

Explainable OSINT-Based Risk Scoring for Companies and Individuals

by

Almaz Amangeldin

Submitted to the Department of Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Data Science

at the

NAZARBAYEV UNIVERSITY

June 2026

© Nazarbayev University 2026. All rights reserved.

Author
Department of Computer Science
08 May 2026

Certified by
Askar Boranbayev
Associate Professor
Thesis Supervisor

Certified by
Talgat Bayan
Postdoctoral Researcher
Thesis Supervisor

Accepted by
Yelyzaveta Arkhangelsky
Dean, School of Engineering and Digital Sciences

Explainable OSINT-Based Risk Scoring for Companies and Individuals

by

Almaz Amangeldin

Submitted to the Department of Computer Science
on 08 May 2026, in partial fulfillment of the
requirements for the degree of
Master of Science in Data Science

Abstract

The purpose of the thesis is to support compliance and due diligence checks by combining sanctions lists, registry information, and adverse media signals in one auditable and traceable workflow. The system integrates data from the Office of Foreign Assets Control (OFAC), the European Union (EU) Consolidated Sanctions List, the United Nations (UN) Consolidated Sanctions List, the United Kingdom Sanctions List (UKSL), the Astana Financial Services Authority (AFSA) public register, the Global Legal Entity Identifier Foundation (GLEIF), and the Global Database of Events, Language, and Tone (GDELT), and processes them through a modular NestJS backend with source-specific parsers, snapshot-based provenance storage, evidence tracking, and a transparent rule-based scoring model. In the proposed approach, sanctions evidence is treated as the strongest signal, while registry anomalies and media mentions are used as secondary and contextual factors, because not all types of evidence should have the same weight. The results show that the system can successfully ingest heterogeneous public datasets, preserve source provenance, and generate structured outputs with scores, explanations, and evidence references, which makes the final result easier to review by a human analyst. At the same time, the work also shows some important limitations, such as heuristic entity matching, noisy media data, and the tradeoff between strict reproducibility and live data freshness. Overall, the thesis demonstrates that a practical and explainable Open-Source Intelligence (OSINT)-based risk screening backend can be built from public sources, although it should still be used as a decision-support tool rather than as a fully automatic final judgment system.

Thesis Supervisor: Askar Boranbayev
Title: Associate Professor

Thesis Supervisor: Talgar Bayan
Title: Postdoctoral Researcher

Contents

1	Introduction	7
1.1	Background and Motivation	7
1.2	Problem Statement	8
1.3	Scope and Assumptions	8
2	Literature Review	11
2.1	The Importance of OSINT	11
2.2	Sanctions Screening of Entities and Individuals	12
2.3	Reputation and Risk Scoring Approaches	13
2.4	Explainable AI (Artificial Intelligence) and Interpretability	15
2.5	News Sentiment Analysis	16
3	Datasets: OSINT Sources for Explainable Company Risk	19
3.1	Overview	19
3.2	Sanctions and Watchlists	20
3.2.1	OFAC SDN (Specially Designated Nationals) List	20
3.2.2	EU Consolidated Sanctions List	23
3.2.3	UN Consolidated Sanctions List	25
3.2.4	UK Sanctions List	28
3.3	Registries	30
3.3.1	AFSA Public Register	30
3.3.2	GLEIF Register	32
3.4	Adverse Media	35

4	Methodology	39
4.1	System Workflow	39
4.2	System Architecture and Implementation Stack	42
4.2.1	Backend Architecture and Organization	42
4.2.2	Persistence and Storage Design	44
4.2.3	Frontend Organization	45
4.3	Scoring Model	47
4.3.1	Sanctions Scoring	48
4.3.2	Registries Scoring	49
4.3.3	Adverse Media Scoring	50
4.3.4	Other Scorings	51
4.4	Explainability and Evidence Tracking	52
5	Results and Discussion	55
5.1	Evaluation Setup, Labeled Benchmark, and Representative Scenarios	55
5.2	Benchmark Results and Example Checks	57
5.3	Discussion and Observed Limitations	62
6	Conclusion	63

Chapter 1

Introduction

1.1 Background and Motivation

Compliance risk scoring is becoming more important because companies deal with many partners and banks across borders, and even small mistakes cause compliance, operational, and reputational problems. In practice, people often rely on "black box" checks or just some screenshots from different websites, but later it is hard to explain why a company was flagged or what exact data led to this result. The main issue is that OSINT data is heterogeneous, noisy, and becoming more fragmented in 2026, which makes it harder to cross-compare and verify facts [20]. For example, the same company name can be written in many different ways because of language differences, spelling variations, or missing identifiers, like BIN (Business Identification Number) or LEI (Legal Entity Identifier) might be missing, which causes entity ambiguity. This leads to false positives, where normal companies are flagged as risky, and also false negatives, where real risks are missed, causing serious problems for compliance work.

Because of these problems, there is a need for a system that is not a "black box" but is transparent and rule-based. A risk score alone is not enough, it should be supported by facts like a sanctions list match, a registry record, or news mentions with a source and date. The project integrates data from 7 different sources. In addition to this, it keeps strict provenance so every score can be traced back to the original record. By storing when the data was collected and where it came from,

the results become reproducible and easier to defend, which is critical for reducing unnecessary restrictions for low-risk businesses. The proposed approach in this thesis considers sanctions evidence as the strongest signal, while registry and media signals are secondary contextual factors.

1.2 Problem Statement

The biggest technical issue is that compliance data is distributed across too many places, like the US, UN, or other databases, and they all use different formats, which is very messy and inconvenient. Because these sources use XML (Extensible Markup Language) or CSV (Comma-Separated Values) with non standard structures, it is very hard to build one system that reads them all correctly without a lot of human help. This makes entity matching a serious problem, as a company name may have spelling variations or be written in a different language which leads to many false hits and wastes time for the analysts [21]. In addition to this, most tools today work like a black box, so they give a risk score but do not show the exact logic or the calculation trace, making it difficult to verify the result later if needed. Without a clear way to see the evidence and the steps taken to get the score, the whole process remains unreliable and hard to defend during a real audit.

1.3 Scope and Assumptions

The thesis is guided by one main research question: how can heterogeneous Open-Source Intelligence (OSINT) sources be integrated into an explainable and auditable risk scoring workflow for companies and individuals? This question is divided into three supporting research questions. First, how can sanctions lists, public registry data, and adverse media signals be normalized and combined in a single backend pipeline? Second, how can evidence tracking, source snapshots, and provenance metadata make the produced risk score transparent and reproducible? Third, can a transparent rule-based scoring model distinguish known high-risk or sanctioned en-

tities from clean entities in a small manually labeled benchmark? Based on these questions, the thesis assumes that the system should be evaluated not only by showing representative cases, but also by checking whether its outputs match expected labels in a controlled benchmark.

The overall plan for the platform is to focus on risk scoring for various companies and individuals by using only publicly available sources, such as international sanction lists and news mentions. In general, the platform will be built as a modular pipeline using NestJS and PostgreSQL that separates different tasks, such as parsing, data fetching, and scoring into separate modules. It will connect to 7 different OSINT sources including OFAC, EU, UN, UKSL, AFSA, GLEIF and GDELT to gather various risk signals. The plan is to use a rule-based model to give a risk score between 0 and 100, while keeping the logic transparent, instead of using a "black box" approach [20]. The scope also includes a full calculation trace and evidence tracking, so that every result can be checked and repeated later if needed. It is assumed that raw data will always be messy and have many typos or language variants, so the platform is designed to handle these inconsistencies through name normalization and prefix matching. Another plan is to keep snapshots and SHA-256 (Secure Hash Algorithm 256-bit) hashes of all raw files, which are necessary to make sure the findings remain auditable for future compliance reviews.

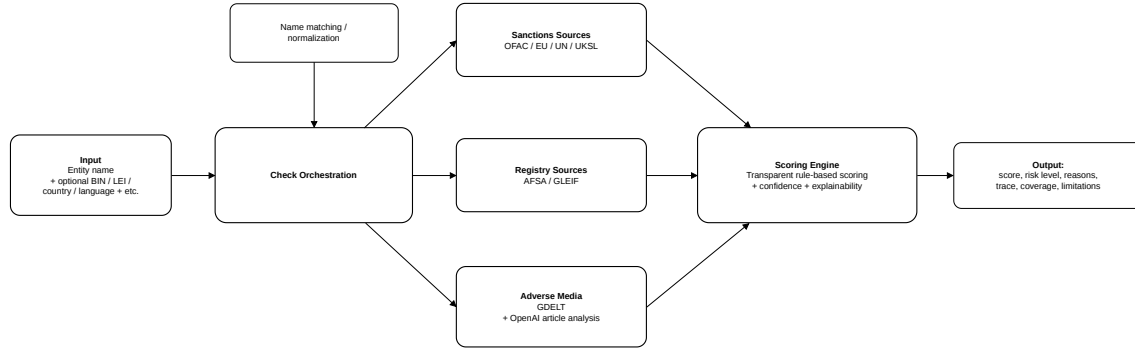


Figure 1-1: High-level architecture of the proposed explainable OSINT-based company risk scoring system. The workflow begins with an entity name and optional identifiers, passes through check orchestration and name normalization, then integrates sanctions, registry, and adverse-media sources, and finally produces a transparent rule-based output including score, risk level, reasons, trace, coverage, and limitations.

Chapter 2

Literature Review

2.1 The Importance of OSINT

The first related work is the research done by Rajamaki et al., where they examined how publicly available web data can be used to validate what risks a corporation possesses, particularly in the area of cyber security [1]. To be more precise, their strategy consisted of three steps: first of all, define the information requirements, then go online and see what is available (social media, search engines, etc.), and then go through all of this data and determine what is useful. They used tools like Maltego and SpiderFoot to help map out company networks and identify people, using data found online and from social media. Their findings show that it is in fact possible to discover risks and weaknesses of an organization by just doing a methodical search of the publicly available information available on the Internet. By contrast, not all online information can be considered credible and, therefore, analysts must verify facts and compare outcomes with various sources. In addition to this, there are also legal and privacy limitations which are related to collecting some form of information and those must be taken into account. Although the human judgment is critical in determining the important findings, even with the help of the software tools, a significant part of the work can be streamlined. Overall, the paper demonstrates that OSINT, when used properly, is a very helpful method of raising concern over the risks that could have otherwise gone unnoticed.

The next work is the paper by D. R. Hayes and F. Cappa, describing how they can assist companies in identifying threats and preventing cyberattacks with the help of OSINT tools [2]. They conducted a vulnerability scan on a certain company that belongs to the U.S. electrical grid. The team used OSINT to characterize the network software, hardware and primary IT employees used at the company. Through searching the Internet, such as legal websites and social platforms, they were able to identify the possible vulnerabilities in the tech and processes of the firm, which internal analysis could overlook. They revealed that OSINT can enhance security by identifying threats that cannot be easily detected by normal internal controls. In addition to this, they imply that organizations should implement OSINT to analyze technical and human conditions and maintain their security regulations. Another conclusion of the study is that corporate organizations must find a balance between all these endeavors and privacy and ethics since certain open-source data may be personal and sensitive. Overall, D. R. Hayes and F. Cappa prove the effectiveness of OSINT in risk assessment and persuade companies to maintain such tools in the continuous vulnerability tests.

2.2 Sanctions Screening of Entities and Individuals

According to Thomson Reuters, sanctions screening is an important and resource-intensive process that helps firms screen their customers, vendors, partners, or intermediaries against watchlists [3]. To be more precise, these lists consist of individuals, entities, vessels, and countries subject to legal or economic restrictions that can be associated with crimes like terrorism, money laundering or fraud. Businesses need to make decisions, such as blocking transactions or breaking relationships in order to remain compliant when a match is found. The article describes the implementation of digital systems of screening that scan all parties concerned on a regular basis, as a way of allowing firms to avoid large fines and reputational damage in case they unknowingly do business with a sanctioned organization. Another aspect that the article highlights is the need to update and maintain lists as accurate as possible

since new names may come up or be dropped off any time as circumstances shift all over the globe. Practically, the screening procedure includes gathering names and other information about business partners and subjecting them to automated software and letting compliance personnel evaluate possible matches. As a result, this two step approach of integrating technology with human verification would result in fewer mistakes, and no company stumbling upon breaking international regulations. The authors emphasize that it is necessary to establish effective sanctions risk management through proper employee training and policies.

The next is the article by The Financial Crime Academy, which discusses sanctions screening as a part of the overall compliance program of the company [4]. It elucidates that in most countries, screening against sanctions lists is not only a good idea but also a mandatory requirement of many industries particularly banks and financial firms. The authors underline that it is necessary to screen all the customers, suppliers and even other business partners and the screening should be repeated on a regular basis as the sanctions lists change very frequently. Their guide notes that not doing due checks may lead to punishment, criminal prosecution or bring damages to the reputation of a company. The article continues by detailing how screening can be done more productively in practical terms using a robust IT infrastructure, record-keeping, and regular training of personnel on how to identify suspicious activity. They state that the company has to establish various roles and responsibilities with clarity so that everybody knows who does what in the event of an eventual match. In addition to this, the authors also remind that compliance does not solely concern technology, but it requires human judgement to explore potential fits and make the correct decision, and thus, it should be reviewed and improved on a regular basis.

2.3 Reputation and Risk Scoring Approaches

Moreover, Yildirim and Uslu examined the determination of the corporate reputation scores and the relationship between the scores and financial performance [5]. To be more precise, their analysis relied on reputation scores released by the Reputation

Institute that rely on a number of major factors, like leadership, performance, products and services, innovation, workplace culture, governance, and citizenship. The researchers selected the companies which were among the top 10 in the list of global reputation reports more than 15 years and conducted comparison with the values of market value and financial statistics of each of the companies. The findings indicate that those companies that score higher on reputation tend to have better long-term financial performance and confirm previous studies which have found that reputation and performance are related. One of the main characteristics of the approach is that both qualitative and quantitative data were used in order to appropriately assess the methodology. More specifically, it uses surveys of experts and ratings by the people and converts these into a simple reputation score of each company. The authors mention that although the pulse scores provide a definite figure, it is necessary to keep in mind that the brand value, popular opinion, and company activity can have an impact on the findings. The paper concludes that good reputation scoring can assist organizations and investors to have a more accurate insight on the connection between public perception and financial gains and future development.

The next source is the work done by Flagright, a practical account of how risk scores are computed and used by businesses, particularly in assessing customers or partners [6]. More specifically, risk scoring systems take into account a variety of aspects, including the background, location, financial history, and any warning indicators of a customer based on previous transactions. The factors are weighted and subjected to some sort of a formula that produces a score that indicates the extent of risk that the customer can bring to the firm. The article shows the importance of risk scoring being adaptable, constantly revised and one that is closely aligned to the real products of the company, geography as well as its risk possibility. In addition to this, transparency and realistic risk estimation also are emphasised by the authors. They advise that companies should not only use the static models or old data but continuously update their scores when new information is received like when a new rule is implemented or an issue of recognized fraud cases. The article also states that risk scoring is not only reducing loss or avoiding compliance issues, but it is also the

cause of risk to promote trust and safer business decisions in the company and to its clients.

2.4 Explainable AI (Artificial Intelligence) and Interpretability

In the paper by Pathak et al., the authors examined an AI system that is supposed to simplify the credit risk assessment process [7]. To be more precise, the authors contrast logistic regression with more advanced tools such as random forests and use explainability techniques like SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) which are effective methods of examining black-box models in more detail. The interesting fact is that the models can actually be more successful in identifying high-risk cases, but the transparent processes assist in seeing deep reasons, which matters a lot in risk assessment when we cannot afford to blindly trust the black box. The authors emphasize that it is very important to add the explainability of the models, particularly, in areas where regulators want to know everything transparently. According to them, they revealed that when interpretability is incorporated into models, decision-makers receive the reasons behind predictions so that they can understand why certain predictions are made, rather than just trusting a black-box system. This approach also helps in meeting regulatory expectations and ensures that AI-based risk assessments are both effective and fair, making the case for explainable AI in everyday business decision-making.

Furthermore, this work by MarkovML shows the role of two large explainability techniques, LIME and SHAP, in assisting us to understand ML models [8]. More specifically, LIME essentially constructs a simple local model which mimics the prediction of a single instance indicating what features are most important to that instance. On the other hand, SHAP is based on the game theory and calculates the sum of each input on a global and local level. SHAP according to the authors provides a more

detailed attribution and can be used in both data-wide and individual explanations, but it is slow, particularly when interacting with larger models. In addition to this, the article notes that LIME is typically quicker and easier to compute on individual explanations, whereas SHAP provides users with a big picture of the large dataset. When applying AI to estimate risk or reputation score of a company, these tools allow users to see clearly how a model’s logic lines up with business understanding. The authors recommend SHAP when the complete transparency is required, and LIME when only quick insights are needed, emphasizing when risk analysis projects have clear results, which are easy to interpret, it enhances trust in the project.

2.5 News Sentiment Analysis

According to Moody’s Analytics, news sentiment analysis is essentially a reliable tool in determining the market risk and predicting the business results [9]. The report shows how by examining the sentiments in news, social media posts, and other written content, analysts acquire preliminary indications on possible risks. It is a kind of a heads-up regarding the negative news of the company before it affects stock prices or brand reputation, and the businesses have a chance to respond. In addition, the authors discovered that when there is a rise in negative sentiments, it is typically an indicator of decreasing stocks or even more volatile markets, and therefore sentiment analysis can be considered as a good addition to risk management. Furthermore, the article also discusses how the existing trading strategies are increasingly becoming sentiment-based. The authors state that when sentiment information is used, in fact, the predictions in standard financial models are improved. By combining the sentiment of news and traditional metrics, firms can be able to make more informed decisions and avoid expensive risks later.

Moreover, the research done by Casigay explains how the newswire and web publication news sentiment scores can really be used to inform the market risk of publicly traded companies [10]. To be more precise, they construct a sentiment scoring model based on years of headlines of S&P 500 companies and inputs those scores into a

Value-at-Risk (VaR) model. The results indicate that when sentiment analysis is added to the analysis, the extreme-loss forecasts become a little sharper, even when typical financial predictors are held unchanged. In addition to this, they support it using statistical instruments such as quantile regression to disaggregate the unique risk forecasting movement that sentiment makes. The study suggests that the improvements are not large, yet sentiment scores do continue to contribute to risk-sensitive decisions. The inclusion of sentiment as an additional component in standard risk models will allow analysts to predict large swings, which result in surprising news or abrupt changes in investor sentiment. They also conclude by indicating that news sentiment analysis is a helpful complement to both quantitative risk managers and policymakers who want to be informed about risk in a fluctuating market.

Chapter 3

Datasets: OSINT Sources for Explainable Company Risk

3.1 Overview

This chapter describes the datasets used to build an explainable, OSINT-based risk scoring pipeline for various companies. The overall idea is to combine three large families of public data, including watchlists, company registries and adverse-media streams while keeping strict provenance so every score is traced back to an original record. The selection of these sources follows practical rules, such as being relevant to compliance risk and having stable identifiers or fields for matching. To make the whole system transparent the ingestion process records a link to the exact source page or API (Application Programming Interface) together with a retrieval timestamp and a SHA-256 hash so that any result can be reproduced later by a reviewer. This way the foundation for the scoring is stable and can actually be used in practice for transparent company risk screening. It will connect to 7 different OSINT sources including OFAC, EU, UN, UKSL, AFSA, GLEIF and GDELT to gather various risk signals.

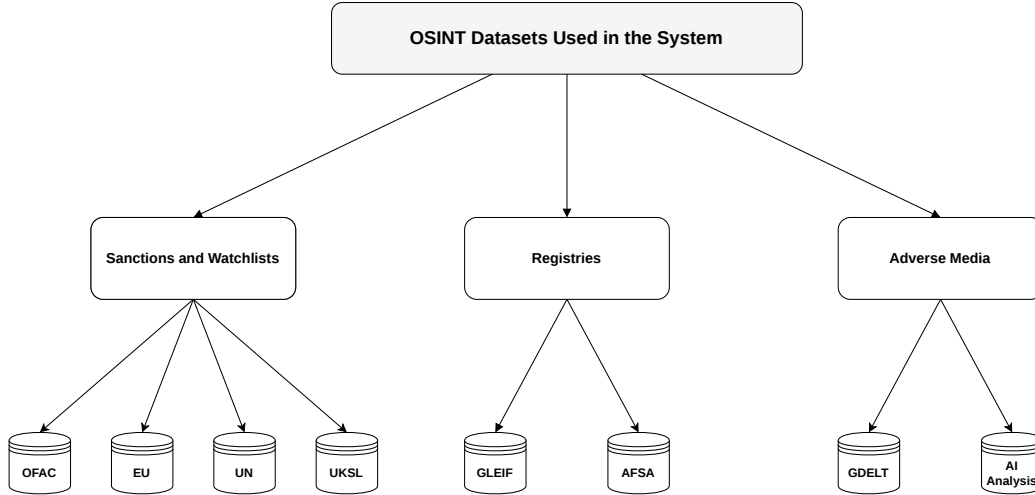


Figure 3-1: Overview of the main OSINT dataset categories used in the system. The selected sources are grouped into three evidence families: sanctions and watchlists (OFAC, EU, UN, and UKSL), registries (GLEIF and AFSA), and adverse media (GDELT with optional AI-based analysis). This grouping reflects the multi-source design of the platform, where different dataset types contribute complementary signals to the final company risk assessment.

3.2 Sanctions and Watchlists

3.2.1 OFAC SDN (Specially Designated Nationals) List

The OFAC data is handled by using two local files, which are `sdn.csv` and `alt.csv`, consisting of 18,637 SDN records and 20,182 alias records, respectively [11]. The system parses these sources together at the same time. The main `sdn.csv` is a headerless CSV file, where the parser reads the entity number (`ent_num`), entity name, raw OFAC type and the sanctions programs by looking at the column index. Subsequently, the `alt.csv` file is used to find the alternate names and each row links back to the main record with that same `ent_num`. This makes the dataset different from other sources, because the aliases are not in the same file and must be joined together from the second source. As a result, the OFAC source produced a total of 38,815

searchable name strings in the system. The system also deals with technical issues like the files being Latin-1 encoded and it strips out uncommon EOF (End-of-file) characters before parsing. In addition to this, in order to handle the missing data in `sdn.csv`, rows are skipped if they contain fewer than two columns or with empty `ent_num`/`name` columns. As regards the `alt.csv`, rows with fewer than 4 columns or with empty `ent_num`/`alt_name` were skipped. After that, the raw type is normalized into categories like individual, entity, ship, or unknown. The aliases are deduplicated using normalized lowercase so the names don't repeat.

Table 3.1: Structure of the OFAC dataset used in the system

Component	Format	Main fields used	Role in the system
<code>sdn.csv</code>	Headerless CSV	<code>ent_num</code> , <code>name</code> , <code>typ</code> , <code>programs</code>	Stores the primary OFAC SDN records. The parser reads the file positionally, extracts the entity number as the stable key, the main display name, the raw subject type, and the sanctions program information.
<code>alt.csv</code>	Headerless CSV	<code>ent_num</code> , <code>alt_num</code> , <code>alt_type</code> , <code>alt_name</code> , <code>remarks</code>	Stores alternate names linked to the main SDN entries through <code>ent_num</code> . During parsing, aliases from this file are attached to the corresponding record from <code>sdn.csv</code> and then deduplicated before storage.

Inside the system, every entry from OFAC dataset is turned into a `ParsedSanctionsRecord`, where the `ent_num` becomes the stable key and the main name becomes the display name. The searchable names are all put into normalized name and aliases fields. The raw payload keeps the original row data, the raw OFAC type, the sanc-

tions program string, and detailed alias objects from the dataset `alt.csv`, including alternate number, alias type, alias name and remarks. When the data is ingested, it is stored in a snapshot model, where the files are copied into local storage, the hashes of the files are calculated, provenance metadata like row counts are attached to the snapshot. The records are added to the database in an atomic transaction. Later, when a user runs a check, the system loads the OFAC snapshot and compares the query against all the names to find hits. Due to the fact that OFAC has the highest weight, it is one of the strongest metrics that changes the final risk score in the system.

Table 3.2: Profile of the OFAC dataset used in the system

Metric	Value	Interpretation
SDN rows	18,637	Number of rows in the primary <code>sdn.csv</code> file.
Alias rows	20,182	Number of rows in the <code>alt.csv</code> file containing alternate names.
Entities with at least one alias	8,699 (46.7%)	Share of OFAC entities that have one or more alternate names.
Entities without aliases	9,938 (53.3%)	Share of OFAC entities represented only by a primary name.
Total searchable names	38,815	Combined number of primary and alias name strings available for matching.
Average aliases per entity	1.08	Mean alias count across all parsed OFAC entities.

Metric	Value	Interpretation
Maximum aliases on one entity	95	Highest alias count observed for a single OFAC entry.

3.2.2 EU Consolidated Sanctions List

The EU data is a bit different, because it comes from a single XML file instead of two CSVs like the OFAC dataset [12]. The system uses a library called "fast-xml-parser" to read the full document. The root element of this file is called "export" and it consists of "sanctionEntity" elements, where the main sanctions information is stored. To be more precise, there are a total of 5,860 "sanctionEntity" records and all of them were successfully parsed without discarding any entities. The parser had to find names inside nameAlias fields and used some fallbacks like wholeName, wholename, and #text in order to handle possible differences in how the XML fields are stored and represented after parsing. In addition to this, the file contains 29,809 nameAlias elements, which after the deduplication process, produce 29,442 unique searchable names. This means that the dataset provides the system with a large number of searchable names, with the average of 5.02 name options for each entity. Moreover, due to the nested structure, the source can be considered as more hierarchical than other datasets. Subsequently, the parser extracts identifiers specific to the source, like euReferenceNumber, logicalId, unitedNationId, and designationDetails, while the generationDate and globalFileId are saved in the metadata of the snapshot.

Table 3.3: Profile of the EU sanctions dataset used in the system

Metric	Value	Interpretation
Root element	export	Top-level XML structure of the dataset.

Metric	Value	Interpretation
Record element	<code>sanctionEntity</code>	Main XML element representing one sanctioned entity.
Total record elements	5,860	Number of <code>sanctionEntity</code> elements in the examined snapshot.
Total <code>nameAlias</code> elements	29,809	Raw number of name and alias entries present in the XML file.
Unique names after deduplication	29,442	Number of searchable names preserved after per-entity case-insensitive deduplication.
Average names per entity	5.02	Mean number of searchable names associated with one EU sanctions record.
Records classified as persons	4,341 (74.1%)	Records mapped to <code>person</code> through <code>classificationCode</code> .
Records classified as entities	1,519 (25.9%)	Records mapped to <code>entity</code> through <code>classificationCode</code> .

One technical part of the EU parser is how it finds out if a record is a person or an organization. The parser identifies subject type through `subjectType` and nested `classificationCode` inside `subjectType`, where P is mapped to person and E to entity. After all these processes, each entity is converted to the simple `ParsedSanctionsRecord`. Later, these normalized EU records are used in the same matching workflow as the other sanctions sources and contribute to the sanctions component of the final risk score with their own source weight.

Table 3.4: Extracted fields from the EU sanctions dataset

Field / element	Role	Use in processing
euReferenceNumber	Primary source identifier	Main stable key for the parsed record.
logicalId	Fallback identifier	Used if the primary identifier is unavailable.
nameAlias	Name and alias container	Provides display name and searchable name variants.
subjectType	Direct subject-type field	Used to classify records as person or entity.
classificationCode	Nested type code	Maps values such as P and E to internal subject types.
unitedNationId	Cross-reference identifier	Preserved in the identifiers structure.
designationDetails	Designation metadata	Stored as source-specific identifier information.
generationDate	Dataset metadata	Used for provenance of the snapshot.
globalFileId	Dataset metadata	Preserved for traceability of the imported source.

3.2.3 UN Consolidated Sanctions List

The UN dataset is implemented in the system as a local XML-based sanctions source. The dataset contains 999 sanctions records, 726 of which are individuals and 273 are entities [13]. As a result of identifying the alias names, 4,109 searchable names are obtained from the dataset, including 999 primary names and 3,110 deduplicated aliases. The parser looks for a root element called CONSOLIDATED_LIST, and then it splits the data between INDIVIDUALS and ENTITIES. This means the subject type is not

obtained heuristically from the metadata like in the EU source, but is known from the structural organization of the UN XML file. The display name is built by joining fields FIRST_NAME, SECOND_NAME, THIRD_NAME, and FOURTH_NAME. Despite being empty in every element, there is a field ENTITY_NAME for entity records, which was replaced by the joint display name. In addition to this, the parser also collects aliases from nested alias blocks, including regular alias names and names written in not a Latin script. Moreover, the parser had to deal with nested VALUE tags recursively, because some XML fields do not store their content directly but wrap it inside child nodes. This makes the UN parser the most specific one structurally in the whole project.

Table 3.5: Profile of the UN sanctions dataset used in the system

Metric	Value	Interpretation
Total records in one file	999	Total number of sanctioned records in the examined snapshot.
Individuals	726 (72.7%)	Records stored under the INDIVIDUALS container.
Entities	273 (27.3%)	Records stored under the ENTITIES container.
Deduplicated aliases	3,110	Alias names preserved after normalization and duplicate removal.
Total searchable names	4,109	Combined number of primary and alias names available for matching.

Metric	Value	Interpretation
Average names per record	4.11	Mean number of searchable names associated with one UN sanctions record.
Non-Latin-script names	378	Number of original-script aliases preserved in the parsed data.
Committee coverage	14 sanctions committees	The snapshot includes records from 14 distinct UN sanctions committees.

After everything is parsed the UN records are moved into the same sanctions structure used by the rest of the platform. This helps to normalize the messy XML into something that is convenient to search. The stable key comes from the REFERENCE_NUMBER or the DATAID as a fallback, and the extracted identifiers also keep other fields like listed_on and last_updated. Instead of just keeping names the system stores the whole original XML node in a payload so everything is traceable later. All these records are added to the database in an atomic transaction with provenance metadata. Later, when the user runs a check the system uses the name matching module to find hits that change the final risk score.

Table 3.6: Key elements of the UN sanctions dataset used in the system

Element	Role in the system
CONSOLIDATED_LIST	Root XML structure of the UN sanctions dataset.
INDIVIDUALS / ENTITIES	Separate containers for sanctioned individuals and sanctioned entities.
REFERENCE_NUMBER / DATAID	Main identifiers used to construct stable keys for parsed records.

Element	Role in the system
Name fields	Display names are assembled from multiple name parts <code>ENTITY_NAME</code> .
Alias fields	Alias blocks provide alternative searchable names, including original-script variants.
<code>LISTED_ON</code> / <code>LAST_DAY_UPDATED</code>	Record-level metadata preserved in the normalized output.
Nested <code>VALUE</code> tags	Require recursive extraction of text content during parsing.
Raw XML payload	The original parsed UN node is preserved for traceability and later inspection.

3.2.4 UK Sanctions List

The UKSL data comes from a single CSV file called UK-Sanctions-List.csv, with a size of 44.6 MB and a total of 55,737 data rows [14]. By contrast, after grouping the records by Unique ID, these rows correspond to 5,738 distinct entities, which shows that the raw CSV row count is much larger than the true entity count. Unlike other sanctions datasets, the biggest technical difficulty of this source is that one sanctioned person or company can have many rows in the table that must be joined together before a single record is made. After parsing and deduplication, the dataset contains 18,020 unique searchable names, with an average of 3.14 names per entity.

Table 3.7: Profile of the UKSL dataset used in the system

Metric	Value	Interpretation
File size	46,727,260 bytes (44.6 MB)	Largest sanctions source file used in the prototype.
Header columns	58	Total number of columns detected in the CSV header.

Metric	Value	Interpretation
CSV data rows	55,737	Raw number of data rows in the file, excluding the preamble and header row.
Distinct entities	5,738	Number of unique sanctions subjects after grouping rows by <code>Unique ID</code> .
Rows with assembled names	55,668	Rows from which a name could be constructed from the six name fields.
Final searchable names	18,020	Distinct searchable names preserved after the two-stage deduplication process.
Average names per entity	3.14	Mean number of searchable names associated with one UKSL record.
Individuals	3,868 (67.4%)	Records classified as <code>Individual</code> through the <code>Designation Type</code> field.
Entities	1,288 (22.4%)	Records classified as <code>Entity</code> .
Ships	582 (10.1%)	Records classified as <code>Ship</code> .

At the record level the parser groups all the rows together by the Unique ID which becomes the stable key for the entry. To get the display name it joins six different fields, where Name 6 is treated as a prefix and put at the front of Name 1 through Name 5. It also extracts other fields like OFSI (Office of Financial Sanctions Implementation) Group ID, UN Reference Number, Designation Type, Designation Source, Regime Name, Sanctions Imposed, Name Type, Alias Strength, and not a Latin script name fields. Since there are many rows for one subject the parser puts everything into sets so it keeps all the different regimes and aliases inside the final

record. Subsequently, the UKSL records are turned into the same format as the rest of the system where the identifiers and the full raw payload are stored in the database. These records are saved using snapshots and then used for matching and scoring where the UKSL has its own weight for the final score.

Table 3.8: Key elements of the UKSL dataset used in the system

Element	Role in the system
Unique ID	Main identifier used for grouping rows and constructing the stable key of the parsed record.
Grouped CSV rows	Multiple rows belonging to the same Unique ID are merged into one internal sanctions record.
Name fields (Name 1–Name 6)	Used to assemble the display name and collect searchable name variants.
Name Type / Alias Strength	Preserve additional metadata about the type and strength of each name variant.
Designation Type	Used as the main indicator of subject type in the normalized record.

3.3 Registries

3.3.1 AFSA Public Register

As the registry enrichment source, the AFSA public Register was used in the system for the companies that have a BIN (Business Identification Number) [15]. Unlike sanctions datasets that are read from local files, AFSA is accessed through live HTTP (HyperText Transfer Protocol) requests to the public AFSA register website when the BIN is provided. The data that is returned is an HTML (HyperText Markup Language) response, not a structured file like CSV, XML, or JSON (JavaScript Object Notation), which makes it technically different from the other sources in the whole

project. After getting the response, the raw HTML is cached and saved as a local snapshot, so the system can keep the original registry evidence for later if the user needs to check it for traceability.

Table 3.9: Structure of the AFSA profile extracted by the system

Profile group	Description	Example extracted fields
Core attributes	Main company registration details parsed from the key-value table on the AFSA detail page.	BIN, Company name, Registration status, Registration date, Registered address
Roles	Corporate role information parsed from a classified matrix-style table.	Role, Name, Effective date
Activities	Operational activities associated with the company.	Code, Activity
Shareholders	Shareholder-related information extracted from the AFSA page.	Name, Status, Comments
Share classes	Share class information, excluding summary rows marked as totals.	Class name, Number of shares, Currency
Collateral	Collateral-related information when such a section is present.	Under collateral / related collateral fields

The AFSA parser parses and processes the HTML with a library called "cheerio" and uses heuristic table detection in order to turn the page into a structured company

profile. First of all, it looks for the core table by searching for a table that has fields like the BIN, the Company Name, or the Registration Status, and then it reads that table as key-value pairs. After this process, the other tables on the page are sorted into different groups based on what the headers mean, such as the important roles in the company, company’s activities, shareholders, share classes, and collateral. Each of these parts is parsed as a matrix-style table, which allows the system to keep the core registration data and all the other corporate information found on the AFSA website. As a result, it can be said that AFSA doesn’t just provide a simple confirmation of whether a company is registered, but it also gives a more detailed view of the legal and operational structure for the entity. After parsing, AFSA’s response is stored through the system’s snapshot mechanism. To be more precise, the returned HTML page is kept locally, a SHA-256 hash is also calculated, and the parsed company profile is persisted as an auxiliary ParsedRecord with recordKind = 'registry_profile'. When the scoring is done, the information from AFSA helps the registry part mostly with two signals, like if the registration status is active/non-active, it adds 15 points to the risk score. On the other hand, if there is any collateral found it adds 5 points. Therefore, due to the very limited impact on the scoring, the information from this source is mainly used in order to show more information about the entity in the platform.

3.3.2 GLEIF Register

The second registry source the system uses is GLEIF and it gets data about the Legal Entity Identifier (LEI) through a live JSON API [16]. This one is different from AFSA because it works in two modes and a BIN or LEI is not required to start it. To be more precise, if the input already has an LEI, then the system just does a direct lookup to find the record right away. However, if there is no LEI, it can also search by the name of the company and even use a country hint if it was given in the check input to make the results better. Because GLEIF integration has this dual access model, it works both as a way to verify identifiers and also as an enrichment tool for companies when the LEI is not known yet.

Technically GLEIF is easier to process than AFSA, due to the fact that it returns a structured JSON:API format, rather than raw HTML. The backend has a special client to query the lei-records endpoint and then it uses a mapper to flatten that nested JSON into a simpler internal version. The extracted fields are the LEI itself, the legal name, entity status, registration status, jurisdiction, legal form, legal address, headquarters address, initial registration date, last update date, and the LEI of the managing Local Operating Unit. This mapping step is very important because it removes the extra fields from the JSON response, but keeps all the fields that are needed for checking identity and risk signals. After getting the data from GLEIF, it is saved using the same snapshot approach like with the other sources, where the raw JSON is stored as a source snapshot and the simplified record is written as an auxiliary parsed record with the recordKind set to registry_profile.

Table 3.10: Main fields extracted from the GLEIF JSON response

JSON:API source path	Flattened field	Role in the system
<code>item.id</code> or <code>item.attributes.lei</code>	<code>lei</code>	Primary identifier of the legal entity and stable key of the parsed record.
<code>item.attributes.entity.legalName</code>	<code>legal_name</code>	Canonical legal name used for display and matching.
<code>item.attributes.entity.status</code>	<code>entity_status</code>	Indicates whether the entity is operationally active or inactive.
<code>item.attributes.registration.status</code>	<code>registration_status</code>	Indicates the current status of the LEI registration.

JSON:API source path	Flattened field	Role in the system
<code>item.attributes.entity.jurisdiction</code>	<code>jurisdiction</code>	Jurisdiction code associated with the legal entity.
<code>item.attributes.entity.legalForm</code>	<code>legal_form</code>	Legal form identifier or textual description.
<code>item.attributes.entity.legalAddress</code>	<code>legal_address</code>	Structured legal address of the entity.
<code>item.attributes.entity.headquartersAddress</code>	<code>hq_address</code>	Structured headquarters address of the entity.
<code>item.attributes.registration.initialRegistrationDate</code>	<code>initial_registration_date</code>	Date when the LEI was first registered.
<code>item.attributes.registration.lastUpdateDate</code>	<code>last_update_date</code>	Date of the most recent LEI record update.
<code>item.attributes.registration.managingLou.lei</code>	<code>managing_lou</code>	Identifier of the managing Local Operating Unit.

As regards the scoring, GLEIF helps the registry part of the total score with two main signals, first one looks if the LEI registration status is not active and the second one checks if the entity status itself is not active. These two things can add up to 18 points total which makes GLEIF a bit less influential than AFSA for the registry score. One important detail is that in search mode, only the first three GLEIF results are considered for the scoring and they use deduplication so the score doesn't get multiplied if many records show the same problem. This means even if there are several records with the same status the points are only counted once.

3.4 Adverse Media

The adverse media part of the system uses GDELT, which is a live JSON API and not a local dataset like the other ones mentioned before [17]. This makes it different, because the system doesn't keep a whole copy of all the dataset of news, but just gets articles for each check depending on the query. Thus, this media data should be considered as a dynamic stream of recent news instead of a static list of records. The repository evidence also shows this clearly because there is no one master file for it but only cached GDELT responses and snapshot files that came from individual searches.

More precisely, at the record level, each GDELT article is a compact JSON object that has the fields such as URL, mobile URL, title, seen date, image, domain, language, and source country. Nevertheless, the internal mapper does not keep all of these fields and reduces every article to five main fields, which are the seendate, sourceCountry, title, url, and a normalized domain derived from the URL. This means the data the system uses is smaller than the raw API response from GDELT, because it only focuses on the fields that are useful for traceability and simple text based analysis. Fields such as the image link, mobile URL, and language are dropped during the normalization process.

Table 3.11: Main fields of the adverse media dataset used in the system

Raw GDELT field	Internal field	Role in the system
title	title	Main textual field used for adverse-media keyword analysis and later optional LLM (Large Language Model) evaluation.
url (Uniform Resource Locator)	url	Preserved as the main article reference and used in evidence links.

Raw GDELT field	Internal field	Role in the system
url	domain	Converted into a normalized source domain for article provenance.
seendate	seendate	Preserved as the timestamp of when the article was seen by the source.
sourcecountry	sourceCountry	Preserved as source-country metadata for the article.
socialimage	—	Discarded during normalization.
url_mobile	—	Discarded during normalization.

The search works in a way that the company name is put in quotes and the system looks for news in English language based on the input. The default lookback window is set to 14 days (can be increased or decreased by the user) and it takes the newest 10 articles, so the sample can be considered as relatively small. Due to these limitations, the media data is fresh, but it also means that older reporting or news that falls outside the cap won't show up in the final dataset for the score. There is also an optional OpenAI layer that can be turned on for the GDELT news, which analyses the article metadata such as title, URL, domain, and seen date [18]. This part is advisory and it evaluates entity relevance, adverse signal, sentiment, and confidence. The additional contribution is capped at only 8, so it doesn't change the final risk score very significantly.

Table 3.12: Main query characteristics of the adverse media source

Property	Value	Role in the system
Data format	JSON	Returned as structured article records rather than HTML pages.

Property	Value	Role in the system
Query type	Exact quoted entity name	The input name is enclosed in quotes before submission to the API.
Language filter	English only	Restricts the query to English-language sources.
Default lookback window	14 days	Limits the media sample to relatively recent articles.
Default maximum records	10 articles	Caps the number of returned records per query.
Sort order	<code>datedesc</code>	Prioritizes the most recent articles first.

Chapter 4

Methodology

4.1 System Workflow

A compliance check in the system starts up when a user sends a request with a company name and optional fields like a BIN, LEI or country hints if needed. This request is handled by the central check orchestration service which is the main part of the NestJS backend that manages the whole system. To be more precise, it creates a new check run in the PostgreSQL database and sets the state to RUNNING before the actual searching begins. From there the workflow follows one pipeline that combines offline sanctions matching with live data from registries and news. In this setup the company name is the most important part of the input, but the identifiers such as the BIN or LEI decide which branches of the workflow actually should get activated for the check.

The next part of the workflow is getting the data from all the sources. For the sanctions lists like OFAC, EU, UN, and UKSL the system does not use remote services during the check, instead it just finds the latest local snapshots that were already saved and loads those parsed records. This is different for AFSA, GLEIF, and GDELT data, because they are treated as live sources that get queried remotely. AFSA is queried only when a BIN is available, GLEIF performs either direct lookup by LEI or search by company name, and GDELT retrieves recent media articles related to the queried entity. This division between local and live sources is crucial, because it lets the

system keep results reproducible for the sanctions part, while also giving the user the newest information for the registry and media parts.

After the system gets all the data it changes the results into simple internal structures so they can be processed in the same way for all the sources. The sanctions records are already stored in normalized form, while the live sources are parsed on the fly. To be more precise, AFSA HTML pages are transformed into structured company profiles, GLEIF JSON responses are flattened into compact registry records, and GDELT article arrays are reduced to a small set of fields such as title, URL, date, and domain. Subsequently, when the records are ready the system checks the company name against all the sanctions names and aliases to see if there is a match, then it generates evidence for both hits and no-hits. In addition to this, the system also saves the raw responses as snapshots with metadata for provenance. This way the workflow does not just search for entities, but also keeps the exact source state used for the check which is good for later reproducibility.

In the final stage all the structured findings are passed to the scoring engine which calculates sanctions, registry, and media scores separately, and then joins them for the overall risk result. The output produced by the system is not just a single score that shows the risk level, but also an array of explanatory factors, evidence references, and when enabled, more verbose trace information about how the score was formed. After the report is ready, the check run is tagged in the database as COMPLETED or FAILED so the pipeline has a clear life cycle. Overall, the workflow of the system is designed not as a black-box risk classifier, but as a traceable sequence of retrieval, normalization, matching, evidence creation, and rule-based scoring steps.

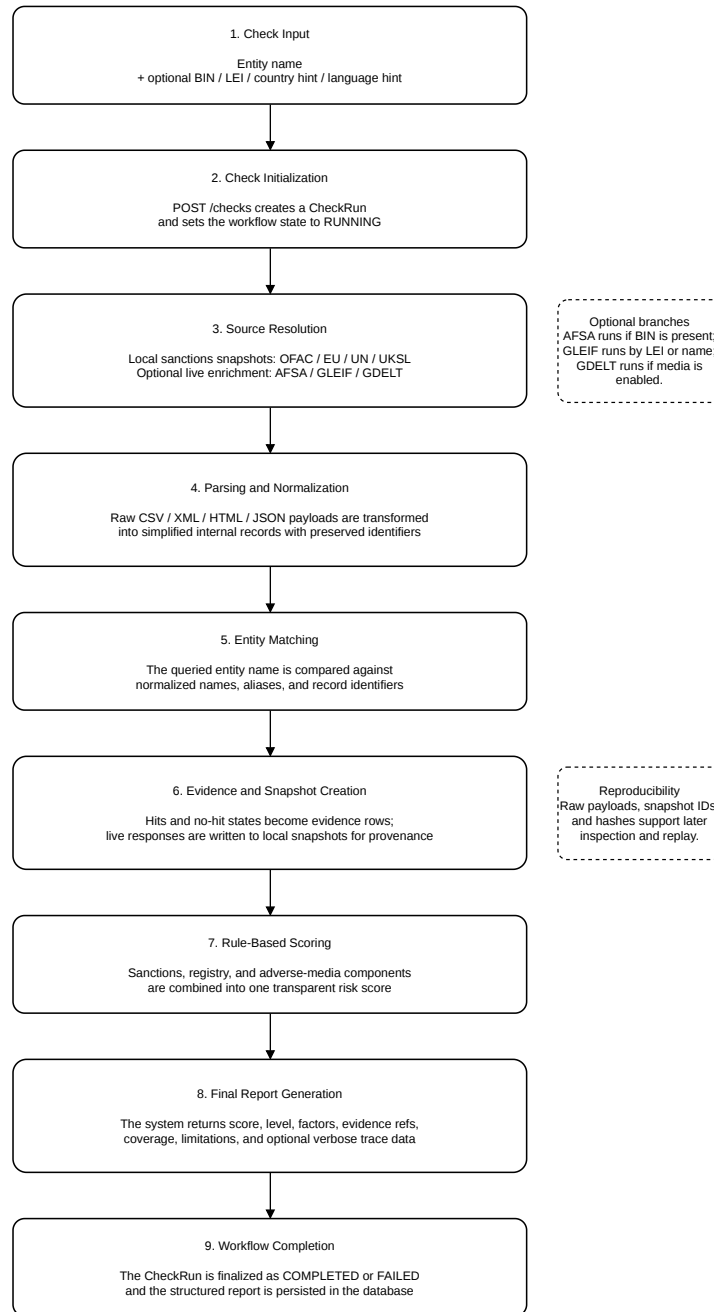


Figure 4-1: End-to-end workflow of an individual compliance check in the proposed system. The process begins with a company name and optional identifiers, proceeds through source resolution, parsing, normalization, and entity matching, then records evidence and source snapshots, and finally computes a transparent rule-based risk score before storing the completed report in the database.

4.2 System Architecture and Implementation Stack

4.2.1 Backend Architecture and Organization

The backend of the system is built as a single NestJS application using TypeScript and it works as the orchestration layer of the whole platform [22, 23]. Its role is not limited to serving API endpoints, because it also coordinates source retrieval, parsing, matching, scoring, evidence creation, and report assembly inside one backend process. Structurally the code is divided into separate modules such as checks, scoring, sources, adapters, and parsers, which makes it easier to maintain the system even though the OSINT sources are all very different. In our case, the ChecksService is like the central orchestrator, while other services do the HTTP requests, the mappers turn the raw payloads into internal structures, and the scoring service computes the final rule-based result. This modular design is important because the system has to work with CSV, XML, HTML, and JSON files without mixing all that logic together in the main part of the code.

Another important point regarding the backend design is how it splits up the offline and live sources. The sanctions lists are handled by a batch ingestion where the local files are parsed before and stored in the database as records, while AFSA, GLEIF, and GDELT are live and get queried right when someone starts a check. Despite the fact that these source families work differently, the backend uses the same pattern where it keeps the raw files and then turns the parsed fields into common records convenient for the scoring. In addition to that, the backend includes several implementation features that support reliability and auditability, such as deterministic filesystem caching for live HTTP requests, atomic ingestion transactions for sanctions data, and scheduled recovery of stale running checks. It is not just a set of API handlers, but the main engine that turns all this messy public data into a real risk screening process.

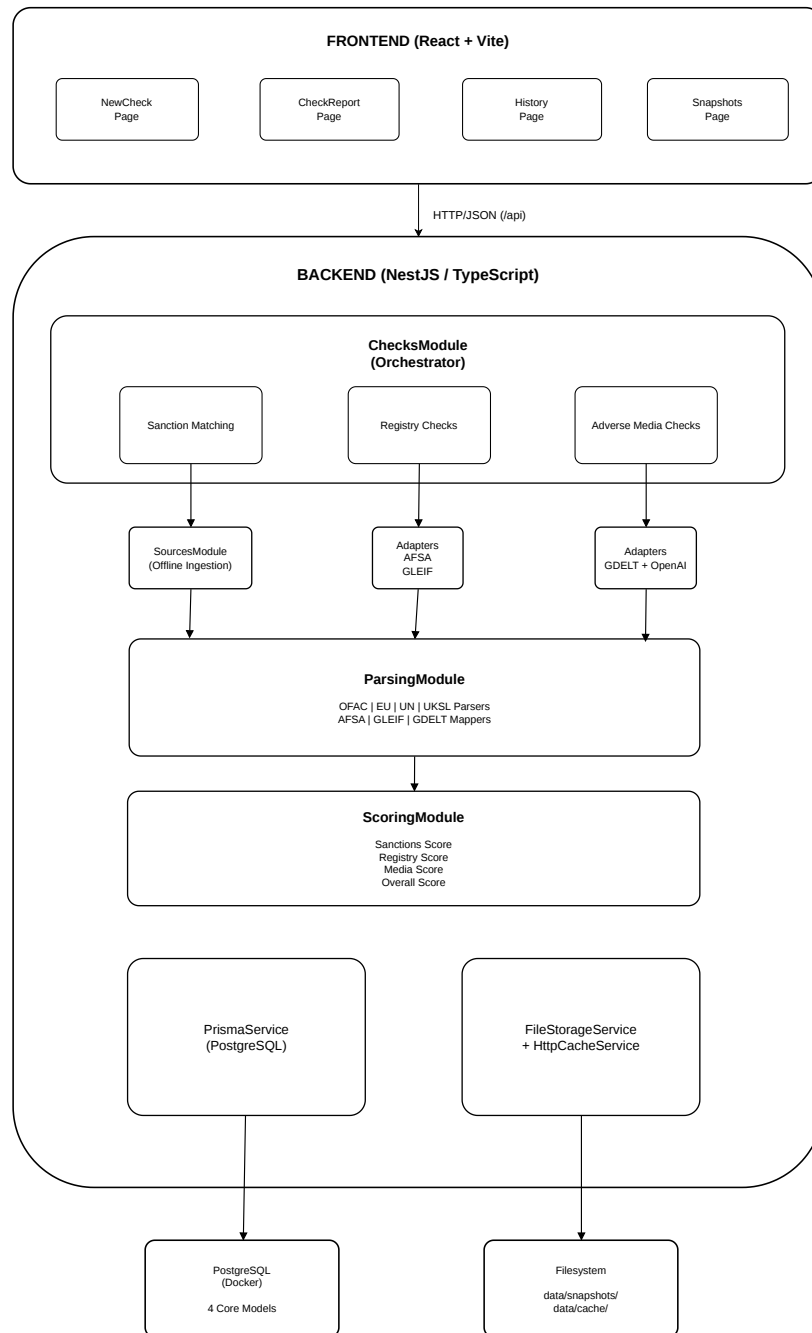


Figure 4-2: Component-level architecture of the proposed OSINT-based risk scoring system. The figure shows the interaction between the frontend interface, the modular NestJS backend, the PostgreSQL persistence layer, and the filesystem-based snapshot and cache storage, illustrating how source ingestion, live adapters, parsing, scoring, and report delivery are organized inside the platform.

4.2.2 Persistence and Storage Design

The database part of the system uses PostgreSQL 16 and Prisma ORM (Object-Relational Mapping) and its main goal is to keep the screening results and the source provenance for later [24, 25]. The design is centered on four main models which are SourceSnapshot, ParsedRecord, CheckRun, and CheckEvidence. SourceSnapshot saves a fixed copy of a source file or a live response with metadata like source type, version label, timestamps, MIME (Multipurpose Internet Mail Extensions) type, and SHA-256 hash. ParsedRecord stores a normalized version of the entity from a snapshot like the stable key, display name, subject type, identifiers, names, and raw payload. Then CheckRun represents one run of the workflow and it stores the input, the snapshots, and the final JSON report. CheckEvidence is where the atomic findings such as sanctions hits, registry information, adverse media results, and the no-hit outcomes are recorded.

The CheckRun model is the central execution record of the screening workflow. When a user submits a request through the POST /checks endpoint, the backend creates a new CheckRun row and stores the original input as JSON, including the entity name, optional identifiers such as BIN or LEI, country and language hints, and selected execution flags. The run is initially marked as RUNNING. During execution, ChecksService resolves the relevant sanctions snapshots, performs local sanctions matching, triggers live registry and adverse-media enrichment when enabled, creates evidence rows, and passes the collected findings to the scoring service. After the workflow is completed, the generated report is stored in the report JSONB field and the run status is updated to COMPLETED. If an error occurs, the status is updated to FAILED and the error message is preserved. The snapshotSelection field records which source snapshots were used during the run, including any user-provided snapshotOverrides, which allows the same check to be repeated against a known historical dataset state. Therefore, CheckRun is not only a log of the final score, but also the persistent container that links the input, selected data versions, evidence rows, provenance metadata, and final explainable report into one auditable screening record.

Beside the relational database, the system also uses the local files to store the raw sources and the cached web responses. Two parallel directory trees are maintained under `backend/data/`: `data/snapshots/` stores timestamped copies of source files and live payloads that were actually used in checks, while `data/cache/` stores deterministic cached HTTP responses for AFSA, GLEIF, and GDELT. Every file gets a SHA-256 hash when it is written and is saved in the snapshot metadata. This dual storage is important, as it joins the PostgreSQL database with the raw files on the disk. As a result, the system is auditable and reproducible, because it is possible to get the records in the database, but the original CSV, XML, HTML, or JSON files are still there if there is a need to check how the result was derived later.

Table 4.1: Main persistence components of the system

Component	Storage	Role
SourceSnapshot	PostgreSQL	Stores one captured source version with metadata and hash.
ParsedRecord	PostgreSQL	Stores one normalized entity extracted from a source.
CheckRun	PostgreSQL	Stores one check execution with input, status, and final report.
CheckEvidence	PostgreSQL	Stores atomic evidence such as hits, no-hits, and registry/media findings.
<code>data/snapshots/</code>	Filesystem	Stores raw CSV, XML, HTML, and JSON source artifacts.
<code>data/cache/</code>	Filesystem	Stores cached HTTP responses for live sources.

4.2.3 Frontend Organization

The front-end of the system is a single-page app built with React 18 and TypeScript and Vite, and its role is mainly to give an analyst interface for communication with

the backend [26, 27]. It does not perform the main analytical logic itself, but instead serves as the presentation and control layer of the platform. With a small API client based on `fetch()` the frontend interacts with the NestJS backend over JSON and it supports the main user actions like check submission, report viewing, check history browsing, and source snapshot management. As a result, this makes the frontend architecturally thin, because the core processes such as parsing, matching, scoring stay fully on the backend part.

At the page level, the frontend is organized around four operational views. To be more precise, there are pages such as `NewCheckPage` for entering a company name and optional identifiers, `CheckReportPage` for reviewing the full result of an executed check, `ChecksHistoryPage` for browsing previous runs, and `SnapshotsPage` for inspecting stored source snapshots and triggering reference ingestion. The report page is the most important part of the interface, because it shows the final score and the structured explanations with the evidence and registry information in a way a human analyst can review it conveniently. Shared UI (User Interface) components like badges, metric cards, and JSON viewers were used, so that the interface stays consistent.

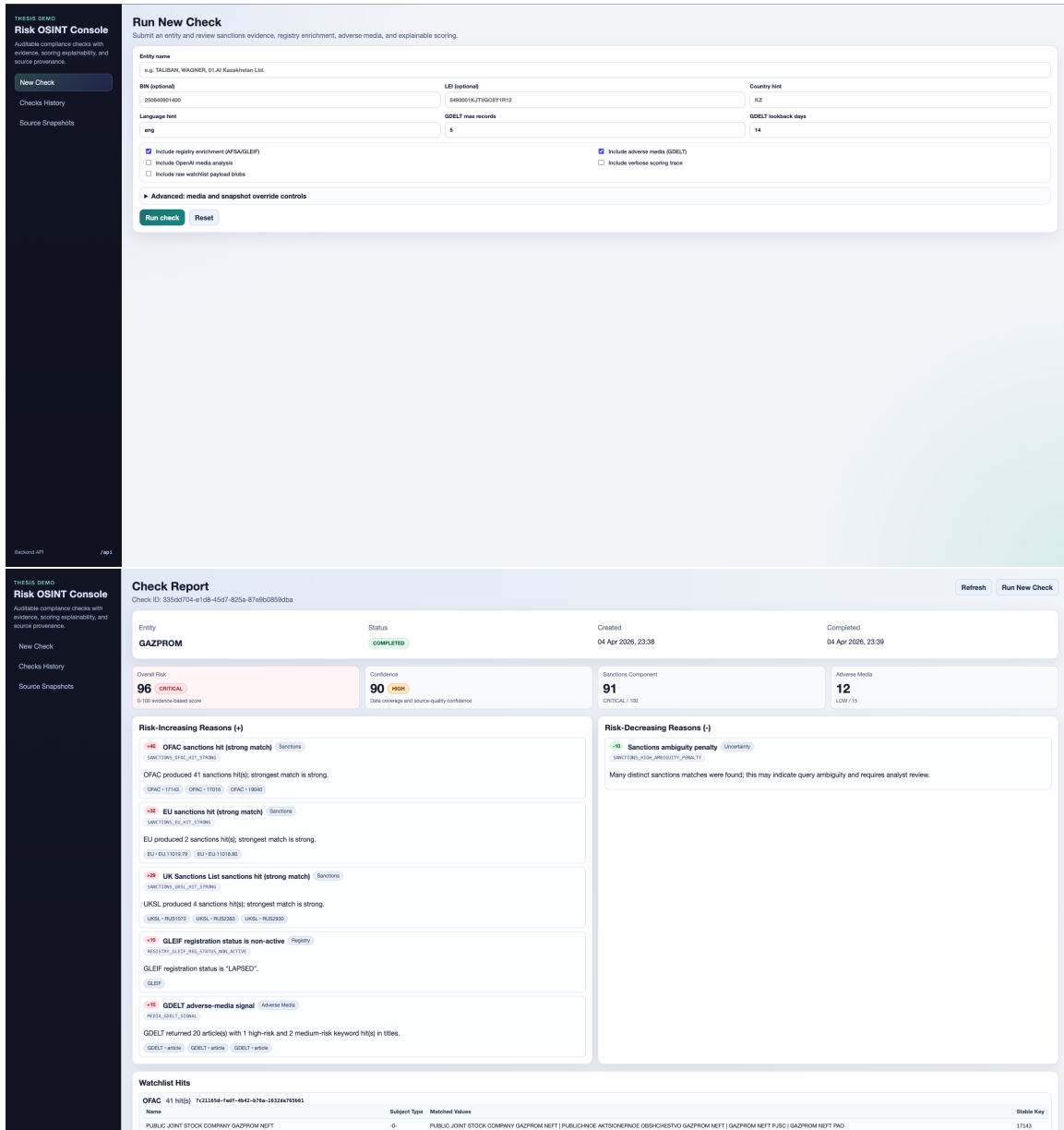


Figure 4-3: Examples of the user interface of the developed system.

4.3 Scoring Model

The scoring model is a rule-based function that turns all the evidence into three different scores for sanctions, registry, and adverse media, then it combines them into one final risk score. In the implementation, the model is versioned as the-`sis_evidence_score` version 1.0.1 and is labeled transparent-rule-based-weighted-evidence.

The three parts are calculated separately and then merged using a branch-dependent rule. To be more precise, this means the same registry or media evidence can have a different influence on the final result depending on whether sanctions evidence is absent, moderate, or already dominant. The final score is mapped to four risk levels, where scores below 25 are classified as LOW, MEDIUM for 25 to 54, HIGH for 55 to 79, and CRITICAL for anything above 80.

Table 4.2: Main scoring components of the model

Component	Max Score	Scoring Scope	Primary Formula
Sanctions	100	Per-source base weight $\times$ match multiplier + hits bonus	<code>sum(source_weight $\times$ match_multiplier + hit_bonus)</code>
Registry	30	Anomaly signals from AFSA/GLEIF status fields	<code>sum(status_penalties)</code>
Adverse Media	15	GDELT base + keyword hits + optional LLM boost	<code>min(6, articles) + highHits$\times$2 + mediumHits + llmBoost</code>

4.3.1 Sanctions Scoring

The sanctions part is the strongest part of the whole model. To be more precise, every source gets its own fixed base weight, so OFAC is 35 and the EU list is 32, while UKSL is 28, and the UN list is 22. For every source the system analyses the match it found and decides if it is strong, moderate, or weak and then multiplies the weight by 1.0 or 0.72 or 0.45 respectively depending on that. In addition to this, repeated matches from the same source can add a logarithmic bonus, while highly ambiguous result sets are penalized. A compact representation of the sanctions subscore is shown below:

$$S_i = \text{round}(w_i \cdot m_i) + b(h_i),$$

$$b(h_i) = \min(8, \max(0, \lceil \log_2(h_i) \rceil - 1)),$$

where w_i is the source weight, m_i is the match-strength multiplier, and h_i is the number of hits from source i . The total sanctions score is then

$$S = \min\left(100, \sum_i S_i - P_{\text{amb}}\right),$$

with an ambiguity penalty $P_{\text{amb}} = 10$ can be activated when the query produces at least 15 total hits across at least 8 distinct matched names. This penalty is used because it helps to reduce the score inflation caused by common name collisions.

Table 4.3: Match strength classification used in sanctions scoring

Strength	Condition	Multiplier
Strong	Exact normalized match (<code>qNorm == cNorm</code>), or all query tokens match exactly and the candidate has at most query tokens + 2	1.00
Moderate	All query tokens match exactly but the candidate is much longer, or all tokens match with some prefix matching and at least 50% of them match exactly	0.72
Weak	Any other partial match	0.45

4.3.2 Registries Scoring

The registry part is basically anomaly-driven and it is capped at 30 points total. For AFSA it can add 15 points if the registration status shows not active and then 5 additional points are added if it finds collateral. GLEIF contributes 10 points for a non-active registration status and 8 points for a non-active entity status. In search mode the signals are deduplicated so the same problem doesn't get counted multiple times for different rows.

Table 4.4: AFSA-based registry signals used in the scoring model

Signal Code	Condition	Points	Strength
REGISTRY_AFSA_ INACTIVE_STATUS	Status contains: revoked, dissolved, terminated, liquidated, inactive, struck	+15	moderate
REGISTRY_AFSA_ COLLATERAL_PRESENT	Collateral array is non-empty	+5	weak
REGISTRY_AFSA_ BIN_CONFIRMED	Returned BIN matches requested BIN	0	info

Table 4.5: GLEIF-based registry signals used in the scoring model

Signal Code	Condition	Points	Strength
REGISTRY_GLEIF_REG_ STATUS_NON_ACTIVE	registration_status is not “issued” or “active”	+10	moderate
REGISTRY_GLEIF_ENTITY_ STATUS_NON_ACTIVE	entity_status does not contain “active”	+8	moderate

4.3.3 Adverse Media Scoring

Subsequently, the adverse media part is capped at 15 points and it uses the GDELT article volume and keywords. In the baseline model, up to 6 points come from the number of returned articles, while predefined high-risk keywords add 2 points each and medium-risk keywords add 1 point each:

$$M_{\text{gdelt}} = \min(15, \min(6, n) + 2H + L),$$

where n is the number of returned articles, H is the number of high-risk keyword hits, and L is the number of medium-risk keyword hits in article titles. In addition to this, when optional OpenAI media analysis is turned on, the model adds a small relevance-gated boost. More precisely, for each article, the adverse label is mapped

to a base value $\{0, 1, 2, 3\}$ and multiplied by an entity-relevance weight $\{0.0, 0.8, 1.0\}$ for low, medium, and high relevance respectively. The resulting sum is rounded and capped at 8 points before being merged back into the media score.

4.3.4 Other Scorings

The overall score is not a simple sum of the three components. Instead, the model uses three different aggregation branches. To be more precise, if there are no sanctions hits, the final score is capped at 40 and is driven only by registry and media evidence. If sanctions are present but remain below 70, registry and media are added with reduced weights. If the sanctions score is already 70 or higher, then the contextual components are down-weighted even more. This logic can be written as:

$$O = \begin{cases} \min(40, \text{round}(0.8R) + \text{round}(0.7M)), & \text{if } S = 0, \\ \min(100, S + \text{round}(0.35R) + \text{round}(0.25M)), & \text{if } 0 < S < 70, \\ \min(100, S + \text{round}(0.30R) + \text{round}(0.20M)), & \text{if } S \geq 70, \end{cases}$$

where S , R , and M denote sanctions, registry, and adverse-media scores. This ensures that registry anomalies or noisy media signals can't create a critical-risk result when no sanctions evidence is present.

Besides the final score the system also computes a separate confidence score to show the data coverage and how stable the result is. More precisely, the confidence score starts from a base of 50, then increases by 10 for each available sanctions source and decreases by 15 for each missing one. In addition to this, it is also penalized by 15 when the result is driven only by media signals without sanctions support and by 5 when live-source variance is present without a sanctions anchor. On the other hand, the presence of BIN and LEI identifiers adds 3 points each. The result is then

bounded to the interval from 5 to 95:

$$C = \text{clip}_{[5,95]} (50 + 10A - 15M_s - 15I_{\text{media-only}} - 5I_{\text{live-var}} + 3I_{\text{BIN}} + 3I_{\text{LEI}}),$$

where A is the number of available sanctions sources, M_s is the number of missing sanctions sources, and the I terms are indicator variables for the corresponding conditions. The confidence level is then mapped to **LOW** for values below 45, **MEDIUM** for values from 45 to 74, and **HIGH** for values of 75 or above.

Table 4.6: Confidence score components used in the model

Property	Value
Base	50
Per available sanctions source	+10
Per missing sanctions source	−15
Media-only signal (no sanctions)	−15
Live-source variance without sanctions anchor	−5
BIN input present	+3
LEI input present	+3
Bounds	[5, 95]
Levels	LOW (< 45), MEDIUM (45–74), HIGH (≥ 75)

4.4 Explainability and Evidence Tracking

The explainability layer of the system is implemented mainly inside the scoring service and is designed to ensure that the final result is not presented as a single number with no other information. To be more precise, every check report has extra information apart from the main score, such as the component level factors, top reasons, negative reasons, calculation traces, confidence estimates, coverage information, limitations and model metadata. In practice, this means that the user does not only see the final

score and risk level, but also all the individual scoring decisions that contributed to the result. When verbose explainability is turned on by the user, the trace becomes deeper by including zero impact steps and evidence references at the arithmetic-step level.

Evidence tracking is developed as a separate persistence layer centered on the CheckEvidence model which records the concrete source interactions that occurred during a check. It doesn't just record the sanctions hits, but also tracks no-hit outcomes, registry lookups, and the OpenAI news analysis. Each evidence row has the source type, category, outcome, matched field, matched value, optional record stable key, and a JSON payload containing the details specific to the source. This approach is important because it creates an audit trail that is more detailed than the final score by itself. More precisely, the system can show later which datasets were searched, which values matched and which source payloads were returned, even when the result was negative. In addition to this, the evidence rows are linked back to the parent check and the exact SourceSnapshot that was used so the result is tied directly to the original source state.

Table 4.7: Main explainability outputs included in the final report

Output	Always present	Role
model	Yes	Stores model ID, version, methodology, and computation time.
overall	Yes	Stores final score, risk level, and confidence.
components	Yes	Stores sanctions, registry, and adverse-media breakdowns.
top_reasons	Yes	Lists the strongest risk-increasing factors.

Output	Always present	Role
<code>negative_reasons</code>	Yes	Lists the strongest mitigating or negative-impact factors.
<code>calculation_trace</code>	Yes	Stores step-by-step score calculation details.
<code>coverage</code>	Yes	Shows source availability and missing-source status.
<code>limitations</code>	Yes	Stores automatically generated caution and boundary statements.
<code>provenance.sources[]</code>	Yes	Stores per-source snapshot metadata and hashes.
Raw payload in hits	No	Includes nested source payloads in detailed responses.

Another layer of transparency is provided by provenance and reproducibility support. More specifically, for each source used in a check, the report includes snapshot metadata such as ID of the snapshot itself, version label, fetch time, publication time if available, SHA-256 hash and stored metadata describing the raw artifact. In addition to this, offline sanctions sources can also be pinned to specific snapshots using `snapshotOverrides`, which makes it possible to rerun a check against a known prior dataset state. On the frontend, all these explainability and evidence parts are shown in a special investigation view with score cards, reason lists, watchlist hits, media and registry panels, a provenance table, evidence rows, a calculation panel, and a full JSON report viewer. The system also shows its own limits by generating limitation strings automatically if sources are missing, if the news is uncertain, or the AI analysis is probabilistic. As a result, the explainability in this system is not limited to score decomposition, but also extends to source traceability, evidence preservation, and explicit communication of uncertainty.

Chapter 5

Results and Discussion

5.1 Evaluation Setup, Labeled Benchmark, and Representative Scenarios

The evaluation was conducted using real CheckRun outputs from the running system API. Since the proposed scoring model is rule-based rather than supervised, labeled data was not used for model training. However, a small manually labeled benchmark was prepared for evaluation purposes. In this benchmark, each case was assigned an expected label based on its known evaluation role. Entities known to appear in sanctions sources were treated as positive cases, while clean registry entities and false-positive control cases were treated as negative cases. For binary evaluation, MEDIUM, HIGH, and CRITICAL outputs were treated as positive predictions, while LOW outputs were treated as negative predictions.

Table 5.1: Manually labeled benchmark cases used for binary evaluation.

Entity	Expected Label	System Level	Correct
AL-QAIDA	Positive	CRITICAL	Yes
GAZPROM	Positive	CRITICAL	Yes
Rosneft	Positive	CRITICAL	Yes
Wagner	Positive	CRITICAL	Yes
Kalashnikov	Positive	CRITICAL	Yes
HARAKAT AL-NUJABA	Positive	MEDIUM	Yes
Astana International Exchange Limited	Negative	LOW	Yes
ALPI S.A.S. DI PUTIN GIULIA & C.	Negative	LOW	Yes

The benchmark included eight completed checks: six positive cases and two negative cases. The positive cases included multi-source sanctions matches, high-risk sanctions cases, and a single-source sanctions case. The negative cases were used to test whether the system avoids risk escalation for clean entities and false-positive controls. For detailed discussion, four representative cases were selected: a multiple-source sanctions case (AL-QAIDA), a high-risk case where sanctions, registry, and adverse media all contributed (GAZPROM), a single-source sanctions case with strong media support (HARAKAT AL-NUJABA), and a clean entity with active registry data and no sanctions risk signals (Astana International Exchange Limited). This setup allows the evaluation to assess not only whether the system detects clearly designated entities, but also how it handles contextual media signals, registry absence or anomalies, false-positive controls, and no-sanctions cases by reporting component scores, confidence values, evidence rows, and provenance metadata.

Table 5.2: Cross-case summary of representative evaluation scenarios (S - sanctions, R - registries, M - media)

Entity	S	R	M	Overall	Level	Confidence	Evidence Rows
AL-QAIDA	100	0	6	100	CRITICAL	90 / HIGH	22
GAZPROM	91	10	12	96	CRITICAL	90 / HIGH	50
HARAKAT AL-NUJABA	35	0	15	39	MEDIUM	90 / HIGH	6
Astana International Exchange Limited	0	0	0	0	LOW	88 / HIGH	7

5.2 Benchmark Results and Example Checks

Using the selected binary threshold, the system correctly classified all eight manually labeled benchmark cases. The benchmark produced 6 true positives, 2 true negatives, 0 false positives, and 0 false negatives. Therefore, on this small benchmark, accuracy, precision, recall, and F1-score were all equal to 1.00. These results should be interpreted carefully because the benchmark is small and manually selected. They show that the implementation behaves as expected on representative positive, negative, and false-positive-control cases, but they do not prove general accuracy across all possible entities, languages, spelling variants, and naming patterns.

No false positives or false negatives were observed in this small benchmark. However, this does not mean that the system is error-free. The main expected error sources are common-name collisions, spelling variants, transliteration differences, and temporal changes in live registry or media sources. The false-positive control case,

ALPI S.A.S. DI PUTIN GIULIA & C., was included to test whether the system escalates a clean entity only because of a risky token in the name. The LOW output for this case suggests that the multi-token matching logic reduces this type of false positive, although broader testing would be needed to confirm this behaviour.

Table 5.3: Binary classification results on the manually labeled benchmark.

Metric	Value	Explanation
True positives	6	Positive benchmark cases classified as MEDIUM or higher.
True negatives	2	Negative benchmark cases classified as LOW.
False positives	0	Negative cases incorrectly classified as MEDIUM or higher.
False negatives	0	Positive cases incorrectly classified as LOW.
Accuracy	1.00	All benchmark cases were classified correctly.
Precision	1.00	All positive predictions corresponded to positive benchmark cases.
Recall	1.00	All expected positive cases were detected.
F1-score	1.00	Harmonic mean of precision and recall.

The results of the checks show that the system works differently depending on which part of the evidence is dominating more for each case. More precisely, in the

high risk cases, the sanctions part basically decides most of the score, while the registry and news components remain as the secondary signals, affecting the score a lot less. We can see this in the AL-QAIDA and GAZPROM checks, both of which were identified as CRITICAL by the system. The AL-QAIDA case demonstrates a sanctions-dominant result, as the system found strong hits with all four sanctions sources and gave a maximum score of 100. On the other hand, GAZPROM demonstrated more complicated high-risk results, where the final score of 96 was obtained mainly by sanctions, but was also impacted by a registry anomaly and adverse-media findings. These two cases confirm that the platform can identify designated entities and preserve an appropriate relationship between high sanctions evidence and high final risk.

Table 5.4: Component dominance across the selected representative cases

Case	Dominant Component	Secondary Signal	Ambiguity Penalty
AL-QAIDA	Sanctions (100)	Media (+6)	Yes (−10)
GAZPROM	Sanctions (91)	Registry (+10), Media (+12)	Yes (−10)
HARAKAT AL-NUJABA	Sanctions (35)	Media (15) — maxed	No
Astana International Exchange Limited	None (clean baseline)	—	No

The other two cases show how the model behaves when there are few sanctions hits or no sanctions hits. For the HARAKAT AL-NUJABA case, the system gave a MEDIUM result with a score of 39, due to the one strong OFAC hit and a strong adverse media component. This is important as it shows that the media component can make a case stronger, despite the fact that it can’t push the result to the

highest risk levels by itself. By contrast, the Astana International Exchange Limited example shows a clean baseline scenario, with zero points in sanctions, registry, and adverse media, which resulted in a LOW classification. Even in this case, the system still created evidence rows and preserved provenance to show what it checked, for transparency and reproducibility reasons. Taken together everything stated above, these examples show that the system can differentiate between a sanctioned entity, high-risk case, a medium-risk case with partial corroboration, and a clean entity, simultaneously preserving explainability using the component scores, confidence values, evidence rows, and source provenances.

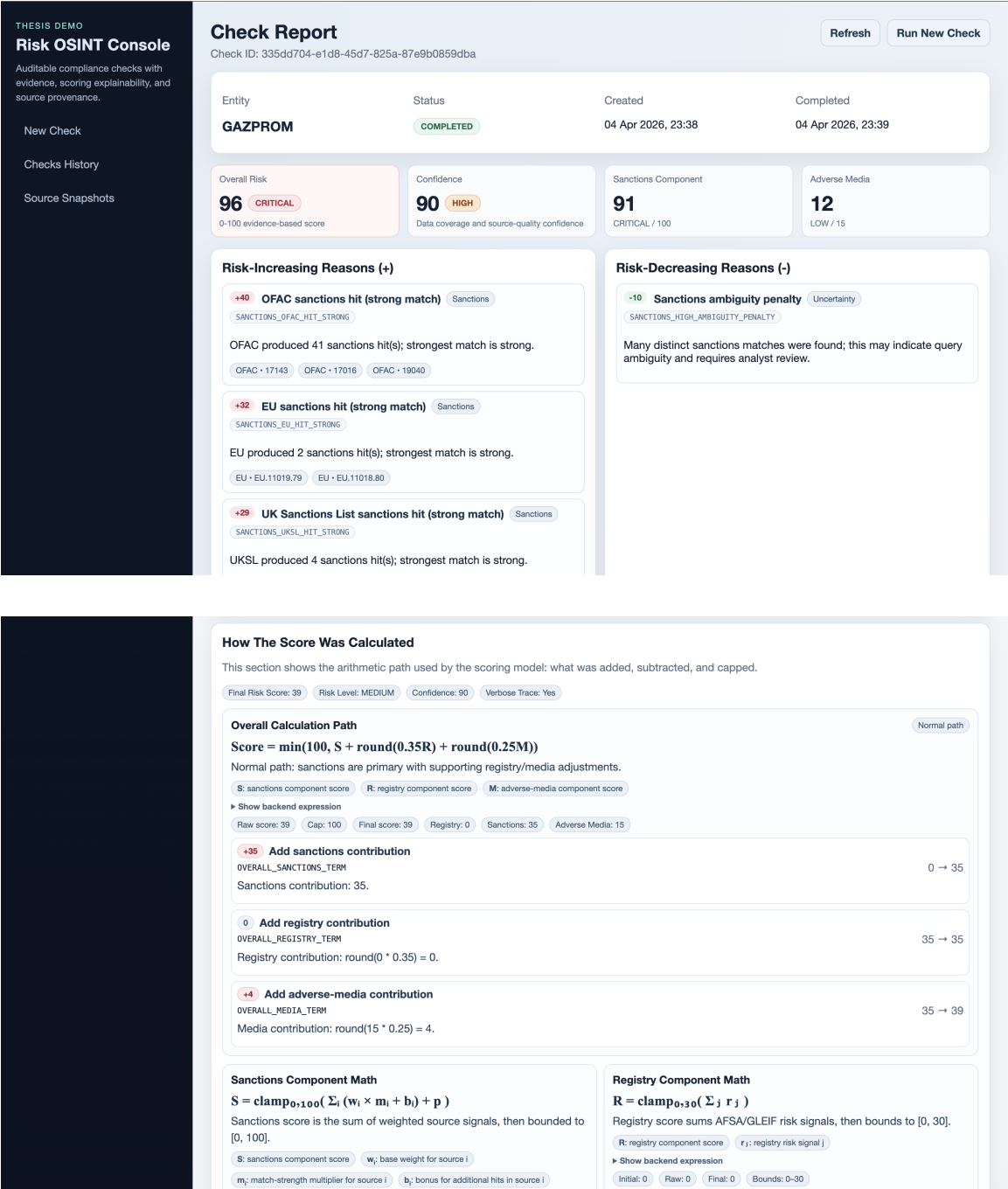


Figure 5-1: Examples of explainable system output for representative checks: the GAZPROM report overview and the HAKAKAT AL-NUJABA scoring-calculation panel.

5.3 Discussion and Observed Limitations

Overall, the results show that the developed system is able to differentiate sanctioned entities from medium-risk and low-risk cases in a way that is consistent with its design goals. In the strongest scenarios, such as AL-QAIDA and GAZPROM, the final result was driven primarily by sanctions evidence, while registry and adverse media components were more like a supporting context rather than overriding the main conclusion. In the medium-risk case of HAKKAT AL-NUJABA, the model combined a single-source sanctions hit with strong adverse media evidence, but still kept the overall result below the critical range. In the clean baseline case like the Astana International Exchange Limited, the system returned zero risk in all three components while still keeping evidence rows, provenance metadata, and no-hit records. This shows that the weighting logic, component caps, and branch-based score combination work as intended. To be more precise, strong sanctions evidence dominates the assessment, while registry and media signals remain corroborative and do not independently escalate a case to the highest risk levels.

On the other hand, the evaluation also shows several important limitations. The matching layer remains heuristic and does not use fuzzy distance, transliteration, or cross-source entity resolution, which means that difficult spelling variants, not a Latin name forms, and common name collisions may still affect the results. The adverse media component is also limited by the fact that it searches mostly on recent GDELT article metadata, while the optional OpenAI layer is probabilistic and therefore cannot be treated as deterministic evidence. In addition to this, live sources such as AFSA, GLEIF, and GDELT introduce temporal variability, so rerunning the same check at a later time may produce different registry or media enrichments even when the sanctions snapshots remain fixed. For these reasons, the system should be interpreted as a transparent decision-support tool rather than a fully automatic final judgment engine. Its main strength lies in organizing heterogeneous OSINT signals into a reproducible, inspectable workflow, but expert review is still necessary in ambiguous or high-impact cases.

Chapter 6

Conclusion

This thesis showed the design and implementation of an explainable OSINT-based risk scoring system for companies and individuals, which was built by integrating seven public data sources. To be more precise, there were three evidence families: sanctions lists, registries, and adverse media. The developed platform contains sanctions datasets with live registry and media enrichments inside a modular NestJS backend, while preserving source snapshots, hashes, structured evidence rows, and transparent scoring outputs for later review. Instead of relying on a black-box prediction model, the system uses a transparent approach that clearly shows component scores, reasons, calculation traces, coverage information, limitations, and provenance metadata. As a result of this, the system demonstrates that messy public information can be transformed into a practical and auditable screening workflow that supports company risk assessment in a structured and reproducible manner.

The research questions stated in Chapter 1 are addressed through the design, implementation, and evaluation of the proposed system. The first question, concerning the integration of heterogeneous OSINT sources, is answered by the modular ingestion and parsing pipeline, which normalizes sanctions lists, registry data, and adverse media into a common workflow. The second question, concerning explainability and auditability, is addressed through source snapshots, SHA-256 hashes, evidence rows, provenance metadata, and calculation traces that connect each score to its underlying data. The third question, concerning the ability of the rule-based model to distin-

guish risky and clean entities, is addressed through the manually labeled benchmark in Chapter 5, where the system correctly classified the selected positive, negative, and false-positive-control cases.

The evaluation on representative completed checks showed that the platform is able to distinguish clearly sanctioned entities, mixed-signal cases, medium risk cases supported by adverse media, and clean baseline entities with no material signals. In the strongest cases, sanctions evidence correctly dominated the final result, while registry and adverse media findings acted mainly as corroborative context, which is consistent with the intended design of the scoring model. At the same time, the work also highlighted several important limitations, including heuristic entity matching, the lack of transliteration and cross source entity resolution, and the temporal variability of live sources such as AFSA, GLEIF, and GDELT. For this reason, the developed system should be treated as a transparent decision support tool rather than a fully automatic final judgment engine. Future improvements could include stronger multilingual matching, richer article-level analysis, better handling of ambiguous names, and broader cross-source linkage, but even in its current form the system shows that explainability, evidence tracking, and reproducibility can be embedded directly into an OSINT-based company risk scoring pipeline.

Bibliography

- [1] J. Rajamäki, S. McMenamin, and K. Tiitta, “The Importance of Open-Source Intelligence in Assessing Risks of Cyberattacks,” *Information & Security: An International Journal*, vol. 55, no. 1, pp. 15–31, 2024.
- [2] D. R. Hayes and F. Cappa, “Open-source intelligence for risk assessment,” *Business Horizons*, vol. 61, no. 5, pp. 689–697, 2018.
- [3] Thomson Reuters, “Sanctions screening: An overview,” 2025. [Online]. Available: <https://legal.thomsonreuters.com/blog/overview-sanctions-screening>
- [4] Financial Crime Academy, “Sanctions Screening In Compliance,” 2025. [Online]. Available: <https://financialcrimeacademy.org/sanctions-screening-in-compliance/>
- [5] E. Yildirim and T. Uslu, “Examination of Corporate Reputation Scores, Market Value and Financial Performance,” *Fiscaeconomia*, vol. 8, no. 2, pp. 606–623, 2024. [Online]. Available: <https://dergipark.org.tr/en/pub/fesa/issue/76546/1114198>
- [6] Flagright, “A Guide to Risk Scoring: Best Practices and Strategies,” 2025. [Online]. Available: <https://www.flagright.com/post/how-to-do-risk-scoring>
- [7] P. Pathak, S. Rajput, and R. Kumar, “Explainable Artificial Intelligence Credit Risk Assessment: Integrating LIME and SHAP with ML Models,” *IOSR Journal of Economics and Finance*, vol. 16, no. 5, pp. 63–72, Sep.

2025. [Online]. Available: <https://www.iosrjournals.org/iosr-jef/papers/Vol16-Issue5/Ser-1/H1605016372.pdf>
- [8] MarkovML Team, “LIME vs SHAP: A Comparative Analysis of Interpretability Tools,” 2024. [Online]. Available: <https://www.markovml.com/blog/lime-vs-shap>
- [9] Moody’s Analytics, “The power of news sentiment in modern financial analysis,” 2024. [Online]. Available: <https://www.moody’s.com/web/en/us/insights/digital-transformation/the-power-of-news-sentiment-in-modern-financial-analysis.html>
- [10] J. Casigay, “Application of Textual Sentiment Scores in Value-at-Risk models,” Master’s thesis, HEC Montréal, 2021. [Online]. Available: https://biblos.hec.ca/biblio/memoires/casigay_jaime_m2021.pdf
- [11] U.S. Department of the Treasury, Office of Foreign Assets Control (OFAC), “Sanctions List Service (SLS),” 2025. [Online]. Available: <https://ofac.treasury.gov/sanctions-list-service>
- [12] European Union, “Consolidated list of persons, groups and entities subject to EU financial sanctions,” [Online]. Available: <https://data.europa.eu/data/datasets/>.
- [13] United Nations Security Council, “Consolidated United Nations Security Council Sanctions List,” 2025. [Online]. Available: <https://main.un.org/securitycouncil/en/content/un-sc-consolidated-list>
- [14] HM Government (FCDO), “UK Sanctions List — Search Service,” 2025. [Online]. Available: <https://search-uk-sanctions-list.service.gov.uk/>
- [15] Astana Financial Services Authority (AFSA), “Public Register,” 2026. [Online]. Available: <https://publicreg.myafsa.com/>

- [16] Global Legal Entity Identifier Foundation (GLEIF), “GLEIF API — LEI and Relationship (Level 2) Data,” 2025. [Online]. Available: <https://www.gleif.org/en/lei-data/gleif-api>
- [17] The GDELT Project, “Data: Querying, Analyzing and Downloading,” 2025. [Online]. Available: <https://www.gdeltproject.org/data.html>
- [18] OpenAI, “OpenAI API documentation,” 2026. [Online]. Available: <https://developers.openai.com/api/docs/>
- [19] Companies House, “Companies House API (REST),” 2025. [Online]. Available: <https://developer.company-information.service.gov.uk/>
- [20] Blackdot Solutions, “OSINT in 2026: Key Trends and What to Expect,” Jan. 20, 2026. [Online]. Available: <https://blackdotsolutions.com/blog/osint-in-2026>
- [21] Flagright, “Overcoming Compliance Data Fragmentation,” Feb. 15, 2026. [Online]. Available: <https://www.flagright.com/post/overcoming-compliance-data-fragmentation>
- [22] NestJS, “NestJS — A progressive Node.js framework,” 2026. [Online]. Available: <https://nestjs.com/>
- [23] Microsoft, “TypeScript: JavaScript with syntax for types,” 2026. [Online]. Available: <https://www.typescriptlang.org/>
- [24] PostgreSQL Global Development Group, “PostgreSQL: The world’s most advanced open source relational database,” 2026. [Online]. Available: <https://www.postgresql.org/>
- [25] Prisma, “Prisma ORM: Next-generation Node.js and TypeScript ORM,” 2026. [Online]. Available: <https://www.prisma.io/orm>
- [26] Meta, “React,” 2026. [Online]. Available: <https://react.dev/>

- [27] VoidZero and Vite contributors, “Vite: Next Generation Frontend Tooling,” 2026.
[Online]. Available: <https://vite.dev/>