**Product Requirements Document (PRD) – Regulatorischer KI-Monitor**

* **Dokumentenversion:** 1.0
* **Datum:** 24. Juli 2025
* **Projektname:** Regulatorischer KI-Monitor (RegMon)

**1. Einleitung**

Der Regulatorische KI-Monitor (RegMon) ist ein automatisiertes Tool, welches mithilfe neuester LLM-Technologien relevante regulatorische Nachrichten identifiziert, klassifiziert, zusammenfasst und verteilt. Ziel ist es, Fachexperten und Entscheidungsträger zeitnah mit kuratierten und relevanten Informationen zu versorgen, um proaktives Risikomanagement und Compliance zu ermöglichen.

**Technologische Basis:**

* Der Regulatorische News-Monitor (RegMon) wird in der Programmiersprache R entwickelt. R bietet umfangreiche Bibliotheken für Datenverarbeitung, statistische Analysen und maschinelles Lernen, die es ermöglichen, regulatorische Nachrichten effizient abzurufen, zu verarbeiten und zu analysieren. Durch den Einsatz von R können zudem bestehende Datenanalyse- und Reporting-Tools nahtlos integriert werden, um die Funktionalität des Systems zu erweitern.
* OpenRouter.ai: Eine Plattform, die den Zugriff auf verschiedene Large Language Models (LLMs) ermöglicht, darunter das Modell von Mistral, welches zur Bewertung der Relevanz von Nachrichten verwendet wird.
* LLM: Mistral-Small-3.2-24B-Instruct-2506 – Leistungsstark und aktuell in kleinem Rahmen gratis nutzbar.
* Brevo API: Eine E-Mail-API, die für den stabilen und verlässlichen Versand der E-Mail-Zusammenfassungen genutzt wird. In kleinem Rahmen ebenso gratis nutzbar.
* Github Actions: GitHub wird zur Speicherung des Codes verwendet, ermöglicht jedoch primär die automatisierte Ausführung des Skripts in der Cloud nach einem festgelegten Zeitplan. Dies gewährleistet eine zuverlässige und kontinuierliche Verarbeitung relevanter regulatorischer Nachrichten ohne manuelle Eingriffe.

**2. Vision und Zweck**

**Vision:** Compliance und Risikomanagement-Abteilungen eine präzise, automatisierte, personalisierte und zeitkritische Quelle für relevante regulatorische Nachrichten zu bieten, die es ihnen ermöglicht, auf sich ständig ändernde Compliance- und Risikolandschaften schnell und fundiert zu reagieren.

**Zweck:**

* **Automatisierung:** Reduzierung des manuellen Aufwands für die Recherche und das Filtern relevanter regulatorischer News.
* **Relevanz:** KI-unterstützte Filterung von relevanten regulatorischen Aktivitäten und News für den persönlichen Arbeits- und Themenschwerpunkt
* **Effizienz:** Bereitstellung von prägnanten Zusammenfassungen und detaillierten Einordnungen der relevanten Nachrichten.
* **Informationsfluss:** Regelmäßiger, automatischer Versand der kuratierten Informationen an definierte Empfänger.
* **Historisierung:** Aufbau einer Wissensbasis über relevante regulatorische Entwicklungen.

**3. Funktionen (Features) und Anforderungen**

**3.1. Datenbeschaffung (RSS-Feeds)**

* **F1.1:** Automatisiertes Abrufen von News aus vordefinierten RSS-Feeds von relevanten Quellen (z.B. BaFin, EZB, ESMA).
  + **Anforderung (REQ-1.1.1):** Unterstützung der RSS 2.0 Spezifikation (oder ähnlicher Standards, die von tidyRSS unterstützt werden).
  + **Anforderung (REQ-1.1.2):** Konfigurierbare Liste von RSS-Feed-URLs.
  + **Anforderung (REQ-1.1.3):** Fehlerbehandlung für nicht erreichbare oder fehlerhafte Feeds.

**3.2. Historisierung und Dublettenprüfung**

* **F2.1:** Speicherung aller abgerufenen Nachrichten in einer historischen Datenbank (CSV-Datei Regulatory\_News\_History.csv).
  + **Anforderung (REQ-2.1.1):** Jede News muss mit einem eindeutigen Identifikator (z.B. Link) gespeichert werden.
  + **Anforderung (REQ-2.1.2):** Sicherstellung, dass nur neue, noch nicht vorhandene News-Einträge zur weiteren Verarbeitung hinzugefügt werden (Dublettenprüfung basierend auf item\_link).

**3.3. Relevanzfilterung und Klassifizierung (LLM-basiert)**

* **F3.1:** Filtern der abgerufenen News nach Relevanz bezüglich des Themenschwerpunkts.
  + **Anforderung (REQ-3.1.1):** Einsatz eines Large Language Models (LLM) zur Bewertung der Relevanz (z.B. mistralai/mistral-small-3.2-24b-instruct:free über OpenRouter.ai API).
  + **Anforderung (REQ-3.1.2):** Das LLM muss einen "Relevanz-Label" (z.B. "Relevant" / "Nicht relevant") und eine "Relevanz-Begründung" liefern.
  + **Anforderung (REQ-3.1.3):** Konfigurierbarer Prompt für das LLM, um den thematischen Fokus klar zu definieren bzw. ändern zu können ohne den Kerncode ändern zu müssen.
  + **Anforderung (REQ-3.1.4):** Fehlerbehandlung für LLM-API-Anfragen (z.B. 400 Bad Request, 500 Internal Server Error, Timeout). Im Fehlerfall soll eine Meldung erfolgen und der Prozess ggf. fortgesetzt oder abgebrochen werden.
  + **Anforderung (REQ-3.1.5):** Die genaue Formulierung des Prompts, der das LLM zur Bewertung der Relevanz anweist, sollte konstant sein. Sobald der thematische Fokus (wie in REQ-3.1.3 konfiguriert) definiert ist, muss er bei allen nachfolgenden Ausführungen exakt gleich an das LLM übermittelt werden. Dies ist entscheidend, um sicherzustellen, dass die Filterung der Nachrichten konsistent über die Zeit erfolgt und die Ergebnisse miteinander vergleichbar sind.

**3.4. E-Mail-Generierung und -Versand**

* **F4.1:** Erstellung einer formatierten E-Mail mit der Tageszusammenfassung und einer detaillierten Liste der relevanten News.
  + **Anforderung (REQ-4.1.1):** Nutzung der Brevo API für den E-Mail-Versand. Die E-Mail-Inhalte sollen im HTML-Format verfasst sein, um eine ansprechende Darstellung zu ermöglichen.
  + **Anforderung (REQ-4.1.2):** Die E-Mail muss einen Betreff enthalten, der das Datum einschließt.
  + **Anforderung (REQ-4.1.3):** Der E-Mail-Body muss eine strukturierte Liste oder Tabelle mit dem Titel, dem Link, der Quelle, dem Datum, dem Relevanz-Status und der Begründung für jeden relevanten Artikel enthalten.
  + **Anforderung (REQ-4.1.4):** Konfigurierbare Absender- und Empfänger-E-Mail-Adressen (idealerweise über Umgebungsvariablen).
  + **Anforderung (REQ-4.1.5):** Die Authentifizierung und der Versand über die Brevo API müssen sicher erfolgen, indem der API-Schlüssel ebenfalls aus einer Umgebungsvariable gelesen wird.
  + **Anforderung (REQ-4.1.6):** Es muss eine einfache Möglichkeit geben, den E-Mail-Versand zu deaktivieren (z.B. über eine Konfigurationsvariable), um das Skript ohne tatsächliche Mail-Zustellung testen zu können.
  + **Anforderung (REQ-4.1.7):** Das Skript muss über eine Fehlerbehandlung verfügen, die bei fehlgeschlagenen API-Aufrufen (z.B. ungültiger API-Schlüssel, Serverfehler) eine aussagekräftige Meldung ausgibt und den Prozess entsprechend handhabt.

**3.5. Historisierung der LLM-Bewertungen**

* **F5.1:** Speicherung der LLM-Bewertungen (Label, Begründung, Zusammenfassung) zusammen mit den historischen News-Daten.
  + **Anforderung (REQ-5.1.1):** Die erweiterten Daten (Relevanz, Begründung, etc.) werden der Regulatory\_News\_History.csv hinzugefügt.

**3.6 Automatisierte Ausführung via Github Actions**

* **F6.1:** Automatisierte Ausführung des R-Skripts über einen GitHub Actions Workflow
  + **Anforderung (REQ-6.1.1):** Der Workflow wird basierend auf einem Cron-Zeitplan einer YAML-Datei ausgelöst. Die aktuelle Konfiguration sieht eine Ausführung zweimal täglich vor (vormittags & nachmittags).
  + **Anforderung (REQ-6.1.2):** Alle sensiblen Daten, wie API-Schlüssel oder E-Mail-Empfängerlisten, werden sicher als GitHub Secrets gespeichert und während der Laufzeit als Umgebungsvariablen bereitgestellt.
  + **Anforderung (REQ-6.1.3):** Die Ausführung erfolgt auf einem bereitgestellten virtuellen Server (ubuntu-latest) innerhalb der GitHub Actions-Plattform.
  + **Anforderung (REQ-6.1.4):** Fehlerbehandlung: Im Falle eines Fehlers im Workflow (z. B. fehlende Umgebungsvariablen) soll der Prozess abbrechen und eine aussagekräftige Warnung ausgeben.
  + **Anforderung (REQ-6.1.5):** Der Code soll so geschrieben werden, dass sowohl die lokale Ausführung als auch die automatisierte Ausführung via Github jederzeit möglich ist. Der Code muss eine klare Trennung zwischen Konfiguration (Umgebungsvariablen) und Logik aufweisen, um eine einfache Anpassung an die jeweilige Ausführungsumgebung zu ermöglichen.

**4. Nicht-funktionale Anforderungen**

* **Performance:** Das Skript sollte in einer angemessenen Zeit ausgeführt werden, um eine tägliche Ausführung zu ermöglichen.
* **Skalierbarkeit:** Das System sollte in der Lage sein, eine wachsende Anzahl von RSS-Feeds und historischen Daten zu verarbeiten.
* **Sicherheit:** API-Schlüssel müssen sicher (z.B. über Umgebungsvariablen) verwaltet werden.
* **Wartbarkeit:** Der Code sollte modular, kommentiert und leicht verständlich sein, um zukünftige Anpassungen und Erweiterungen zu erleichtern.
* **Zuverlässigkeit:** Das System sollte robust gegenüber temporären Netzwerkproblemen oder API-Ausfällen sein (durch tryCatch-Blöcke).

**5. Offene Punkte / Verbesserungspotenziale (To-Do / Backlog)**

* **Automatisierung der Ausführung:** Einrichtung eines Cron-Jobs (Linux) oder Task Schedulers (Windows) für die tägliche, unbeaufsichtigte Ausführung.
* **Fehlerbenachrichtigung:** Implementierung eines Mechanismus zur Benachrichtigung des Betreibers bei Skriptfehlern oder API-Problemen (z.B. separate E-Mail).
* **Konfigurationsmanagement:** Externe Konfigurationsdatei (z.B. YAML), um die Anpassung von Feeds, Parametern etc. zu erleichtern, ohne den Code ändern zu müssen.
* **Feinjustierung des LLM-Prompts:** Kontinuierliche Optimierung der Prompts für Relevanzfilterung und Zusammenfassung, um die Qualität der LLM-Ausgabe zu verbessern.
* **Modellwechsel-Fähigkeit:** Erleichterung des Wechsels zwischen verschiedenen LLM-Modellen oder Anbietern.
* **Erweiterung der Datenquellen:** Integration weiterer Datenquellen (z.B. direkte Webseiten-Scraping, Twitter-Monitoring).
* **Erweiterte Auswertung der Historie:** Möglichkeiten zur Analyse der historischen Daten (z.B. Trends in regulatorischen Schwerpunkten).
* **User Interface (Optional):** Langfristig könnte ein einfaches Web-Interface (z.B. mit Shiny in R) für Konfiguration, Übersicht der Historie und manuelle Trigger nützlich sein.
* **Ausgabeformat:** Optionaler Export in weitere Formate (z.B. PDF).

**Rechtliche Einordnung und Ausblick im Kontext des EU AI Act**

**1. Einordnung des Prototyps (RegMon) als KI-System**

Der Regulatorische KI-Monitor (RegMon) ist als KI-System im Sinne des EU AI Act zu klassifizieren. Die Software nutzt ein Large Language Model (LLM) zur Inferenz aus Eingabedaten (RSS-Feeds), um Vorhersagen zu treffen und Inhalte (Relevanz-Label und -Begründung) zu generieren. Da es sich um ein maschinenbasiertes System handelt, das mit einem gewissen Grad an Autonomie agiert, um aus Eingabedaten Outputs zu erzeugen, fällt es unter die gesetzliche Definition eines KI-Systems.

* Relevanter Artikel: Art. 3 (1) + Leitlinie zum [Die Kommission veröffentlicht Leitlinien zur Definition von KI-Systemen, um die Anwendung der Vorschriften des ersten KI-Gesetzes zu erleichtern | Gestaltung der digitalen Zukunft Europas](https://digital-strategy.ec.europa.eu/de/library/commission-publishes-guidelines-ai-system-definition-facilitate-first-ai-acts-rules-application)

**2. Risikoklassifizierung des RegMon Prototyps**

Die entscheidende Frage ist, ob RegMon als Hochrisiko-KI-System eingestuft wird, da dies strenge Auflagen nach sich ziehen würde.

* Potenzielles Hochrisiko: Der AI-Act klassifiziert Systeme als hochriskant, wenn sie in bestimmten Bereichen wie dem Finanzsektor eingesetzt werden und dort erhebliche Auswirkungen auf die Grundrechte von Personen haben könnten (z.B. bei der Bewertung der Kreditwürdigkeit). Da RegMon potenziell auch bei Banken zum Einsatz kommen soll, die in regulierten Märkten agieren, besteht grundsätzlich die Möglichkeit, dass es unter die Kategorie der Hochrisiko-Systeme fällt.
* Relevanter Artikel: Art. 6 (2b) + aufgeführte Bereiche in Anhang III
* Entscheidende Streitfrage: Der Ausnahmefall für Unterstützungswerkzeuge: Die wichtigste Argumentationslinie, die gegen eine Hochrisiko-Einstufung spricht, ist die Funktion von RegMon als reines Unterstützungswerkzeug. Ein KI-System, das normalerweise als hochriskant eingestuft würde, gilt nicht als solches, wenn es eine enge prozedurale oder vorbereitende Aufgabe erfüllt. Das Tool ist darauf ausgelegt, menschliche Experten zu entlasten, indem es ihnen relevante regulatorische Nachrichten filtert und voranalysiert. Die finale Bewertung, ob eine Nachricht tatsächlich relevant ist und welche Maßnahmen daraus abzuleiten sind, bleibt in der Verantwortung des menschlichen Experten. Dies wird durch die Ausgabe einer "Relevanz-Begründung“ unterstützt, die eine menschliche Überprüfung und Korrektur ermöglicht. Dieser Aspekt der "menschlichen Aufsicht" ist entscheidend für die Anwendung der Ausnahmeregelung
* Relevanter Artikel: Art. 6 (2a)

3. **Pflichten bei einer limited Risk Einwertung des Prototyps/des zu verkaufenden Tool**s

* Transparenz-/ Informationspflicht (Begrenztes Risiko): Notwendige Sicherstellung, dass die Nutzer wissen, dass sie mit einem KI-System interagieren.

Umsetzung: Eindeutige Kennzeichnung der E-Mail als KI-Output. Disclaimer einfügen, welcher besagt, dass die Relevanzeinwertung nur als Vorarbeit und Vorfilterung dienen soll. Die finale Relevanz-Entscheidung verbleibt beim Mitarbeiter.

* Relevanter Artikel: Art. 50

4. Pflichten für Anbieter von GPAI-Modellen

Da „RegMon“ ein KI-System ist, das ein General Purpose AI (GPAI)-Modell (Mistral) als Komponente verwendet, ist die Unterscheidung der Rollen nach dem EU AI Act entscheidend. Die jüngsten Leitlinien und Markteinschätzungen bestätigen die folgende Einordnung:

* Rolle von RegMon: Mit der Nutzung oder dem Vertrieb von RegMon ist man Anbieter eines KI-Systems, das ein GPAI-Modell als Komponente verwendet. Die spezifischen GPAI-Pflichten liegen beim ursprünglichen Anbieter des Modells (Mistral) und nicht bei Ihnen.
* Relevanter Artikel: Art. 52 (a) und 52 (b) + [Guidelines on the scope of obligations for providers of general-purpose AI models under the AI Act | Shaping Europe’s digital future](https://digital-strategy.ec.europa.eu/en/library/guidelines-scope-obligations-providers-general-purpose-ai-models-under-ai-act)
* Entscheidende Abgrenzung: Die Leitlinien stellen klar, dass die strengen Pflichten für GPAI-Modellanbieter nur dann übernehmen werden müssen, wenn man eine wesentliche Änderung am Basismodell vornimmt. Die Steuerung eines Modells durch Prompts, wird ausdrücklich nicht als wesentliche Änderung betrachtet. Die kürzlich veröffentlichte Einschätzung von Branchenexperten unterstreicht dies und gibt an, dass eine wesentliche Änderung erst bei einem Retraining von über einem Drittel des ursprünglichen Rechenaufwands vorliegt.
* Fazit: Die GPAI-Regeln treten zwar ab dem 2. August 2025 in Kraft, aber im Falle von RegMon muss man sich ausschließlich auf die Anforderungen konzentrieren, die sich aus der Risikoklassifizierung des Tools ergeben.