all secrets (API keys, AI tokens) are stored safely (Vault or AWS Secrets Manager) – no hardcoding.	DevSecOps (with QA)

Acceptance Gates

Before declaring the release candidate ready for GA, the following **acceptance criteria** must be objectively met:

- **Functional Completion:** 100% of P0 backlog items are implemented and tested. All high-priority requirements from the PRD (must-haves in scope table) are demonstrably working in the staging environment. E.g. evidence from GitHub and one cloud provider is auto-pulling nightly, audit log records all critical events, cookie consent is captured, etc.
- **Quality Assurance:** All test suites pass (unit, integration tests). No *Critical* or *High* severity bugs remain open in the bug tracker. Medium/Low bugs are triaged with a plan; none of them block core user flows. QA sign-off obtained for each module (Policy, Evidence, Tasks, etc.).
- **Performance & Load:** The system meets the performance target of 95th percentile response time <500 ms for normal operations under 200 req/sec load. Specifically, evidence listing, task updates, and report exports should stay within this limit. Load testing results and monitoring dashboards are reviewed and approved by the engineering lead.
- **Security & Compliance:** Security testing is passed – the external pen-test reported no **Critical/ High** issues remaining (report attached), and medium issues have workarounds. The app adheres to OWASP Top 10 (verified via automated scans). Data encryption is verified (DB and S3 evidence files encrypted at rest with AES-256). MFA enforced for admin logins. A GDPR compliance check is done: e.g., we have a privacy notice, consent mechanism, and the team completed a DPIA for our platform (to ensure we comply with UK GDPR for our own processing).

- **User Acceptance (Beta Trials):** At least two pilot customers (UK SMBs) have used the system in UAT and provided feedback. Their core use cases (evidence collection, generating a report for an audit) were successfully completed. CSAT from these pilot users is ≥4/5 for key admin tasks. Any critical feedback from them has been addressed in a patch.
- **Documentation & Support:** All user-facing documentation is ready – including an Admin guide (with steps to set up integrations, run reports), and an Auditor read-only guide. The AI features have tooltip explanations so users trust the suggestions ("Generated by AI – review before use" etc.). Internal runbooks are prepared for on-call (covering backup restore, incident response). The support team (or persons covering support) have been trained on the new features.
- **Regulatory Approval (if needed):** If any feature required external sign-off (e.g. legal approval of cookie banner text, or compliance advisor sign-off on content accuracy), those approvals are obtained in writing. The **Compliance SME signs off** that the product covers ISO 27001 Annex A controls and core UK GDPR obligations per PRD mapping (we can demonstrate mapping of each requirement to a feature in the app).
- **Gate Review:** A final Go/No-Go meeting checklist is ticked off, and stakeholders agree the release candidate is ready. This includes checking that all the above gates are met and that there are no open "blocking" issues in Jira. Only after this review will we tag the release as **GA 1.0**.

Appendix

- **Code Refs – Key Implementations:** The plan aligns with existing code structure. For example, the `Evidence` model already includes an `ai_metadata` field to store AI analysis of evidence [1]. The integration system supports multiple providers (e.g. `'aws', 'okta', 'google_workspace'`) as seen in the config [13], which we will utilize to add AWS integration. The Google Workspace integration example in code shows how logs are mapped to compliance controls (e.g. mapping admin activity to ISO 27001 A.12.4.1) [2] – we will follow this pattern for other integrations. The `ComplianceAssistant` class in `services/ai/assistant.py` is designed to orchestrate model calls with context and tools; it even defines content categories like evidence classification and policy generation [4] – a foundation we will build upon for our AI features.

- **Competitor Insights:** OneTrust's recent platform emphasizes automation and AI, offering **50+ frameworks out-of-the-box** and claiming to cut compliance effort by *"up to 60%"* [26]. This underlines the need for our broad framework coverage and automated evidence collection. Also, OneTrust uses *"AI-powered questionnaire response capabilities"* [27] – our inclusion of an AI policy assistant and risk advisor will be on par or better, but tailored for UK needs. Vanta and Drata leverage integrations to monitor systems continuously; we match that with GitHub/AWS connectors and go a step further by using AI to interpret evidence (not just collect it). By learning from these, our plan doubles down on automation and intelligent insights as key differentiators.

- **Regulatory Citations:** Our feature mapping ensures coverage of UK regulations: e.g. GDPR Article 32 on *security of processing* is addressed via both technical measures and evidence (see Google log integration mapping to "GDPR: Article 32" control) [2]. PECR compliance is met with an explicit consent module (ICO guidance 2023). Cyber Essentials requirements (like access control, patch management) are covered through corresponding ISO controls and evidence tasks. We will maintain a **traceability matrix** linking each implemented feature to the regulatory requirement or standard control it satisfies – this will be reviewed by the compliance SME as part of acceptance.

---

1 8 models.py
https://github.com/OmarA1-Bakri/ruleIQ/blob/16b5f72582bfedfb938258e5e465377d1d02cfdb/database/models.py

2 12 22 google_workspace_integration.py
https://github.com/OmarA1-Bakri/ruleIQ/blob/16b5f72582bfedfb938258e5e465377d1d02cfdb/api/integrations/google_workspace_integration.py

3 10 11 25 integrations.py
https://github.com/OmarA1-Bakri/ruleIQ/blob/16b5f72582bfedfb938258e5e465377d1d02cfdb/database/models/integrations.py

4 9 20 21 assistant.py
https://github.com/OmarA1-Bakri/ruleIQ/blob/16b5f72582bfedfb938258e5e465377d1d02cfdb/services/ai/assistant.py

5 6 7 ai-insights-widget.tsx
https://github.com/OmarA1-Bakri/ruleIQ/blob/16b5f72582bfedfb938258e5e465377d1d02cfdb/frontend/components/dashboard/ai-insights-widget.tsx

13 19 integration_service.py
https://github.com/OmarA1-Bakri/ruleIQ/blob/16b5f72582bfedfb938258e5e465377d1d02cfdb/database/services/integration_service.py

14 implementation_plan.py
https://github.com/OmarA1-Bakri/ruleIQ/blob/16b5f72582bfedfb938258e5e465377d1d02cfdb/database/implementation_plan.py

15 dashboard.ts
https://github.com/OmarA1-Bakri/ruleIQ/blob/16b5f72582bfedfb938258e5e465377d1d02cfdb/frontend/types/dashboard.ts

16 17 smart_evidence_collector.py
https://github.com/OmarA1-Bakri/ruleIQ/blob/16b5f72582bfedfb938258e5e465377d1d02cfdb/services/ai/smart_evidence_collector.py

18 user.py
https://github.com/OmarA1-Bakri/ruleIQ/blob/16b5f72582bfedfb938258e5e465377d1d02cfdb/database/user.py

23 24 26 27 OneTrust Launches AI-Powered Compliance Automation Platform
https://www.channelinsider.com/news-and-trends/us/onetrust-compliance-automation/