FSD AKUT

KIRAS/K-PASS Ausschreibung 2023-2 - F&E Dienstleistungen

Wissenschaftlicher Bericht

31.07.2026

Antrag-ID: 59826759

Inhalt

D1.1. Fortschritte bei der Einführung des neuen Produkts (Report/Bericht)

Der Bericht des Konsortiums über Mittelverwendung, Ergebnis- und Zielerreichung wurde im FSD Akut Endbericht abgeliefert (siehe e-Call).

D2.1. Fahrplan zur Umsetzung der ausgearbeiteten Sicherheitsstrategie für die Skalierung des FSD (Report/Bericht)

Das Projekt FSD Akut soll einen skalierten Einsatz des FSD ermöglichen, um den FSD sowohl im Bereich der Strafverfolgung als auch über Teil-Kommerzialisierungen nutzen zu können. Dafür muss an Robustheit, Sicherheit und Skalierung genauso gearbeitet werden wie an der Implementierung von rechtlichen und ethischen Guardrails.

Ziel ist es daher, den technischen und rechtlich-ethischen Rahmen einer Sicherheitsstrategie für die Skalierung des FSD zu definieren – konkret soll die Frage beantwortet werden, welche Voraussetzungen für den robusten, modularen und qualitätsgesicherten Betrieb eines skalierten Systems im Praxiseinsatz gegeben sein müssen. Die Sicherheitsstrategie wird dabei entlang technischer, organisatorischer und rechtlich-ethischer Anforderungen entwickelt. Im Einzelnen werden folgende Teilziele verfolgt:

- **Identifikation von Anforderungen für einen robusten und qualitätsgesicherten Betrieb:** Analyse der Voraussetzungen, die für den zuverlässigen, modularen und skalierbaren Betrieb KI-basierter Betrugserkennungssysteme in der Praxis erforderlich sind.
- **Bewertung unterschiedlicher Einsatz- und Skalierungsszenarien:** Untersuchung der Anforderungen und Rahmenbedingungen für den Einsatz des FSD im öffentlichen sowie im kommerziellen Sektor und Ableitung entsprechender Verantwortlichkeiten und Betriebsmodelle.
- **Stärkung der technischen Sicherheit und Resilienz des Systems:** Definition von Sicherheitsanforderungen und Schutzmaßnahmen für den Betrieb in kommerzieller und nicht-kommerzieller Umgebungen, einschließlich der Berücksichtigung potenzieller Cyberangriffe und Manipulationsversuche auf KI-Systeme.
- **Sicherstellung regulatorischer und ethischer Konformität:** Analyse relevanter rechtlicher Anforderungen, insbesondere im Bereich Datenschutz und KI-

Regulierung, sowie Bewertung ethischer Fragestellungen im Zusammenhang mit dem Einsatz automatisierter Risikobewertungssysteme.

- **Erstellung eines Umsetzungsfahrplans für die Skalierung:** Entwicklung eines strukturierten Fahrplans mit Empfehlungen zu technischen, organisatorischen und regulatorischen Maßnahmen als Grundlage für die weitere Implementierung und Verwertung des Systems.

1. Skalierungsszenarien und Anforderungsanalyse

Definition der zwei Szenarien

SZENARIO 1: Nutzung des FSD durch Polizeibehörden

Ein Anwendungsszenario für den Fake-Shop Detector (FSD) ist der Einsatz im Rahmen der Aufnahme und Erstbearbeitung von Internetbetrugsanzeigen durch Polizeibehörden. Bei der Meldung verdächtiger Online-Shops, Finanzangebote oder anderer betrügerischer Online-Angebote ist ein Einsatz des FSD als unterstützendes Werkzeug angedacht, insbesondere um eine erste automatisierte Risikoeinschätzung der zur Anzeige gebrachten Domain vorzunehmen und Beamt:innen bereits im Zuge der Anzeigenaufnahme mit zusätzlichen Hinweisen auf Betrug und Seriosität zu unterstützen.

Für den operativen Einsatz, vor allem auf Ermittlungsebene liegt eine zentrale Einschränkung vor: die vom FSD generierten Risikobewertungen sind nur begrenzt nachvollziehbar. Das System liefert einen Risikoscore, jedoch keine vollständige Begründung, welche konkreten Faktoren zu der jeweiligen Einstufung geführt haben. Für behördliche Entscheidungen, die dokumentiert, überprüfbar und gegebenenfalls vor Gericht nachvollziehbar sein müssen, stellt dies eine wesentliche Hürde dar. Die Ergebnisse des FSD könnten daher nicht als Beweismittel oder alleinige Entscheidungsgrundlage verwendet werden.

Die Anforderungen hinsichtlich Transparenz, Reproduzierbarkeit und Qualitätssicherung sind in der behördlichen Anwendung entsprechend besonders hoch. Falsch-positive Bewertungen könnten legitime Unternehmen betreffen, während falsch-negative Bewertungen dazu führen könnten, dass betrügerische Angebote nicht erkannt werden. Ebenso müssen organisatorische und technische Anforderungen berücksichtigt werden, insbesondere wenn das System in bestehende behördliche IT-Infrastrukturen integriert werden soll.

SZENARIO II: Nutzung der KI-Detektionsmodelle durch Internet Service Provider

Das zweite Szenario beschreibt eine kommerzielle Verwertung, bei der die Detektionsmodelle des FSD bei Internet Service Provider (ISP) integriert werden. Ziel ist es dabei durch eine Teil-Kommerzialisierung des FSD eine langfristige Finanzierung des kostenlosen Echtzeitschutzes für Konsument:innen zu ermöglichen.

Als privatwirtschaftlicher Betreiber unterliegt ein ISP uneingeschränkt der DSGVO, die Rolle des ISP (Verantwortlicher, Auftragsverarbeiter oder gemeinsam Verantwortlicher) ist dabei vorab zu klären, da sie Vertragsgestaltung und Haftungsverteilung bestimmt. Zentral ist die Frage, ob das bestehende Datenschutzversprechen des FSD – die lokale, gerätebasierte Risikobewertung ohne Erfassung personenbezogener Nutzer:innendaten auch bei einer zentralisierten Integration auf ISP-Ebene erhalten bleibt. Eine Verlagerung der Bewertung weg vom Endgerät könnte dieses Versprechen unterlaufen und ist daher kritisch zu prüfen.

Hinzu kommen im kommerziellen Betrieb vertragliche Zusagen zu Verfügbarkeit und Reaktionszeiten (Service Level Agreements), die im behördlichen Szenario in dieser Form nicht bestehen.

Vergleich SZENARIO I und SZENARIO II

Die zwei skizzierten Szenarien unterscheiden sich insbesondere in der Finanzierungslogik, den Datenschutzanforderungen sowie den zentralen Rahmenbedingungen (siehe Tabelle 1).

Kriterium	Nicht-kommerziell (BMI/Polizei)	Kommerziell (ISP)
Betreiber	Strafverfolgungsbehörde (Bedarfsträger BMI)	Privatwirtschaftliches Unternehmen (Internet Service Provider)
Zweck / Einsatz	Unterstützung des Anzeigeprozesses; Erstbewertung und Priorisierung von Verdachtsfällen	Breitenwirksamer, automatisierter Basisschutz vor betrügerischen Websites im laufenden Betrieb
Finanzierungslogik	Öffentliche Hand; gemeinwohlorientiert	Teil-Kommerzialisierung; trägt zur Finanzierung des kostenlosen Echtzeitschutzes bei
Datenschutz	JI-Richtlinie / 3. Hauptstück DSG im Anzeigekontext; Zweckbindung und Trennung der Ermittlungsdaten vom Warnlisten-Datenbestand	DSGVO uneingeschränkt; Rolle des ISP (Verantwortlicher / Auftragsverarbeiter) zu klären; bei netzseitiger Bewertung Einwilligung nach § 167 Abs 4 TKG 2021 erforderlich

Zentrale Anforderung	Nachvollziehbarkeit, Dokumentierbarkeit und Erklärbarkeit der Einschätzungen	Skalierbarkeit, Verfügbarkeit, Erhalt des Datenschutzversprechens bei Zentralisierung
AI Act Einstufung	Anhang III Nr. 6 einschlägig, aber Ausnahme nach Art 6 Abs 3 KI-VO (nur vorbereitende Aufgabe) – voraussichtlich nicht hochriskant; Einstufung zu dokumentieren	Voraussichtlich minimales Risiko (vergleichbar Spam-Filter)
Service Level Agreement (SLA)	Keine kommerziellen SLA; behördliche Verfügbarkeitsanforderungen	Vertragliche SLA zu Verfügbarkeit und gestaffelten Reaktionszeiten

Tabelle 1: Gegenüberstellung der beiden Skalierungsszenarien

Anforderungen je Verantwortungsdomäne

Aufbauend auf der Unterscheidung der beiden Szenarien werden die Anforderungen an einen sicheren und skalierbaren Betrieb des FSD im Folgenden entlang dreier Verantwortungsdomänen strukturiert: dem Machine Learning System, der Infrastruktur und der Qualitätssicherung. Ziel ist es die technischen, rechtlichen und ethischen Verantwortlichkeiten je Domäne zu definieren und die entsprechenden Anforderungen klar zu zuordnen.

ML-System

Auf Ebene des ML-Systems ergeben sich für den skalierten Betrieb drei zentrale Anforderungen:

(1) Relearning-Schleifen: Die manuellen Korrekturen vonseiten der Expert:innen der Watchlist Internet fließen in die Trainingsdaten zurück und verbessern so das Modell. Dieser Rückfluss muss auch bei höherem Durchsatz zuverlässig funktionieren.

(2) Erklärbarkeit: Insbesondere im behördlichen Szenario ist die Erklärbarkeit von Einschätzungen eine zentrale Anforderung: Eine in einem Anzeigeprozess einfließende ML-Bewertung muss für die ermittelnde Person transparent und nachvollziehbar sein. Dafür müssen neben den Risiko-Score Einschätzungen zu den ausschlaggebenden Merkmalen in verständlicher Form dargestellt werden.

(2) Generalisierung: Insbesondere im Kontext polizeilicher Anzeigen ist eine Ausweitung des FSD-Ansatzes auf weitere Betrugskategorien sinnvoll, da unterschiedliche Maschen unter Online-Betrug fallen. Zu nennen ist dabei insbesondere Cyber Trading Fraud, der mit enormen finanziellen Schäden für Betroffene einhergeht. Neue Betrugskategorien müssen dabei aufgenommen werden können, ohne dass ein vollständiges Re-Deployment des Systems notwendig ist.

Infrastruktur

Die FSD-Infrastruktur ist auf horizontale Skalierung ausgelegt und kann durch Hinzufügen weiterer virtuelle Server erweitert werden, um eine hohe Systemstabilität zu gewähren. Mitgedacht werden muss jedoch, dass eine Skalierung auch mit einem deutlich höheren Aufkommen an Support-Anfragen zu rechnen ist. Zudem muss ein zentraler Bestandteil des sicheren Betriebs auf einer laufenden Überwachung der eingesetzten Komponenten auf Aktualität und bekannte Sicherheitslücken (inkl. regelmäßiger Sicherheitspatches und Upgrades) basieren.

Auf Infrastrukturebene ist zudem der bereits bestehende Privacy-by-Design-Ansatz eine maßgebliche Anforderungen. Dazu zählt, dass Risikobewertungen lokal bzw. offline erfolgen, und die Datenerhebung werden auf das technisch notwendige Minimum beschränkt bleibt: Verarbeitet werden ausschließlich URL und Bewertungszeitpunkt sowie zu Sicherheitszwecken die IP-Adresse des anfragenden Geräts, diee wird nach 14 Tagen gelöscht. Dieser Ansatz darf durch die Skalierung nicht erodieren. Insbesondere im kommerziellen Szenario darf eine zentralisierte Bereitstellung der Modelle nicht dazu führen, dass Profile einzelner Endnutzer:innen entstehen.

Im behördlichen Szenario kommt die Einbindung in bestehende Intranet-Infrastrukturen hinzu. Das webbasierte Tool muss sich dabei in die IT-Umgebung des Bedarfsträgers integrieren lassen und dabei eine klare Trennung von Ermittlungsdaten gegenüber dem öffentlichen Warnlisten-Datenbestand gewährleisten.

Qualitätssicherung

Die manuelle Qualitätssicherung durch Expert:innen ist konstitutiver Bestandteil des FSD: Sie wirkt als Korrektiv gegen Fehlklassifikationen und systematische Verzerrungen und speist gleichzeitig über die Korrekturen die Verbesserung der Modelle. Mit wachsender Datenmenge stößt dieser Human-in-the-Loop-Ansatz jedoch an personelle Grenzen. Die Skalierung der manuellen Qualitätssicherung ist daher ein offenes Risiko.

Ein möglicher Entlastungsweg ist die Auslagerung von einfachen Prüfaufgaben an Data Worker, dieser ist wiederum mit erheblichen arbeitsrechtlichen und ethischen Problemen behaftet (prekäre Vergütung, fehlende Absicherung, psychische Belastung durch potenziell problematische Inhalte) und steht im Spannungsverhältnis zu Fairness- und Verantwortungsprinzipien des Konsortiums.

Die Verortung der Qualitätssicherung unterscheidet sich je Szenario: Im BMI-Szenario verschiebt sich ein Teil der inhaltlichen Prüfung in den Anzeigeprozess, eingehende Anzeigen werden durch Expert:innen geprüft, bevor diese in Warnlisten überführt werden. Im kommerziellen Szenario bleibt die Verantwortung der Qualitätssicherung bei der Redaktion der Watchlist Internet. Hier wäre eine priorisierte Qualitätssicherung anzudenken – etwa durch die Fokussierung auf bestimme Domain-Räume (z. B. .at/.de).

2. Security-Komponenten und Belastungstests

Aufbauend auf den in Kapitel 2 beschriebenen Skalierungsszenarien werden im Folgenden die technischen Anforderungen an die Security-Komponenten für den skalierten Betrieb des Fake-Shop Detector (FSD) hinsichtlich Zielsetzungen und Sicherheitsanforderungen abgeleitet. Während im behördlichen Einsatzszenario insbesondere die Integrität, Nachvollziehbarkeit sowie sichere Integration in bestehende IT-Infrastrukturen im Vordergrund stehen, entstehen im kommerziellen Szenario mit Internet Service Providern zusätzliche Anforderungen hinsichtlich Verfügbarkeit, Skalierbarkeit und Schutz vor missbräuchlicher Nutzung.

Mit zunehmender Verbreitung des FSD steigt gleichzeitig dessen Attraktivität als Angriffsziel. Die Sicherheitsstrategie muss daher sowohl klassische Bedrohungen auf Ebene der IT-Infrastruktur als auch spezifische Risiken KI-basierter Systeme berücksichtigen. Während Angriffe wie Distributed-Denial-of-Service-Attacken (DDoS) primär die Verfügbarkeit des Dienstes gefährden, können Angriffe auf KI-Komponenten die Integrität und Vertrauenswürdigkeit der Analyseergebnisse beeinträchtigen. Dazu zählen insbesondere Versuche, die Entscheidungslogik durch Reverse Engineering zu rekonstruieren oder Webseiten gezielt so anzupassen, dass eine fehlerhafte Bewertung erzeugt wird.

Für ein System zur automatisierten Betrugserkennung ist die Sicherstellung der Ergebnisintegrität von besonderer Bedeutung. Eine erfolgreiche Manipulation könnte dazu führen, dass betrügerische Angebote nicht erkannt werden oder legitime Unternehmen fälschlicherweise als betrügerisch bewertet werden. Dies kann neben technischen Auswirkungen auch rechtliche und wirtschaftliche Konsequenzen nach sich ziehen.

Die Security-Strategie verfolgt daher einen mehrschichtigen Ansatz, bei dem technische Schutzmaßnahmen auf Ebene der Infrastruktur, Schnittstellen, Datenverarbeitung und KI-Komponenten kombiniert werden. Ziel ist ein robuster, modularer und qualitätsgesicherter Betrieb des FSD auch bei deutlich steigenden Nutzerzahlen.

Die Anforderungen werden im Folgenden anhand unterschiedlicher Skalierungsszenarien betrachtet: neben einer Baseline des Status Quo bis zu 30.000 Nutzer:innen werden

zukünftige mittlere Skalierungen mit etwa 50.000 Nutzer:innen sowie langfristig größere Deployment-Umgebungen mit bis zu 100.000 Nutzer:innen berücksichtigt. Mit steigender Nutzung verändern sich insbesondere die Anforderungen an Lastverteilung, Überwachung, Zugriffsschutz und Angriffserkennung.

Security-by-Design Architektur des skalierten FSD

Die technische Grundlage für einen sicheren und skalierbaren Betrieb des FSD bildet eine modulare, event-basierte Architektur auf Basis von Apache Pulsar. Die Architektur wurde so gestaltet, dass einzelne Verarbeitungsschritte unabhängig voneinander betrieben, überwacht und skaliert werden können. Dadurch werden Abhängigkeiten zwischen Komponenten reduziert und einzelne Systembereiche können bei Fehlern oder Sicherheitsvorfällen isoliert behandelt werden.

Externe Zugriffe erfolgen ausschließlich über definierte Entry Points. Dazu zählt insbesondere die FSD Middleware API, die als Schnittstelle zwischen dem FSD Browser Plugin und der Backend-Infrastruktur dient. Darüber hinaus verarbeitet ein Certstream-Ingest neu registrierte beziehungsweise verlängerte HTTPS-Zertifikate und filtert relevante Domains anhand eines DACH-spezifischen Filters für die weitere Analyse.

Nach Eingang einer Analyseanfrage erfolgt die Verarbeitung innerhalb der geschützten FSD-Pipeline. Apache Pulsar übernimmt dabei die Verteilung der Nachrichten zwischen den einzelnen Verarbeitungsschritten. Hierzu zählen insbesondere die URL-Frontier-Komponente mit Scheduling- und Retry-Logik, die Scraper-Komponenten zur Erzeugung von WARC-Dateien sowie die anschließende Klassifikation durch verschiedene KI-basierte Analysekomponenten.

Die Classifier-Komponenten stellen dabei einen zentralen Bestandteil der Pipeline dar. Neben der Berechnung des Fake-Shop-Risikos und der Shop-Klassifikation werden weitere Metadaten wie Kategorie oder Sprache einer Webseite bestimmt. Die modulare Architektur ermöglicht die Erweiterung um zusätzliche Analysefunktionen, beispielsweise die Extraktion von Impressumsinformationen, Telefonnummern oder Unternehmenskennungen, ohne dass die grundlegende Systemarchitektur verändert werden muss.

Die Speicherung der Analyseergebnisse erfolgt über eine MongoDB-basierte Data-Lake-Architektur. Diese ermöglicht eine flexible Speicherung unterschiedlicher Datenstrukturen und unterstützt sowohl Forschungsanforderungen als auch produktive Betriebsprozesse. Gleichzeitig ermöglicht die Kombination aus Apache Pulsar und MongoDB eine verbesserte Nachvollziehbarkeit, da Verarbeitungsschritte, Klassifikationsergebnisse und Änderungen historisiert werden können.

Die Architektur unterstützt unterschiedliche Nutzungsszenarien durch getrennte Workflows. Dadurch können beispielsweise Forschungsaktivitäten, der Tagesbetrieb der Watchlist Internet sowie zukünftige kommerzielle Nutzungsszenarien getrennt verarbeitet werden. Diese Trennung ist insbesondere für einen sicheren Betrieb in unterschiedlichen organisatorischen Kontexten relevant.

Notwendige Security-Komponente

Für den Betrieb des FSD in großen Umgebungen sind zusätzliche technische Schutzmaßnahmen erforderlich, die sich aus den erwarteten Nutzungsszenarien und Bedrohungen ableiten.

Eine zentrale Rolle spielt dabei die Absicherung der öffentlich erreichbaren Schnittstellen. Die Middleware API stellt einen potenziellen Angriffspunkt dar, da sie aus externen Netzen erreichbar ist und eine große Anzahl an Analyseanfragen verarbeiten muss. Für große Deployment-Szenarien ist daher eine vorgeschaltete Schutzschicht erforderlich, die eingehende Anfragen kontrolliert, missbräuchliche Nutzung erkennt und die Verfügbarkeit des Dienstes sicherstellt.

Der FSD verfolgt hierbei bereits einen Security-by-Design-Ansatz. Die externe Schnittstelle stellt ausschließlich jene Informationen bereit, die für die Nutzung erforderlich sind. Anstelle detaillierter Risiko-Scores oder interner Modellinformationen werden abstrahierte Risikobänder ausgegeben. Dadurch wird verhindert, dass externe Nutzer:innen ausreichende Informationen erhalten, um die Modelllogik systematisch zu analysieren oder betrügerische Angebote gezielt gegen das Modell zu optimieren.

Ein vorgeschalteter Reverse Proxy ermöglicht zusätzliche Schutzmechanismen wie Rate Limiting, Zugriffsbeschränkungen und die Erkennung ungewöhnlicher Zugriffsmuster. Diese Maßnahmen sind insbesondere bei großen Nutzerzahlen relevant, da automatisierte Anfragen oder missbräuchliche Nutzung der Schnittstellen einen erheblichen Einfluss auf die Systemverfügbarkeit haben können.

Für mittlere und große Deployment-Szenarien ist insbesondere die Skalierbarkeit der Verarbeitungskomponenten entscheidend. Die Classifier-Komponenten des FSD sind vollständig containerisiert und können unabhängig voneinander betrieben und skaliert werden. Dadurch können zusätzliche Instanzen bei steigender Nachfrage bereitgestellt werden, ohne dass Änderungen an anderen Systemkomponenten erforderlich sind.

Die bekannte Ressourcencharakteristik der einzelnen Komponenten ermöglicht zudem eine gezielte Dimensionierung der Infrastruktur. Durch die Optimierung der KI-Komponenten konnte der Ressourcenbedarf bereits deutlich reduziert werden. Beim Shop-No-Shop Classifier konnte beispielsweise der Speicherbedarf unter Burst-Last von 2,4 GiB auf 467,9 MiB reduziert werden. Beim Fake-Shop-Risk Classifier wurde eine

Reduktion von 4,6 GiB auf 600,4 MiB erreicht. Diese Verbesserungen ermöglichen einen effizienteren Betrieb zusätzlicher Instanzen und erhöhen die Resilienz gegenüber Lastspitzen.

Ein weiterer wesentlicher Bestandteil der Security-Architektur ist die kontinuierliche Überwachung der Infrastruktur und Systemkomponenten. Die technische Infrastruktur wird mittels Checkmk überwacht, während zentrale FSD-Komponenten und Pipeline-Metriken über Grafana-Dashboards analysiert werden. Dadurch können ungewöhnliche Systemzustände, erhöhte Last oder Fehlverhalten einzelner Komponenten frühzeitig erkannt werden.

Schutz vor Reverse Engineering und Manipulation

Neben klassischen Angriffen auf die Infrastruktur müssen bei einem KI-basierten System wie dem FSD spezifische Bedrohungen berücksichtigt werden. Ein relevantes Angriffsszenario besteht darin, dass kriminelle Akteur:innen versuchen, die Funktionsweise der KI-Komponenten zu analysieren und gezielt auszunutzen.

Beim Reverse Engineering versucht ein Angreifer, durch eine große Anzahl gezielter Anfragen Rückschlüsse auf die Entscheidungslogik des Modells zu gewinnen. Ziel kann sein, Entscheidungsgrenzen zu rekonstruieren oder Webseiten gezielt so anzupassen, dass diese nicht mehr als betrügerisch erkannt werden.

Die Schutzmaßnahmen des FSD setzen daher insbesondere auf eine Begrenzung des Informationsgehalts öffentlicher Schnittstellen. Die Middleware API stellt keine einzelnen Modellfeatures, Gewichtungen oder internen Entscheidungsinformationen bereit. Zusätzlich ist über die öffentliche Schnittstelle keine automatisierte Re-Klassifikation bereits bewerteter Domains vorgesehen. Diese Maßnahmen sollen den Informationsgewinn eines potenziellen Angreifers deutlich reduzieren; ein Restrisiko verbleibt, da die Echtzeitanalyse unbekannter Domains prinzipiell auch für gezielt registrierte Test-Domains angefragt werden kann.

Neben Reverse Engineering müssen auch Manipulationsversuche gegen die Eingabedaten berücksichtigt werden. Betreiber betrügerischer Webseiten könnten versuchen, bekannte Erkennungsmerkmale gezielt zu verändern, beispielsweise durch künstlich erzeugte Vertrauenssignale oder die Nachbildung seriöser Webseitenstrukturen.

Diesen Risiken wird durch die Kombination aus kontinuierlicher Modellweiterentwicklung, Qualitätssicherung und Human-in-the-Loop-Prozessen begegnet. Die manuelle Prüfung durch Expert:innen stellt sicher, dass Fehlklassifikationen erkannt und in die Weiterentwicklung der Modelle sowie

Trainingsdaten zurückgeführt werden können. Die Skalierung dieser Prozesse ist daher ein wesentlicher Bestandteil der Sicherheitsstrategie.

Belastungstests und Angriffssimulationen

Zur Validierung der definierten Security-Komponenten werden Belastungs- und Angriffstests spezifiziert. Ziel dieser Tests ist es, die Skalierbarkeit der Architektur sowie die Widerstandsfähigkeit gegenüber technischen Angriffen und Manipulationsversuchen zu überprüfen.

Die Skalierungstests betrachten die definierten Nutzungsszenarien mit 30.000, 50.000 und 100.000 Nutzer:innen. Dabei werden insbesondere die Verarbeitungskapazität, Antwortzeiten, Fehlerraten sowie die Auslastung zentraler Systemkomponenten untersucht. Ein besonderer Fokus liegt auf den KI-Classifiers, da diese zentrale Verarbeitungsschritte innerhalb der Analysepipeline darstellen.

Neben regulären Lasttests werden Szenarien mit kurzfristig stark erhöhtem Anfragevolumen betrachtet. Diese simulieren beispielsweise Situationen, in denen aktuelle Betrugswellen oder mediale Aufmerksamkeit zu einem starken Anstieg von Analyseanfragen führen. Dabei wird überprüft, ob die horizontale Skalierung der Architektur ausreichend ist und ob einzelne Komponenten ohne Beeinträchtigung des Gesamtsystems erweitert werden können.

Ein wesentliches Angriffsszenario stellt die Simulation von DDoS-Angriffen dar. Dabei wird untersucht, wie das System auf eine große Anzahl paralleler Anfragen reagiert und ob vorhandene Schutzmechanismen wie Reverse Proxy, Rate Limiting und Monitoring ausreichend wirksam sind. Neben der Aufrechterhaltung der Systemverfügbarkeit wird insbesondere geprüft, ob legitime Nutzeranfragen weiterhin verarbeitet werden können.

Ein weiteres Testszenario betrifft Reverse Engineering. Hierbei wird simuliert, wie ein Angreifer versucht, durch systematische Anfragen Informationen über die Klassifikationslogik abzuleiten. Bewertet wird, welche Informationen aus der Schnittstelle gewonnen werden können, ob ungewöhnliche Anfrageprofile erkannt werden und ob bestehende Schutzmaßnahmen ausreichend verhindern, dass die Modelllogik praktisch rekonstruiert werden kann.

Zusätzlich werden Manipulationsszenarien betrachtet, bei denen Webseiten gezielt verändert werden, um die KI-Klassifikation zu beeinflussen. Hierbei wird untersucht, wie robust die Modelle gegenüber veränderten Eingabedaten sind und wie schnell neue Angriffsmuster durch Qualitätssicherung und Retraining berücksichtigt werden können.

Zusammenfassung der Security-Strategie

Die Sicherheitsstrategie für den skalierten Betrieb des FSD basiert auf einer Kombination aus modularer Systemarchitektur, kontrollierten Schnittstellen, horizontal skalierbaren KI-Komponenten sowie kontinuierlichem Monitoring.

Die Verwendung von Apache Pulsar als zentrale Verarbeitungsschicht ermöglicht eine robuste und flexible Architektur, in der einzelne Komponenten unabhängig skaliert und überwacht werden können. Die Absicherung öffentlicher Schnittstellen, die Begrenzung verfügbarer Informationen sowie technische Schutzmechanismen gegen automatisierte Angriffe reduzieren das Risiko von Missbrauch und gezielter Manipulation.

Durch definierten Belastungs- und Angriffstests sollen zukünftig wesentlichen Sicherheitsanforderungen für große Deployment-Szenarien überprüfbar gemacht werden. Damit wird eine Grundlage geschaffen, den FSD auch bei steigender Nutzung robust, verfügbar und vertrauenswürdig im praktischen Einsatz betreiben zu können.

3. Rechtliche Risiken und Compliance Anforderungen

Die in Kapitel 2 ausgearbeiteten Anforderungen an einen sicheren und skalierbaren Betrieb des FSD müssen mit einem Geflecht rechtlicher und ethischer Vorgaben in Einklang stehen. In diesem Kapitel werden daher die rechtlicher Risiken und Compliance-Anforderungen analysiert.

Kreditschädigungen durch Warnhinweis

Bei der Nutzung des FSD kann ein Warnhinweis zu Webseiten angezeigt werden, der lauten kann: „*Vorsicht: Fake-Shop: Es handelt sich um einen Fake-Shop oder das Fake-Shop Risiko ist sehr hoch*“. Soweit die Aussage, dass es sich bei einer Webseite bzw. einem Onlineshop um einen Fake-Shop handle, nicht der Warheit entspricht, kann es sich bei dem Warnhinweis um eine kreditschädigende Äußerung im Sinne des § 1330 Allgemeines Bürgerliches Gesetzbuch (ABGB) handeln. Der Betreiber der Webseite bzw. Shopbetreiber, der in seiner wirtschaftlichen Reputation verletzt wird, könnte entsprechende Ansprüche geltend machen.

Nutzung des FSD durch Polizeibehörden

Im Rahmen der Nutzung des FSD durch Polizeibehörden bei der Aufnahme und Erstbearbeitung von Strafanzeigen würde die Polizeibehörde keine (eigene) öffentliche Äußerungen vornehmen. Die Mitteilung der Einschätzung an ein Opfer, dass es sich bei einer Webseite um einen Fakeshop handle, könnte eine kreditschädigende Verbreitung unwahrer Tatsachen sein. Es ist nämlich ausreichend, dass die Mitteilung an eine Person erfolgt (6 Ob 97/06t). Für eine nicht öffentlich vorgebrachte Mitteilung, deren Unwahrheit der Mitteilende nicht kennt, besteht aber keine Haftung, wenn der Empfänger der

Mitteilung an ihr ein berechtigtes Interesse hatte (siehe § 1330 Abs 2 Satz 2 ABGB). Das rechtliche Risiko hinsichtlich möglicher Ansprüche gegen die Polizeibehörden bzw. den Bund auf Grundlage von § 1330 ABGB ist also als gering einzustufen.

Nutzung des FSD durch ISPs

In dem angedachten Szenario der Nutzung des FSD durch einen Internet Service Provider (ISPs) würde der ISP beim Versuch des Besuchs einer bestimmten Webseite auf einen Warnhinweis umleiten, in dem die Webseite als betrügerisches Angebot bezeichnet wird. Soweit der Warnhinweis zu Unrecht angezeigt wird, würde der ISP eine eigene kreditschädigende öffentliche Äußerung tätigen und hätte ein entsprechendes Haftungsrisiko.

Ein Shopbetreiber könnte unter anderem die Unterlassung und den öffentlichen Widerruf dieses Warnhinweises sowie Schadenersatz für eine allfällig eingetreten Schaden vom ISP fordern. Der Schaden könnte zum Beispiel in einem Umsatzrückgang und einem entsprechenden entgangenen Gewinn liegen, wenn der Umsatzrückgang auf den Warnhinweis zurückzuführen ist. Rein praktisch kann sich der Nachweis der Kausalität zwischen dem Warnhinweis und dem Umsatzrückgang im Detail für den Shopbetreiber als herausfordernd darstellen.

Soweit der ISP für solche Ansprüche haftet, stellt sich die Frage, ob er sich an den Betreibern des FSD regressieren kann. Allfällige Regressansprüche könnten zwischen den Betreibern des FSD und dem ISP vertraglich ausgeschlossen oder betraglich beschränkt werden. Nur der vollständige Ausschluss von Regressansprüchen auch im Fall des Vorsatzes oder einer groben Fahrlässigkeit der Betreiber der FSD würde wohl wegen des sittenwidrigen Regelungsgehalts an § 879 Abs 3 ABGB (gröbliche Benachteiligung eines Teils) scheitern.

Datenschutzrechtliche Fragen

Im Rahmen der Echtzeitanalyse des FSD werden von der IP-Adresse des genutzten Geräts die Webadresse (URL) einer aktuell besuchten Webseite an die FSD Datenbank gesendet. Die Webseite wird analysiert und das Ergebnis der Analyse wird über das FSD-Plugin ausgegeben. Es werden keine Informationen über das Such-, Surf- oder Einkaufsverhalten der Nutzer:innen verarbeitet. Im Rahmen der Echtzeitanalyse des FSD kann die aufgerufene Website mit der IP-Adresse des entsprechenden Geräts in Verbindung gesetzt werden. Die Datenverarbeitung dient dem Zweck, die (zuvor noch nicht analysierte) besuchte Webseite über den FSD zu bewerten, und erfolgt auf Grundlage der von dem/der Nutzer:in erteilten Einwilligung (Art 6 Abs 1 lit a DSGVO) bzw. zu dem Zweck, die vertragliche Verpflichtung gegenüber der:die Nutzer:in zur Bereitstellung des FSD zu erfüllen (Art 6 Abs 1 lit b DSGVO).

Beim Betrieb des FSD werden (im Zuge der Aktualisierung des Plugin-Caches oder im Zuge der Echtzeitanalyse) die IP-Adresse des anfragenden Geräts, der Zeitpunkt der Verbindung für 14 Tage in Server-Log-Dateien gespeichert und danach automatisiert gelöscht. Die Datenspeicherung erfolgt allein zum Zweck der Abwehr von Cyber-Angriffen (z. B. DDoS-Attacken) und ist aufgrund berechtigter Interessen (Art 6 Abs 1 lit f DSGVO) der Verantwortlichen gerechtfertigt.

Nutzung des FSD durch Polizeibehörden

Im Rahmen der Nutzung des FSD durch Polizeibehörden bei der Aufnahme und Erstbearbeitung von Strafanzeigen werden durch den FSD selbst keine über das oben beschriebene Ausmaß hinausgehenden Daten erhoben oder gespeichert. Die Organe der Polizeibehörde würden den FSD wie andere Nutzer:innen verwenden. Zu beachten ist jedoch der Verarbeitungskontext: Bei einer Integration des FSD in behördliche IT-Infrastrukturen sind entsprechend Zweckbindung und eine klare Trennung der Ermittlungsdaten vom öffentlichen Warnlisten-Datenbestand sicherzustellen, da nicht die DSGVO, sondern die Richtlinie (EU) 2016/680 (JI-Richtlinie) gilt.

Nutzung des FSD durch ISPs

Für das angedachte Szenario der Nutzung des FSD durch Internet Service Provider (ISPs) wäre die datenschutzrechtliche Bewertung zentral vom Ausmaß der Datenverarbeitung auf ISP-Ebene abhängig. Grundsätzlich gilt: Nach § 167 Abs 1 TKG 2021 dürfen österreichische ISPs Verkehrsdaten (wie z. B. die IP-Adresse) oder den Zeitpunkt der Verbindung nur dann und insoweit verarbeiten, als dies für die Verbindungsherstellung, die Abrechnung oder eine Störungsbehebung notwendig sind. Danach müssen die Daten gelöscht oder anonymisiert werden. Nach § 167 Abs 4 TKG darf der ISP nur mit Zustimmung des Nutzers die Daten für die Bereitstellung von Diensten mit Zusatznutzen verwenden. Eine Verarbeitung der IP-Adresse zwecks Anzeige eines Warnhinweises durch den ISP hinsichtlich einer durch den FSD analysierten (besuchten) Webseite würde also eine vorherige Einwilligung durch den Nutzer gegenüber dem ISP voraussetzen.

Anforderungen aus KI-Verordnung

Die Verordnung (EU) 2024/1689 über künstliche Intelligenz (KI-VO) verfolgt einen risikobasierten Ansatz, um ein verhältnismäßiges, wirksames und verbindliches Regelwerk für KI-Systeme einzuführen. KI-Systeme werden entsprechend ihrem Risikopotential nach inakzeptablem, hohem, geringem und minimalem Risiko kategorisiert. Risiko definiert die KI-VO als „die Kombination aus der Wahrscheinlichkeit des Auftretens eines Schadens und der Schwere dieses Eintritts“ (Art. 3) und bewertet insbesondere mögliche Schäden an individuellen und/oder öffentlichen Interessen (Gesundheit, Sicherheit, Grundrechte, einschließlich Demokratie, der Rechtsstaatlichkeit und des Umweltschutzes). Die Einordnung richtet sich nicht nach der

verwendeten Technik, sondern nach dem Verwendungszweck des KI-Systems (Art. 6 KI-VO).

Nutzung des FSD durch Polizeibehörden

Im Rahmen der Nutzung des FSD durch Polizeibehörden bei der Aufnahme und Erstbearbeitung von Strafanzeigen müsste geprüft werden, ob der FSD als Hochrisiko-KI-System qualifiziert werden könnte.

Nach Nr. 6 Nr. 2 KI-VO gelten unter anderem die in Anhang III genannten KI-Systeme als mit hohem Risiko behaftet (Hochrisiko-KI-Systeme). Im Anhang III Nr. 6 des zur KI-VO wird der Bereich der Strafverfolgung folgendes als Hochrisiko-KI-System genannt: „KI-Systeme, die bestimmungsgemäß von Strafverfolgungsbehörden [...] zur Bewertung des Risikos einer natürlichen Person, zum Opfer von Straftaten zu werden, verwendet werden sollen“. Der FSD bewertet, ob ein Online-Shop betrügerisch ist und ob Konsument:innen Opfer einer (Betrugs-)Straftat werden könnten. Damit dürfte er in den Anwendungsbereich von Anhang III Nr. 6 KI-VO fallen.

Ein in Anhang III genanntes KI-System gilt nach Art 6 Abs 3 KI-VO wiederum nicht als hochriskant, wenn es kein erhebliches Risiko der Beeinträchtigung in Bezug auf die Gesundheit, Sicherheit oder Grundrechte natürlicher Personen birgt, indem es unter anderem nicht das Ergebnis der Entscheidungsfindung wesentlich beeinflusst (z. B. nur dazu bestimmt ist, eine vorbereitende Aufgabe für eine Bewertung durchzuführen). Der FSD erfüllt wohl nur eine vorbereitende Aufgabe für die Bewertung, ob eine strafbare Handlung begangen wurde, weil die genauen Tatbestandsvoraussetzungen (insb. die Täuschung) gesondert von den Strafverfolgungsbehörden ermittelt werden müssten. Dementsprechend wird der FSD wohl auch in diesem Kontext nicht als Hochrisiko-KI-System zu bewerten sein.

Nutzung des FSD durch ISPs

In dem angedachten Szenario der Nutzung des FSD durch Internet Service Provider (ISPs) würde der FSD nur eingesetzt, um Nutzer:innen einen Warnhinweis anzuzeigen und sie in ihrer individuelle Kaufentscheidung der Nutzer:innen zu unterstützen. Damit fiele der Detektor wahrscheinlich nur unter minimales Risiko. Das System trifft keine Entscheidungen über Personen, sondern informiert Nutzer:innen über externe Risiken. Vergleichbar sind Spam-Filter, die explizit als Beispiel für minimales Risiko gelten, bei dem keine spezifischen Pflichten der KI-VO anfallen.

Spannungsfeld zu Netzneutralität

Nach Art 3 Abs 3 Netzneutralitäts-VO (TSM-VO) sind Access-Provider grundsätzlich zur Gleichbehandlung des gesamten Datenverkehrs verpflichtet. Es gibt zwar die Möglichkeit, transparente, nichtdiskriminierende und verhältnismäßige Einschränkungen

(„Vekehrsmanagementmaßnahmen“) vorzunehmen. Solche Einschränkungen dürfen allerdings nur auf Basis einer gesetzlichen Grundlage oder einer gerichtlichen oder behördlichen Verfügung vorgenommen werden.

Nutzung des FSD durch Polizeibehörden

Im Rahmen der Nutzung des FSD durch Polizeibehörden würde der FSD im Rahmen der Aufnahme und Erstbearbeitung von Internetbetrugsanzeigen genutzt. Es besteht für Polizeibehörden soweit ersichtlich keine Rechtsgrundlage, um eine Sperre von Webseiten anzuordnen. Daher kann es im Rahmen der Nutzung des FSD durch Polizeibehörden auch zu keinem Konflikt mit der Netzneutralität kommen.

Nutzung des FSD durch ISPs

Für das angedachte Szenario der Nutzung des FSD durch Internet Service Provider (ISPs): Ein ISP dürfte eine Einschränkung des Datenverkehrs nur auf Basis einer gerichtlichen oder behördlichen Verfügung vornehmen. Auf Wunsch des Nutzers bzw. der Nutzerin könnte der ISP aber eine Beschränkung des Datenverkehrs insoweit vornehmen, als der:die Nutzer:in beim Besuch einer bestimmten Webseite auf einen Warnhinweis umgeleitet wird. Entsprechende Kundenprodukte (Internetsicherheitsprodukte) werden bereits angeboten.

Compliance Anforderungen

Die ethischen Anforderungen an einen skalierten Betrieb des FSD werden im Folgenden entlang der einschlägigen IEEE AI Ethics Standards (7000er-Reihe) analysiert. Herangezogen werden die Standards IEEE 7000 (ethischer Systemdesign-Prozess), 7001 (Transparenz), 7002 (Datenschutz), 7003 (algorithmische Verzerrungen), 7009 (Fail-Safe-Design) und 7010 (Wohlergehen). Je Standard werden die relevanten Anforderungen sowie der aktuelle Umsetzungsstand des FSD dargestellt; die Dimension Rechenschaftspflicht wird als Querschnittsthema unter IEEE 7000 behandelt.

IEEE 7000 – Model Process for Addressing Ethical Concerns During System Desing

IEEE 7000 liefert einen übergreifenden Prozess, um betroffene Stakeholder zu identifizieren, relevante Werte abzubilden und daraus messbare, testbare ethische Anforderungen abzuleiten. Mit steigender Nutzung und breiteren Einsatzszenarien gewinnen klare Verantwortlichkeiten, transparente Prozesse und wirksame Mechanismen zur Fehlerkorrektur an Bedeutung. Der FSD verfügt bereits über entsprechende Prozesse – etwa die Möglichkeit für Shop-Betreiber:innen, fehlerhafte Bewertungen direkt zu melden, mit einer angestrebten Bearbeitungszeit von maximal 72 Stunden. Für den skalierten Betrieb sind darüber hinaus klar definierte Zuständigkeiten für Betrieb, Qualitätssicherung, Datenpflege und technische Weiterentwicklung erforderlich. Mit zunehmender Systemgröße steigt der Bedarf an dokumentierten Prozessen und Governance-Strukturen, um Verantwortlichkeiten nachvollziehbar zu

regeln und eine effiziente Bearbeitung von Beschwerden und Korrekturanfragen sicherzustellen.

I EEE 7001 – Transparency of Autonomous Systems

IEEE 7001 definiert abgestufte, messbare Transparenzanforderungen für unterschiedliche Stakeholder-Gruppen und ist für das behördliche Szenario besonders relevant.

Nutzer:innen müssen nachvollziehen können, wie Bewertungen zustande kommen, welche Datenquellen verwendet werden und welche Grenzen das System aufweist. Der FSD verfolgt dazu einen gestuften Transparenzansatz: Es wird kenntlich gemacht, ob eine Bewertung auf einer KI-Analyse oder auf manuell kuratierten Informationen beruht, ergänzt um Kontextinformationen etwa aus Impressumsprüfungen. Funktionswiese und Limitationen sollten zusätzlich in einer Model Card dokumentiert werden.

Gleichzeitig besteht in der Betrugsprävention ein Spannungsfeld zwischen Transparenz und Sicherheit: Eine vollständige Offenlegung der Erkennungsmechanismen könnte von kriminellen Akteur:innen genutzt werden, um die Detektion gezielt zu umgehen. Detaillierte technische Informationen werden daher nur ausgewählten Partnerorganisationen bereitgestellt, während öffentlich zugängliche Informationen auf die wesentlichen Funktionsprinzipien beschränkt bleiben. Eine zusätzliche Herausforderung ergibt sich aus der begrenzten Erklärbarkeit der Modelle: Analysen mittels LIME und SHAP zeigen, dass Risikobewertungen nicht auf einzelnen Merkmalen beruhen, sondern auf dem Zusammenspiel zahlreicher Faktoren. Diese Erkenntnisse werden für interne Expert:innen aufbereitet und unterstützen die Nachvollziehbarkeit der Bewertungen.

IEEE 7002 – Data Privacy Process

IEEE 7002 stützt den Privacy-by-Design Ansatz prozessual ab und ist insbesondere für das kommerzielle Szenario relevant. Der folgt dem Grundsatz strikter Datenminimierung: Auch bei aktivierter Echtzeitanalyse werden keine Informationen zum Such-, Surf- oder Einkaufsverhalten erfasst und keine Nutzungsprofile gebildet. Verarbeitet werden ausschließlich Website-URLs und der Zeitpunkt ihrer Analyse, sofern die Domain noch nicht in der Datenbank vorhanden ist. Zur Abwehr von Angriffen und DDoS-Attacken werden Serveranfragen samt IP-Adressen ausschließlich zu Sicherheitszwecken in Log-Files erfasst, eine Zusammenführung mit anderen Datenquellen oder eine Weitergabe an Dritte findet nicht statt. Da die Risikobewertung lokal auf dem Endgerät erfolgt, können selbst bei einem Angriff auf Backend-Systeme keine individuellen Nutzerdaten entnommen werden.

Neben dem Schutz der Nutzer:innendaten ist die Datenqualität ein wesentlicher Faktor: Die Leistungsfähigkeit des FSD hängt von Qualität und Aktualität der Trainingsdaten ab,

die durch das Team der Watchlist Internet kontinuierlich kuratiert und qualitätsgesichert werden. Die Datenbasis umfasst aktuell mehrere zehntausend als betrügerisch bzw. vertrauenswürdig klassifizierte Online-Shops.

IEEE 7003 – Algorithmic Bias Considerations

IEEE 7003 adressiert systematische Verzerrungen in KI-Systemen hinsichtlich Nichtdiskriminierung und Fairness. Bei breitem Einsatz des FSD muss sichergestellt sein, dass Bewertungen keine bestimmten Gruppen oder legitimen Angebote benachteiligen. Im Bereich der Betrugserkennung besteht das Risiko, dass Trainingsdaten bestimmte Betrugsarten oder legitime Geschäftsmodelle unzureichend abbilden. Der FSD begegnet dem durch menschliche Qualitätssicherung, in der Bewertungen überprüft und Fehleinschätzungen korrigiert werden.

Offen bleibt, inwieweit die Trainingsdaten selbst Verzerrungen enthalten: Mit zunehmender Skalierung wird eine regelmäßige Evaluierung von Datenbasis und Modellperformance entsprechend wichtiger, um Biases frühzeitig zu erkennen und zu minimieren.

IEEE 7009 – Fail-Safe Design of Autonomous and Semi-Autonomous Systems

IEEE 7009 definiert Methoden und Nachweise für das sichere Ausfallverhalten autonomer Systeme. KI-Systeme müssen auch – und insbesondere – bei steigender Nutzung zuverlässig, präzise und widerstandsfähig gegenüber Störungen und Angriffen bleiben. Der FSD verfügt über eine solide technische Grundlage; in der aktuellen Implementierung erreicht das System eine Genauigkeit von 90,38%. Besonderes Augenmerk liegt dabei auf der Reduktion von False-Positive-Raten, da fehlerhafte Warnungen für legitime Online-Shops wirtschaftliche Schäden und Reputationsverluste verursachen können. Das mehrstufige Warnsystem mit unterschiedlichen Konfidenzniveaus ermöglicht eine differenziert Risikobewertung.

Mit der Skalierung steigen die Anforderungen an Verfügbarkeit, Performance und Ausfallsicherheit, zugleich erhöht sich die Attraktivität des Systems als Angriffsziel. Neben klassischen Cyberangriffen wie DDoS-Attacken sind daher auch Angriffe auf die KI-Komponenten zu berücksichtigen – etwa Manipulationsversuche oder Reverse-Engineering mit dem Ziel, fehlerhafte Bewertungen zu erzeugen. Für technische Ausfälle ist ein Fail-Safe-Verhalten vorgesehen. Dabei wird ein entsprechender Hinweis ausgespielt und eine Anleitung zur manuellen Prüfung bereitgestellt. Fehlerhafte Bewertungen können von Nutzer:innen wie betroffenen Shop-Betreiber:innen gemeldet werden. Bei steigenden Fallzahlen ist sicherzustellen, dass diese Prozesse nach wie vor effizient funktionieren und ausreichend personelle Ressourcen bereitstehen.

IEEE 7010 – Wellbeing Metrics for Ethical AI

IEEE 7010 dient der Bewertung der Auswirkungen von KI-Systemen auf das menschliche Wohlergehen. Durch die Skalierung des FSD kann der gesellschaftliche Nutzen deutlich erweitert werden: Durch die automatisierte Erkennung betrügerischer Online-Shops lassen sich finanzielle und emotionale Schäden für Konsument:innen reduzieren und das Vertrauen in digitale Angebote stärken – insbesondere für Personen mit erhöhtem Betrugsrisiko. Zugleich gewinnt die Qualität der Risikobewertungen an Bedeutung, um unbegründete Warnungen und negative Auswirkungen auf legitime Unternehmen zu vermeiden. Die gesellschaftliche Wirkung hängt damit wesentlich von einer verantwortungsvollen Ausgestaltung und kontinuierlicher Qualitätssicherung ab.

Auch ökologische Aspekte zählen zum Wohlergehen: Eine steigende Zahl von Nutzer:innen und Analysen führt zu höherem Ressourcenbedarf. Der FSD wirkt dem aktuell durch effiziente Datenverarbeitung entgegen – bereits bekannte Websites werden nicht erneut analysiert und die Nutzung externer Black- und Whitelists ermöglicht viele Bewertungen ohne rechenintensive Prozesse. Diese Daten- und Ressourcenminimierung trägt zu einem möglichst ressourcenschonenden Betrieb bei wachsender Nutzung bei.

Zusammenfassend adressiert der FSD die zentralen Anforderungen der IEEE-7000er-Standards bereits in seiner aktuellen Implementierung in weiten Teilen; insbesondere bei Datenschutz (lokale Verarbeitung, strikte Datenminimierung, keine Profilbildung), Transparenz (gestufter Ansatz) und Fail-Safe-Verhalten (definiertes Ausfallverhalten mit Hinweis und Anleitung zur manuellen Prüfung, Meldeprozesse für Fehlbewertungen, mehrstufiges Warnsystem). Offen bleiben demgegenüber die systematische Evaluierung der Trainingsdaten auf Verzerrungen (IEEE 7003) sowie der Ausbau von Governance-Strukturen, dokumentierten Verantwortlichkeiten und ausreichenden personellen Ressourcen. Diese Punkte sind die entscheidenden Voraussetzungen dafür, dass Qualität und gesellschaftlicher Nutzen bei wachsender Nutzung erhalten bleiben.

Fazit

Die ausgearbeitete Sicherheitsstrategie zeigt, dass der skalierte Betrieb des Fake-Shop Detectors technisch machbar, rechtlich gestaltbar und ethisch vertretbar ist – allerdings nur unter klaren Voraussetzungen.

Die Analyse der beiden Skalierungsszenarien ergibt unterschiedliche, teils gegenläufige Anforderungsprofile: Der behördliche Einsatz verlangt vor allem Nachvollziehbarkeit und Erklärbarkeit der Bewertungen, der kommerzielle Einsatz über Internet Service Provider primär Verfügbarkeit, Skalierbarkeit und den Erhalt des bestehenden Datenschutzversprechens. Die modulare, event-basierte Architektur, die Absicherung der Schnittstellen und die spezifizierten Belastungs- und Angriffstests bilden dafür eine

tragfähige technische Grundlage; ihre tatsächliche Belastbarkeit ist durch die geplanten Tests noch nachzuweisen.

Rechtlich bestehen für keines der beiden Szenarien unüberwindbare Hürden. Das Haftungsrisiko aus § 1330 ABGB ist im behördlichen Szenario gering, im ISP-Szenario vertraglich beherrschbar; eine Einstufung als Hochrisiko-KI-System nach der KI-VO ist nach derzeitiger Einschätzung in beiden Szenarien nicht zu erwarten, sollte angesichts der Auslegungsspielräume von Art. 6 Abs. 3 KI-VO jedoch dokumentiert begründet und laufend überprüft werden. Im ISP-Szenario setzen TKG 2021 und Netzneutralitäts-VO eine ausdrückliche Einwilligung der Nutzer:innen voraus.

Als zentrale offene Punkte für die Umsetzung verbleiben: (1) die Skalierung der manuellen Qualitätssicherung, deren personelle Grenzen derzeit das größte operative Risiko darstellen; (2) die Erklärbarkeit der Modellbewertungen im behördlichen Kontext, die über den reinen Risikoscore hinaus entwickelt werden muss; (3) die Klärung, ob die lokale, gerätebasierte Bewertung bei einer Integration auf ISP-Ebene erhalten bleibt; und (4) die regelmäßige Evaluierung der Trainingsdaten auf systematische Verzerrungen. Diese Punkte sind im Umsetzungsfahrplan zu priorisieren, da sie über die Tragfähigkeit beider Szenarien entscheiden.

D3.1. Installation eines laufenden Monitorings zur Erkennung von Zwischenfällen (Software)

Ein Monitoring der Systemumgebung und Clusterinfrastruktur hinsichtlich der in Task 2.2. identifizierten Risiken und Metriken (z.B. Clusterauslastung, Modell-Degradation im zeitlichen Verlauf, etc.) wurde fristgerecht auf der FSD-Infrastruktur ausgerollt. Das Ergebnis wurde in der Endpräsentation demonstriert.

Technische Dokumentation

1. fsd-monitor

1.1. FSD Monitoring Stack

Ein zentraler Monitoring-Stack für die Überwachung der FSD-Plattform mit Prometheus und Grafana.

Der Stack sammelt Metriken verschiedener Komponenten und stellt vorkonfigurierte Dashboards zur Visualisierung von Infrastruktur-, Messaging- und Anwendungsmetriken bereit.

1.2. Features

- Grafana für Visualisierung und Dashboarding
- Prometheus für Metrik-Erfassung und Speicherung
- Automatisches Provisioning von Dashboards und Datenquellen
- Containerisierte Bereitstellung mit Docker Compose
- Vorkonfigurierte Dashboards für:
  - MongoDB
  - Apache Pulsar
  - FSD Crawler
  - FSD Classifier
  - Infrastruktur-Metriken

1.3. Projektstruktur

```
./ .2 docker-compose.yaml
├── grafana.ini
├── .env
├── prometheus/
│   └── prometheus.yml
└── grafana/
    ├── dashboards/
    │   ├── mongodb-metrics.json
    │   ├── pulsar-broker-metrics.json
    │   ├── pulsar-bookie-metrics.json
    │   ├── pulsar-functions-metrics.json
    │   ├── pulsar-messaging-metrics.json
    │   ├── pulsar-topic-metrics.json
    │   ├── fsd-crawler.json
    │   ├── fsd-classifier-workers.json
    │   └── fsd-classifier-infrastructure.json
    └── provisioning/
        ├── dashboards/
        └── datasources
```

Abbildung 5: Projektstruktur von Grafana/Prometheus

1.4. Voraussetzungen

- Docker
- Docker Compose

Überprüfen:

```
docker --version
docker compose version
```

1.5. Installation

1.5.1. 1. Repository klonen

```
git clone <repository-url>
cd monitor
```

1.5.2. 2. Umgebungsvariablen konfigurieren

Falls noch keine `.env` Datei existiert:

```
cp copy.env .env
```

Anschließend die gewünschten Grafana-Zugangsdaten und Umgebungswerte anpassen.

1.5.3. 3. Monitoring Stack starten

```
docker compose up -d
```

1.5.4. 4. Status prüfen

```
docker compose ps
```

1.6. Zugriff auf die Dienste

1.6.1. Grafana

Standardmäßig erreichbar unter:

`http://localhost:3000`

Anmeldung mit den in der `.env` Datei definierten Zugangsdaten.

1.6.2. Prometheus

Standardmäßig erreichbar unter:

`http://localhost:9090`

1.7. Überwachte Systeme

Aktuell sind folgende Prometheus Targets konfiguriert:

Job	Beschreibung
`prometheus`	Prometheus Self-Monitoring
`mongodb_exporter`	MongoDB-Metriken
`url-frontier`	URL Frontier Service
`pushgateway`	Push-basierte Metriken

Die Targets können in `prometheus/prometheus.yml` erweitert oder angepasst werden.

1.8. Dashboards

Die Dashboards werden beim Start automatisch durch Grafana provisioniert.

1.8.1. Verfügbare Dashboards

1.8.2. MongoDB Metrics

Überwachung von:

- Speicherverbrauch
- Verbindungen
- Operationen
- Performance-Kennzahlen

1.8.3. Pulsar Metrics

Dashboards für:

- Broker
- BookKeeper
- Topics
- Messaging
- Functions

1.8.4. FSD Crawler

Visualisierung von:

- Crawl-Aktivitäten
- Verarbeitungsgeschwindigkeit
- Fehlern
- Queue-Status

1.8.5. FSD Classifier

Visualisierung von:

- Worker-Auslastung
- Infrastruktur-Metriken
- Verarbeitungsstatus

1.9. Konfiguration

1.9.1. Prometheus

Die Konfiguration befindet sich unter:

`prometheus/prometheus.yml`

Nach Änderungen:

```
docker compose restart prometheus
```

1.9.2. Grafana

Die Grafana-Konfiguration befindet sich unter:

```
grafana.ini
```

Nach Änderungen:

```
docker compose restart fsd-grafana
```

1.10. Persistente Daten

Folgende Docker Volumes werden verwendet:

Volume	Zweck
`grafana_data`	Grafana Datenbank, Dashboards und Einstellungen
`prometheus-data`	Prometheus Zeitreihendaten

1.11. Logs

Grafana Logs:

```
docker logs fsd-grafana
```

Prometheus Logs:

```
docker logs prometheus
```

Live verfolgen:

```
docker compose logs -f
```

1.12. Update

Container aktualisieren:

```
docker compose pull
docker compose up -d
```

1.13. Stoppen

```
docker compose down
```

Mit Entfernung aller gespeicherten Monitoring-Daten:

```
docker compose down -v
```

2. fsd-clustering-frontend

2.1. FSD Clustering Frontend

Interaktives Visualisierungs-Frontend zur Untersuchung der Clustering-Ergebnisse aus dem Fake-Shop Detector (FSD)-Ökosystem. Erstellt mit Vue 3 + Vite + TypeScript, gestaltet mit PrimeVue und TailwindCSS und basierend auf Chart.js.

2.2. Überblick

Das Frontend ermöglicht es Analysten und Ermittlern:

2.3. Cluster nach Algorithmus erkunden

Unterstützt drei Clustering-Algorithmen:

- KMeans
- Agglomerative Clustering
- HDBSCAN

Wechseln Sie den Algorithmus über ein Dropdown-Menü, um Ihre Daten sofort neu zu gruppieren.

2.4. Suchen u. Markieren

Das Frontend unterstützt derzeit drei Clustering-Verfahren:

- URLs in allen Clustern mit einem Live-Filter durchsuchen.
- Übereinstimmende Begriffe werden zur schnellen Erkennung gelb hervorgehoben.

2.5. Den Cluster einer URL ermitteln

- Gib einen Teil einer URL ein, um herauszufinden, zu welchem Cluster bzw. welchen Clustern sie gehört.
- Die Ergebnisse werden zur leichteren Erkennung in blauen Karten hervorgehoben.
- Wird keine Übereinstimmung gefunden, wirst du durch eine Toast-Benachrichtigung darauf hingewiesen.

2.6. Benutzerdefinierte Cluster

- Erstellen Sie Ihre eigenen benannten Cluster, unabhängig von der Ausgabe des Algorithmus.

- Fügen Sie Elemente aus bestehenden Clustern zu Ihren manuellen Gruppen hinzu.
- Jede URL kann zu mehreren benutzerdefinierten Clustern gehören.
- Cluster zeigen die Anzahl der Elemente und die dazugehörigen URLs an.

2.7. Schnellstart

Du kannst dieses Projekt lokal mit Node 20+ oder über eine reproduzierbare Conda-Umgebung ausführen. Für konsistente Installationen wird der Conda-Workflow empfohlen.

2.7.1. Repository klonen

```
git clone https://mal2git.x-t.at/fsd-v3/fsd-clustering.git
cd fsd-clustering-frontend
```

2.7.2. Conda-Umgebung erstellen

Eine vorkonfigurierte Datei environment.yml ist enthalten.

```
name: fsd-clustering-frontend
channels:
- conda-forge
dependencies:
- nodejs=20
- git
```

Umgebung erstellen und aktivieren:

```
conda env create -f environment.yml
conda activate fsd-clustering-frontend
```

2.7.3. Abhängigkeiten installieren

```
npm install
```

Wenn du VirtualBox oder WSL2 verwendest, richte freigegebene Ordner ein, um Symlink-Fehler zu vermeiden:

```
npm install --no-bin-links
```

2.7.4. Entwicklungsserver starten

Standard:

```
npm run dev
```

VirtualBox / WSL – Freigegebene Ordner Dateiabfrage aktivieren:

```
CHOKIDAR_USEPOLLING=1 npm run dev
```

Beispielausgabe:

```
VITE v7.x ready in 2s
[OK] loaded config from vite.config.ts
-> Local: http://localhost:5173/
-> Network: http://192.168.x.x:5173/
```

Öffne http://localhost:5173 in deinem Browser.

2.8. Produktionspaket erstellen und in der Vorschau anzeigen

Erstelle eine Produktionsversion:

```
npm run build
```

Lokale Vorschau:

```
npm run preview
```

Öffne anschließend http://localhost:4173, um den statischen Build zu testen.

2.9. Beispiel für das Format der Eingabedaten

Die App geht davon aus, dass die Datei public/cluster_data.json vorhanden ist und gültige Clustering-Ergebnisse enthält.

Beispiel:

```
{
      "meta": {
            "cluster_run_id": 1,
            "n_input_samples": 100,
            "timestamps": {
                  "clustering_started_at": "2025-02-15T10:12:00Z",
                  "clustering_completed_at": "2025-02-15T10:14:12Z"
            },
            "notes": ["Example clustering run"]
      },
      "metrics": [
      {
            "algorithm": "kmeans",
            "n_clusters_param": 8,
            "silhouette_score": 0.61,
            "davies_bouldin_score": 0.88,
            "n_clusters_formed": 8
```

```
      }
      ],
      "data": [
      {
            "row_id": 1,
            "url": "https://example-shop.com",
            "label_kmeans": 0,
            "label_agglomerative": 1,
            "label_hdbscan": null
      }
      ]
}
```


2.10. Mit Docker ausführen (optional)

Das Frontend lokal erstellen:

```
conda activate fsd-clustering-frontend
npm run build
```

```
docker build -t fsd-clustering-frontend:latest .
docker run --rm -p 8080:80 fsd-clustering-frontend:latest
```

Öffne anschließend http://localhost:8080.

2.11. Vollständige Befehlsübersicht

Schritt	Befehl	Beschreibung
1.	`conda env create -f environment.yml`	Umgebung erstellen.
2.	`conda activate fsd-clustering-frontend`	Conda-Umgebung aktivieren.
3.	`npm install` oder `npm install –no-bin-links`	Abhängigkeiten installieren.
4.	`CHOKIDAR_USEPOLLING=1 npm run dev`	Entwicklungsserver starten.
5.	`npm run build`	Produktions-Bundle erstellen.
6.	`npm run preview`	Vorschau des Produktions-Builds.

Schritt	Befehl	Beschreibung
7.	`docker build ... && docker run ...`	Erstellen und Ausführen über Docker.

D4.1. Back-End entspricht TRL 8 (Software)

Die Systemarchitektur wurde laut Technischer Roadmap fristgerecht umgesetzt und live geschaltet. Das Ergebnis wurde in der Endpräsentation demonstriert.

Technische Dokumentation

1. Einleitung

Fake-Shop Detector-Dokumentation Version 1.0 in LaTeX und in ein Wort Dokument konvertiert.

Die Dokumentation ist aus den in GitLab vorhandenen Repositories generiert und bietet eine vollständige technische Übersicht über die Komponenten des Fake-Shop Detectors.

Jedes Repository im Projekt wird in einem separaten .tex-Dokument dargestellt, um Änderungen und Versionierungen nachvollziehbar zu machen.

2. Übersicht über die Komponenten

2.1. Systemübersicht

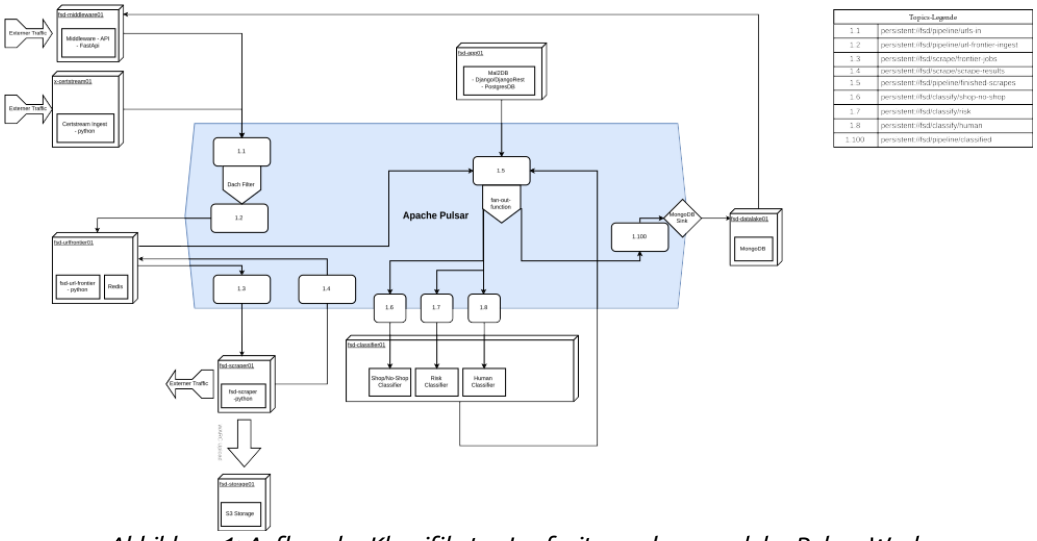

Abbildung 1: Aufbau der Klassifikator-Laufzeitumgebung und des Pulsar-Workers.

2.2. Server und Pakete

- **fsd-middleware01**
  - `fsd-middleware`
- **x-certstream01**
  - `x-certstream`
- **x-pulsar-br01**
  - `DachFilter`
  - `FanOutFunction`
- **fsd-urlfrontier01**
  - `fsd-urlfrontier`
  - `redis`
- **fsd-scraper01**
  - `fsd-browsercrawl`
- **fsd-storage01**
  - `S3 Storage`
- **fsd-classifier01**
  - `fsd-classifiers-predict`
  - siehe Kapitel fsd-classifiers-predict.
- **fsd-datalake01**
  - `mongod.service` (Systemd)
  - `fsd-mongopy-sink`
  - Der Sink wird zukünftig als Apache-Pulsar-Function umgesetzt und entspricht derzeit noch nicht der Architekturzeichnung.
- **fsd-app01**
  - `fsd-portal-api`
  - `fsd-portal-ui`

2.3. Themenschemata:

Das jeweilige JSON zum Datenschema liegt immer unter der Topic Referenz angehängt von der Version des Schemas. e.g.: 1.1/0.json

- **1.1** = Topic-Referenz
  - `0.json` = Versionsnummer (`<Version>.json`)

Nr.	Apache Pulsar Topic
1.1	`persistent://fsd/pipeline/urls-in`
1.2	`persistent://fsd/pipeline/url-frontier-ingest`
1.3	`persistent://fsd/scrape/frontier-jobs`
1.4	`persistent://fsd/scrape/scrape-results`
1.5	`persistent://fsd/pipeline/finished-scrapes`
1.6	`persistent://fsd/classify/shop-no-shop`
1.7	`persistent://fsd/classify/risk`

Nr.	Apache Pulsar Topic
1.8	persistent://fsd/classify/human
1.100	persistent://fsd/pipeline/classified

2.4. Schemas einspielen

```
/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin topics create-partitioned-topic persistent://fsd/pipeline/urls-in -p 3
/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin --admin-url=http://x-pulsar01.x:8080 schemas upload --filename /data/pulsar/apache-pulsar/schemas/1.1/0.json persistent://fsd/pipeline/urls-in

/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin topics create-partitioned-topic persistent://fsd/pipeline/url-frontier-ingest -p 3
/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin --admin-url=http://x-pulsar01.x:8080 schemas upload --filename /data/pulsar/apache-pulsar/schemas/1.2/0.json persistent://fsd/pipeline/url-frontier-ingest

/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin topics create-partitioned-topic persistent://fsd/scrape/frontier-jobs -p 3
/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin --admin-url=http://x-pulsar01.x:8080 schemas upload --filename /data/pulsar/apache-pulsar/schemas/1.3/0.json persistent://fsd/scrape/frontier-jobs

/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin topics create-partitioned-topic persistent://fsd/scrape/scrape-results -p 3
/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin --admin-url=http://x-pulsar01.x:8080 schemas upload --filename /data/pulsar/apache-pulsar/schemas/1.4/0.json persistent://fsd/scrape/scrape-results

/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin topics create-partitioned-topic persistent://fsd/pipeline/finished-scrapes -p 3
/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin --admin-url=http://x-pulsar01.x:8080 schemas upload --filename /data/pulsar/apache-pulsar/schemas/1.5/0.json persistent://fsd/pipeline/finished-scrapes

/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin topics create-partitioned-topic persistent://fsd/classify/shop-no-shop -p 3
/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin --admin-url=http://x-pulsar01.x:8080 schemas upload --filename /data/pulsar/apache-pulsar/schemas/1.6/0.json persistent://fsd/classify/shop-no-shop
```

```
/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin topics create-partitioned-topic persistent://fsd/classify/risk -p 3
/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin --admin-url=http://x-pulsar01.x:8080 schemas upload --filename /data/pulsar/apache-pulsar/schemas/1.7/0.json persistent://fsd/classify/risk

/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin topics create-partitioned-topic persistent://fsd/classify/human -p 3
/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin --admin-url=http://x-pulsar01.x:8080 schemas upload --filename /data/pulsar/apache-pulsar/schemas/1.8/0.json persistent://fsd/classify/human

/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin topics create-partitioned-topic persistent://fsd/pipeline/classified -p 3
/data/pulsar/apache-pulsar-4.1.1/bin/pulsar-admin --admin-url=http://x-pulsar01.x:8080 schemas upload --filename /data/pulsar/apache-pulsar/schemas/1.100/0.json persistent://fsd/pipeline/classified
```

2.5. Server und Netzwerk

Dieses Kapitel beschreibt die Infrastruktur des FSD-Systems. Neben den einzelnen Servern werden deren Hardwareausstattung, Standort, Netzwerkzuordnung sowie die darauf betriebenen Dienste dokumentiert.

2.6. Serverübersicht

2.6.1. Fsd-app01

Eigenschaft	Wert
Standort	Rechenzentrum Linz
VLAN	2059
RAM	8 GB
CPU	4 Kerne
Speicher	30 GB
Beschreibung	• Webserver (Nginx) • shop-check-dev.malzwei.at • Docker • Grafana (Mit Prometheus und Loki) • dekaf (GUI für Apache Pulsar)

Eigenschaft	Wert
	• Postgresql • Memcached

2.6.2. fsd-urlfrontier01

Eigenschaft	Wert
Standort	Rechenzentrum Linz
VLAN	2059
RAM	4 GB
CPU	2 Kerne
Speicher	30 GB

Beschreibung	• Docker
	  ○ URL Frontier
	  ○ Redis
	  ○ Prometheus Pushgateway
	• Fanout Service

2.6.3. fsd-datalake01

Eigenschaft	Wert
Standort	Rechenzentrum Linz
VLAN	2059
RAM	8 GB
CPU	4 Kerne
Speicher	110 GB
Beschreibung	• Docker ○ MongoDB Exporter • MongoDB

2.6.4. x-certstream01

Eigenschaft	Wert
Standort	Rechenzentrum Linz
VLAN	2053
RAM	2 GB
CPU	2 Kerne
Speicher	50 GB
Beschreibung	• Certstream Server

2.6.5. fsd-scraper01

Eigenschaft	Wert
Standort	Hetzner
RAM	8 GB
CPU	4 Kerne
Speicher	80 GB
Beschreibung	• Docker • Crawler Worker • Prometheus Pushgateway

2.6.6. fsd-classifier01

Eigenschaft	Wert
Standort	Hetzner
RAM	16 GB
CPU	8 Kerne
Speicher	80 GB

Beschreibung	• Docker
	  ○ classifiers-predict-api
	  ○ classifiers-predict-pulsar-worker
	  ○ cadvisor

Eigenschaft	Wert
	○ Grafana Alloy (Mit Prometheus Node-exporter)

2.6.7. fsd-middleware01

Eigenschaft	Wert
Standort	Hetzner
RAM	8 GB
CPU	4 Kerne
Speicher	80 GB
Beschreibung	○ Webserver(Nginx) ○ api.infra.fakeshop.at ○ fsd-middleware Service

2.6.8. fsd-storage01

Eigenschaft	Wert
Standort	Hetzner
Resourcen	○ Bucket fsd-screenshots: 30GB (Stand: 29.06.2026) ○ Bucket fsd-warcfiles: 260GB (Stand: 29.06.2026)
Beschreibung	S3 Buckets für Speicherung von WARC Dateien und Screenshots

2.7. Apache-Pulsar-Cluster

Eigenschaft	Wert
Standort	Rechenzentrum Linz und Wien
Beschreibung	○ Cluster besteht aus 9 virtuellen Maschinen (3 Broker, 3 Bookkeeper, 3 Zookeeper) ○ VMs werden in unseren Rechenzentren betrieben und sind netzwerkmäßig getrennt in eigenen VLANs ○ Cluster ist nicht öffentlich erreichbar und nur notwendige Services haben Zugriff

2.8. Netzwerkplan

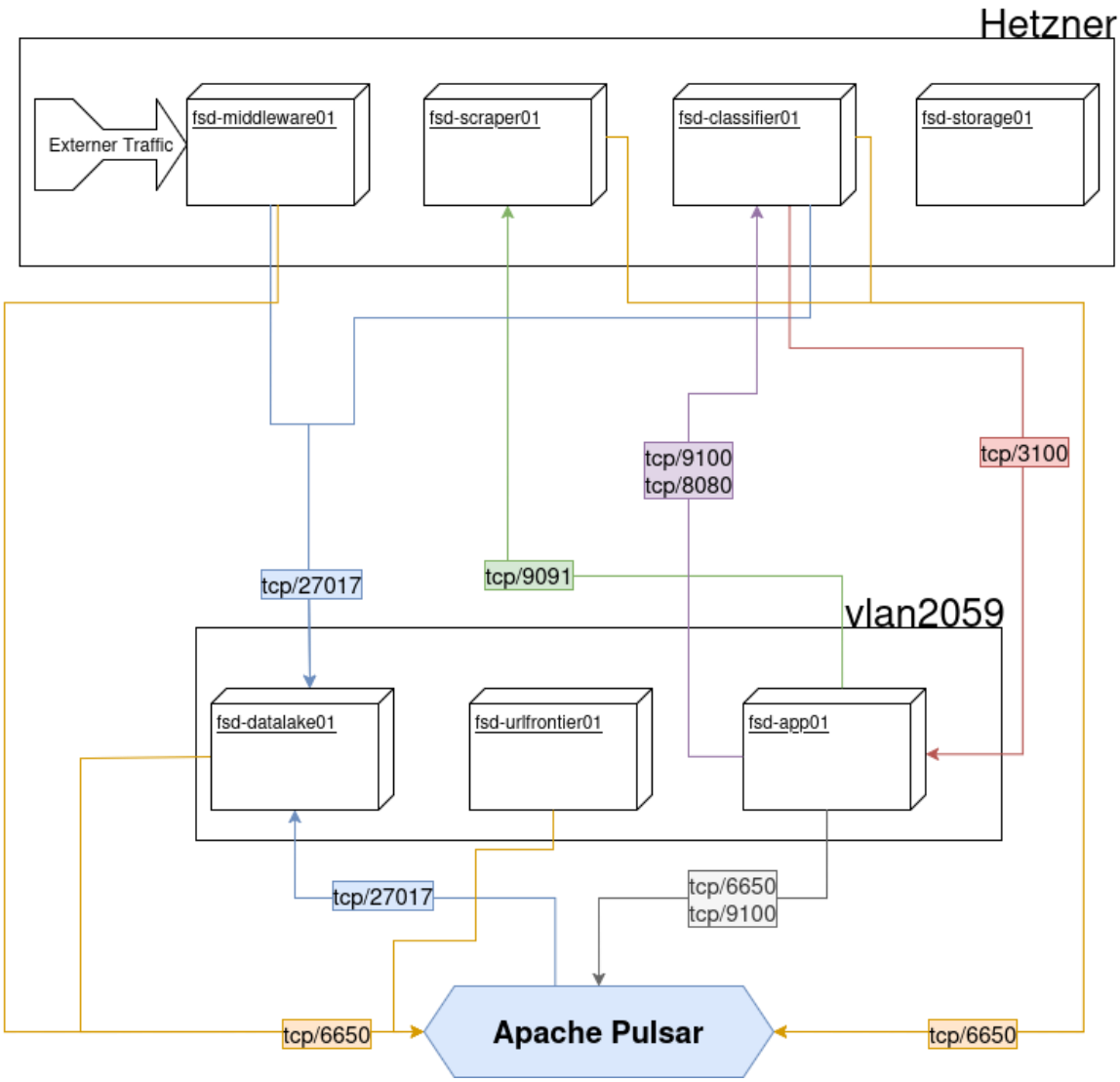

Abbildung 2: Logischer Netzwerkaufbau der FSD-Infrastruktur

2.9. Services und Komponenten

2.9.1. fsd-middleware (FastAPI)

Eigenschaft	Beschreibung
Package/Pfad	fsd-middelware
Technologie	Python, FastAPI, NGINX
Beschreibung	Dieser Service nimmt URL-Anfragen entgegen, prüft diese

Eigenschaft	Beschreibung
	gegen eine Datenbank und spielt neue URLs in die Verarbeitungs-Pipeline ein. Zusätzlich verwaltet er regelmäßig aktualisierte Filterlisten, die über einen Memory based cache ausgeliefert werden.

2.9.2. fsd-certstream-server

Eigenschaft	Beschreibung
Package	x-certstream
Technologie	https://certstream.calidog.io/
Beschreibung	CertStream ist ein Intelligence-Feed, der uns Echtzeit-Updates aus dem Certificate Transparency Log-Netzwerk liefert. Damit können wir CertStream als Baustein verwenden, um auf neu ausgestellte Zertifikate in Echtzeit zu reagieren.

2.9.3. fsd-certstream-ingest

Eigenschaft	Beschreibung
Package	fsd-certstream-ingest
Technologie	Python
Beschreibung	Dieses Skript empfängt Echtzeit-Zertifikatsdaten von CertStream und sendet sie als JSON-Nachrichten an ein Pulsar-Topic. Jede Domain aus den Zertifikaten wird auf das Format https:// normalisiert und mit Metadaten wie not_before versehen. Die Daten können dann von anderen

Eigenschaft	Beschreibung
	Systemen weiterverarbeitet werden.

2.9.4. fsd-apache-pulsar-filter-function

Eigenschaft	Beschreibung
Package	fsd-apache-pulsar-filter-function
Technologie	Java
Beschreibung	Diese Pulsar Funktion filtert alle Domains die nicht mit .de .at oder .ch enden, außer reported_by ist Middleware oder Plugin. Dann reagiert dieser Filter nicht.

2.9.5. fsd-urlfrontier

Eigenschaft	Beschreibung
Package	`fsd-url-frontier`
Technologie	Python, Redis, Docker
Beschreibung	Der URL-Frontier empfängt Seeds von der Außenwelt. Seeds werden über pulsar eingemeldet und in einem sorted set in redis priorisiert. Sobald der Seed an der Reihe ist wird er als scrap-job an die crawler gegeben. Die Scraper melden den Status zurück an den Frontier, welcher entweder bis zu 5 mal erneut versucht oder das Ergebnis an die restliche Pipeline sendet. Der Frontier ist ein "Monolit" und besteht aus 3 Modulen: • Ingest: Empfängt URLS und setzt sie in Queues in redis. • Scheduler: Verteilt scrap jobs an die Crawler Agents

Eigenschaft	Beschreibung
	Status: Empfängt scrap results von den Crawler Agents

2.9.6. fsd-fanout

Eigenschaft	Beschreibung
Package	fsd-fanout
Technologie	Python
Beschreibung	Dieses Skript ist ein Pulsar Message Router, der eingehende Klassifizierungsjobs verteilt. Es abonniert ein Apache Pulsar Topic und empfängt dort Nachrichten vom Typ ClassificationJob, die jeweils eine URL und eine Liste von Klassifizierern enthalten. Je nachdem ob noch Klassifizierer in der Liste vorhanden sind, wird die Nachricht an den nächsten zuständigen Classifier-Topic weitergeleitet – oder, wenn die Liste leer ist, direkt an den „classified"-Topic als finalen Schritt.

2.9.7. fsd-scraper

Eigenschaft	Beschreibung
Package	fsd-scraper
Technologie	Python, Docker, Playwright
Beschreibung	fsd-crawler ist ein skalierbarer Web-Crawler, der besuchte Seiten inklusive aller Assets als komprimierte WARC-Archive speichert. Jobs werden über Apache Pulsar verteilt – ein

Eigenschaft	Beschreibung
	Producer schickt Crawl-Anfragen, ein oder mehrere Worker erledigen das eigentliche Crawlen mit Playwright (Chromium) und veröffentlichen die Ergebnisse wieder auf Pulsar. Die fertigen WARC-Dateien landen entweder lokal oder werden optional in Minio/S3 hochgeladen, und können anschließend direkt im Browser über replayweb.page wiedergegeben werden. Laufzeit-Metriken wie Erfolgsrate, Crawl-Dauer und WARC-Größe werden per Pushgateway an Prometheus gemeldet und lassen sich in Grafana visualisieren.

2.9.8. Fsd-portal

Eigenschaft	Beschreibung
Packages	fsd-portal-ui fsd-portal-api
Technologie	Python, Django REST, VueJS
Beschreibung	Verwaltung von Webseiten für die Watchlist Internet und dem Fake-Shop Detector. Einsicht in den Verlauf einer Webseite.

2.9.9. fsd-xylem-classifier

Eigenschaft	Beschreibung
Package	fsd-xylem-classifier
Technologie	Python
Beschreibung	Abholen von bereitgestellten CSV Dateien von einem FTP Server. CSV Datei lesen und in die Datenbank

Eigenschaft	Beschreibung
	einspielen. In Bestimmten Use-Cases wird dann eine Webseite auf die Liste automatisierte Ausspielung gesetzt.

2.9.10. fsd-classifiers-predict

Eigenschaft	Beschreibung
Package	fsd-classifiers-predict
Technologie	Python, Flask, Docker
Beschreibung	Das FSD Website Classifier Repository stellt eine ML-Pipeline bereit, mit der Websites automatisch als Shop oder Nicht-Shop eingestuft und auf Fake-Shop-Risiken bewertet werden können. Es unterstützt drei Betriebsmodi – CLI, REST API und Pulsar Worker – und kann Websites entweder live scrapen, bereits gescrapte Verzeichnisse oder WARC-Archive als Eingabe verwenden. Aufgrund eines Python-Versionskonflikts zwischen dem Legacy-Classifier-Stack (Python 3.7) und dem modernen Pulsar-Worker (Python 3.12) ist das Projekt in zwei separate Komponenten aufgeteilt, die über HTTP miteinander kommunizieren. Die Klassifizierungsergebnisse werden als strukturiertes ClassificationResult-Objekt zurückgegeben, das neben den Model-Scores und Risiko-Labels auch Metadaten und

Eigenschaft	Beschreibung
	Fehlerinformationen enthält. Deployment erfolgt über Docker und Docker Compose, wobei die vortrainierten Modelle (v0.25) als Read-only-Volume eingebunden werden.

2.9.11. fsd-mongopy-sink

Eigenschaft	Beschreibung
Package	fsd-mongopy-sink
Technologie	Python
Beschreibung	fsd-mongopy-sink ist ein Apache Pulsar Consumer, der eingehende Website-Klassifizierungsergebnisse aus einem Pulsar-Topic liest und in eine MongoDB-Datenbank schreibt – dabei werden bestehende Dokumente aktualisiert oder neue angelegt. Nach jedem verarbeiteten Eintrag wird automatisch eine Aggregation ausgeführt, die Scores, Risikobewertungen und Crawler-Daten verschiedener Classifier zu einem einheitlichen, sortierbaren Website-Dokument zusammenführt. Fehlerhafte Nachrichten werden nach 5 Wiederholungsversuchen automatisch in ein Dead Letter Topic verschoben, und ein separates Skript (flatten_all_websites.py) ermöglicht das nachträgliche

Eigenschaft	Beschreibung
	Reprocessing aller bestehenden Dokumente.

2.9.12. fsd-datalake

Eigenschaft	Beschreibung
Package	MongoDB
Technologie	-
Beschreibung	MongoDB ist eine dokumentenorientierte NoSQL-Datenbank, die Daten nicht in klassischen Tabellen wie relationale Datenbanken, sondern in flexiblen JSON-ähnlichen Dokumenten (BSON) speichert. Das macht sie besonders geeignet für Anwendungsfälle wie diesen, wo die Struktur der Dokumente variieren kann – z.B. weil manche Websites Scrape-Einträge haben und andere nicht. MongoDB bietet außerdem eingebaute Unterstützung für horizontale Skalierung, schnelle Abfragen über Indizes, und mit pymongo einen gut integrierten Python-Treiber, der direkt in diesem Projekt verwendet wird.

3. apache-pulsar-filter-function

3.1. Allgemeines:

Allgemeines: Diese Funktion filtert alle Domains heraus, die nicht auf .at, .de oder .ch enden.

- Tenant: fakeshop
- namespace: scraping
- FunctionName: DomainFilter

- Die Daten in der Ausgabe sehen wie folgt aus:
  Das Eingabethema lautet persistent://fakeshop/scraping/certstream_input
- Das Ausgabethema lautet persistent://fakeshop/scraping/certstream_filtered

Die Daten in der Ausgabe sehen wie folgt aus:

```
b'{"domains":["DOMAIN"],"notBefore":UNIXTIMESTAMP,"notAfter":UNIXTIMESTAMP}
'
```

3.2. Einrichtung eines eigenständigen Apache-Pulsar-Servers:

1. curl -LO "https://www.apache.org/dyn/closer.lua/pulsar/pulsar-4.0.5/apache-pulsar-4.0.5-bin.tar.gz?action=download"
1. tar xvfz apache-pulsar-4.0.5-bin.tar.gz
2. cd apache-pulsar-4.0.5
3. bin/pulsar standalone

3.3. tenant und namespace erstellen:

1. bin/pulsar-admin tenants create fakeshop
1. bin/pulsar-admin namespaces create fakeshop/scraping

3.4. Erstelle diese Funktion:

```
sudo apt install maven
mvn package ...
```

WICHTIG: Du musst «jar» in der Konfigurationsdatei «function.yml» auf deinen spezifischen Pfad einstellen.

3.5. Funktion in tenant/namespac erstellen:

- Um diese Funktion zu erstellen, Java Version 17 verwenden.
- sudo apt install openjdk-17-jdk

3.6. Funktion erstellen

```
bin/pulsar-admin functions create --function-config-file
PATH_TO_THE_FUNCTION_YML_CONFIG_FILE
```

3.7. Funktionsdetails abrufen

- bin/pulsar-admin functions get –tenant fakeshop –namespace scraping –name DomainFilter

Hier siehst du alle Details zur Funktion, zum Beispiel, wie oft sie ausgeführt wurde.

4. fsd-browsercrawl

4.1. FSD E-Commerce crawler based on playwright

4.1.1. Config, build and installation

Config must be placed under /etc/ecommerce_crawler/settings.toml. See settings.toml.example for details

```
docker build -t crawler .
cp ./systemd/fsd\_ecommerce\_crawler.service /etc/systemd/system/
```

4.2. Manual Run

4.2.1. Run and build

```
docker compose up --build --force-recreate --scale crawler=<num-instances>
```

4.2.2. Run only

```
docker compose up --scale crawler=<num-instances>
```

5. certstream-ingest

5.1.1. Installation

Um die Python-Abhängigkeiten zu installieren, führe den folgenden Befehl aus:

```
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt...
```

5.1.2. Konfiguration

Der Dienst wird über die Datei «settings.toml» in diesem Repo konfiguriert.

```
[general]
log\_level="INFO"

[ingester]
topic="persistent://fsd/pipeline/urls-in"
producer\_name="fsd-certstream-ingest"
reported\_by="Certstream"

[certstream]
url="ws://localhost:8080/"

[pulsar]
```

```
url="pulsar://x-pulsar01.x:6650"
auth\_enabled=false

[pulsar.auth]
issuer\_url                                =                                ""
private\_key                               =                                ""
audience = ""
```

5.1.3. Laufzeit

Der Python-Dienst ist von certstream-go abhängig, das hier zu finden ist: https://mal2git.x-t.at/fsd-v3/fsd-certstream-server

```
./bin/certstream-server-go\_1.7.0\_linux\_amd64 -config certstream.yaml
```

Den Ingest-Dienst ausführen (mit aktivierter venv):

```
python ingest/main.py
```

6. Certstream Server

6.1. Certstream config

Die Standardkonfiguration von certstream befindet sich im Stammverzeichnis dieses Repos und sollte wie erwartet funktionieren.

6.2. Running

```
./bin/certstream-server-go\_1.7.0\_linux\_amd64 -config certstream.yaml
```

7. fsd-classifiers-predict

7.1. FSD Website Classifier

Dieses Repository enthält die Vorhersage-Laufzeitumgebung für den FSD-Website-Klassifikator. Das Stammprojekt umfasst die CLI und die REST-API des Klassifikators. Das separate Pulsar-Worker-Bereitstellungsprojekt befindet sich unter pulsar/ und verfügt über eine eigene, auf Worker ausgerichtete Dokumentation. Es unterstützt zwei Klassifikatorfamilien:

- `shop-no-shop`
- `fake-shop-risk`

Das Projekt kann:

- eine Zieldomäne scrapen, bereits gescrapte Websites verwenden oder ein WARC-Archiv klassifizieren
- für jeden Scrape ein .warc.gzWebarchiv unter scraped_data/_warc/ schreiben

- einen Merkmalsvektor aus den gescrapten Artefakten extrahieren
- die Website mit einem oder mehreren vortrainierten Modellen bewerten
- ein strukturiertes ClassificationResultmit Metadaten, Vorhersagen und Fehlern zurückgeben

Die derzeit bereitgestellten, eingecheckten Modellfamilien sind:

- **shop-no-shop** v0.25
- **fake-shop-risk** v0.25

7.2. Schnellstart

Je nach Anwendungsfall empfiehlt sich folgender Einstieg:

- Eine einzelne Webseite lokal über die CLI klassifizieren: siehe Abschnitt **## CLI.**
- Den Klassifikator über HTTP aufrufen: siehe Abschnitt **## REST API**.
- Den Pulsar-Worker starten: siehe Abschnitt **## Pulsar Worker Docker Mod** und anschließend `pulsar/README.md`.
- Die vollständige Testsuite ausführen: siehe Abschnitt **## Run the tests**.

Schnellstes lokales CLI-Beispiel:

```
python predict.py
--warc-path scraped_data/_warc/shop.modelltech.ch.warc.gz
--site shop.modelltech.ch
--all-models
--json
```

7.3. Repository-Struktur

`classifier_modes.py`

Zentrale Modusdefinitionen für `shop-no-shop` und `fake-shop-risk`.

`classifier_workflow.py`

Implementiert den übergeordneten Klassifikationsablauf für Einzel- und Mehrmodell-Klassifikationen.

`classification_result.py`

Dataklassen für ClassificationResult und ModelPrediction.

`predict.py`

CLI-Einstiegspunkt sowie Pretty-Printer und JSON-Ausgabe.

`api.py`

Flask-API-Einstiegspunkt für /healthund /predict.

`features/extract.py`

Erstellt den Merkmalsvektor der Website aus gescrapten Artefakten.

`scraping/runner.py`

Blockierender Scraping-Einstiegspunkt, der vom Workflow verwendet wird.

`scrapy_spider/spider.py`

Scrapy-Spider für HTML, CSS/JS und optionales Bild-Scraping.

`model/`

Versionierte, vortrainierte Modelle und Artefakte wie Vektorisierer und Pickles des Trainingsdatensatzes.

`pulsar/`

Separates Pulsar-Worker-Projekt für die Themenverwertung, den S3-gestützten WARC-Download und die Weiterleitung von Klassifizierungsergebnissen.

`tests/unit/`

Unit-Tests für Übersetzung, Vorhersagelogik, Ergebnisobjekte und simulierte Workflows.

`tests/functionality/`

End-to-End-Tests anhand realer Domänen.

7.4. Aufteilung der Laufzeitumgebung

Dieses Repository enthält nun zwei eng miteinander verbundene Komponenten:

- die Haupt-Classifier-Laufzeitumgebung im Stammverzeichnis des Repositorys
- die Pulsar-Worker-Laufzeitumgebung unter pulsar/

Sie wurden aufgrund eines Konflikts zwischen Python- und Abhängigkeitsversionen bewusst aufgeteilt:

- Der aktuelle Classifier-Stack ist an Python 3.7 gebunden
- Mehrere bestehende ML- und Scraping-Abhängigkeiten sind veraltet und begrenzen die Classifier-Laufzeitumgebung faktisch auf Python 3.7
- Der neue Pulsar-Worker benötigt pulsar-client==3.6.1
- pulsar-client==3.6.1 ist in der aktuellen Python-Laufzeitumgebung 3.7 des Klassifikators nicht verwendbar. Ältere Pulsar-Clients wie 3.3.0 schlagen bei diesem Schema fehl, da classifications[].result ein rohes bytes-Feld innerhalb von Pulsar JsonSchema(...)

Aus diesem Grund wird die geplante Bereitstellung in zwei Komponenten/Projekte aufgeteilt:

- Klassifikator-API-/CLI-Komponente: Behält die bestehende Klassifikator-Laufzeitumgebung und den Modell-Stack bei
- Pulsar-Worker-Komponente: Verwendet eine neuere Python-Laufzeitumgebung für die Pulsar- und S3-Integration

Das Ziel dieser Aufteilung ist es, eine sofortige vollständige Migration der älteren Klassifikator-Abhängigkeiten auf modernes Python zu vermeiden und gleichzeitig die Verwendung eines neueren Pulsar-Clients für die Nachrichtenverarbeitung zu ermöglichen.

7.5. Architektur

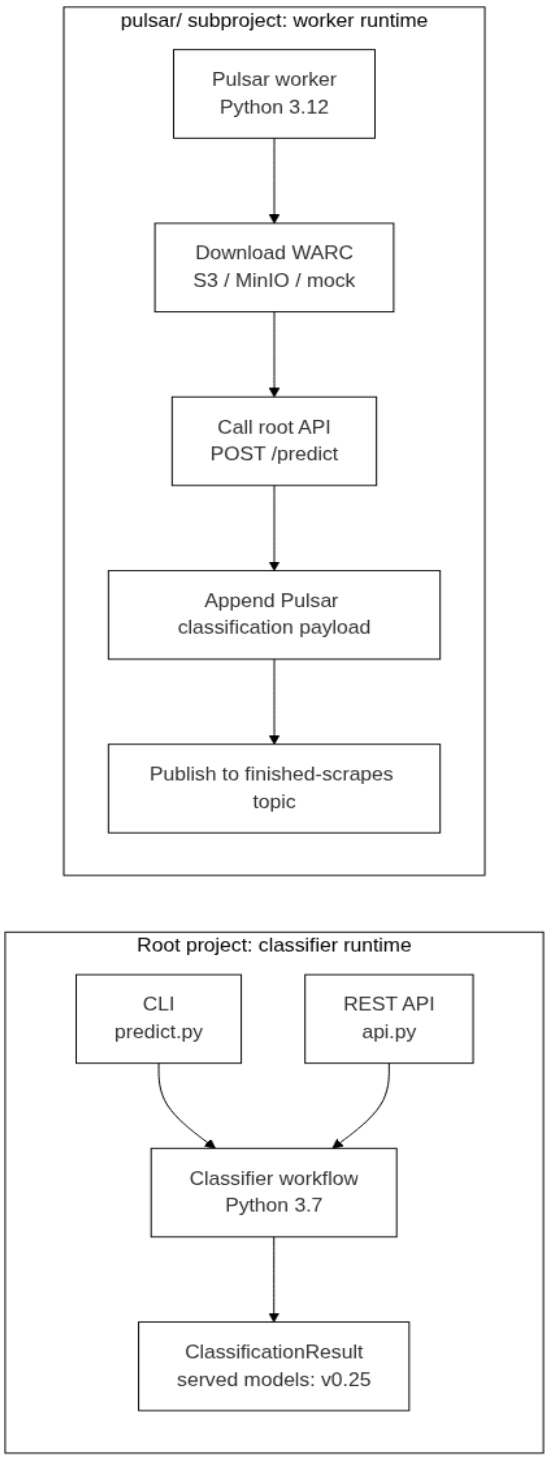

Abbildung 3: Aufbau der Klassifkator-Laufzeitumgebung und des Pulsar-Workers.

7.6. Vorhersageablauf

Der eigentliche ML-Score wird in model_code/predict.py durch predict_from_vector() erzeugt. Allgemeiner Ablauf:

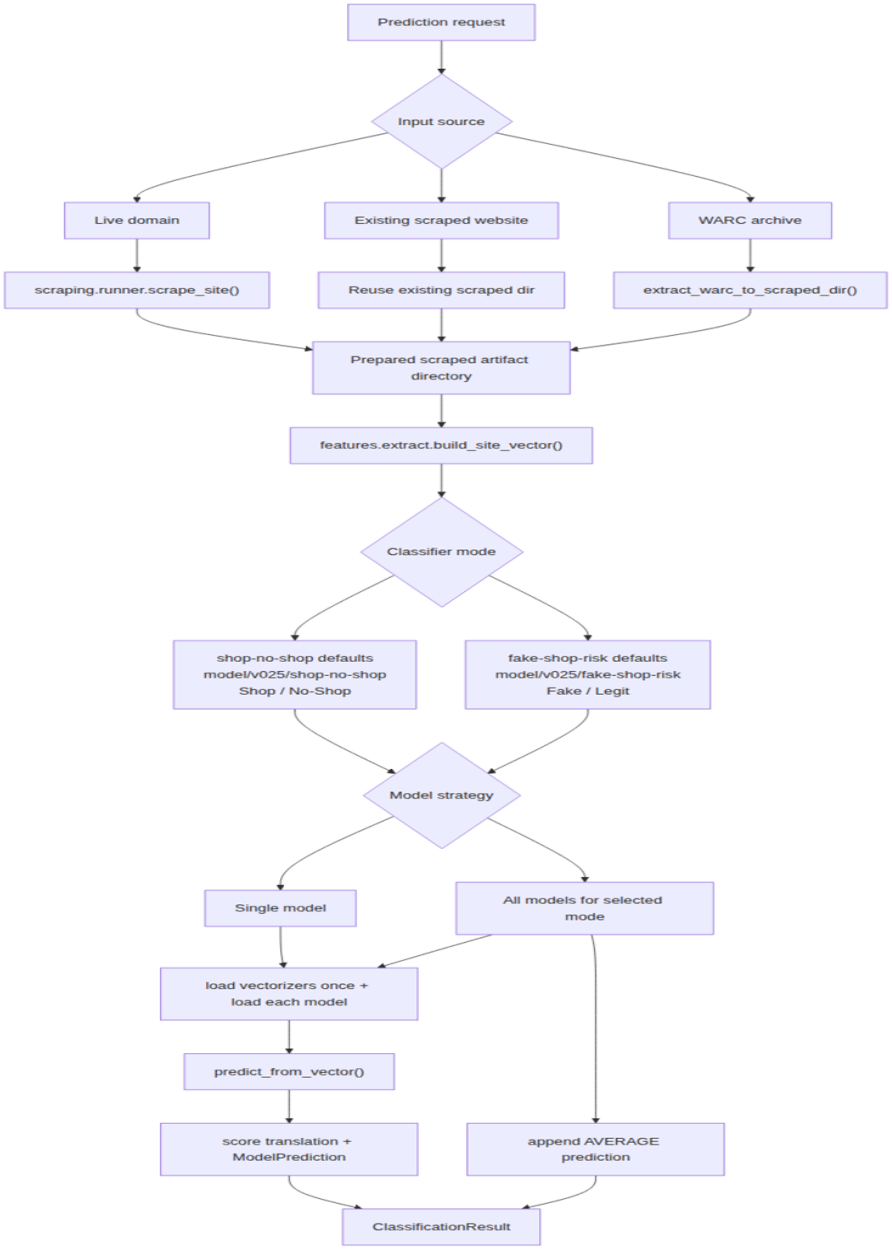

Abbildung 4: Prognoseablauf

Kurz gesagt:

- Live-Domain-Vorhersage: erst scrapen, dann Merkmale erstellen, dann bewerten

- Vorhersage anhand gescrapter Verzeichnisse: vorhandene index.html und cssjs/wiederverwenden, dann bewerten
- WARC-Vorhersage: temporäre Scrape-Artefakte aus dem WARC rekonstruieren, dann bewerten
- shop-no-shop und fake-shop-risk haben denselben Workflow-Ablauf, verwenden jedoch unterschiedliche Standard-Artefakte und Bezeichnungen
- Die numerische Vorhersage selbst erfolgt innerhalb von predict_from_vector()

7.7. Unterstützte modi

Zurzeit werden zwei Modellfamilien unterstützt.

7.7.1. shop-no-shop

- Standard-Artefakt für ein einzelnes Modell: `model/v025/shop-no-shop/xgboost_shopnoshop.model`
- Standardverzeichnis für alle Modelle: `model/v025/shop-no-shop/`
- Labels: **Shop** und **No-Shop**
- Modellfamilie: Dateien mit der Endung `_shopnoshop`.

7.7.2. fake-shop-risk

- Standard-Artefakt für ein einzelnes Modell: `model/v025/fake-shop-risk/xgboost_fsd.model`
- Standardverzeichnis für alle Modelle: `model/v025/fake-shop-risk/`
- Labels: **Fake** und **Legit**
- Familie aller Modelle: Dateien, die auf `_fsd` enden.

Die eingecheckten Artefakte sind derzeit gespeichert unter:

```
model/v025/shop-no-shop/
model/v025/fake-shop-risk/
```

Dies sind die derzeit bereitgestellten v0.25 Modellverzeichnisse, die von der Root-Klassifikator-Laufzeitumgebung verwendet werden.

Verwende die expliziten symmetrischen Namen für beide Klassifikatorfamilien.

```
classify_domain_shop_no_shop()  and
classify_domain_all_models_shop_no_shop()
```

Dies sind die expliziten «Shop-No-Shop»-Einstiegspunkte für die Vorhersage auf Live-Domains.

```
from classifier_workflow import (
    classify_domain_shop_no_shop,
    classify_domain_all_models_shop_no_shop,
)

single_result = classify_domain_shop_no_shop("shop.modelltech.ch")
```

```
all_model_result = classify_domain_all_models_shop_no_shop("shop.modelltech.ch")
```

```
classify_domain_fake_shop_risk()and classify_domain_all_models_fake_shop_risk()
```

Dies sind die Gegenstücke zum Risiko durch gefälschte Shops für die Vorhersage bei Live-Domains.

```
from classifier_workflow import (
    classify_domain_fake_shop_risk,
    classify_domain_all_models_fake_shop_risk,
)

single_result = classify_domain_fake_shop_risk("example.com")
all_model_result = classify_domain_all_models_fake_shop_risk("example.com")
```

```
classify_scraped_dir_shop_no_shop() and classify_warc_shop_no_shop()
```

Diese Hilfsfunktionen klassifizieren ein bereits vorhandenes Verzeichnis mit gescrapten Artefakten oder ein .warc.gz-Archiv, ohne Scrapy erneut ausführen zu müssen.

```
from classifier_workflow import classify_scraped_dir_shop_no_shop, classify_warc_shop_no_shop

scraped_result = classify_scraped_dir_shop_no_shop("scraped_data/shop.modelltech.ch")
warc_result = classify_warc_shop_no_shop("scraped_data/_warc/shop.modelltech.ch.warc.gz")
```

```
classify_scraped_dir_all_models_shop_no_shop() and classify_warc_all_models_shop_no_shop()
```

Diese Helfer sind die Multi-Modell-Entsprechungen für bestehende gescrapte Artefakte und WARC-Archive. Sie geben alle erkannten Modellvorhersagen sowie den angehängten AVERAGE-Eintrag zurück.

```
from classifier_workflow import (
    classify_scraped_dir_all_models_shop_no_shop,
    classify_warc_all_models_shop_no_shop,
```

```
)

scraped_result = classify_scraped_dir_all_models_shop_no_shop
("scraped_data/shop.modelltech.ch")
warc_result = classify_warc_all_models_shop_no_shop
("scraped_data/_warc/shop.modelltech.ch.warc.gz")
```

«Fake-Shop-Risk»-Wrapper für bestehende Artefakte

Der «Fake-Shop-Risk»-Modus stellt außerdem explizite Wrapper für gescrapte Verzeichnisse und WARC-Archive bereit:

```
from classifier_workflow import (
    classify_scraped_dir_fake_shop_risk,
    classify_scraped_dir_all_models_fake_shop_risk,
    classify_warc_fake_shop_risk,
    classify_warc_all_models_fake_shop_risk,
)

scraped_result = classify_scraped_dir_fake_shop_risk("scraped_data/example.com")
scraped_all = classify_scraped_dir_all_models_fake_shop_risk("scraped_data/example.com")

warc_result = classify_warc_fake_shop_risk
("scraped_data/_warc/example.com.warc.gz")

warc_all = classify_warc_all_models_fake_shop_risk
("scraped_data/_warc/example.com.warc.gz")
```

7.8. Ergebnis Format

Der aktuelle Workflow gibt ein ClassificationResult-Objekt mit folgenden Feldern zurück:

- site
- classifier_version
- prediction_timestamp
- scrape_dir_hash
- verifiedFlag für den Erfolg des Workflows: Truebedeutet, dass die Pipeline abgeschlossen wurde und mindestens eine Vorhersage erzeugt hat; dies bedeutet nicht, dass die Ergebnisse von Menschen oder anhand von Ground-Truth-Daten verifiziert wurden
- models
- errors

Jedes ModelPrediction enthält:

- model_path
- model_name
- score
- risk_id
- risk_label
- threshold
- label
- extra

Beispiel-JSON aus ClassificationResult.to_json():


```
{
      "site":                                    "shop.modelltech.ch",
      "classifier_version":                                   "v0.25",
      "prediction_timestamp":     "2026-03-28T21:39:09.280678+00:00",
      "scrape_dir_hash":
"9417ec7750f05cbf27d0a9b956de8d29cc0efcae7661fb22ca6a58aa678f28d4",
      "verified":                                                true,
      "models":                                                    [
      {
            "model_path":                        "model/v025/shop-no-
shop/random_forest_shopnoshop.model",
            "model_name":           "random_forest_shopnoshop.model",
            "score":                                           0.81,
            "risk_id":                                            3,
            "risk_label":                                    "high",
            "threshold":                                       0.5,
            "label":                                         "Shop",
            "extra":                                             {}
      },
      {
            "model_path":                        "model/v025/shop-no-
shop/xgboost_shopnoshop.model",
            "model_name":                "xgboost_shopnoshop.model",
            "score":                                      0.721161,
            "risk_id":                                            7,
            "risk_label":              "above              average",
            "threshold":                                         0.5,
            "label":                                           "Shop",
            "extra":                                             {}
      },
```

```
        {
            "model_path":                                    "",
            "model_name":                                "AVERAGE",
            "score":                                      0.765581,
            "risk_id":                                           7,
            "risk_label":              "above              average",
            "threshold":                                       0.5,
            "label":                                        "Shop",
            "extra":                                             {
                "n_models":                                      2
            }
        }
        ],
        "errors":                                               []
    }
```


Wichtig:

- Es gibt keine Felder der obersten Ebene average_score oder average_risk.
- Bei Läufen mit mehreren Modellen wird der Durchschnitt als zusätzlicher Eintrag in models angehängt.
- Ein Score von genau 0.5 wird als Shop klassifiziert.
- verified=True bedeutet, dass der Workflow erfolgreich abgeschlossen wurde und mindestens eine Vorhersage erstellt wurde.
- verified=False bedeutet, dass der Workflow nicht fehlerfrei abgeschlossen wurde; den Grund dafür finden Sie unter errors.
- verified ist ein Flag für den Zustand der Pipeline bzw. des Ergebnisses, kein Flag für eine manuelle Überprüfung oder die Ground-Truth.

7.9. Risikoübersetzung

Beide Klassifikatorfamilien verwenden derzeit dieselbe alte MAL2-DB-Zuordnung von Score zu ID. In den aktuellen Ergebnisobjekten ist risk_id nicht nur eine generische Bucket-Nummer: Sie entspricht den alten MAL2-DB-Score-Tabellen-IDs, die von folgenden Klassifikatoren verwendet werden:

- fake-shop-risk:
  - mal2_db.websiteriskscore
- shop-no-shop:
  - mal2_db.websiteshopscore

Die aktuelle Implementierung verwendet also eine gemeinsame numerische ID-Zuordnung, während die beiden Klassifikatorfamilien weiterhin auf unterschiedliche Bedeutungen in den alten MAL2-DB-Tabellen verweisen.

Die Hilfsfunktionen in risk_translation.py lauten:

- translate_shop_no_shop_model_score()

- translate_fake_shop_risk_model_score()

Kompatibilitätsaliase wie translate_sc_model_score() sind für ältere Aufrufer weiterhin vorhanden.

Die gemeinsame Zuordnung lautet:

- `0.00 <= score < 0.10` → 5 (very low)
- `0.10 <= score < 0.25` → 1 (low)
- `0.25 <= score < 0.50` → 2 (below average)
- `0.50 <= score < 0.80` → 7 (above average)
- `0.80 <= score < 0.90` → 3 (high)
- `0.90 <= score <= 1.00` → 6 (very high)
- Otherwise → 4 (unknown)

7.10. Installation

7.10.1. Conda

```
conda env create -f environment.yml
conda activate fsdakut-classifiers-predict-py37
```

7.10.2. Docker

```
docker build -t fsd-akut-classifiers-predict:latest .
```

Der Standard-Docker-Build zielt auf das CLI-Image ab. Der Docker-Build-Kontext schließt nun model/aus, sodass Laufzeitcontainer den Modellbaum unter /app/modelgemountet benötigen. Erstelle das dedizierte API-Image:

```
docker build --target api -t fsd-akut-classifiers-predict-api:latest .
```

Erstellen Sie das dedizierte Pulsar-Worker-Image:

```
docker build --target pulsar-worker -t fsd-akut-classifiers-predict-pulsar-worker:latest .
```

Führe eine Vorhersage innerhalb des Containers durch:

```
docker run --rm \
-v $(pwd)/model:/app/model:ro \
fsd-akut-classifiers-predict:latest -u shop.modelltech.ch
```

Den «fake-shop-risk»-Modus im Container ausführen:

```
docker run --rm \
-v $(pwd)/model:/app/model:ro \
fsd-akut-classifiers-predict:latest --mode fake-shop-risk -u example.com
```

Alle verfügbaren Modelle ausführen:

```
docker run --rm -it \
-v $(pwd)/model:/app/model:ro \
fsd-akut-classifiers-predict:latest -u shop.modelltech.ch --all-models
```

Führe alle verfügbaren Modelle mit JSON-Ausgabe aus:

```
docker run --rm -it \
-v $(pwd)/model:/app/model:ro \
fsd-akut-classifiers-predict:latest -u shop.modelltech.ch --all-models --
json
```

Ausgabe von «fake-shop-risk» für alle Modelle ausführen:

```
docker run --rm -it \
-v $(pwd)/model:/app/model:ro \
fsd-akut-classifiers-predict:latest --mode fake-shop-risk -u example.com --
all-models --json
```

Bind-Mounts für persistente Daten:

```
docker run --rm \
-v $(pwd)/model:/app/model:ro \
-v $(pwd)/scraped_data:/app/scraped_data \
fsd-akut-classifiers-predict:latest -u shop.modelltech.ch
```

Mit diesen Bind-Mounts:

- Host ./model wird zu /app/model innerhalb des Containers
- Host ./scraped_data wird zu /app/scraped_data innerhalb des Containers
- Der Modellbaum sollte schreibgeschützt gemountet werden

Das bedeutet:

- Vom Container geschriebene Dateien bleiben auf dem Host verfügbar
- Vorhandene Scrape-Artefakte und WARC-Dateien des Hosts können innerhalb des Containers wiederverwendet werden
- API-Anfragen mit results_dir: "'scraped_data/..."' oder warc_path: "'scraped_data/_warc/..."' funktionieren mit den gemounteten Host-Daten

Die Verwendung einer Live-S3-WARC-Datei mit dem Root-CLI- oder API-Container erfolgt daher derzeit in zwei Schritten:

1. Lade die WARC-Datei in ein lokales Host-Verzeichnis herunter
2. Rufe den CLI- oder API-Container mit dieser heruntergeladenen Datei über warc_path auf

Die Root-CLI und die Root-API akzeptieren derzeit keinen s3_key direkt.

Beispiel: Herunterladen einer Live-WARC-Datei aus S3 unter Wiederverwendung des Pulsar-Worker-Images:

```
mkdir -p live-warcs

docker run --rm \
-e S3_ENDPOINT="$S3_ENDPOINT" \
-e S3_ACCESS_KEY="$S3_ACCESS_KEY" \
-e S3_SECRET_KEY="$S3_SECRET_KEY" \
-e S3_BUCKET="$S3_BUCKET" \
-e S3_SECURE="${S3_SECURE:-true}" \
-v $(pwd)/live-warcs:/downloads \
fsd-akut-classifiers-predict-pulsar-worker:latest \
    verify_s3_access.py \
    --s3-key 7bc7d01d-64c9-4e36-a062-476021382b4b.warc.gz \
    --warc-download-dir /downloads
```

Ordne das heruntergeladene Archiv anschließend dem CLI-Container zu:

```
docker run --rm \
-v $(pwd)/model:/app/model:ro \
-v $(pwd)/live-warcs:/live-warcs:ro \
fsd-akut-classifiers-predict:latest \
--warc-path /live-warcs/7bc7d01d-64c9-4e36-a062-476021382b4b.warc.gz \
--site <site-from-warc> \
--all-models \
--json
```

Oder stellen Sie dasselbe heruntergeladene Archiv über den API-Container bereit:

```
docker run --rm -it -p 8000:8000 \
-v $(pwd)/model:/app/model:ro \
-v $(pwd)/live-warcs:/live-warcs:ro \
fsd-akut-classifiers-predict-api:latest

curl -X POST http://localhost:8000/predict \
-H "Content-Type: application/json" \
-d '{
```

```
"warc_path": "/live-warcs/7bc7d01d-64c9-4e36-a062 476021382b4b.warc.gz",
"site": "<site-from-warc>",
"all_models": true
}'
```

Bindungsmodelle mit Vorhersagen für mehrere Modelle:

```
docker run --rm -it \
-v $(pwd)/model:/app/model:ro \
-v $(pwd)/scraped_data:/app/scraped_data \
fsd-akut-classifiers-predict:latest \
-u shop.modelltech.ch --all-models
```

Bind-Mounts mit JSON-Ausgabe für mehrere Modelle:

```
docker run --rm -it \
-v $(pwd)/model:/app/model:ro \
-v $(pwd)/scraped_data:/app/scraped_data \
fsd-akut-classifiers-predict:latest \
-u shop.modelltech.ch --all-models --json
```

Starte den REST-API-Container:

```
docker run --rm -it -p 8000:8000 \
-v $(pwd)/model:/app/model:ro \
-v $(pwd)/scraped_data:/app/scraped_data \
fsd-akut-classifiers-predict-api:latest
```

7.10.3. Pulsar Worker Docker Mode

Der Pulsar-Worker befindet sich unter pulsar/, doch bei der empfohlenen Bereitstellung werden das Dockerfile im Repository-Stammverzeichnis und der API-Container gemeinsam verwendet.

Dieser Abschnitt fasst lediglich zusammen, wie das Stammprojekt mit der Worker-Bereitstellung verbunden ist. Die workerspecifischen betrieblichen Details finden sich in pulsar/README.md und pulsar/OPERATIONS.md.

Warum diese Aufteilung erfolgt:

- Die Classifier-Laufzeitumgebung bleibt auf dem älteren Python-3.7-Stack
- Die Pulsar-Worker-Laufzeitumgebung verwendet Python 3.12 mit pulsar-client==3.6.1
- Der Worker ruft daher die Classifier-API über HTTP auf, anstatt die Classifier-Laufzeitumgebung im Standard-Docker-Modus direkt zu importieren

Die Compose-Datei enthält einen fsd-akut-classifiers-predict-pulsar-worker-Dienst, der:

- WARCs von S3 oder aus dem Mock-Fixture-Pfad herunterlädt
- sie in einem gemeinsam genutzten Docker-Volume speichert, das unter /shared-warcs eingebunden ist
- http://127.0.0.1:8000/predict aufruft
- die angereicherte Pulsar-Nachricht veröffentlicht, nachdem die API eine Antwort zurückgegeben hat
- den versionierten Modellbaum aus /app/model liest

Empfohlene Bereitstellungsstruktur für Modelle:

- Einen gemeinsamen, versionierten Modellordner auf dem Host führen
- Diesen schreibgeschützt in jeden API-Container einbinden
- Ihn auch schreibgeschützt in Worker-Container einbinden, damit der direkte Worker-Modus weiterhin nutzbar bleibt
- Den Pfad innerhalb des Containers fest auf /app/model setzen
- Die aktuell bereitgestellten Verzeichnisse der v0.25-Familie unter diesem eingebundenen Stammverzeichnis belassen

Beispiel für eine gemeinsame Host-Struktur:

- /srv/fsd-models/v025/shop-no-shop/…
- /srv/fsd-models/v025/fake-shop-risk/…

Das Worker-Image ist generisch. Der tatsächliche Worker-Modus wird wie folgt ausgewählt:

- PULSAR_WORKER_ENTRYPOINT=shop_no_shop_worker.py
- PULSAR_WORKER_ENTRYPOINT=fake_shop_risk_worker.py

Die entsprechenden Pulsar-Standardwerte sollten an diese Worker-Auswahl angepasst werden:

- shop-no-shop:
  - PULSAR_CONSUME_TOPIC=persistent://fsd/classify/shop-no-shop
  - PULSAR_SUBSCRIPTION=fsd-classifier-shop-no-shop
  - PULSAR_CLASSIFIER_ID=urn:classifier:fsd:shop-no-shop:v025
- fake-shop-risk:
  - PULSAR_CONSUME_TOPIC=persistent://fsd/classify/risk
  - PULSAR_SUBSCRIPTION=fsd-classifier-risk
  - PULSAR_CLASSIFIER_ID=urn:classifier:fsd:fake-shop-risk:v025

Starte die API und den Worker gemeinsam:

```
docker compose up fsd-akut-classifiers-predict-api fsd-akut-classifiers-predict-pulsar-worker
```

Der Worker und die API nutzen ein gemeinsames benanntes Volume, sodass der Worker eine WARC-Datei einmal herunterladen kann und die API denselben Dateipfad klassifizieren kann:

- Download-Verzeichnis des Workers: /shared-warcs
- Einbindungspfad der API: /shared-warcs

Heruntergeladene WARC-Dateien werden als temporäre Verarbeitungsartefakte behandelt. Der Worker löscht die lokale WARC-Kopie aus /shared-warcs nach Abschluss jeder Nachricht, damit das gemeinsame Volume im Laufe der Zeit nicht unbegrenzt anwächst.

Wichtige Umgebungsvariablen des Workers in Compose:

- CLASSIFIER_MODE=api
- CLASSIFIER_API_URL=http://127.0.0.1:8000
- WARC_DOWNLOAD_DIR=/shared-warcs
- WARC_DOWNLOAD_MODE
- S3_ENDPOINT
- S3_ACCESS_KEY
- S3_SECRET_KEY
- S3_BUCKET
- S3_SECURE
- PULSAR_BROKER
- PULSAR_WORKER_ENTRYPOINT
- PULSAR_CONSUME_TOPIC
- PULSAR_PRODUCE_TOPIC
- PULSAR_SUBSCRIPTION
- PULSAR_CLASSIFIER_ID
- PULSAR_MODEL_DIR
- ALL_MODELS
- THRESHOLD

Umgebungsvariable für das Einbinden von Modellen in Compose:

- MODEL_VOLUME_SOURCE
  - Standard: ./model
  - empfohlen auf dem Server: /srv/fsd-models
  - erforderlich für Docker- und Compose-Läufe, da Images model/ nicht mehr enthalten

In Umgebungen, in denen der Pulsar-Broker interne Broker-Adressen bekanntgibt, benötigt der Worker möglicherweise network_mode: host, damit er demselben Routing- und DNS-Verhalten wie der Host-Rechner folgt. Der bereitgestellte Compose-Service verwendet diesen Host-Netzwerkmodus für den Worker und kommuniziert daher über den vom Host veröffentlichten Port 8’000mit der API. In dieser Umgebung benötigt die native pulsar-client-Bibliothek zudem ein gelockertes Docker-Seccomp-Profil, weshalb der Compose-Dienst den Worker mit seccomp:unconfinedausführt. Das Compose-Setup startet die API zudem neu, falls sie beendet wird, und startet den Worker bei Fehlern neu. Wenn die Classifier-API während einer Anfrage hängen bleibt, blockiert der Worker, bis sein konfiguriertes HTTP-Timeout abgelaufen ist; anschließend sendet er eine negative Bestätigung für die Nachricht und beendet sich, damit Docker ihn neu starten kann. Die Compose-Datei verwendet absichtlich eine ältere, kompatible Healthcheck-Syntax, damit sie weiterhin mit älteren docker-compose-Installationen funktioniert.

Bewährte Konfiguration für dieses Repository:

- verwende network_mode: host
- verwende den in Compose konfigurierten internen DNS-Server
- behalte seccomp:unconfinedauf dem Worker bei

Informationen zur Einführung mehrerer Instanzen und zu Vorgängen auf Host-Ebene findest du unter OPERATIONS.md.

Empfohlener Ort für die eigentlichen S3- und Pulsar-Einstellungen:

- Speichere sie in einer lokalen .env-Datei im Repo-Stammverzeichnisneben docker-compose.yml
- Beginne mit .env.example
- Lasse die eigentliche .env-Dateiuncommitted; sie wird von Git ignoriert
- Die .env-Dateibefindet sich im übergeordneten Verzeichnis von pulsar/, nicht innerhalb von pulsar/
- Der Live-Pulsar-Integrationstest in test_live_pulsar_integration.py liest diese lokale .env-Datei ebenfalls automatisch ein
- MODEL_VOLUME_SOURCE ist dort für Docker- und Compose-Läufe erforderlich
- Verwende MODEL_VOLUME_SOURCE=./model für die lokale Entwicklung aus diesem Repository-Checkout
- Verwende MODEL_VOLUME_SOURCE=/srv/fsd-models auf dem Server, wenn du zum gemeinsam genutzten Host-Modellbaum wechselst

Typische .env-Werte auf dem Server:

```
InhaltMODEL_VOLUME_SOURCE=/srv/fsd-models
WARC_DOWNLOAD_MODE=real
S3_ENDPOINT=s3.emarketshield.at
S3_ACCESS_KEY=YOUR_ACCESS_KEY
S3_SECRET_KEY=YOUR_SECRET_KEY
S3_BUCKET=fsd-warcs
S3_SECURE=true
```

Typischer Wert für die lokale Entwicklung in der Datei .env:

```
MODEL_VOLUME_SOURCE=./model
```

Dann starte den Stack wie gewohnt:

```
docker compose up fsd-akut-classifiers-predict-api fsd-akut-classifiers-predict-pulsar-worker
```

7.11. CLI Usage

Das Repository unterstützt drei Nutzungsmodi:

- CLI via predict.py
- REST API via api.py

- Pulsar worker via the scripts under pulsar/

Die CLI und die API nutzen dieselbe Klassifikator-Laufzeitumgebung. Der Pulsar-Worker ist eine separate Laufzeitumgebung, die WARCs herunterlädt, standardmäßig die API aufruft und die Klassifikationsergebnisse an Pulsar-Meldungen anhängt.

Einzelnes Modell:

```
python predict.py -u shop.modelltech.ch
```

Einzelmodell zum Risiko durch gefälschte Shops:

```
python predict.py --mode fake-shop-risk -u example.com
```

Ein bestehendes Verzeichnis mit gescrapten Artefakten klassifizieren:

```
python predict.py --results-dir scraped_data/shop.modelltech.ch --site
shop.modelltech.ch
```

Ein WARC-Archiv klassifizieren:

```
python predict.py --warc-path scraped_data/_warc/shop.modelltech.ch.warc.gz
--site
shop.modelltech.ch
```

Alle verfügbaren Modelle sowie die angehängte AVERAGE-Prognose:

```
python predict.py -u shop.modelltech.ch --all-models
```

Risiko von Fälschungen – alle verfügbaren Modelle:

```
python predict.py --mode fake-shop-risk -u example.com --all-models
```

JSON-Ausgabe:

```
python predict.py -u shop.modelltech.ch --all-models --json
```

JSON-Ausgabe in verschiedenen Formaten aus einem bestehenden, mit einem Scraper erfassten Verzeichnis:

```
python predict.py --results-dir scraped_data/shop.modelltech.ch --site
shop.modelltech.ch --all-models --json
```

JSON-Ausgabe in verschiedenen Formaten aus einem WARC-Archiv:

```
python predict.py --warc-path scraped_data/_warc/shop.modelltech.ch.warc.gz
--site

shop.modelltech.ch                    --all-models                    --json
```

Das Docker-Image verwendet „predict.py“ als Einstiegspunkt, sodass Argumente wie „–mode“, „–all-models“ und „–json“ direkt an die Befehlszeilenschnittstelle (CLI) übergeben werden.

Die CLI-Protokollausgabe zeigt außerdem an, welcher Eingabemodus gerade verwendet wird, zum Beispiel:

- Input source=live-site-scrape
- Input source=existing-scraped-dir
- Input source=warc-archive

Die API-Anwendung nutzt „api.py“ als Einstiegspunkt und startet einen kleinen Flask-basierten REST-Dienst auf Port 8000.

7.12. REST API

Die API stellt folgende Funktionen bereit:

- GET /health
- POST /predict

POST /predict akzeptiert genau eine Eingabequelle:

- site
- results_dir
- warc_path

Zu den optionalen Anfragefeldern gehören:

- mode
  - shop-no-shop
  - fake-shop-risk
  - Aus Gründen der Abwärtskompatibilität ist die Standardeinstellung „shop-no-shop“
- all_models
- threshold
- modelloc, vectorizersloc, trainsetloc
- model_dir für die Erkennung aller Modelle

Beispiel für einen Health-Check:

```
curl http://localhost:8000/health
```

Beispiel für eine Domain-Vorhersage:

```
curl -X POST http://localhost:8000/predict \
-H "Content-Type: application/json" \
-d '{
"site": "shop.modelltech.ch"
}'
```

Beispiel für die Vorhersage von Domains mit «Fake-Shop-Risiko»:

```
curl -X POST http://localhost:8000/predict \
-H "Content-Type: application/json" \
-d '{
"site": "example.com",
"mode": "fake-shop-risk"
}'
```

Beispiel für eine Domänenvorhersage mit mehreren Modellen

```
curl -X POST http://localhost:8000/predict \
-H "Content-Type: application/json" \
-d '{
"site": "shop.modelltech.ch",
"all_models": true
}'
```

Beispiel für eine WARC-basierte Prognose:

```
curl -X POST http://localhost:8000/predict \   -H "Content-Type:
application/json" \
-d '{
"warc_path": "scraped_data/_warc/shop.modelltech.ch.warc.gz", "site":
"shop.modelltech.ch",
"all_models": true
}'
```

Beispiel für eine Vorhersage anhand eines durchsuchten Verzeichnisses:

```
curl -X POST http://localhost:8000/predict \
-H "Content-Type: application/json" \
-d '{
```

```
"results_dir": "scraped_data/shop.modelltech.ch",
"site": "shop.modelltech.ch"
}'
```

Regeln:

- Geben Sie eine primäre Eingabequelle an:
  - site, or
  - results_dir, or
  - warc_path
- site kann auch als optionale Metadaten zusammen mit results_dir oder warc_path angegeben werden
- Geben Sie nicht sowohl results_dir als auch warc_path in derselben Anfrage an
- Setzen Sie all_models auf true, um den Multi-Modell-Workflow zu verwenden
- Der Antworttext ist das serialisierte ClassificationResult

Nützliche Flags:

- -m / –model
- -v / –vectorizers
- -t / –trainset
- -u / –url
- –results-dir
- –warc-path
- –site
- –use-cache
- –scrape-images
- –threshold
- –model-dir

Beispiel für eine ansprechende Ausgabe:

```
[INFO] Running prediction for shop.modelltech.ch...
[META] site=shop.modelltech.ch | version=v0.25 | verified=True
[META] prediction_timestamp=2026-03-28T21:39:09.280678+00:00
[META]
scrape_dir_hash=9417ec7750f05cbf27d0a9b956de8d29cc0efcae7661fb22ca6a58aa678
f28d4
[MODELS]
- model=random_forest_shopnoshop.model score=0.810000 label=Shop
risk=high(3)
- model=xgboost_shopnoshop.model score=0.721161 label=Shop risk=above
```

```
average(7)
- model=AVERAGE score=0.765581 label=Shop risk=above average(7)
```

7.13. Testen

Unit-Tests ausführen:

```
pytest tests/unit -v
```

Funktionalitätstests ausführen:

```
pytest tests/functionality -v -s
```

Docker-Integrationstests ausführen:

```
pytest tests/integration -v
```

Die Live-Docker-Integrationstests mit S3-Unterstützung ausführen:

```
RUN_LIVE_S3_DOCKER_TESTS=1 pytest tests/integration/test_docker_images.py -
v
```

Nur die Tests ausführen, die von den jüngsten Workflow-Änderungen betroffen sind:

```
pytest                                  -v                                  \
tests/unit/test_model_predict.py                                            \
tests/unit/test_entrypoint_warc_equivalence.py                              \
tests/functionality/test_predict_fakeshop_at.py                             \
tests/functionality/test_predict_shop_modelltech_ch_modelscore_translation.
py                                                                          \
tests/functionality/test_classify_domain_single_model_bader.py
```

Hilfsskript für Funktionstests:

```
./run_tests.sh
./run_tests.sh                                                        unit
./run_tests.sh                                                 functionality
./run_tests.sh all
```

Hinweise:

- Funktionstests greifen auf echte Domains zu und benötigen daher Netzwerkzugang.

- Integrationstests erstellen und führen die Docker-CLI-/API-Images aus und erfordern daher einen funktionierenden Docker-Daemon.
- RUN_LIVE_S3_DOCKER_TESTS=1 fügt Live-S3-Docker-Abdeckung hinzu: Das Worker-Image lädt das echte WARC von S3 herunter, anschließend klassifizieren die CLI-/API-Container dieses heruntergeladene Archiv über ihre normale lokale warc_path-Schnittstelle.
- tests/unit/test_entrypoint_warc_equivalence.pyvergleicht den CLI-Stammpfad, den API-Stammpfad und den Pulsar-Worker-Pfad auf derselben erhaltenen WARC-Fixture für beide unterstützten Modi.
- Der blockierende Scrape-Wrapper isoliert jeden Scrape in einem eigenen untergeordneten Python-Prozess, sodass wiederholte synchrone Aufrufe sicher in einer pytest-Sitzung ausgeführt werden können.
- Jeder Scrape schreibt außerdem ein WARC-Webarchiv in scraped_data/_warc/<site>.warc.gz.
- pytest-xdist steht zur Verfügung, falls Sie parallele Ausführungen wünschen.
- ./run_tests.sh ist standardmäßig auf all eingestellt, wodurch zuerst Unit-Tests und anschließend Funktionstests ausgeführt werden.

7.14. Hinweise

- Der Standardpfad für das Einzelmodell «shop-no-shop» lautet model/v025/shop-no-shop/xgboost_shopnoshop.model.
- Der Standardpfad für das Einzelmodell «fake-shop-risk» lautet model/v025/fake-shop-risk/xgboost_fsd.model.
- Die Multi-Modell-Erkennung durchsucht das ausgewählte Modellverzeichnis, zum Beispiel model/v025/shop-no-shop/, und ignoriert «vectorizer»-/«trainset»-Artefakte.
- verified=Truebedeutet, dass der Workflow erfolgreich abgeschlossen wurde und mindestens eine Vorhersage erstellt wurde.
- verified=False bedeutet, dass die Pipeline kein brauchbares Vorhersageergebnis erzielt hat; überprüfen Sie ClassificationResult.errors.
- verified bezieht sich auf den Abschluss des Workflows, nicht auf eine manuelle Überprüfung.
- Fehler werden in ClassificationResult.errors gesammelt, anstatt über die öffentliche Workflow-API ausgelöst zu werden.

7.15. Bekannte Probleme

Wiederholte API-Anfragen können nach wie vor zu einem erheblichen Anstieg des Speicherbedarfs führen, insbesondere bei all_models=True WARC-basierten Vorhersagen. Die aktuelle Implementierung enthält eine wichtige Maßnahme zur Risikominderung: Vectorizer-Artefakte werden pro Anfrage neu geladen, anstatt in einem prozessinternen Cache gespeichert zu werden. Unsere Messungen haben gezeigt, dass dies den belegten Speicher für shop-no-shoperheblich senkt, es ist jedoch keine endgültige Lösung für das übergeordnete Problem.
Was dies in der Praxis bedeutet:

- shop-no-shop belegt deutlich weniger Speicher als zuvor, zeigt aber unter anhaltender Auslastung immer noch einen Anstieg des RSS von Anfrage zu Anfrage.
- fake-shop-risk weist auch nach dem Entfernen des Vectorizer-Caches immer noch einen erheblichen Anstieg des belegten Speichers auf.
- Das verbleibende Problem liegt wahrscheinlich im Feature-Building- bzw. Transformationspfad oder im High-Watermark-Verhalten des Allokators und nicht nur im Artefakt-Caching.

Für Untersuchungen können die optionalen API_MEMORY_DEBUG, API_MEMORY_DEBUG_GC und API_MEMORY_DEBUG_MALLOC_TRIMUmgebungsvariablen aktiviert werden, um stufenweise Speichersnapshots in den API-Logs auszugeben. Als operative Schutzmaßnahme kann API_MEMORY_RESTART_THRESHOLD_GB so eingestellt werden, dass die API sich selbst entleert und neu startet, sobald ihr RSS einen konfigurierten Schwellenwert überschreitet.
In der Beispielumgebung ist diese Schutzmaßnahme standardmäßig auf 2 GB eingestellt. Wenn der Schwellenwert überschritten wird, schließt die API die aktuelle /predict-Anfrage ab, beginnt, HTTP-503-Antworten mit restart_pending=true für neue /predict-Anfragen zurückzugeben, und wird anschließend beendet, damit Docker den Container neu starten kann.
Diese 503-Antworten enthalten außerdem retryable=true, ein restart_behavior-Objekt und eine next_action-Zeichenkette, damit Aufrufer erkennen können, dass die aktuell laufenden Vorgänge abgewickelt werden und sie es erneut versuchen sollten, sobald die API wieder einwandfrei funktioniert. Diese Schutzmaßnahme soll die Auswirkungen des bekannten Problems des Speicherwachstums verringern; sie ist kein Ersatz für eine echte Behebung. Der Pulsar-Worker behandelt die temporären Classifier-API-Status 502, 503und 504 als «retryable» und wartet eine gewisse Zeit ab, bevor er die Anfrage erneut versucht. In der Praxis bedeutet dies, dass der Worker das kurze, durch den Speicher-Guard verursachte API-Neustartfenster überbrücken kann. Sind diese Wiederholungsversuche ausgeschöpft oder hängt sich die API bis CLASSIFIER_TIMEOUTauf, quittiert der Worker die Nachricht mit einer negativen Bestätigung, damit Pulsar sie später erneut zustellen kann.

7.16. Lizenz

Basierend auf der European Union Public Licence V. 1.2 des mal2-model-Projekts.

D5.1. Verwertungskonzepte (Report/Bericht)

Die Verwertung des Fake-Shop Detectors wurde in Zielgruppenspezifischen Dokumenten beschreiben mit Fokus auf a) Sicherstellung des laufenden FSD-Betriebs für Konsument:innen und b) Go-To-Market-Strategie für die Verwertung des Produktes FSD durch die öffentliche Hand (Bedarfsträger) oder Internet Service Provider (kommerzielle Verwertung).

1. Sicherstellung des langfristigen Betriebs des Fake-Shop Detectors für Konsument:innen (Bedarfsträger: ÖIAT)

Anforderungen aus Sicht des Bedarfsträgers Watchlist Internet/ÖIAT

Aus der Perspektive des Bedarfsträgers Watchlist Internet ergeben sich Anforderungen in Zwei Prioritätsstufen.

Vorrangige Anforderungen (Priorität 1) betreffen den Betrieb und die Wartung der zentralen Datenbank (Data Lake) und die Konfiguration und den Betrieb der API-Anbindungen für die Watchlist Internet, Crawler und weiterer Services. Mit dieser neu implementierten Architektur sind alle Datenquellen über den zentralen Data Lake angebunden und steuerbar und eine Unabhängigkeit von der bisherigen Middleware gegeben.

Ziel ist dabei die flexible Anbindung und die Steuerung von unterschiedlichen Datenquellen kombiniert mit einer für Watchlist-Internet-Mitarbeiter:innen übersichtlichen Darstellung der vorhandenen Informationen je Domaineintrag. Wesentlich dabei ist u. a., dass sämtliche verdächtigen Domains, die aus unterschiedlichen Quellen gesammelt werden, mit allen uns verfügbaren Methoden analysiert werden. Damit soll verhindert werden, dass z. B. nur via Crawler gesammelte Domains mit Kontextinformationen wie Impressumsdaten oder Trustpilot-Analysen angereichert werden, diese jedoch keine KI-basierte Bewertung erhalten oder umgekehrt via FSD-Plugin gesammelte Domains nicht mit Kontextinformationen angereichert werden. Aktuell liefert die Machine-Learning-Bewertung nur in wenigen Fällen eindeutige und somit für die Watchlist Internet verwertbare Ergebnisse. Zusätzliche Daten erhöhen die Anzahl der auf automatisierte Weise eindeutig klassifizierten Ergebnisse deutlich.

Zudem soll sichergestellt werden, dass Falscheinträge, die über das FSD-Plug-in ausgespielt werden, von Watchlist-Internet-Mitarbeiter:innen selbst gelöscht werden können.

Weitere wesentliche Anforderungen (Priorität 2) umfassen den Betrieb und das regelmäßige Retraining des Machine-Learning-Modelss, den Betrieb und die Wartung des Browser-Plugins und des browserbasierten Shop-Checks (einschließlich der Weiterentwicklung des Umfangs und der visuellen Aufbereitung der dargestellten Ergebnisse), die Optimierung des Datenbank-User-Interfaches sowie die Möglichkeit automatisierte Prozesse für die Analyse und Bewertung neu erfasster Domains mit unterschiedlichen Methoden zu definieren (z. B. Impressums-Check).

Laufender Betrieb des FSD

Laufende Betriebskosten

Infrastruktur- und Hostingkosten

Die laufenden Kosten für Infrastruktur und Hosting setzen sich aus folgenden Positionen zusammen:

- Hosting von 26 virtuellen Servern und Cloud Servern für Crawling, Pulsar, App, Data Lake
- Internet-Anbindung über mehrere unterschiedliche IP Adressen
- Monitoring des laufenden Betriebs
- Überwachung der eingesetzten Komponenten auf Aktualität und Sicherheitslücken gemäß CRA, NIS-2 und weiteren relevanten Richtlinien
- Update und Upgrade der eingesetzten Infrastruktur (umfasst Sicherheitsupdates und Aktualisierung von Komponenten)
- Backup der Systeme und Dienste

Die monatlichen Betriebskosten für Infrastruktur und Hosting belaufen sich in der im Projekt evaluierten Infrastruktur auf 1.050€. Die Servicierung der Cluster im Hinblick auf Monitoring, Konformität gemäß relevanter Richtlinien sowie Updates der Infrastruktur ist mit 13 Stunden pro Monat anzunehmen. Diese Kalkulation versteht sich exklusive Installationskosten - pro Installation eines virtuellen Servers ist von einem Aufwand von etwa einer Stunde auszugehen. Die Infrastruktur ist so aufgebaut, dass sie einfach um neue virtuelle Server erweitert und somit einfach skaliert werden kann.

KI-Modellbetrieb und Wartung

Die laufenden Kosten für KI-Modellbetrieb und Wartung setzen sich aus folgenden Positionen zusammen:

- Bestehende MLOps Infrastruktur
    - Monitoring Model-Degratation & Performance Evaluierung (Dashboard inkl. KPI Anzeige, DataLake)
    - 2x jährlich Re-Training (Hardware: 4 CPUs, 16GB Ram, 24 Stunden) + Zusammenstellung des Trainingsdatensatzes (Personalkosten, siehe unten)
- Training und Evaluierung von neuen Klassifikationsmodellen: z.B. Risikoklassifikation basierend auf rein textuellen Merkmalen (Domainnamen), Metadatenextraktion bestehender Datensätze (z.B. Impressum, Telefonnummern, Whois, IBAN), Detektion der Websitesprache, Klassifikation der Shopkategorien (Elektro, Textilien, etc.), andere Arten von Betrugsseiten (Finanzbetrug, Lovescams).

Die monatlichen Kosten für Betrieb und Wartung (Hardware) der im Projekt evaulierten Modelle belaufen sich auf 820€. Personalaufwände sind im folgenden Abschnitt gesondert ausgewiesen.

Personalaufwand für Support, Entwicklung und Qualitätssicherung

Der Personalaufwand des ÖIAT umfasst (1) den laufenden Basisbetrieb der Qualitätssicherung (Betrugstrends-Monitoring zur Sicherstellung der Relevanz, Analysen diverser Datenquellen, Monitoring der Ausspielung der FSD-Ergebnissen über die Watchlist-Internet-Kanäle, Fehler-Management etc.) mit jährlichen Kosten von 28.000€, (2) die laufende Überprüfung und Analyse der mit hohen Risikoscores bewerten KI-Ergebnissen nach bestimmten Priorisierungskriterien (z. B. .at- und .de-Domains) mit jährlichen Kosten von 30.000€ sowie (3) Aufwände im Zusammenhang mit Re-Training, insbesondere die systematische Aufbereitung von Trainginsdaten und die Qualitätskontrolle für aktuelle Betrugsfallen mit Kosten von 80.000€ für zwei jährliche Re-Trainings.

Der Personalaufwand von X-Net betrifft den 2nd und 3rd Level-Support, für den je nach Nutzer:innen-Anzahl mit wachsendem Aufwand zu rechnen ist. In der Beta- und Evaluierungsphase zeigte sich eine hohe Stabilität des Systems bei einem Anfrage-Aufkommen von etwa zwei Stunden / Monat. Im Zuge einer Skalierung des Systems ist mit einer steigenden Anzahl an Support-Requests zu Fehlern und Störungen zu rechnen. Eine wesentliche Kostenposition stellt zudem die Qualitätssicherung der eingesetzten Dienste und Komponenten dar: Notwendig ist die laufende Überwachung der SBOM auf Aktualität sowie auf bekannte Sicherheitslücken (regelmäßige Sicherheitspatches, Updates, Upgrades). Zu berücksichtigen sind zudem Anpassungen, der Ersatz von Komponenten und ggf. Weiterentwicklungen zur Herstellung eines höheren Sicherheitslevels.

Der Personalaufwand des AIT beinhaltet (1) das kontinuierliche Monitoring (2 Arbeitstage / Monat bzw. 1 Personenmonat / Jahr), (2) Re-Training und Evaluierung von vier Modellen (zweimal jährlich eine Arbeitswoche pro Modell bzw. 2 Personenmonate / Jahr) sowie (3) Training und Evaluierung neuer Modelle (1 Personenmonat / Modell).

Kosten für laufenden Betrieb gesamt

Daraus ergeben sich für einen sicheren Weiterbetrieb (ohne Weiterentwicklungen) folgender Bedarf:

Position	Träger	Jährliche Kosten
Infrastruktur und Hosting	X-Net	12.600€

Servicierung der Cluster und laufender Support	X-Net	25.000€
KI-Modellbetrieb: Hardware	AIT	9.840€
KI-Modellbetrieb: kontinuierliches Monitoring	AIT	22.000€
KI-Modell-Updates: 2x jährliches Re-Training & Evaluierung	AIT	44.000€
Qualitätssicherung	ÖIAT	65.000€
Aufbereitung Trainingsdaten & Qualitätskontrolle für 2x jährliches Re-Training	ÖIAT	80.000€
Gesamt		**258.440€**

Tabelle 2: Geschätzte jährliche Kosten für den laufenden Betrieb. Nicht Teil dieser Kostenschätzung, aber bei Bedarf zusätzlich einzuplanen sind Training und Evaluierung neuer KI-Modelle mit 22.000 € pro Modell.

Nachweis des Nutzens: verhinderte Schäden

Die durch das FSD-System verhinderten Schäden lassen sich näherungsweise abschätzen. Die zugrundeliegende Berechnungsmethode folgt der Logik, dass Betrugswarnungen aus dem FSD-System überwiegend im Moment des Zweifels („Ist ein Shop echt oder fake?“) konsumiert werden und daher eine hohe Wirkung in der Betrugsprävention entfalten.

Für die Berechnung werden folgende konservativ geschätzte Annahmen getroffen: (1) 400.000 Nutzer:innen rufen pro Monat die Watchlist-Internet-Warnungen auf (via Plugin des Fake-Shop Detectors sowie über die Watchlist-Internet-Kanäle); (2) bei 5% der Personen, denen eine Warnmeldung bereitgestellt wird, wird konkret ein Schaden verhindert; (3) die durchschnittliche Schadensumme beträgt 50€.

Daraus folgt die konservative Schätzung, dass **jährlich Schäden im Wert von rund 12.000.000 € verhindert werden** können (Berechnung: 400.000 Personen × 5 % × 50€ × 12 Monate).

Notwendige Ressourcen für Weiterentwicklungen

Über den Mindestbetrieb hinaus werden folgende Weiterentwicklungen identifiziert und mit Aufwänden in Personentagen (PT) bzw. Personenmonaten (PM) geschätzt:

- Dynamischer Data-Lake-Output Server für einfacheren List Export (5 PT)
- Automatisiertes Training auf Basis des Datalakes für alle KI Classifier (8 PT)
- Separates Test-System um neue Classifier ausprobieren zu können. (6 PT)
- Annotations UI: Ziel: menschlich annotierte Ground-Truth für Classifier (14 PT)
- Role based UI: Je nach Bedarfsträger konfigurierbar (Eine Art Data-Portal). (14 PT)
- Öffentliches Widget für Webseiten (Webseiten innerhalb der letzten 24 Stunden verarbeitet, Fake-Shops, etc.) (10 PT)
- Public Grafana Dashboard (5 PT)
- Integration weiterer Classifier: Domain Kategorie, Risikoanalyse auf Basis von Domain-Wort-Extraktion, externe Domain Blacklists (15 PT)
- Definition der Serviceangebote außerhalb des Browser-Plugins und Shop-Check. (15 PT)
- Ausbau der bestehenden ML-Ops Architektur:
    - Ausbau der Infrastruktur für automatisiertes Monitoring und Qualitätssicherung, etc. (1-4 PM).
    - Erweiterung der bestehenden Clusteranalyse für den Bedarfsträger zur rascheren Auffindbarkeit von Fake Shops und Verbesserung der Prävention für Konsument:innen. Geschätzter Aufwand Erweiterung Clusteringanalyse (1 PM).
    - Explainability bestehender Modelle erweitern (AI Act: Nachvollziehbarkeit für Expert:innen und Nutzer:innen verbessern) (2-6 PM).

Business Plan

Nachhaltigkeitsstrategie

Langfristige Betreiberstruktur

Der langfristige Betrieb des FSD stützt sich auf die drei Schlüsselpartner ÖIAT (Qualitätssicherung und Ausspielung über die Watchlist Internet), AIT (KI-Modelle und Re-Training) und X-Net (Architektur, Data Lake, API und Infrastrukturbetrieb). Die

kommerzielle Verwertung könnte über eine eigens dafür gegründete GmbH erfolgen, während der gemeinwohlorientierte Betrieb beim ÖIAT verbleibt.

Die technische Grundlage dieser Betreiberstruktur bildet die neu implementierte Architektur, die sich durch Modularität und Skalierbarkeit auszeichnet. Der Message-Bus (Apache Pulsar) ermöglicht eine effiziente, skalierbare Kommunikation zwischen den Modulen; die Round-Robin-Subscription erlaubt es, neue Instanzen einfach hinzuzufügen und den Arbeitsaufwand zu verteilen. Die Funktionen sind zudem in dedizierte Services aufgeteilt (Separation of Concerns). Ein FastAPI-Endpunkt für die URL-Eingabe, ein URL-Frontier zur Aufgabenverwaltung sowie eigene Server für Scraping, KI-Klassifikation und Datenbearbeitung.- Diese Trennung erhöht die Wartbarkeit und erleichtert zukünftige Erweiterungen. Die Migration von PostgreSQL zu MongoDB ermöglicht eine wesentlich höhere Flexibilität in der Datenstruktur, sodass sich Datenmodelle im Laufe der Zeit anpassen lassen, ohne dass komplexe Datenbankmigrationen erforderlich sind. Durch die dezentrale Architektur kann jedes Modul unabhängig skaliert werden. Bei steigendem Bedarf können z. B. mehrere Scraper- oder KI-Server gestartet werden, ohne die gesamte Anwendung neu aufbauen zu müssen. Der URL-Frontier verarbeitet URLs aus unterschiedlichen Quellen über einen zentralen Service, wodurch bereits bekannte URLs am Beginn der Pipeline ausgefiltert werden.

Insgesamt verbessert die Architektur nicht nur Performance und Skalierbarkeit erheblich, sondern erhöht auch Wartbarkeit und Flexibilität der Anwendung, während ei Kosten durch eine effizientere Nutzung der Infrastruktur gesenkt werden. Dies entspricht den Anforderungen eines marktreifen Produkts und erlaubt eine Skalierung des Systems auf das Produktreifestadium TRL 8.

Open Source und proprietäre Kompenenten

Open-Source-Lösungen ermöglichen es Unternehmen und Institutionen, ihre Daten verantwortungsvoll, unabhängig und transparent zu verwalten. Durch den öffentlich einsehbaren Quellcode bietet Open Source ein hohes Maß an Transparenz, Nachvollziehbarkeit und Unabhängigkeit im Vergleich zu kommerziellen Hersteller:innen. Mit dem Einsatz von Open Source ist es möglich, professionelle Systeme dezentral und autonom aufzubauen.

Für die Endnutzer:innen ergeben sich daraus zahlreiche Vorteile – u.a. die vollständige Kontrolle über die eigenen Daten, Sicherheit und Transparenz, Flexibilität durch Anpassungen an die eigenen Bedürfnisse sowie die Unabhängigkeit von einzelnen kommerziellen Anbietern. Proprietäre Komponenten sind demgegenüber an bestimmte Hersteller gebunden, der Quellcode ist nicht einsehbar und damit einher gehen Risiken wie Lock-in-Effekte, die plötzliche Einstellung der Software, nicht überprüfbare

schadhafte Code-Teile oder Datenlecks. Die Nutzung von proprietärer Software ist oft nur durch die Bezahlung von Lizenzgebühren möglich.

In der Entwicklung des Fake-Shop Detectors wurde besonders auf Transparenz und Nachvollziehbarkeit gegenüber den Nutzer:innen geachtet. Daher fiel die Wahl bei der Entwicklung der grundlegenden Funktionalitäten der App sowie der Dienste zur Erfassung, Aufbereitung und Speicherung der Website-Daten auf Open-Source Komponenten. Über das Browser-Plugin ist klar ersichtlich, welche Daten verarbeitet werden und wie die Software funktioniert. Die Datenverarbeitung folgt dem Grundsatz strikter Datenminimierung: Verarbeitet werden ausschließlich die für die Analyse der Domains erforderliche Daten. Eine Bildung von Nutzungsprofilen findet nicht statt (siehe im Detail Abschnitt 1.4.2 Datenschutz und Compliance).

Partnerschaften und Fördermöglichkeiten

Der FSD-Datenkorpus kann als Teil der öffentlichen Sicherheitsinfrastruktur betrachtet und der Betrieb als Aufgabe der öffentlichen Hand verstanden werden. Mehrere Argumente sprechen dafür (1) Die Schutzwirkung kommt allen Konsument:innen zugute – unabhängig davon, ob sie für ein kommerzielles Produkt bezahlen oder nicht. (2) Der FSD als Instrument gegen Cyberkriminalität berührt unmittelbar die innere Sicherheit und ist mit anderer Infrastruktur wie CERT- und Frühwarnstrukturen vergleichbar. (3) Jeder verhinderte Betrugsfall reduziert Schäden bei Bürger:innen und Unternehmen. Angesichts der hohen Verbreitung von Fake-Shops überwiegt der Nutzen die Kosten deutlich.1 (4) Im Sinne der digitalen Souveränität verringert ein im Inland betriebener, neutraler und deutschsprachiger Korpus die Abhängigkeit von internationalen Anbietern und stärkt die Glaubwürdigkeit gegenüber der Bevölkerung.

Die öffentliche Hand stellt damit einen wichtigen Partner für den FSD dar. Ihre Leistungen kommen allen Bürgerinnen und Bürgern zugute. Anknüpfungspunkte für eine öffentliche Finanzierung bestehen insbesondere beim Bundesministerium für Inneres, beim Bundesministerium für Arbeit, Soziales, Gesundheit, Pflege und Konsumentenschutz sowie beim Bundeskanzleramt (Staatssekretariat für Digitalisierung).

Neben der öffentlichen Hand stellen Unternehmenspartnerschaften im Sinne eines Sponsorings eine realistische Finanzierungsmöglichkeit dar. Anknüpfungspunkte bestehen vor allem in folgenden Branchen: Für Banken und Zahlungsdienstleister verschärft die beschlossene, aber noch nicht in Kraft getretene EU-Payment-Services-Regulation (PSR, gemeinsam mit der PSD3) die Haftungsregelungen bei Betrugsfällen.

[1] Vgl. Handelsverband Österreich: Sicherheitsstudie 2025 (27 % der Bevölkerung hatten bereits Kontakt mit Fake-Webshops) sowie ÖIAT/KMU Forschung Austria: Dunkelfeldstudie „Stop Fraud – Bedrohungen für den österreichischen Onlinehandel", 2025 (jedes vierte österreichische KMU im Online-Handel war bereits von Fake-Shops, Markenfälschungen oder Produktpiraterie betroffen).

Künftig besteht die Verpflichtung, angemessene Betrugspräventionsmechanismen zu implementieren. Wer dies versäumt, haftet für die Verluste der Kund:innen, bei bestimmten Fällen von Identitätsbetrug („Spoofing“) besteht sogar eine Erstattungspflicht.2 Frühzeitige Betrugserkennung gewinnt für diese Branche damit zusätzlich an Wert. E-Commerce-Plattformen und Preisvergleichsportale profitieren, weil unseriöse Anbieter das Vertrauen in den gesamten Online-Handel schädigen und eine glaubwürdige Positionierung zum Thema Sicherheit unterstützt wird. Versicherungen (Anbieter von Online-Kauf- oder Cyber-Schutz) reduzieren mit weniger Betrugsfällen ihre Schadenslast. Im Bereich der Personenmobilität existiert mit dem „Kuratorium für Verkehrssicherheit“ eine vergleichbare, von der Versicherungswirtschaft getragene Initiative. Logistik- und Versanddienstleister sind direkt von Betrugsmaschen betroffen und leiden indirekt unter Fake-Shops durch ausbleibende oder problematische Lieferungen. Ein vertrauenswürdiger Online-Handel sichert ihr Sendungsvolumen und ihren Ruf.

Risiko- und Sicherheitsmanagement

Umgang mit Fehlklassifikationen

Für den Umgang mit False Positives gilt folgendes Vorgehensmodell: (1) Eingang der Beschwerde; (2) manuelle Überprüfung durch Expert:innen der Watchlist Internet nach dem Vier-Augen-Prinzip; (3) Entscheidung über die Löschung gemeinsam mit der Leitung der Watchlist Internet; (4) im Falle einer Löschentscheidung Änderung des Status auf „Don’t show” in der Datenbank und Löschung aus der Middleware; (5) Meldung an Suchmaschinen zur Entfernung der fehlklassifizierten Domains aus den Suchindizes. Präventiv wirken Maßnahmen des Qualitätsmanagements, insbesondere Schulungen der Mitarbeiter:innen, um Fehlklassifikationen von vornherein zu vermeiden.

Updates der KI-Modelle

Um die Einhaltung der regulatorischen Anforderungen des AI Acts bei gleichzeitiger Steigerung von Vertrauen und Akzeptanz der KI-Lösungen zu gewährleisten, werden im Rahmen der Anpassung der KI-Modelle folgende Maßnahmen umgesetzt: ein Qualitätsmanagement für KI-Modelle zur Sicherstellung von Transparenz, Robustheit und Zuverlässigkeit durch systematische Prüfung und Validierung; Explainable AI (XAI) zur Nachvollziehbarkeit und Interpretierbarkeit der KI-Entscheidungen; sowie Human-in-the-Loop-Prozesse (HITL), die menschliche Überprüfung und Steuerung in kritische KI-Prozesse integrieren, um Compliance und ethische Standards zu gewährleisten.

[2] Vgl. https://www.europarl.europa.eu/legislative-train/theme-an-economy-that-works-for-people/file-revision-of-eu-rules-on-payment-services

Datenschutz und Compliance

Der Betrieb des FSD für Konsument:innen folgt einem Privacy-by-Design-Ansatz, der den datenschutzrechtlichen Aufwand von vornherein geringhält. Die Risikobewertung erfolgt lokal auf dem Endgerät. Verarbeitet werden ausschließlich die aufgerufene URL und der Zeitpunkt der Analyse, sofern die Domain noch nicht in der Datenbank vorhanden ist. Informationen zum Such-, Surf- und Einkaufsverhalten werden nicht erfasst. Lediglich zur Abwehr von Angriffen werden Serveranfragen samt IP-Adresse für 14 Tage in Log-Dateien gespeichert und anschließend automatisiert gelöscht; eine Zusammenführung mit anderen Datenquellen oder eine Weitergabe an Dritte findet nicht statt. Da im reinen Konsument:innen-Betrieb keine Entscheidungen über Personen getroffen werden, sondern lediglich über externe Risiken informiert wird, ist der FSD in diesem Einsatz regulatorisch als KI-System mit minimalem Risiko einzuordnen.

Sonstige Risiken

Zum Schutz der Expert:innen vor verstörenden Inhalten werden entsprechende Inhalte im Expert:innen-Dashboard/Portal gefiltert und nicht angezeigt. Weitere identifizierte Risiken sind: das Risiko der Skalierung; das Risiko der Sperrung (durch Cloud-Anbieter infolge von Abuse-Meldungen und durch Fake-Shop-Betreiber, die selbst Abwehrmechanismen gegen den FSD implementieren) sowie das Risiko eines Angriffs durch Masseninstallation, bei dem sich Fake-Shop-Betreiber selbst als FSD-Nutzer:innen tarnen. Dem begegnet das Sicherheitsmanagement durch ein Monitoring der unterschiedlichen Faktoren; skalierbare und modulare Dienste erlauben ein schnelles Reagieren auf unterschiedliche Angriffs- und Abwehrmechanismen.

KPIs und Erfolgsmessung

Zentrale KPIs können über die Plugin-Zahlen nachgezeichnet werden: 2025 haben im Durchschnitt 12.150 Nutzer:innen pro Woche weltweit das Plugin im Browser Chrome verwendet, die durchschnittliche Nutzung für Firefox liegt bei über 3.150 Nutzer:innen pro Tag. Zudem wurden 2025 über 5.400 Neu-Installationen (Chrome & Firefox) durchgeführt.

Der Erfolg ist zudem hinsichtlich Reaktionszeiten und Systemverfügbarkeit zu messen: Reaktionszeiten sind üblicherweise in SLA-Vereinbarungen geregelt und hängen von der Kritikalität einzelner Dienste und der Art der Störung ab. Vorzusehen ist eine Unterteilung in allgemeine Anfragen, Störungen mit eingeschränkt verfügbarem Service sowie dringende Störungen, bei denen der Service nicht mehr erreichbar ist. Die Einordnung der Meldungen erfolgt anhand der Fehlerbeschreibungen; die Reaktionszeit beginnt grundsätzlich mit dem Eingang der Meldung über das Monitoring-System oder das Ticket-System. Die Systemverfügbarkeit ist ebenso in SLA-Vereinbarungen zu regeln; im Mittel kann eine Verfügbarkeit von mehr als 99 % garantiert werden.

Fazit

Der Weiterbetrieb des Fake-Shop Detectors für Konsument:innen ist technisch gesichert und wirtschaftlich klar begründbar: Einem jährlichen Mindestbedarf in der Größenordnung von rund € 275.000,– bis € 370.000,– steht ein geschätztes verhindertes Schadensvolumen von rund € 12 Mio. pro Jahr gegenüber. Die neu implementierte, modulare Architektur schafft die Voraussetzung für einen stabilen, skalierbaren und von der bisherigen Middleware unabhängigen Betrieb. Die entscheidende offene Frage ist damit keine technische, sondern eine der nachhaltigen Finanzierung: Da die Schutzwirkung allen Konsument:innen unabhängig von einer Zahlung zugutekommt, ist eine dauerhafte Grund- oder Mitfinanzierung durch die öffentliche Hand – ergänzt um Unternehmenspartnerschaften – die naheliegendste Grundlage für den langfristigen gemeinwohlorientierten Betrieb.

2. Verwertungskonzept für den Betrieb durch die öffentliche Hand (Bedarfsträger: BMI)

Ergebnis der Testimplementierung

Anforderungsanalyse

Auf Wunsch des Bedarfsträgers BMI und in enger Abstimmung mit diesem wurde die Testimplementierung des Fake-Shop-Detector-Dashboards (FSD-Dashboard) für die zentrale Datenanalyse beim BMI umgesetzt. Der ursprünglich angedachte Fokus auf den Einsatz auf Polizeidienststellen wurde nicht weiterverfolgt, da sich in der ersten Projektphase herausstellte, dass eine Integration des FSD-Dashboards in den bestehenden Workflow der Anzeigenaufnahme nur schwer umsetzbar ist – beispielsweise haben Opfer zum Zeitpunkt der Anzeige nur selten eine betrügerische URL zur Hand, die mit den vorhandenen FSD-Daten abgeglichen werden kann.

Um die Testimplementierung im Rahmen von FSD Akut neu auszurichten, wurden zwischen September 2025 und Januar 2026 drei Stakeholder-Workshops organisiert. In diesen Treffen diskutierten die Konsortiumspartner in enger Zusammenarbeit mit dem Bedarfsträger drei mögliche Anwendungsszenarien, identifizierten die Anforderungen an die Testimplementierung und legten die notwendigen Umsetzungsschritte fest.

Integrierbarkeit in bestehende Prozesse: Die Testimplementierung wurde für die Anwendung in der zentralen Datenanalyse von Lagebild Betrug umgesetzt. Diese erarbeitet eine tägliche Übersicht der eingegangenen Fake-Shop-Anzeigen und führt monatliche bzw. jährliche Auswertungen und Gesamtschauen für ganz Österreich durch.

Nutzen für den Bedarfsträger: Die Integration automatischer Clusteringverfahren anhand bekannter Fake-Shop-Seiten unterstützt die Ermittler:innen bei der Analyse und der Erkennung relevanter Synergien und Entitäten.

Anwendungsfall und Anforderungen: Für die Ermittlungsarbeit geht es in erster Linie darum herauszufinden, wer eine Fake-Shop-Seite betreibt. Daher sind alle Informationen essenziell, die Rückschlüsse auf Täter:innen oder Täter:innengruppen ermöglichen. Informationen zu den Herstellern der Seite – etwa welche Seiten mit demselben Baukasten erstellt wurden – sind demgegenüber nicht unmittelbar relevant. An zentraler Stelle ist die Analyse dieser Informationen besser möglich, da hier die Daten aus verschiedenen Akten zusammenfließen; von dort können ermittlungsrelevante Erkenntnisse aus dem FSD-Dashboard weiter geteilt werden (z. B. Einsichten zu möglicherweise zusammengehörenden Akten).

Im Zuge der Workshops wurden zudem weitere für den Bedarfsträger relevante Anwendungsfelder identifiziert: die (Teil-)Automatisierung der Datenanalyse und Aktverifizierung für Lagebild Betrug sowie ein KI-Assistent/Chatbot für die Online-Anzeige.

Organisatorisches und Verwertungsmöglichkeiten: Innerhalb des BMI ist die Direktion Digitale Services für die Umsetzung von KI-Tools verantwortlich. Identifizierte Möglichkeiten zur Finanzierung und Beschaffung umfassen die Einreichung eines Folgeprojekts (FFG) oder eine interne Finanzierung.

Use Cases

Use Case 1: Die URLs aus einer kuratierten Liste, zu denen weitere Informationen zu Täter:innen, Tatverdacht etc. vorhanden sind, werden im FSD-Dashboard gesucht. Idealerweise bestehen diese aus bereits vorgeclusterten Adressen oder solchen, bei denen Zusammenhänge gefunden wurden (z. B. derselbe Täter). Evaluiert werden sollte, ob das Clustering der URLs den Behörden hilft, zusätzliche relevante Informationen zu finden, die auf ähnliche oder dieselben Täter:innen bei verschiedenen Betrugsfällen hinweisen. Dazu können die URLs, falls noch existent, im Browser aufgerufen oder in der Wayback Machine nach einer gespeicherten Version durchsucht werden.

Use Case 2: Die in Buckets zusammengefassten URLs weisen eine hohe Ähnlichkeit auf, etwa weil sie mit demselben oder ähnlichen Templates erstellt wurden. Der Bedarfsträger wurde eingeladen, die geclusterten URLs z. B. nach Topics wie „Pellet" oder „Clinic" oder nach wiederkehrenden Mustern wie „geschaft", „billig", „unterwaeschede" oder „eheringede" zu durchsuchen und zu erproben, ob diese Suchfunktion die Ermittlungsarbeit unterstützt.

Zur Unterstützung bei Testung und Evaluierung des Dashboards wurden dem Bedarfsträger für beide Use Cases folgende Leitfragen mitgegeben:

- In welche Cluster wurden die URLs gruppiert, und lassen sich sinnvolle Zusammenhänge der Einstiegs-URLs mit anderen URLs im Cluster identifizieren?
- Sind mehrere bereits als zusammenhängend identifizierte URLs im selben Cluster wiederzufinden, und welche Algorithmen waren dabei hilfreicher?
- War es möglich, die getesteten ermittlungsrelevanten URLs aus der Vergangenheit auch in Buckets – der stärksten Form des Clusterings, bei der alle drei Algorithmen übereinstimmen – zu finden, und ergibt sich dadurch ein deutlicheres Bild bezüglich der anderen URLs im Bucket?
- Sind die durch das Clustering gewonnenen Informationen für die Ermittlungs- oder Dokumentationsarbeit am BMI sinnvoll? Welche Informationen fehlen oder könnten verbessert werden? Welche Funktionalitäten werden im Dashboard vermisst?

Feedback des BMI

Nach der Testimplementierung wurde Feedback des BMI zur getesteten Anwendung eingeholt. Das Feedback konzentrierte sich auf Nutzbarkeit, Relevanz der implementierten Funktionen sowie Verbesserungswünsche.

Das FSD-Dashboard wurde mit 70 URLs relativ aktueller Fake-Shops getestet, die in den drei Wochen vor der Testphase zur Anzeige gebracht worden waren. Diese mussten von den Ermittler:innen einzeln eingegeben werden, da es keine Möglichkeit gab, die gesamte Liste in das Tool einzulesen.

Für die durchgespielten Anwendungsfälle erwies sich aus Sicht der Ermittler:innen die Suche unter dem Reiter „Buckets" als am sinnvollsten. Einige der eingegebenen URLs konnten im Dashboard gefunden werden; lediglich die aktuellsten Adressen waren zu diesem Zeitpunkt noch nicht in das Clustering eingeflossen. Die getesteten URLs wurden in einem oder mehreren Buckets gefunden. Durch die Suche nach Schlagwörtern bzw. Textelementen aus den URLs konnten potenzielle Zusammenhänge der Einstiegs-URLs mit anderen URLs im selben Cluster identifiziert werden, die mit demselben Anbieter assoziiert werden können. Es fehlten den Ermittler:innen jedoch weitere Metainformationen, um über die Ähnlichkeit der Websites hinausgehende Zusammenhänge genauer erschließen zu können – insbesondere zusätzliche ermittlungsrelevante Metainformationen, die zur Erstellung von Netzwerkstrukturen genutzt werden können.

Bezüglich der Usability ist der derzeitige Workflow aufwendig, da die URLs händisch per Copy-and-paste eingegeben werden müssen. Zudem zeigt das Interface nicht an, ob eine Seite noch online ist. Die Buckets enthalten teilweise Hunderte URLs, die derzeit

händisch durchgesehen bzw. nur nach Zeichenketten in der URL durchsucht werden können; wünschenswert wäre eine Grobselektion anhand zusätzlicher Metadaten direkt im Tool sowie die Möglichkeit, geclusterte URLs nach einer Liste bekannter URLs zu filtern (z. B. um mehrere in Österreich zur Anzeige gebrachten URLs im selben Bucket zu identifizieren).

Das Info-Icon ist geeignet, weitere ermittlungsrelevante Metainformationen anzuzeigen. Der derzeit angezeigte Risikoscore ist für die Ermittler:innen allerdings nicht relevant, da sie das Tool für bereits zur Anzeige gebrachte Fake-Shops verwenden, durch die bereits Schaden entstanden ist – es handelt sich also um URLs, die erwiesenermaßen Fake-Shops sind.

Wünschenswerte Informationen für das Info-Icon sind: Whois-Daten, Hosting-Provider, Angaben zum Impressum sowie auf der Website genannte Bankverbindungen/IBAN, E-Mail-Adressen, Firmennamen und sonstige Informationen, die zur Erstellung von Netzwerkstrukturen genutzt werden können – letztere sind besonders relevant. Weitere in der derzeitigen Implementierung fehlende Informationen betreffen den Status der Seite: Ist die Seite noch online, wann war sie zuletzt online, und wie lange war sie online? Da die Buckets zum Teil sehr groß sind, wären zudem eine Klarstellung ihrer Organisation sowie weitere Filteroptionen von Vorteil.

Zur Usability wurde angemerkt, dass die Buckets derzeit nur über den Pfeil ganz rechts angesteuert werden können; die Möglichkeit, direkt auf die Bucket-ID zu klicken, wäre intuitiver. Ergibt die Suche nach URLs in Buckets keinen Treffer, sollte eine entsprechende Meldung erscheinen, da es andernfalls wirkt, als liefe die Suche noch.

Als wünschenswerte Features wurden genannt: eine Import-Funktion für URL-Listen oder Dokumente, um mehrere URLs gleichzeitig durchsuchen und danach filtern zu können (ein Import eigener Listen könnte beispielsweise unter dem Reiter „Manuelle Cluster" angesiedelt werden); die Möglichkeit, gefundene Links direkt anzuklicken und die Oberfläche der gefundenen Websites direkt im Tool zu betrachten; ein Icon, das anzeigt, ob eine Seite noch aktiv oder bereits offline ist; für nicht mehr aktive Seiten die direkte Einbindung einer archivierten Version (z. B. Wayback Machine), da andernfalls nur die URL ohne weitere Informationen zum Clustering-Grund vorliegt; weitere Filtermöglichkeiten anhand einzelner Metainformationen sowie eigener URL-Listen – ermittlungsrelevant sind etwa die Fragen, welche der im Cluster/Bucket aufgelisteten Seiten in Österreich zur Anzeige gebracht wurden, auf welche gemeinsamen IBANs überwiesen wurde und wo weitere Synergien bzw. dieselben Entitäten dahinterstehen; sowie die Option, eigene Cluster anhand der Ergebnisse erstellen zu können.

Wenn für einen bekannten Täter bereits eine Cluster-Evidenz vorhanden ist, ist der getestete Ansatz zielführend. Geht es jedoch darum, anhand der im Dashboard

auffindbaren Informationen Täter:innen und Tätergruppen erst zu identifizieren, sind die derzeit fehlenden Metainformationen essenziell. Je mehr Synergien, gemeinsame Entitäten bzw. Hinweise darauf direkt über das Tool auffindbar sind, umso zielführender ist dessen Anwendung für die Erschließung ermittlungsrelevanter Informationen; diese direkt mit den Ermittlungsakten abgleichen zu können, bringt einen großen Mehrwert für die Ermittlungsarbeit.

Eine Ermittlung zu Fake-Shops findet erst statt, wenn es bereits Opfer gibt; auch Crime-as-a-Service wird ermittelt. Für die Ermittlung relevant ist die Identifizierung von Täterkreisen, nicht die der Hersteller der Seiten. Inwieweit das Clusteringverfahren ohne zusätzliche Metainformationen Zusammenhänge herstellen kann, die auf die Betrüger:innen hinweisen, ist unklar. Die zusätzliche Bereitstellung der oben genannten Metadaten ist daher zentral, um Rückschlüsse auf Täterkreise ziehen zu können.

Verwertungskonzept

Das Verwertungskonzept zielt darauf ab, den Fake-Shop Detector als zentrales Analysewerkzeug für das Bundesministerium für Inneres (BMI) zu etablieren, um Ermittlungsprozesse zu optimieren und die Betrugsprävention zu stärken. Bevor der FSD in eine Produktivumgebung überführt werden kann, sind weitere Tests und Evaluierungen erforderlich, um seine Zuverlässigkeit und Skalierbarkeit unter realen Bedingungen zu bestätigen. Des Weiteren ist die Einhaltung der rechtlichen Rahmenbedingungen sicherzustellen, insbesondere im Hinblick auf Datenschutz (DSGVO bzw. für den Strafverfolgungskontext die Richtlinie (EU) 2016/680) und KI-Regulierung (AI Act), um eine rechtssichere und ethische Nutzung des Systems zu garantieren.

Wirtschaftliche Betrachtung

Die Integration des FSD in die zentrale Datenanalyse des BMI ermöglicht eine automatisierte Voranalyse von Fake-Shop-Anzeigen und kann den manuellen Aufwand für Ermittler:innen deutlich reduzieren. Durch das KI-gestützte Clustering können – nach erfolgreicher Umsetzung der noch fehlenden Features – Zusammenhänge zwischen Betrugsfällen potenziell schneller erkannt werden, was zu einer effizienteren Ressourcenverteilung führt.

Anfallende Kosten betreffen Investitionen in Infrastruktur und Weiterentwicklung des FSD-Dashboards, um die im Rahmen von FSD Akut identifizierten Lücken zu schließen, sowie Personal (inklusive Schulungen) und Wartung im laufenden Betrieb. Für die Überführung des aktuellen FSD-Dashboards in den Produktionsbetrieb ist eine Aufwandsschätzung erforderlich, sobald die umzusetzenden Anforderungen festgelegt sind.

Der daraus resultierende Nutzen liegt – auf Basis der Testimplementierung geschätzt – in einer Zeitersparnis von 25–30 % dank Teilautomatisierung sowie in einer potenziell höheren Aufklärungsquote und einer geringeren Belastung der Ermittler:innen.

Eine flächendeckende Einführung des FSD-Dashboards bietet darüber hinaus mehrere Einsparpotenziale durch erhöhte Betrugsprävention: Durch die frühzeitige Erkennung von Fake-Shops – unterstützt durch das im Rahmen von FSD Akut getestete Clusteringverfahren – können Betrugsverluste für Verbraucher:innen und Unternehmen minimiert und dadurch finanzielle Einsparungen in Höhe mehrerer Millionen Euro erreicht werden.[3] Dank der Zeitersparnis steigt zudem die Effizienz des an der Ermittlung beteiligten Personals; bei flächendeckender Einführung sind weitere Kosteneinsparungen durch Standardisierung erzielbar. Nicht zuletzt erzielt das FSD-Dashboard einen gesellschaftlichen Mehrwert durch die Sichtbarmachung von Betrugsnetzwerken, die einerseits die Awareness bei Verbraucher:innen erhöht und andererseits eine abschreckende Wirkung auf Täter:innen entfalten kann.

Rechtliche Rahmenbedingungen

Im Bereich Datenschutz wird die Rechtskonformität durch verschiedene Maßnahmen sichergestellt. Zugangskontrollen stellen sicher, dass nur berechtigte Ermittler:innen Zugriff auf sensible Daten erhalten. Wo kein unmittelbarer Tatverdacht besteht, erfolgt eine Anonymisierung von Nutzerdaten in der Analyse, um die Privatsphäre zu wahren. Die Löschfristen für gespeicherte Daten richten sich nach den gesetzlichen Vorgaben, um eine rechtssichere Verarbeitung zu gewährleisten. Zur Schaffung von Transparenz werden klare Datenverarbeitungsrichtlinien für Behörden und Ermittler:innen definiert. Der Datenschutz wird damit durch technische und organisatorische Maßnahmen sowie durch die Rechtmäßigkeit der Verarbeitung nach DSGVO bzw. – im Strafverfolgungskontext – nach der Richtlinie (EU) 2016/680 und dem 3. Hauptstück des DSG sichergestellt.

Hinsichtlich der KI-Verordnung (AI Act) ist zu berücksichtigen, dass KI-Systeme im Bereich der Strafverfolgung grundsätzlich in den Anwendungsbereich von Anhang III der KI-VO fallen können. Der FSD erfüllt im dargestellten Einsatz jedoch nur eine vorbereitende bzw. unterstützende Aufgabe für die behördliche Bewertung und beeinflusst das Ergebnis der Entscheidungsfindung nicht wesentlich, sodass er nach Art. 6 Abs. 3 KI-VO voraussichtlich nicht als Hochrisiko-KI-System zu qualifizieren ist; diese Einstufung ist zu dokumentieren und laufend zu überprüfen. Zur Compliance mit den KI-Regularien

[3] Eine Dunkelfeldstudie geht von 320.000 direkt betroffenen Konsumentinnen und Konsumenten in Österreich aus und beziffert die Schadenshöhe auf 16 Millionen Euro (https://www.ait.ac.at/blog/kuenstliche-intelligenz-fake-shop-detector).

werden Schulungen für Behördenmitarbeiter:innen durchgeführt, und die Systemarchitektur wird an die Vorgaben des AI Acts angepasst.

Bei Haftungsfragen gilt es, die Verantwortlichkeiten klar zu regeln. Zu klären ist insbesondere, wer die Haftung bei Fehlklassifizierungen trägt, falls ein legitimer Shop fälschlich als Fake-Shop markiert wird. Für Nutzer:innen des FSD-Browser-Plugins gelten klare Nutzungsbedingungen, die einen Haftungsausschluss für den Umstand beinhalten, dass keine hundertprozentige Erkennungsgenauigkeit garantiert werden kann. Zur Absicherung möglicher Risiken wird empfohlen, den Abschluss einer Cyber-Haftpflichtversicherung für den Betreiber (BMI) oder den Service-Provider (FSD ARGE) zu evaluieren. Ebenfalls empfohlen werden weitere rechtliche Absicherungen bei Kooperationen mit Internet Service Providern (ISP), um mögliche Haftungsansprüche im Vorfeld zu klären.

Fazit

Das Verwertungskonzept für den FSD schafft die Grundlage für einen nachhaltigen, gesellschaftlich wertvollen und rechtssicheren Einsatz des Systems. Durch die Etablierung als zentrales Analysewerkzeug für das BMI kann nicht nur die Effizienz von Ermittlungen gesteigert, sondern auch ein wirksamer Beitrag zur Betrugsprävention geleistet werden. Die Einhaltung der rechtlichen Rahmenbedingungen (DSGVO, AI Act) gewährleistet eine verantwortungsvolle und transparente Anwendung.

Langfristig trägt das Konzept dazu bei, das Vertrauen in den digitalen Handel zu stärken, finanzielle Verluste durch Online-Betrug zu reduzieren und Österreich in der Bekämpfung von Cyberkriminalität gut zu positionieren. Durch kontinuierliche Weiterentwicklung, Schulungen und öffentliche Aufklärung kann sich der Fake-Shop Detector als wichtiges Instrument für mehr Sicherheit im E-Commerce etablieren.

3. Verwertungskonzept für kommerzielle Nutzung (Bedarfsträger: Internet Service Provider u.a.)

Im Rahmen dieses Verwertungskonzepts wurde das kommerzielle Potenzial des Fake-Shop Detectors zunächst mit Internet Service Providern (ISP) als angenommenem primärem Kundensegment untersucht. Die Marktanalyse zeigte jedoch, dass die Zahlungsbereitschaft der ISP gering ist und betrügerische Shops für sie nur einen von vielen abzudeckenden Bedrohungstypen darstellen. Als eigentliches Zielsegment wurden daher Anbieter von Sicherheitssoftware identifiziert, die den Datenkorpus samt Zusatzinformationen in eigene Produkte integrieren. ISP werden folglich als sekundäres Segment eingeordnet. Das folgende Kapitel bildet diesen Erkenntnisweg ab: Es beginnt mit der ursprünglich ISP-fokussierten Marktanalyse und leitet daraus die tatsächlich tragfähige Segmentierung und Verwertungsstrategie ab.

Marktanalyse

Bestehende Anknüpfungspunkte

Für eine kommerzielle Verwertung des Fake-Shop Detectors bei Internet Service Provider bestehen bereits mehrere konkrete Anknüpfungspunkte. Dazu zählen der laufend aktualisierte Datenkorpus samt der zusammenhängenden Services, die technischen Integrationsmöglichkeiten über eine API, die auf horizontale Skaklierung angelegte Infrastruktur sowie die spezifischen Differenzierungsmerkmale des FSD gegenüber vergleichbaren Lösungen.

Datenkorpus

Kern des kommerziellen Angebots ist der laufend gepflegte und aktualisierte Datenkorpus mit den als betrügerisch bzw. vertrauenswürdig klassifizierter Domains, inklusive den damit zusammenhängenden Services (Crawler-Ergebnisse, Analysepipelines, Kontextinformationen wie Impressums- oder Trustpilot-Analysen).

Technische Integrationsmöglichkeiten

ISP können die vom FSD-Ökosystem zur Verfügung gestellten Daten und Services auf drei Wegen integrieren: über per API bereitgestellte Black- und Whitelists, über ein integriertes System beim Anbieter selbst oder über die Integration in Drittsysteme, die ISP bereits im Einsatz haben. Das Management der APIs für die unterschiedlichen Kunden und Dienstleister erfolgt über ein eigenes FSD-Kundenportal, das die Zugriffe überwacht, ein Pay-per-Use-Modell erlaubt und die Sperrung von Zugängen jederzeit ermöglicht.

Die Schnittstellen zum Abruf der Black- und Whitelists sind als REST-API-Calls umgesetzt und mit Zertifikaten abgesichert. Das Zertifikatsmanagement ist mit dem Userverwaltungs-Portal gekoppelt und ermöglicht dadurch die nutzungsbasierte Abrechnung. Das laufende Monitoring erlaubt zudem, dass die tatsächlich abgefragten und eingesetzten Services nachvollzogen werden können.

Skalierbarkeit der Infrastruktur

Alle kuratierten Listen des FSD werden aktuell über eine einzelne, mit API-Key abgesicherte Schnittstelle an das Browser-Plugin und an Stakeholder ausgeliefert. Zusätzlich steht der Analyze-Endpoint zur Prüfung einzelner Websites zur Verfügung. Zukünftig sollen spezialisierte Exports für spezifische Stakeholder möglich sein.

Das Browser-Plugin nutzt alle vorhandenen Endpoints und kann entsprechend skaliert werden: Sobald ein:e Nutzer:in eine Website besucht, wird geprüft, ob die URL auf einer Black-, White- oder Greylist liegt und – sollte dem nicht so sein – eine Analyse angestoßen. Da die Infrastruktur zudem für eine horizontale Skalierung ausgelegt ist, kann auch der URL-Ingest des Browser-Plugins hochskaliert werden.- Sollte eine

Datenquelle künftig schneller (oder langsamer verarbeitet werden müssen als das Browser-Plugin, kann dies über eine Priorisierung umgesetzt werden.

Risikoanalyse

Der kommerziellen Verwertung des Fake-Shop Detectors stehen neben den beschriebenen Anknüpfungspunkten auch Risiken gegenüber, die für eine realistische Einschätzung des Verwertungspotenzials zu berücksichtigen sind. Im Folgenden werden diese entlang der drei Dimensionen Marktakzeptanz, Abhängigkeiten von Partnern sowie technologische Risiken analysiert.

Risiko Marktakzeptanz

Die Nachfrageseite ist grundsätzlich tragfähig, denn Online-Betrug ist in Österreich ein Massenphänomen: Laut der Sicherheitsstudie 2025 des Handelsverbands hatten bereits 27 % der Bevölkerung Kontakt mit Fake-Webshops, und jedes vierte österreichische KMU war bereits von Fake-Shops, Markenfälschungen oder Produktpiraterie betroffen.[4] Das zentrale Akzeptanzrisiko liegt damit weniger im Bedarf als in der Zahlungsbereitschaft und der Wettbewerbssituation. Der breitere Markt für Threat Intelligence ist nach Einschätzung einzelner Marktforschungshäuser (etwa MarketsandMarkets) stark konsolidiert: Demnach vereinen fünf Großanbieter (IBM, Palo Alto Networks, CrowdStrike, Cisco und Google) rund 60–70 % des Marktvolumens auf sich, wobei der Trend zu integrierten Komplettplattformen die Konsolidierung zusätzlich vorantreibt.[5]

Für ein spezialisiertes Produkt wie den FSD ergibt sich daraus weniger eine direkte Konkurrenz – die genannten Anbieter adressieren primär Enterprise-Security und nicht die Erkennung betrügerischer Online-Shops – als vielmehr eine strukturelle Chance: Die globale Plattformkonsolidierung lässt gerade im Bereich kleinteiliger, sprachraumspezifischer Betrugserkennung eine Lücke. Hohe Datenqualität, anreichernde Zusatzinformationen und eine konsequente Fokussierung auf den DACH-Raum sind hier die entscheidenden Differenzierungsmerkmale, denn regionale Spezialisierung entlang von Sprachgrenzen stellt im Internetbetrugsbereich ein zentrales Qualitätsmerkmal dar. Das Akzeptanzrisiko wird folglich primär über den Nachweis dieses regionalen Qualitätsvorsprungs sowie über tragfähige Preis- und Lizenzmodelle adressiert.

[4] Vgl. Handelsverband Österreich: Sicherheitsstudie 2025 (27 % der Bevölkerung bereits Opfer von Fake-Webshops; 64 % der heimischen Webshops durch Cybercrime und Bestellbetrug geschädigt). Online: https://www.handelsverband.at/publikationen/studien/sicherheitsstudie/sicherheitsstudie-2025/ (abgerufen am 24.06.2026).

[5] Vgl. MarketsandMarkets: „Threat Intelligence Market", 2025. Wachstum von 11,55 Mrd. USD (2025) auf 22,97 Mrd. USD (2030), CAGR 14,7 %; das Teilsegment der Threat-Intelligence-Feeds hält rund 29,6 % Marktanteil. Online: https://www.marketsandmarkets.com/blog/ICT/threat-intelligence-market (abgerufen am 24.06.2026).

Risiko Abhängigkeiten von Partnern

Betrieb und Verwertung des FSD beruhen auf wenigen Schlüsselpartnern – ÖIAT (Qualitätssicherung und Ausspielung), AIT (KI-Modell und Retraining) sowie X-Net (Architektur, Data Lake, API). Der Ausfall auch nur eines dieser Partner würde zentrale Funktionen unmittelbar beeinträchtigen. Da sich der Wert eines Daten-Feeds maßgeblich aus Aktualität und Abdeckungsbreite speist, ist zudem die kontinuierliche Verfügbarkeit der Eingangsquellen (Crawler, Plug-in-Meldungen, Impressums- und Trustpilot-Daten) erfolgskritisch: Fallen Quellen weg oder ändern sich deren Formate, sinkt unmittelbar die Produktqualität. Erschwerend wirkt, dass uneinheitliche Datenformate und mangelnde Interoperabilität branchenweit als wiederkehrendes Problem gelten und die Anbindung an Partnersysteme entsprechend aufwendig machen. Hinzu treten die Abhängigkeit von der Integrationsbereitschaft kommerzieller Partner sowie – in der Übergangsphase – von einer fortgesetzten öffentlichen Finanzierung. Zur Risikominderung tragen eine klar dokumentierte Betreiberstruktur, vertraglich abgesicherte Schlüsselpartnerschaften, diversifizierte und nach gängigen Standards aufbereitete Datenquellen sowie die konsequente Ablösung der Middleware bei, um Klumpenrisiken systematisch zu reduzieren.

Technologische Risiken

Zu den größten technischen Risiken zählt die schleichende Verschlechterung des KI-Modells, insbesondere bedingt durch Veränderungen der Betrugsschemata. Eine kommerzielle Nutzung erhöht überdies die Anforderungen an Verfügbarkeit, Antwortzeiten und Durchsatz der API und macht verbindliche Service-Level notwendig; die erfolgte Architekturumstellung auf den zentralen Data Lake samt Middleware-Ablösung birgt weiterhin Migrationsrisiken wie Ausfälle oder Inkonsistenzen. Unersetzbar bleibt jedenfalls ein mehrschichtiger Ansatz aus automatisierter Klassifikation, Regeln und menschlicher Prüfung, wobei insbesondere die menschliche Prüfung weiterhin ein zentraler Erfolgsfaktor der Qualitätssicherung sowie ein Alleinstellungsmerkmal des FSD ist.

Bedarfsanalyse

Die Analysen bestätigen, dass Internet Service Provider (ISP) einen Bedarf an Blacklists von Domainnamen bzw. an Fake-Shop Detektionsservices haben, um ihre Kund:innen bestmöglich vor Internetbetrug zu schützen.

Wie sich in der Marktanalyse allerdings gezeigt hat, erzielen ISP mit Sicherheitspaketen als Premium-Feature trotz niedriger Endkundenpreise nur einen geringen zusätzlichen Umsatz. ISP setzen daher auf einen ausreichenden Basisschutz vor schädlichen Websites, den sie allen Kund:innen zukommen lassen und zwar unabhängig davon, ob diese ein zusätzliches Sicherheitspaket kaufen. An zugekaufte Blacklists stellen ISP dabei

vor allem zwei Anforderungen: Zum einen suchen sie einen Anbieter, der aus einer Hand ein möglichst breites Spektrum schädlicher Websites abdeckt. Betrügerische Online-Shops, die der FSD detektiert, sind dabei nur einer von zahlreichen Typen neben Phishing-Websites, Malware-verbreitenden Websites, betrügerischen Investmentplattformen etc. Zum anderen priorisieren ISP, da sie diese Sicherheitsmaßnahmen ihren Kund:innen nur schwer weiterverrechnen können, den Faktor Kosten gegenüber dem Faktor Qualität. Die Zahlungsbereitschaft für Blacklists, die mit einer für den Markt nicht üblichen hohen Qualitätssicherung einhergehen, ist entsprechend gering.

Entgegen der ursprünglichen Annahme folgt daraus, dass Sicherheitssoftware-Anbieter das relevantere Marktsegment für den FSD darstellen als ISP selbst. Relevant sind insbesondere zwei Arten von Anbietern:

- im B2B-Bereich Anbieter, die ISP bei der Identifikation schädlicher Websites unterstützen (z. B. Whalebone, Allot, Cisco, DNSFilter, Akamai)
- im B2C-Bereich Anbieter von Sicherheitssoftware, die sich direkt an Endkund:innen wenden (z. B. ESET, Bitdefender, Norton, Avast & AVG).

Für Sicherheitssoftware-Anbieter sind, wie sich im Zuge der Marktanalyse herausgestellt hat, über die als schädlich klassifizierten Domainnamen hinausgehende Zusatzinformationen von Nutzen, um darauf aufbauend eigene Analysen und Bewertungen vorzunehmen. Beispiele für im FSD-Umfeld vorliegende Zusatzinformationen sind der Impressumscheck (Impressumsdaten und damit verbundene Risikoscores, punktuell für Crawler-Analyseergebnisse vorliegend) sowie die Trustpilot-Analyse mit zugehörigen Risikoscores.

Verwertungsmöglichkeiten

Geschäftsmodelle

Grundsätzlich stehen für den FSD vier Geschäftsmodelle zur Verfügung: das SaaS-, das API-, das Lizenz- und das White-Label-Modell.

SaaS-Modell: Bereitstellung eines fertigen, einsatzbereiten Dienstes inklusive Oberfläche, Hosting und Wartung. Die Kund:innen nutzen eine laufende Funktionalität ohne eigene Entwicklungsleistung und die Abrechnung erfolgt typischerweise als wiederkehrendes Abonnement.

API-Modell: Bereitstellung einer technischen Schnittstelle, über die gezielt Domains, Bewertungen und Zusatzinformationen abgerufen und in eigene Produkte integriert werden können. Die Abrechnung erfolgt meist nutzungsbasiert (pro Abfrage bzw. Volumenpaket). Dieses Modell ist besonders relevant für Sicherheitssoftware-Anbieter, die den Datenkorpus in eigene Analysen einspeisen.

Lizenzmodell: Einräumung eines Nutzungsrechts am Datenkorpus, der den Lizenznehmer:innen als Datensatz bzw. Feed überlassen wird. Die Abrechnung erfolgt als Pauschale oder periodische Lizenzgebühr, gegebenenfalls gestaffelt nach Umfang oder Einsatzgebiet.

White-Label-Modell: Ergänzend besteht die Möglichkeit, den FSD-Dienst als White-Label-Lösung bereitzustellen. Dabei werden Funktionalität und Datenkorpus unter der Marke des jeweiligen Partners betrieben, während die zugrunde liegende Technologie und Datenpflege beim Betreiber verbleiben. Dies erlaubt es etwa Internet Service Providern oder Sicherheitssoftware-Anbietern, einen Schutz vor betrügerischen Websites als integralen Bestandteil ihres eigenen Produktportfolios anzubieten, ohne eine eigene Erkennungsinfrastruktur aufbauen zu müssen. Für den FSD eröffnet dieses Modell eine Skalierung über etablierte Vertriebskanäle der Partner und erhöht die Reichweite des Schutzes, ohne dass eine eigene Endkundenmarke am Markt etabliert werden muss.

Allen Varianten liegt derselbe Datenkorpus zugrunde. Sie unterscheiden sich im Grad der erbrachten Eigenleistung auf Kundenseite sowie in der Abrechnungslogik und können je nach Zielsegment kombiniert werden. Für die Preisgestaltung ist ein Stufenmodell mit Basis- und Premiumvarianten (mit bzw. ohne Zusatzleistungen) sowie einem Exklusivitätszuschlag angedacht - als indikative Größenordnung wurde ein Basispreis von 5.000 € pro Monat diskutiert.

Kundensegmente

Die Analyse des Marktumfelds (siehe auch Abschnitt 3.1) zeigt, dass sich für den FSD zwei Kundensegmente unterscheiden lassen, die in ihrer strategischen Relevanz, wie oben ausgeführt, unterschiedlich zu gewichten sind. Das primäre Zielkundensegment bilden Anbieter von Sicherheitssoftware im B2B- wie im B2C-Bereich. Im B2B-Bereich ist zudem zwischen zwei Typen von Anbietern zu unterscheiden:

Anbieter von Schutzlösungen für ISP und Netzbetreiber: Unternehmen, die Internet Service Provider und Telekommunikationsanbieter dabei unterstützen, schädliche Websites bereits auf Netzwerk- bzw. DNS-Ebene zu identifizieren und zu blockieren bzw. zu flaggen – noch bevor diese die Endgeräte der Kund:innen erreichen. Beispiele sind Whalebone, Allot, DNSFilter, Akamai oder Cisco (Umbrella). Dieser Bereich des B2B-Segments ist für den FSD besonders relevant, da betrügerische Shops und Fake-Domains unmittelbar in deren Abdeckungsspektrum fallen und der FSD hier als spezialisierter, regional fokussierter Daten-Feed andocken kann.

Anbieter allgemeiner Threat Intelligence: Unternehmen, deren Bedrohungsdaten nicht nur von ISP, sondern auch von Unternehmen, Behörden, Finanzinstituten sowie Security Operations Centern (SOC) und Managed-Security-Dienstleistern eingesetzt werden. Dieser Markt ist stark konsolidiert; die dominierenden Anbieter sind IBM, Palo Alto

Networks, CrowdStrike, Cisco und Google (Mandiant), ergänzt um spezialisierte Anbieter wie Recorded Future.[6] Für den FSD ist dieses Teilsegment relevant, aber deutlich umkämpfter, da der FSD nur einen kleinen Ausschnitt (Fake-Shops) eines sehr breiten Angebotsspektrums abdeckt.

In den **B2C-Bereich** fallen Anbieter von Sicherheitssoftware, die sich mit Endkundenprodukten (Virenschutz, Internet-Security-Suiten) direkt an Konsument:innen wenden, etwa ESET, Bitdefender, Norton, Avast & AVG oder G DATA.

Für all diese Unternehmen ist nicht nur der Datenkorpus der als schädlich klassifizierten Domains von Wert, sondern auch die im FSD-Umfeld verfügbaren Zusatzinformationen – etwa Impressumsdaten samt Risikoscore oder die Ergebnisse der Trustpilot-Analyse. Diese Daten ermöglichen es den Anbietern, eigene Analysen und Bewertungen vorzunehmen und das FSD-Angebot in bestehende Produkte zu integrieren. Hier besteht tendenziell eine höhere Zahlungsbereitschaft, da Informationstiefe und Qualität einen unmittelbaren Wettbewerbsvorteil darstellen.

Das sekundäre Zielkundensegment umfasst Internet Service Provider (ISP). Sie setzen FSD-Daten primär als Basisschutz vor schädlichen Websites ein, den sie ihren Kund:innen unabhängig von kostenpflichtigen Sicherheitspaketen zur Verfügung stellen. Da ISP diese Kosten nur schwer weiterverrechnen können und betrügerische Shops für sie nur eines von vielen abzudeckenden Angriffssegmenten darstellen, priorisieren sie tendenziell niedrige Kosten gegenüber Informationstiefe. Der FSD ist für ISP daher vor allem als Teil eines breiteren Blacklist-Angebots interessant. In der Praxis bedienen sie sich überdies häufig B2B-Anbietern aus dem primären Zielsegment.

Ergänzend kommen perspektivisch weitere Segmente in Betracht, etwa Betreiber von Online-Marktplätzen, Zahlungsdienstleister oder Anbieter von Brand-Protection-Lösungen, für die der Schutz ihres Ökosystems vor betrügerischen Akteuren ebenfalls einen konkreten Nutzen stiftet.

Marktpotenzial

Das Marktpotenzial des FSD ergibt sich aus dem Zusammenspiel zweier Faktoren: dem Ausmaß des Problems, das der FSD adressiert (Bedarfs- bzw. Nachfrageseite) und der Größe jener Märkte, in denen die zuvor beschriebenen Kundensegmente agieren (adressierbarer Markt). Beide werden im Folgenden anhand aktueller Marktdaten eingeordnet.

---

[6] Vgl. MarketsandMarkets: „Threat Intelligence Market – Top Companies", 2025. Der Markt ist konsolidiert; fünf Anbieter (IBM, Palo Alto Networks, CrowdStrike, Cisco und Google) vereinen rund 60–70 % des Marktvolumens auf sich. Online: https://www.marketsandmarkets.com/ResearchInsight/threat-intelligence-security-market.asp (abgerufen am 24.06.2026).

Ausmaß und Dynamik des Problems

Online-Betrug ist in Österreich ein Massendelikt mit hoher gesellschaftlicher Relevanz und bildet die grundlegende Bedarfsbasis für ein Werkzeug wie den FSD:

- Im Jahr 2025 wurden in Österreich 31.001 Fälle von Internetbetrug angezeigt, bei einer Aufklärungsquote von lediglich 28,1 %.[7]
- 27 % der Bevölkerung waren bereits Opfer von Fake-Webshops und 64 % der heimischen Webshops wurden durch Cybercrime und Bestellbetrug geschädigt.[8]
- Jedes vierte österreichische KMU im Online-Handel war bereits von Fake-Shops, Markenfälschungen oder Produktpiraterie betroffen.[9]

Die niedrige Aufklärungsquote macht deutlich, dass die nachträgliche Strafverfolgung an Grenzen stößt und der Prävention – und damit präventiv wirkenden Werkzeugen wie dem FSD – eine zentrale Rolle zukommt. Cybercrime zählt zu den fünf Kriminalitätsfeldern mit dem größten Einfluss auf das Sicherheitsempfinden der Bevölkerung.

Adressierbare Märkte je Kundensegment

Für das primäre Segment (Sicherheitssoftware und Threat Intelligence) wächst der globale Markt von rund 11,55 Mrd. USD (2025) auf 22,97 Mrd. USD (2030) bei einer jährlichen Wachstumsrate von 14,7 %; auf das Teilsegment der Threat-Intelligence-Feeds, dem der FSD am nächsten steht, entfallen davon rund 29,6 %.10 Der übergeordnete Cybersecurity-Markt ist in Österreich auf rund 1,2 Mrd. USD (2024, mit Prognose 2,8 Mrd. USD bis 2032) und in Deutschland auf rund 13 Mrd. USD (2025) zu beziffern; der deutsche Consumer-Cybersecurity-Softwaremarkt (B2C) erreicht bis 2030 etwa 1,1 Mrd. USD.[11] Zentral für die Einordnung ist dabei, dass der FSD adressiert nicht diese Gesamtbudgets adressiert, sondern ein spezialisiertes Teilsegment – die Erkennung betrügerischer Shops als ein Baustein innerhalb breiterer Sicherheitslösungen.

---

[7] Vgl. Bundesministerium für Inneres (BMI): Polizeiliche Anzeigenstatistik 2025 (31.001 Anzeigen wegen Internetbetrug; Aufklärungsquote 28,1 %), zitiert nach Watchlist Internet, watchlist-internet.at (abgerufen am 24.06.2026). Cybercrime wird vom Bundeskriminalamt zu den „Big Five" der Kriminalitätsfelder mit dem größten Einfluss auf das Sicherheitsempfinden gezählt.

[8] Vgl. Handelsverband Österreich: Sicherheitsstudie 2025 (27 % der Bevölkerung bereits Opfer von Fake-Webshops; 64 % der heimischen Webshops durch Cybercrime und Bestellbetrug geschädigt). Online: handelsverband.at (abgerufen am 24.06.2026).

[9] Vgl. ÖIAT/KMU Forschung Austria: Dunkelfeldstudie „Stop Fraud – Bedrohungen für den österreichischen Onlinehandel", 2025.

[10] Vgl. MarketsandMarkets: „Threat Intelligence Market", 2025. Wachstum von 11,55 Mrd. USD (2025) auf 22,97 Mrd. USD (2030), CAGR 14,7 %; das Teilsegment der Threat-Intelligence-Feeds hält rund 29,6 % Marktanteil. Online: marketsandmarkets.com (abgerufen am 24.06.2026).

[11] Vgl. Verified Market Research: „Austria Cybersecurity Market", 2025 (1,2 Mrd. USD 2024 → 2,8 Mrd. USD 2032, CAGR 11,2 %); MarketsandMarkets: „Germany Cybersecurity Market", 2025 (13,03 Mrd. USD 2025 → 20,61 Mrd. USD 2030); KBV Research: „Germany Consumer Cybersecurity Software Market" (rund 1,1 Mrd. USD bis 2030, CAGR 9,9 %). Angaben in USD, indikativ.

Im sekundären Segment (ISP und Netzbetreiber) ist die Nachfrage nach netzseitigem Schutz (Protective DNS) international etabliert. Allein der Anbieter Whalebone versorgt nach eigenen Angaben mehr als 400 Telekommunikationsanbieter, ISP und Institutionen weltweit.[12] Für den FSD ist dieses Segment vor allem indirekt relevant – als Datenlieferant für jene B2B-Anbieter, die ISP mit Blacklists und DNS-Schutz beliefern.

Eine weitere Finanzierungsperspektive stellt wie bereits in Kapitel 1 dargestellt die öffentliche Hand als drittes Segment dar: Der gesellschaftliche Nutzen des FSD (Schadensvermeidung, Konsument:innenschutz, Bezug zur inneren Sicherheit) rechtfertigt eine öffentliche Grund- oder Mitfinanzierung des Datenkorpus als Teil der öffentlichen Sicherheitsinfrastruktur (siehe auch Abschnitt 1.6.2).

Einordnung des FSD-Potenzials

Aus diesen Kennzahlen lässt sich das Potenzial des FSD entlang der gängigen Marktabgrenzung (TAM/SAM/SOM) realistisch einordnen. Der Gesamtmarkt (TAM) umfasst jene Threat-Intelligence-, Netzwerk- und Consumer-Security-Segmente, in die Fake-Shop- und Betrugsdaten einfließen können – ein Volumen im mehrstelligen Milliarden-USD-Bereich, jedoch ganz überwiegend von breit aufgestellten Anbietern besetzt. Deutlich kleiner ist der für den FSD tatsächlich bedienbare Markt (SAM): Anbieter mit Bezug zum deutschsprachigen Raum, die spezifisch betrugs- bzw. fake-shop-bezogene Daten mit regionalem Fokus (.at-/.de-Domains) und angereicherten Zusatzinformationen benötigen. Der realistisch erreichbare Markt (SOM) ist kurz- bis mittelfristig über Pilot- und Folgepartner (z. B. A1, IKARUS, perspektivisch G DATA oder DNS-Schutz-Anbieter) sowie über eine öffentliche Grundfinanzierung erschließbar. Dieses Segment ist überschaubar, aber stabil und wenig austauschbar.

Entscheidend für die Hebung dieses Potenzials ist die Differenzierung. So positioniert sich der FSD nicht als Plattformkonkurrent der großen Anbieter, sondern als spezialisierter, regional fokussierter Enrichment-Feed. Dafür sprechen zwei Markttrends: Datenresidenz und digitale Souveränität werden im DACH-Raum zunehmend zu zentralen Kaufkriterien, und spezialisierte Anbieter bleiben dort relevant, wo sektorspezifische Intelligence gefragt ist.[13]

Das absolute Marktpotenzial des FSD ist durch seine Spezialisierung und den regionalen Fokus begrenzt – innerhalb dieser Nische ist es jedoch tragfähig, durch die hohe gesellschaftliche Relevanz des Problems gut abgesichert und gegenüber den breit

[12] Vgl. Whalebone s.r.o.: Unternehmensangaben – DNS-basierter Schutz für mehr als 400 Telkos, ISP und Institutionen weltweit (Indikator für die Nachfrage im ISP-Umfeld). Online: whalebone.io (abgerufen am 24.06.2026).
[13] Vgl. Mordor Intelligence: „Germany Cybersecurity Market", 2026 (Datenresidenz und souveräne Lösungen als zunehmend zentrales Kaufkriterium) sowie „Threat Intelligence Market", 2026 (spezialisierte Anbieter bleiben dort relevant, wo sektorspezifische Intelligence gefragt ist).

aufgestellten Wettbewerbern schwer substituierbar. Die wirtschaftlich sinnvollste Verwertung verbindet daher die kommerzielle Erschließung des bedienbaren Segments mit einer öffentlichen Grundfinanzierung des Datenkorpus.

Vertriebskanäle und Partnerschaften

Der Vertrieb des FSD stützt sich derzeit nicht auf eine eigene Vertriebsabteilung, sondern auf das etablierte Netzwerk des ÖIAT sowie auf die hohe Reputation von ÖIAT und Watchlist Internet im Bereich Konsument:innenschutz. Diese Bekanntheit und Glaubwürdigkeit im deutschsprachigen Raum wirkt als Türöffner und Differenzierungsmerkmal gegenüber potenziellen Partnern und ersetzt in der aktuellen Phase aktive Kaltakquise weitgehend durch eingehende Anfragen.

Da das ÖIAT als gemeinnützige Organisation kommerziellen Vertrieb nicht uneingeschränkt selbst durchführen kann, erfolgt die kommerzielle Verwertung über eine eigens dafür gegründete Gesellschaft mit den Projektpartnern AIT und X-Net. Diese Konstruktion trennt die kommerzielle Tätigkeit klar von der gemeinnützigen Kernaufgabe: Das ÖIAT verantwortet weiterhin Qualitätssicherung und gemeinwohlorientierten Betrieb, während die Verwertungsgesellschaft Vertragsabschluss, Lizenzierung und Kundenbetreuung übernimmt. Die technische Integration und das Onboarding der Partner werden durch AIT und X-Net umgesetzt.

Als Vertriebs- bzw. Bezugskanäle stehen die bereits beschriebenen Geschäftsmodellvarianten zur Verfügung (insbesondere API-Anbindung, SaaS sowie Lizenz- und White-Label-Modelle) über die Partner den Datenkorpus und die zugehörigen Zusatzinformationen in eigene Produkte integrieren können.

Die Tragfähigkeit dieses Ansatzes zeigt sich bereits an konkreten eingehenden Anfragen, die sämtliche im Verwertungskonzept identifizierten, Zielsegmente abdecken – aus dem primären Zielsegment zum Beispiel die Sicherheitssoftware-Anbieter Total Security (B2C), IKARUS Security Software (B2C und B2B) sowie Nord Security (B2C und B2B) und aus dem sekundären Zielsegment der österreichische Internet Service Provider A1.

Im Zusammenhang mit dem ISP-Segment können zudem auch Versicherungen im Sinne einer Fake-Shop-Versicherung eine Rolle spielen. Versicherungsprodukte entstehen bei Rückversicherern, die das eigentliche Versicherungsrisiko tragen und mit den Versicherungsnehmer:innen nicht direkt in Kontakt stehen. Im Zuge des Projekts wurde mit dem größten europäischen Rückversicherer (Munich Re) eine Fake-Shop-Versicherung auf Basis des FSD evaluiert; die KI-Abteilung der Munich Re hat dabei bewertet, ob das KI-Modell die Grundlage für eine derartige Rückversicherung bieten kann, und das FSD-Konzept positiv beurteilt. Für die Umsetzung bedarf es lediglich eines ISP, der ein solches Versicherungsprodukt anbietet. ISP könnten damit eine Fake-Shop-Versicherung anbieten, bei der Geschädigte den entstandenen Schaden innerhalb

kürzester Zeit ersetzt bekommen; Grundlage wäre allein der Nachweis, dass der Fake-Shop Detector im Einsatz war. Derzeitige Fake-Shop-Versicherungen der ISP bieten einen derart zügigen Service nicht, sondern zahlen erst, wenn das Verfahren und alle möglichen Schritte abgeschlossen sind.

Dass Interesse aus beiden Zielsegmenten besteht, ist ein starkes Signal für die Marktrelevanz des FSD und bestätigt die in der Marktanalyse (siehe Abschnitte 3.2.2 und 3.2.3) abgeleitete Segmentierung. Der weitere Auf- und Ausbau von Partnerschaften – etwa durch Pilotprojekte mit den genannten Interessenten – bildet den nächsten Schritt, um das vorhandene Interesse in tragfähige Vertragsbeziehungen zu überführen.

Fazit

Die kommerzielle Verwertung des Fake-Shop Detectors ist tragfähig, erfordert jedoch eine realistische Einordnung: Das absolute Marktpotenzial ist durch die Spezialisierung auf betrügerische Shops und den regionalen Fokus auf den deutschsprachigen Raum begrenzt, innerhalb dieser Nische aber stabil und schwer substituierbar. Die Marktanalyse hat gezeigt, dass nicht ISP, sondern Anbieter von Sicherheitssoftware und Threat Intelligence Provider das eigentliche Zielsegment bilden – für sie sind die hohe, menschlich gesicherte Datenqualität und die anreichernden Zusatzinformationen der entscheidende Mehrwert. Dass bereits aus allen identifizierten Segmenten konkrete Anfragen vorliegen, bestätigt diese Segmentierung und die grundsätzliche Marktrelevanz. Die wirtschaftlich sinnvollste Strategie kombiniert daher die kommerzielle Erschließung dieses bedienbaren Segments – über API-, Lizenz- und White-Label-Modelle sowie die eigens gegründete Verwertungsgesellschaft – mit einer öffentlichen Grundfinanzierung des Datenkorpus als gemeinsame Basis beider Verwertungswege.

Zusammenfassung und Ausblick

Die vorliegenden Verwertungskonzepten mit je unterschiedlichen Bedarfsträgern zeigen, dass der Fake-Shop Detector über die Projektlaufzeit hinaus auf drei sich ergänzenden Wegen weiterbetrieben und verwertet werden kann: als gemeinwohlorientierter Konsument:innenschutz über die Watchlist Internet, als behördliches Analysewerkzeug für das BMI und als kommerzielles Datenprodukt für Anbieter von Sicherheitssoftware. Diese Wege stehen nicht in Konkurrenz zueinander, sondern teilen mit dem laufend gepflegten Datenkorpus dieselbe technische und inhaltliche Grundlage; die neu implementierte Architektur trägt alle drei Szenarien bis zur Produktreife.

Als verbindendes Element tritt in allen drei Verwertungswegen die Rolle der öffentlichen Hand hervor: Der Datenkorpus betrügerischer Domains ist faktisch ein öffentliches Sicherheitsgut, dessen Schutzwirkung allen Bürger:innen zugutekommt und dessen Pflege sich nicht allein kommerziell refinanzieren lässt. Eine öffentliche Grundfinanzierung, ergänzt um kommerzielle Erlöse und Unternehmenspartnerschaften, ist daher die tragfähigste Grundlage für einen dauerhaften Betrieb.

Voraussetzung für die Umsetzung bleiben in allen Szenarien einige zu klärende Punkte: die Sicherstellung der laufenden Qualitätssicherung mit ausreichenden personellen Ressourcen, die Weiterentwicklung der im BMI-Test identifizierten Funktionen sowie die Einhaltung der datenschutz- und KI-rechtlichen Rahmenbedingungen. Werden diese adressiert, verfügt der Fake-Shop Detector über eine belastbare Grundlage, um dauerhaft einen messbaren Beitrag zum Schutz vor Online-Betrug zu leisten.