Funded by the European Union
Horizon Europe
(HORIZON-CL4-2021-HUMAN-01-27
AI to fight disinformation)

AI4TRUST

D5.6

AI4TRUST Platform v2

PARTNERS

FONDAZIONE BRUNO KESSLER · CERTH CENTRE FOR RESEARCH & TECHNOLOGY HELLAS · UNIVERSITÀ DI TRENTO · NATIONAL CENTRE FOR SCIENTIFIC RESEARCH "DEMOKRITOS" · CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

GDI Global Disinformation Index · sky · SAHER EUROPE · DEMAGOG · MALDITA.ES

ASTIKI MI KERDOSKOPIKI ETAIRIA KENTRO KATAPOLEMISIS TIS PARAPLIROFORISIS / CIVIL NON-PROFIT COMPANY KENTRO KATAPOLEMISIS TIS PARAPLIROFORISIS · EURACTIV · ASOCIATIA DIGITAL BRIDGE · EUROPEJSKIE MEDIA SP ZOO · FINCONS GROUP · UNIVERSITY OF CAMBRIDGE

Project acronym	AI4TRUST
Project full title:	AI-based-technologies for trustworthy solutions against disinformation
Grant info:	ID 101070190-AI4TRUST
Funding:	HORIZON-CL4-2021-HUMAN-01-27 - AI to fight disinformation (RIA)
Version:	1.0
Status	Final version
Dissemination level:	Public
Due date of deliverable:	31/03/2025
Actual submission date:	31/03/2025
Work Package:	WP5
Lead partner for this deliverable:	FINC
Partner(s) contributing:	FBK
Main author(s):	Marco Giovanelli (FINC), Gabriel H. Carraretto (FINC), Riccardo Corrias (FINC)
Contributor(s):	Marcello Paolo Scipioni (FINC), Vasileios Mezaris (CERTH), Damianos Galanopoulos (CERTH), Christos Koutlis (CERTH), Antonios Leventakis (CERTH), Horia Cucu (POLITEHNICA), Matteo Saloni (FBK), Riccardo Gallotti (FBK), Marco Guerini (FBK), Stefano Menini (FBK), Alan Ramponi (FBK), Benedetta Liberatori (UNITN), Lina Livdane (GDI), Georgios Petasis (NCSR-D)
Reviewer(s):	Evlampios Apostolidis (CERTH), Elisa Ricci (UNITN), Serena Bressan (FBK), Danilo Giampiccolo (FBK)

Summary of modifications

VERSION	DATE	AUTHOR(S)	SUMMARY OF MAIN CHANGES
0.1	07/02/2025	Marco Giovanelli (FINC)	First draft
0.2	10/02/2025	Marco Giovanelli (FINC)	Platform Overview & Data Flow
0.3	11/02/2025	Gabriel H. Carraretto (FINC)	Front-end, Implementation, & Monitoring
0.4	20/02/2025	Gabriel H. Carraretto (FINC), Riccardo Corrias (FINC)	Monitoring & Human Validation, Database, Integration and Deployment, API Gateway, Dispatcher
0.5	26/02/2025	Evlampios Apostolidis (CERTH), Elisa Ricci (UNITN)	First internal review of the deliverable
0.6	06/03/2025	Marco Giovanelli (FINC), Gabriel H. Carraretto (FINC)	Platform Overview & Data Flow review
0.7	26/03/2025	Vasileios Mezaris (CERTH), Damianos Galanopoulos (CERTH), Antonios Leventakis (CERTH), Christos Koutlis (CERTH), Horia Cucu (POLITEHNICA), Matteo Saloni (FBK), Riccardo Gallotti (FBK), Marco Guerini (FBK), Stefano Menini (FBK), Alan Ramponi (FBK), Benedetta Liberatori (UNITN), Lina Livdane (GDI), Georgios Petasis (NCSR-D)	Contributions on AI-Driven Data Analysis Methods, Disinformation Warning System, Data Platform, Data Collector, Streaming Platform, Serverless Platform, Data Lakehouse, and Data Pipelines sections
0.8	27/03/2025	Evlampios Apostolidis (CERTH), Elisa Ricci (UNITN), Riccardo Gallotti (FBK)	Final internal review of the full document

VERSION	DATE	AUTHOR(S)	SUMMARY OF MAIN CHANGES
0.9	28/03/2025	Marco Giovanelli (FINC), Gabriel H. Carraretto (FINC), Riccardo Corrias (FINC)	Finalisation of the deliverable according to the final internal review
1.0	31/03/2025	Riccardo Gallotti (FBK), Serena Bressan (FBK), Danilo Giampiccolo (FBK)	Final review, formatting, and preparation for submission

Table of contents

List of acronyms

ACRONYMS	MEANING
AI	Artificial Intelligence
API	Application Programming Interface
CIB	Coordinated Inauthentic Behaviour
DAG	Directed Acyclic Graph
DoA	Description of Action
DWS	Disinformation Warning System
IP	Intellectual Property
PS	Platform Specification
SW	Software
UI	User Interface
WP	Work Package

List of figures

List of tables

Executive summary

This deliverable of the **"AI4TRUST - AI-based technologies for trustworthy solutions against disinformation"** project, titled **"D5.6 - AI4TRUST Platform v2"**, is the third deliverable of **Work Package 5 "Technical implementation of the platform & Security Framework"** (hereinafter referred to as WP5). It is closely interconnected with other Work packages, specifically **WP2** ("Methodological design, data gathering and pre-processing"), **WP3** ("AI-driven data analysis methods"), **WP4** ("Human-Centred Explainability, Interpretation and Policy"), and **WP6** ("Piloting, Assessment & Fact-checking"). The deliverable marks the **intermediate release of the AI4TRUST platform (AI4TRUST Platform v2)**. It also incorporates the recommendations outlined in the "General Project Review Consolidated Report (HE)", dated 28 June 2024, following the project's first Review Meeting.

In particular, the following **sections** have been implemented to address the recommendations:

- **Section 2** details the **Platform Roadmap** and shows how the AI4TRUST Platform v2 updates and extends the AI4TRUST Platform v1;
- **Section 2.1** describes the **Toolbox** functionality from the user's perspective and outlines how this functionality has been improved in AI4TRUST Platform v2;
- **Section 2.2** shows the new added **Monitoring and Human Validation** functionality in AI4TRUST Platform v2;
- **Section 3** (and the related subsections) describes from a technical point of view **how the components are integrated, implemented and updated** in AI4TRUST Platform v2 to support the Toolbox and the Monitoring and Human Validation functionalities.

The **AI4TRUST Platform v2** builds on the foundation established in AI4TRUST Platform v1, incorporating advancements to enhance its effectiveness in combating disinformation through AI-driven solutions. It implements the Key Exploitable Results (KER) #1 (the Toolbox) and #2 (the Monitoring and Human Validation) described in deliverable D7.4[1] and it will be tested by the end users of the project (i.e., fact-checkers and journalists) during the second piloting phase. This will be documented in a dedicated internal report by the WP6 Leader ahead of the scheduled Review Meeting on 20 May 2025. Following the release of the third version of the platform in August 2025 (see Milestones), a third piloting phase will take place in the autumn, leading to the submission of the forthcoming **deliverable D6.3 - Piloting sessions report v2**[2]. This deliverable will serve as the foundation for the final updates to the platform, ensuring the necessary refinements for the preparation and submission of the deliverable **D5.7 – Final AI4TRUST Platform**.

The AI4TRUST Platform v2 can be accessed by selected users at the following URL:

https://pilot.platform.ai4trust.eu/

[1] D7.4 - Innovation, Exploitation, and Sustainability Plan v2 (https://ai4trust.eu/public-deliverables/)
[2] D6.3 - Piloting sessions report v2-3

1. Introduction

The **AI4TRUST** project, formally known as **AI4TRUST - AI-based Technologies for Trustworthy Solutions Against Disinformation**, is a pioneering initiative aimed at addressing the pervasive issue of misinformation and disinformation within the European Union (EU). This project is designed to empower a diverse group of stakeholders, including researchers, fact-checkers, media professionals, and policymakers, by equipping them with advanced AI technologies that are tailored to tackle the multifaceted challenges of combating mis/disinformation (hereinafter also "disinformation") through a comprehensive approach that includes multichannel, multilingual, and multimodal monitoring. The AI4TRUST platform is structured to achieve **three primary objectives**:

- **To establish a robust monitoring system** that spans various communication channels, languages, and content types.
- **To support users with objective analyses** to allow them to evaluate the potential risks linked to unreliable information.
- **To contribute to a more trustworthy online environment** that encourages collaboration among key stakeholders, helping to mitigate the spread of unreliable information and potentially guiding the development of effective counter-narratives.

Deliverable D5.6, titled **AI4TRUST Platform v2**, represents a significant milestone in this project. As the third deliverable of Work Package 5 (WP5) - "Technical Implementation of the Platform & Security Framework," D5.6 signifies the intermediate release of the AI4TRUST platform. This document provides an in-depth look at the platform's functionality, highlighting its capabilities and operational framework. This document is organised with the following structure:

- The **second section** provides an overview of the AI4TRUST Platform, detailing its main parts and the roadmap to implement them in AI4TRUST Platform v1, v2 and v3. Then it describes AI4TRUST Platform v2 from the user perspective, detailing its updated and extended Toolbox (in respect to AI4TRUST Platform v1) and its newly introduced Monitoring and Human Validation Dashboard;
- The **third section** shows how the different components are integrated in AI4TRUST Platform v2 to support the Toolbox and the Monitoring and Human Validation functionalities shown in the second section, and details, in the relative subsections, the implementation and the updates of each component;
- The **fourth section** describes the software development (Dev) and IT operations (Ops), providing details about how applications and the related dependencies are packaged and how and where the software is deployed;
- The **fifth section** recaps the main achievements obtained implementing the AI4TRUST Platform v2 and highlights the connections with the forthcoming AI4TRUST Platform v3.

The **AI4TRUST Platform v2** introduces key improvements aimed at **strengthening the platform's analytical capabilities ensuring a more seamless integration of AI-driven solutions**. By improving its approach based on **insights from the initial piloting phase** (see deliverable D6.2 -

Piloting sessions report v1), this version enhances the **platform's ability to process and access content while reinforcing privacy and ethical standards**. Furthermore, the AI4TRUST Platform v2 introduces a **new Monitoring and Human Validation Dashboard** that enables the user to filter and rank the automatically collected Social Media and News items in order to promptly scout potential mis/disinformation and perform further investigation on selected pieces of information, enabling manual rating of the content.

2. Overview of the functionalities of the AI4TRUST Platform

AI4TRUST aims to combat mis/disinformation by leveraging advanced **AI-driven data analysis methods** that continuously assess online content. The platform also features **automated data collection and analysis of content**, serving as a valuable resource for future research on fake claims, enabling a more comprehensive understanding of mis/disinformation techniques and supporting ongoing efforts to improve detection methodologies. It also provides **analysis at-scale of collected data**, to provide an analytical view on disinformation, providing new insights on a more general view of the disinformation patterns. To achieve these objectives, the AI4TRUST Platform follows an incremental development approach, with three planned iterations (AI4TRUST Platform v1, v2, and v3) as detailed in D5.8[3] and summarised in the paragraphs below (Figure 1).

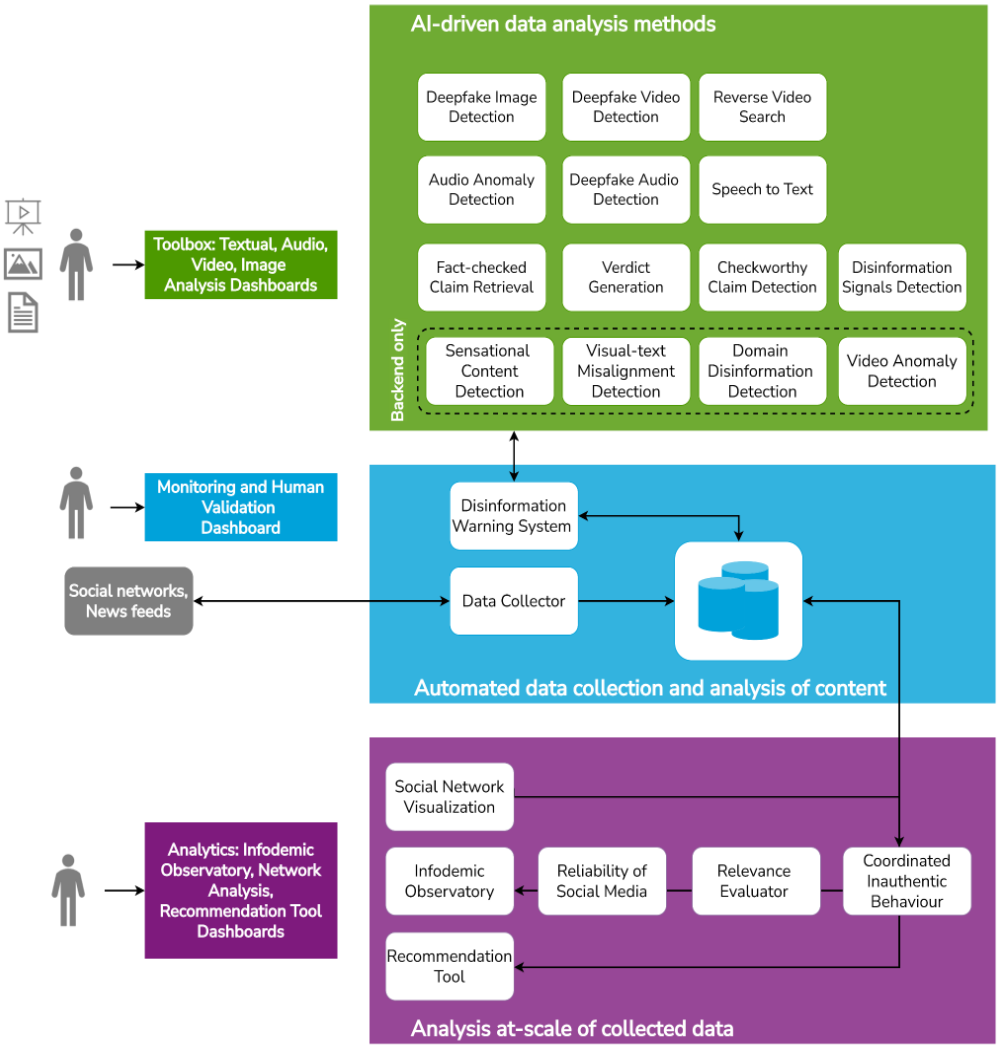

Figure 1: *Overview of the functionalities of AI4TRUST Platform (from D5.8)*

[3] D5.8 - AI4TRUST Platform Specification – Revised version (https://ai4trust.eu/public-deliverables/)

The AI4TRUST Platform is designed to offer users a variety of functionalities, with analyses performed by a comprehensive set of components displayed across different dashboards.

The **first section of the AI4TRUST Platform, shown in green** in Figure 1, focuses on analysing individual content items using AI-driven data analysis methods developed during the first R&D iteration. Users can submit textual, audio, video, or image content through specific dashboards, where AI-driven data analysis methods are used to support the detection of misleading content. This includes examining visual content to identify deepfake images and videos or videos reused out of their original context, analysing the audio stream for detecting deepfake audio, transcribing speech to text, and evaluating text to spot disinformation signals such as hate speech, offensive, sensational language.

The second section, depicted in blue in Figure 1, provides dashboards for exploring social media and news feed content that is automatically collected and evaluated about its checkworthiness by a subset of AI technologies that support large-scale data analysis.. A processing pipeline handles the automatic collection and analysis of content from the Web, and the flagging of checkworthy and potentially misleading content items by the Disinformation Warning System. The automatically flagged content items are presented to the users through the Monitoring Dashboard, where expert users can provide their feedback about the trustworthiness of the content item through the Human Validation Dashboard (see Section 2.2).

The third section, shown in purple in Figure 1, introduces visualisation dashboards for displaying advanced analytics capabilities performed on the data collected by the automated pipelines from the second section. This capability is made possible through the synthesis of relevant indicators from the collected data, tailored for AI4TRUST Platform end-users, such as fact-checkers, journalists, media practitioners, and policy-makers. Users will have the ability to filter the analysis across various dimensions, including topic, language, source platform, and time-period. Specific dashboards will address various types of large-scale analysis of the collected data, such as the Social Network Visualisation tool that will allow the end-users to evaluate how hierarchical, inequal or biased the social network is, or the Infodemic Observatory tool, which will be capable of tracking aggregated statistical information on the quantity of misleading news circulating in a certain period of time on different topics and across various social media platforms, both in absolute terms and relative exposure to the public.

Each version of the AI4TRUST Platform (v1, v2, and v3) progressively incorporates the described sections to enhance its overall effectiveness. The **incremental development of the AI4TRUST Platform** follows a logical and strategic process, aligned with the three Key Exploitable Results of the project: #1 - the Toolbox, #2 - the Monitoring and Human Validation, #3 - the Analytics about the collected data (for further details, please refer to D7.4[4]). Each of the three versions of the AI4TRUST Platform is described in more detail in the paragraphs below.

[4] D7.4 - Innovation, Exploitation, and Sustainability Plan v2 (https://ai4trust.eu/public-deliverables/)

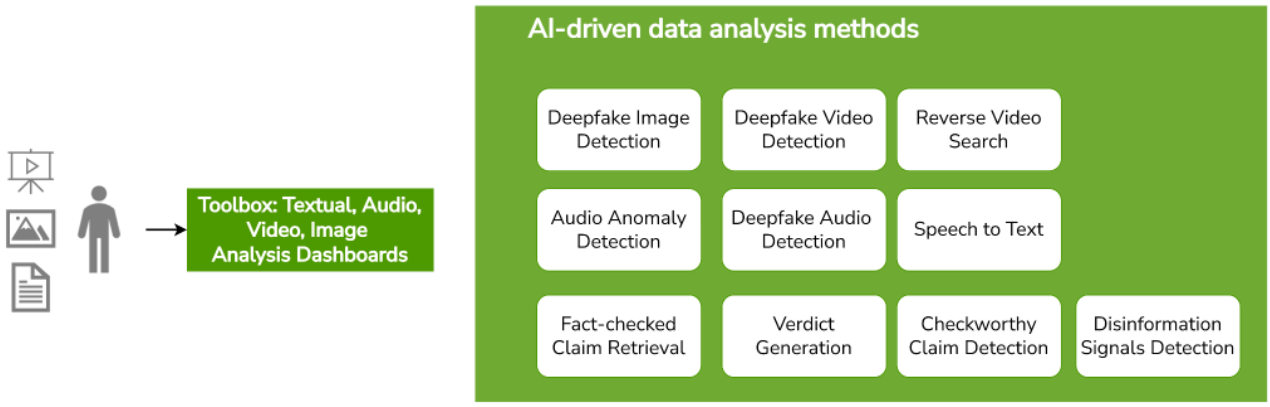

Figure 2: *Overview of the functionalities of AI4TRUST Platform v1*

The **AI4TRUST Platform v1** (see Figure 2), described in detail in D5.5[5], implements the Toolbox that provides end-users with access to AI-driven data analysis methods through dedicated dashboards. End-users can submit social media or news items, which are processed by the appropriate AI-driven analysis methods. The system then collects the results and presents them in a user-friendly format. Some of the AI-driven analysis methods introduced in v1 also serve as a foundation for more advanced automated analysis services in subsequent versions (v2, v3).

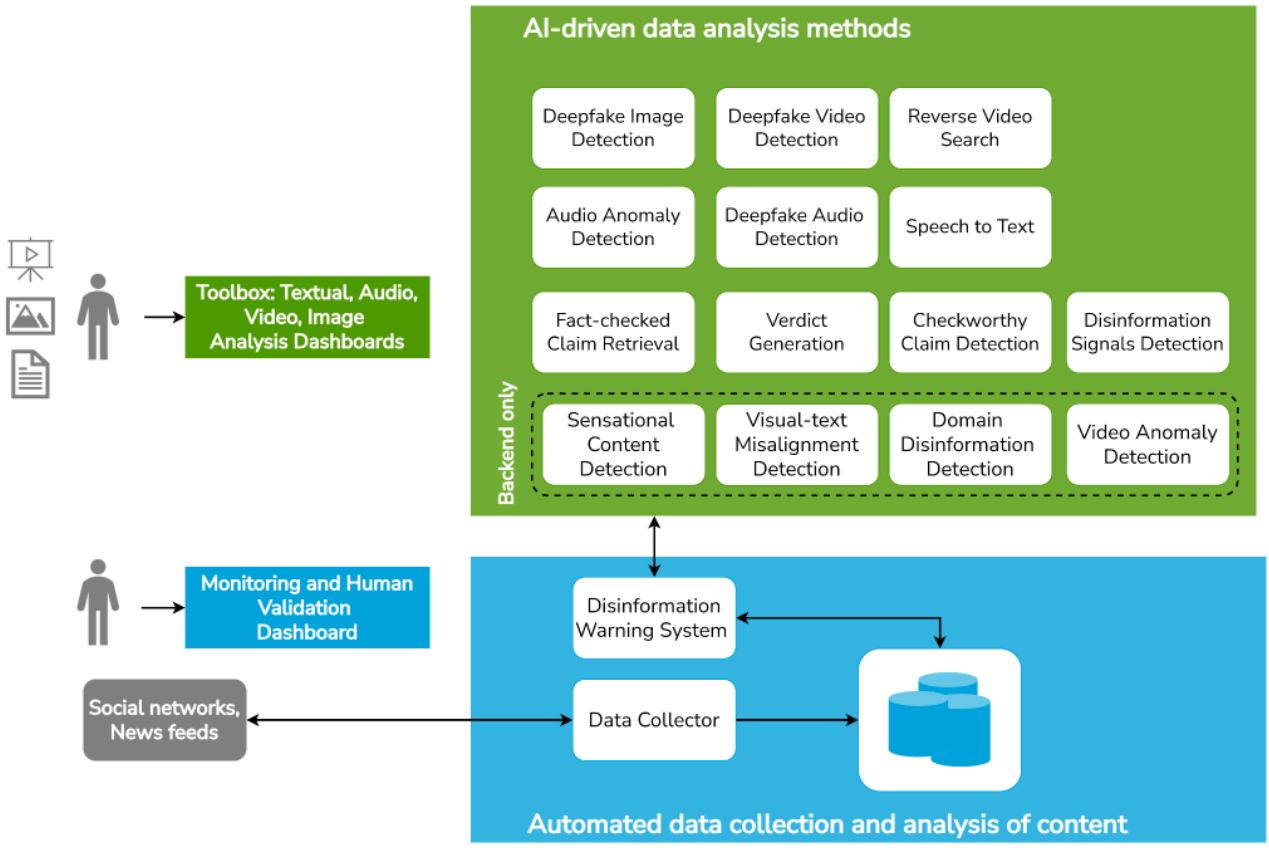

Figure 3: *Overview of the functionalities of AI4TRUST Platform v2*

[5] D5.5 - AI4TRUST Platform v1 (https://ai4trust.eu/public-deliverables/)

The **AI4TRUST Platform v2** (see Figure 3), the focus of this deliverable, enhances its predecessor by introducing **automated content collection and analysis from social media and news feeds**. It continuously gathers and processes content on predefined topics (Public Health, Climate Change, and Migrants) using a selection of **AI methods for text, visual, and multimodal analysis** (see Section 3.1) alongside the **Disinformation Warning System (DWS)** (see Section 3.8). The DWS evaluates these outputs to flag potentially misleading content, enabling real-time analysis, automated detection of checkworthy content, and timely insights for fact-checkers. End-users can access data and analysis results via a monitoring dashboard, while fact-checkers can leverage a human validation dashboard to manually review and validate flagged content. For further details, see Section 2.2.

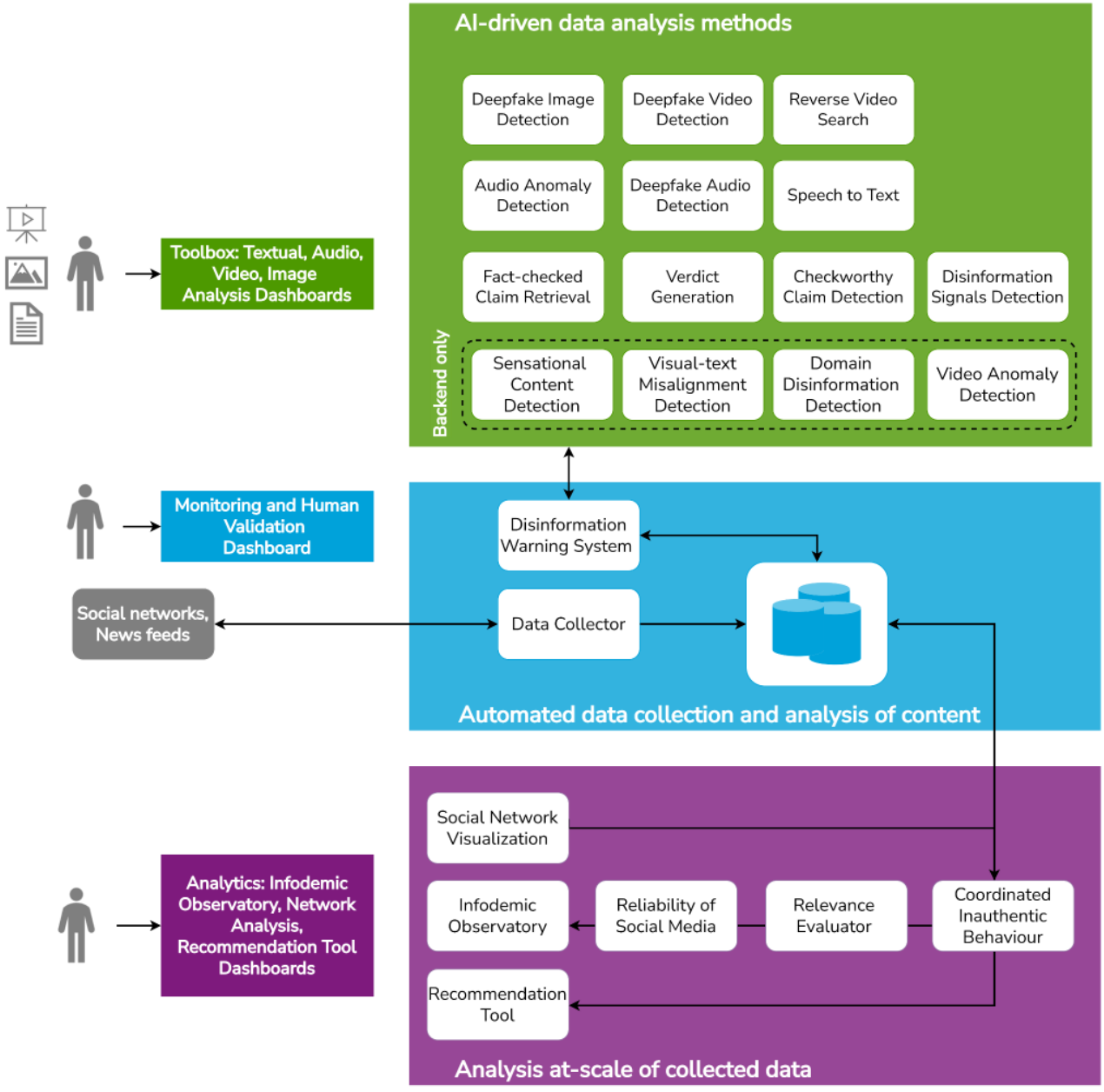

Figure 4: *Overview of the functionalities of AI4TRUST Platform v3*

The **AI4TRUST Platform v3** (see Figure 4), described in the forthcoming D5.7[6], will extend the previous one, building upon the automated data collection and analysis pipeline and adding an advanced **analysis-at-scale of collected data**, which will provide users with contextual and analytical insights based on historical data. This analysis will be employed to track the spread of disinformation, through advanced analytics, including source reliability assessments and infodemic trend monitoring. Additionally, **policy-oriented services** will be introduced to monitor disinformation risks and assist policymakers in developing more effective countermeasures. The results will be seamlessly integrated into **interactive dashboards**, offering end-users with visualisations and trend analyses across the entire dataset.

After providing an overview of the platform's iterations to present a comprehensive picture, this deliverable focuses on **AI4TRUST Platform v2**, as shown above. It details the **updated and newly introduced AI-driven data analysis methods** — building on the previous iteration — and **the new automated data collection and analysis pipeline**. These aspects are explored through the **two scenarios** described in the following sections:

- **Fact-checking media items using the AI4TRUST Toolbox:** highlighting the key advancements from the earlier version, focusing on improvements in the AI-driven data analysis methods interfaces, usability, and functionalities.
- **Accessing automatically flagged contents using AI4TRUST Monitoring and Human Validation Dashboard**: focusing on the identification of potentially misleading content, based on the automated collection and analysis of content from social media and news feeds.

2.1. Fact-checking media items using the AI4TRUST Toolbox

The **AI4TRUST Toolbox** allows users to analyse content from different data modalities (i.e., text, audio, image, video), get clues about the trustworthiness of the associated media item, and spot cased of mis/disinformation (for further details about the Toolbox implemented in AI4TRUST Platform v1 please see D5.5[7]). In AI4TRUST Platform v2, **several data analysis methods of the Toolbox have been advanced** based on the received users' feedback after the first piloting session (see D6.2[8]). Moreover, the **updated dashboard** features a refreshed design, incorporating updated colours and improved descriptions about the project and the integrated AI methods in the Toolbox. An overview can be seen in Figure 5.

[6] D5.7 - AI4TRUST Platform v3 (due by M32 - August 2025, https://ai4trust.eu/public-deliverables/)
[7] D5.5 - AI4TRUST Platform v1 (https://ai4trust.eu/public-deliverables/)
[8] D6.2 - Piloting sessions report v1 (https://ai4trust.eu/public-deliverables/)

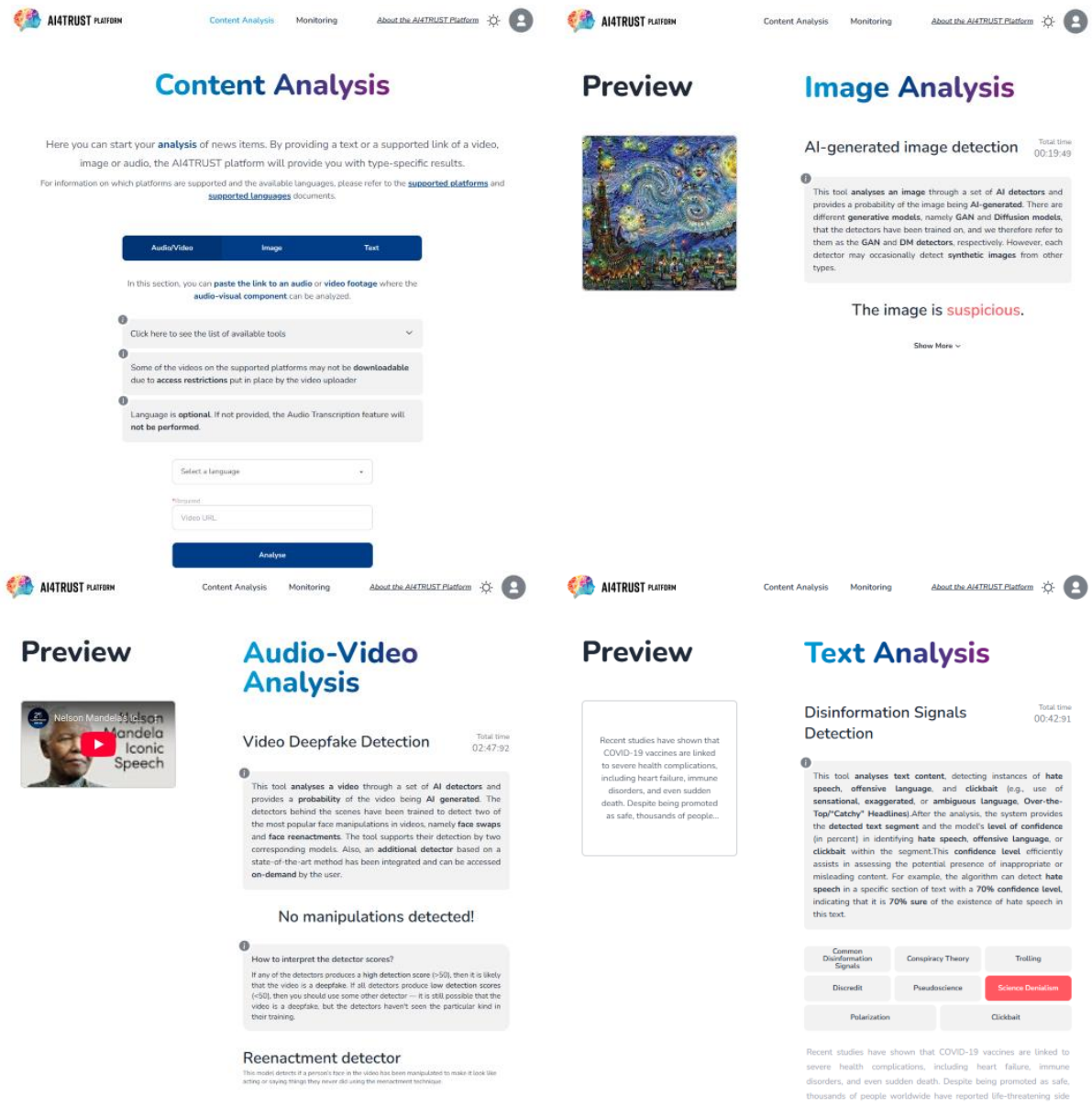

Figure 5: *Toolbox in AI4TRUST Platform v2*

The Toolbox offers a **structured and intuitive interface** tailored to the type of input content, enabling users to select between audio/video, image, and text analysis. This structure ensures a seamless and efficient experience when submitting content for evaluation.

- **Audio/Video**: Requires a URL to the content along with an optional language selection, which is used for audio transcription purposes.
- **Image**: Requires a URL pointing to the image.
- **Text**: Requires textual content and a language. Additionally, an optional title can be included to provide further context for analysis.

When the content is provided, an AI-driven analysis is conducted, leveraging multiple state-of-the-art tools to extract insights and detect potential manipulation. This enhances the

mis/disinformation debunking process by allowing end-users to explore various aspects of the analysed content in detail.

A key enhancement in AI4TRUST Platform v2 is the **integration of video and audio analysis**, which were previously separated within version 1. This unification enhances the platform's ability to assess multimedia content more comprehensively. By processing both visual and audio data within a single workflow, the system enables a **more seamless and holistic evaluation of video content**.

Beyond **visual improvements**, the Toolbox of AI4TRUST Platform v2 integrates a r**efined and updated set of tools**, enhancing the **overall usability**. These updates were guided by the received users' feedback after the first piloting session (see D6.2[9]), ensuring a more intuitive experience and streamlined workflow. A **detailed comparison** of the toolset of AI4TRUST Platform v1 and the UI-based improvements introduced in AI4TRUST Platform v2 is provided in **Table 1**.

Table 1: *Tools available in AI4TRUST Platform v1 vs AI4TRUST Platform v2*

Content Type	AI-Driven Data Analysis Methods	AI4TRUST Platform v1	AI4TRUST Platform v2
Video/Audio	Deepfake Video Detection	YES	Updated UI (Added multimodal detection model)
Video	Reverse Video Search	YES	YES
Image	Deepfake Image Detection	YES	YES
Audio	Audio Anomaly Detection	YES	YES
Audio	Deepfake Audio Detection	YES	YES
Audio	Speech to Text	YES	Updated UI (Supported languages increased from 5 to 7)
Text	Check-worthy Claim Detection	YES	Updated UI (Supported languages increased from 1 to 3, and score for each segment of the text)
Text	Disinformation Signals Detection	YES	Updated UI (number of signals increased from 3 to 38)

[9] D6.2 - Piloting sessions report v1 (https://ai4trust.eu/public-deliverables/)

Content Type	AI-Driven Data Analysis Methods	AI4TRUST Platform v1	AI4TRUST Platform v2
Text	Verdict Generation	YES	Updated UI (Supported languages increased from 1 to 8)
Text	Fact-checked Claim Retrieval	NO	YES

AI4TRUST Platform v2 enhances version 1 by **expanding language support** for most language-dependent tools, making them accessible to a wider user base. Additionally, **key UI improvements** have been made to enhance functionality and usability. The following sections present a **subset of Toolbox tools**, focusing on the newly implemented **Fact-checked Claim Retrieval** and those with updated UIs (see Table 1). For details on tools introduced in Platform v1, such as **Reverse Video Search, Deepfake Image Detection, Audio Anomaly Detection, and Deepfake Audio Detection**, refer to D5.5 [10]. For technical details on all AI-driven data analysis methods, see Section 3.1.

2.1.1. Deepfake Video Detection

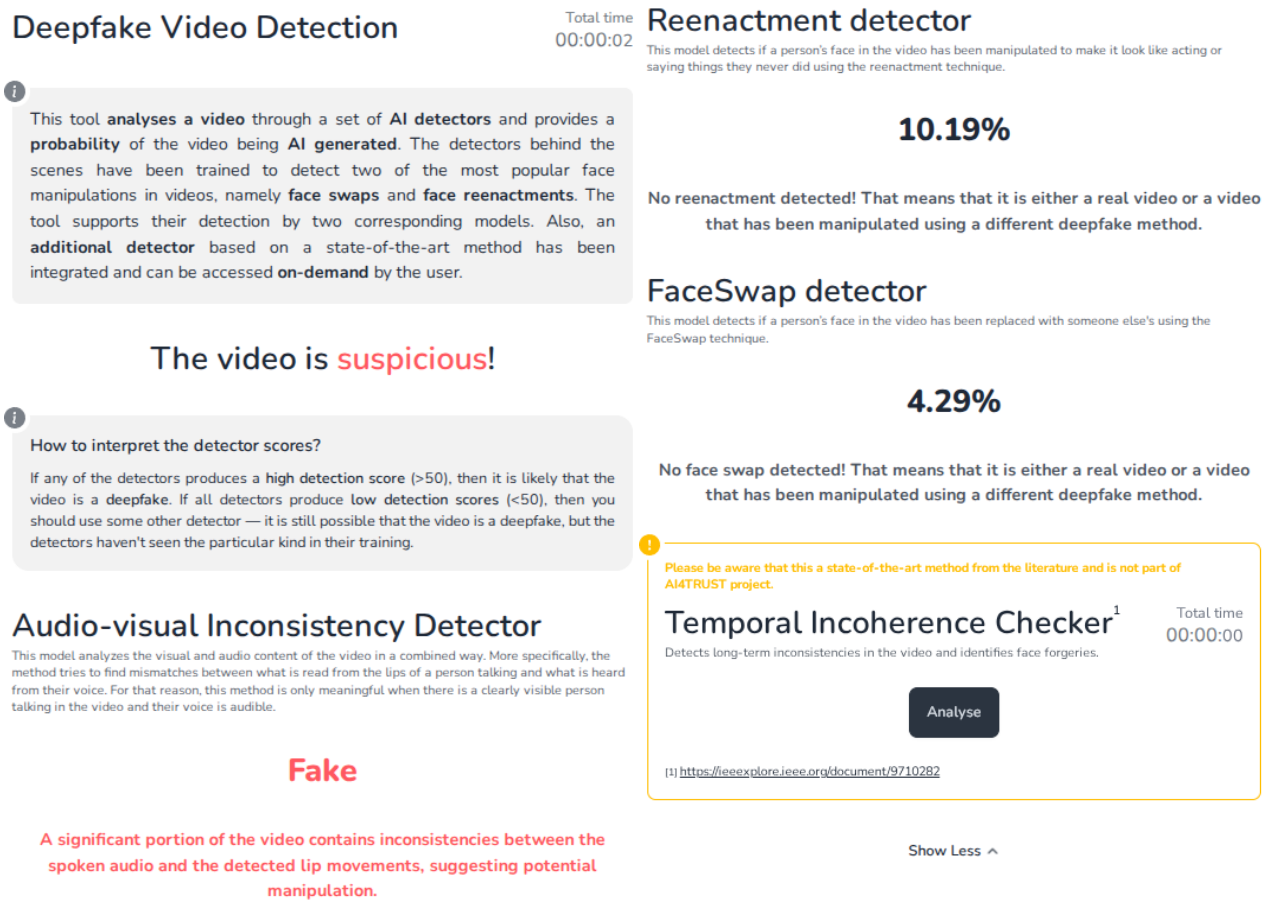

Figure 6: *Deepfake Video Detection in AI4TRUST Platform v2*

[10] D5.5 - AI4TRUST Platform v1 (https://ai4trust.eu/public-deliverables/)

The Deepfake Video Detection tool uses AI detectors to analyse videos for manipulation (D5.6). In AI4TRUST Platform v1, it included the Face Swap detector, which identifies face replacements, and the Reenactment detector, which detects facial alterations to simulate different speech or actions. In AI4TRUST Platform v2 it also introduced the Audio-visual Inconsistency detector, which examines both audio and video for mismatches between mouth movements and spoken audio, enhancing the tool's ability to detect deepfakes. Figure 6 shows the updated UI.

2.1.2. Speech to Text

The Speech to Text tool converts spoken content into text for further analysis. In AI4TRUST Platform v2, it now supports transcription in all project-defined languages: German, Greek, English, Spanish, French, Italian, Polish, and Romanian. This update significantly expands the tool's language coverage. An example is shown in Figure 7, where a French video is transcribed into text.

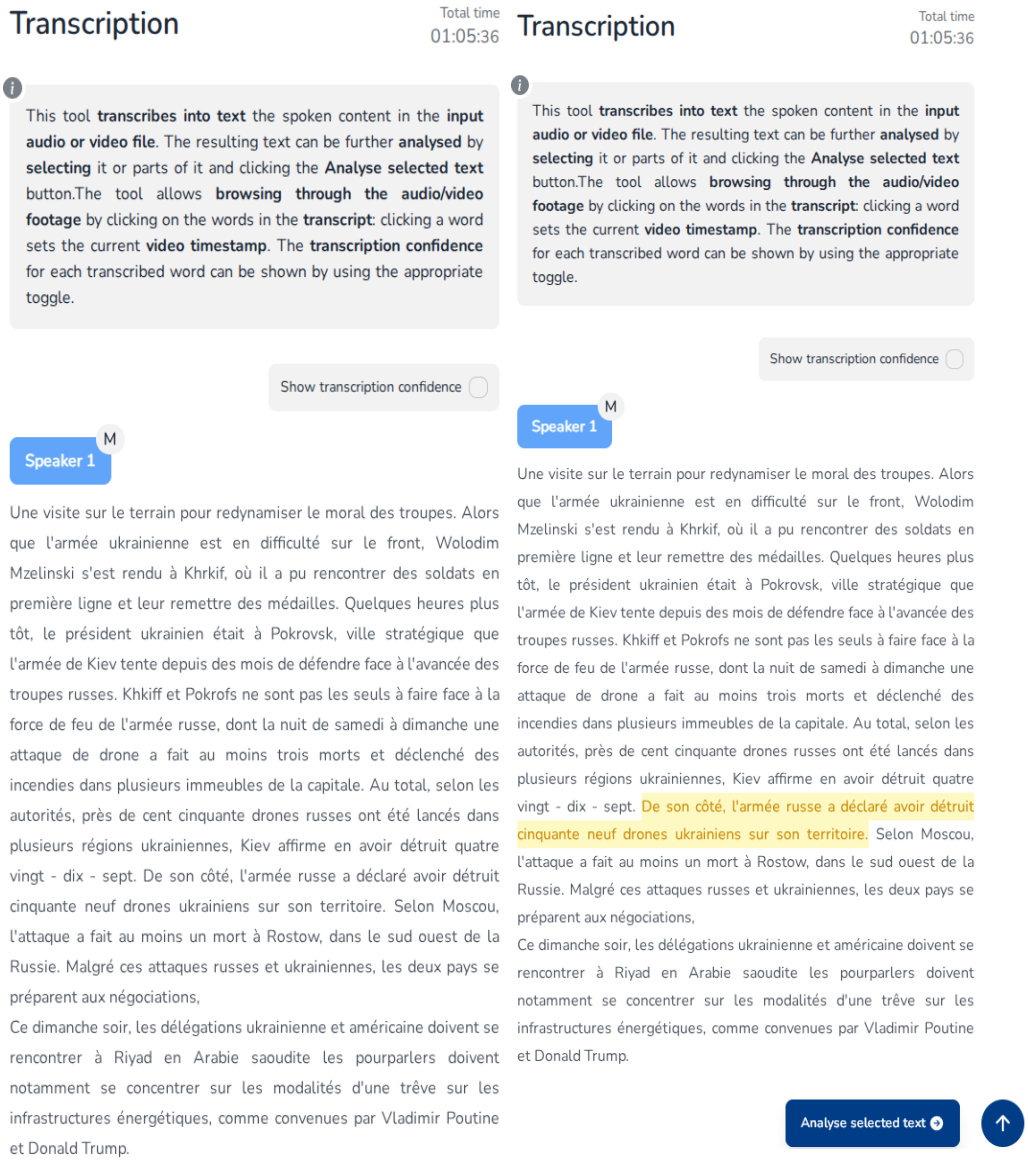

Figure 7: *Speech to Text in AI4TRUST Platform v2*

2.1.3. Check-worthy Claim Detection

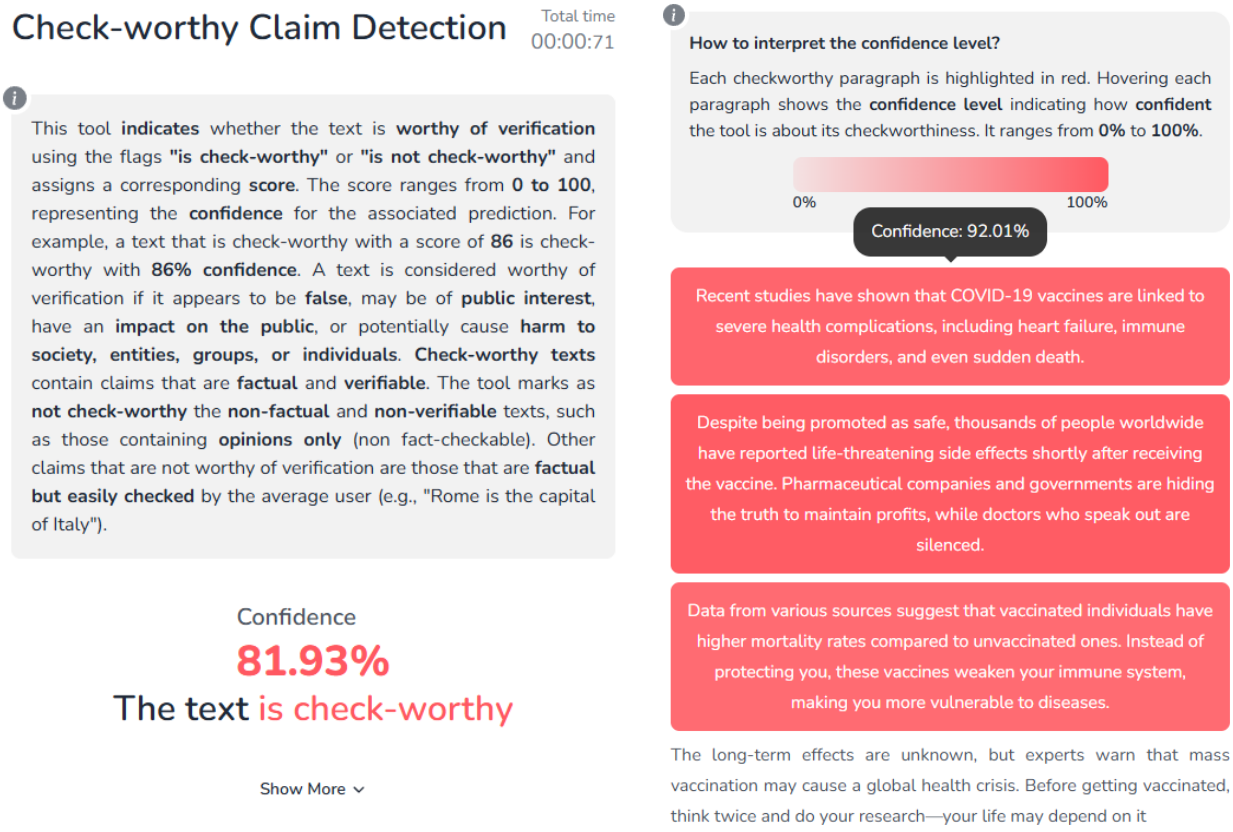

Figure 8: *Check-worthy Claim Detection in AI4TRUST Platform v2*

The Check-worthy Claim Detection tool evaluates content for check-worthiness by assigning labels and confidence scores, identifying potentially false, impactful, or harmful claims while excluding opinions and easily verifiable facts. In AI4TRUST Platform v2, it also offers an enhanced view (Figure 8) with detailed segmentation of text, highlighting check-worthy segments with intensity-based colours and confidence scores. Users can access deeper insights by clicking "Show More" and hovering over highlighted text. The tool now supports additional languages, including Spanish and Italian. See Section 3.1.5 for more details.

2.1.4. Disinformation Signals Detection

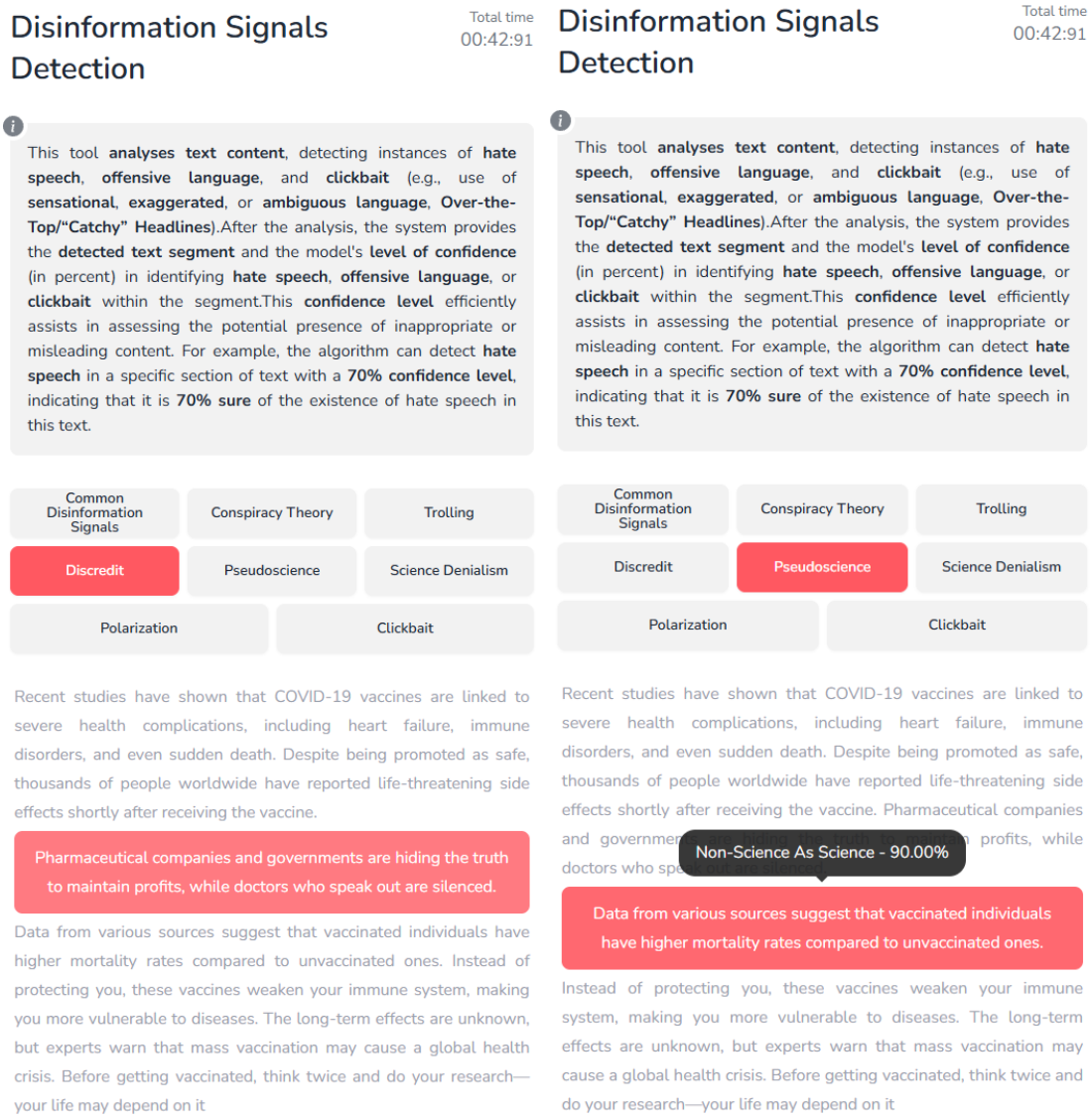

Figure 9: *Disinformation Signals Detection in AI4TRUST Platform v2*

The Disinformation Signals Detection tool analyses text and labels segments to identify misleading, ambiguous, harmful, or false content. In AI4TRUST Platform v2, it features an improved UI and supports over 30 additional signals, grouped into Tactics (Common Disinformation Signals, Conspiracy Theory, Trolling, Discredit, Pseudoscience, Science Denialism, Polarisation, and Clickbait). Users can select Tactics via buttons and view signal titles and confidence by hovering over highlighted text segments (see Figure 9).

2.1.5. Verdict Generation

The Verdict Generation tool evaluates textual input against a provided source of truth to generate a verdict. In AI4TRUST Platform v2 the tool has been advanced to support all eight project-defined languages: German, Greek, English, Spanish, French, Italian, Polish, and Romanian. Additionally, it

has undergone minor UI improvements: the style and sentence selectors from the AI4TRUST Platform v1 have been removed, improving the user experience and providing more intuitive guidance. An example of the updated interface is shown in Figure 10.

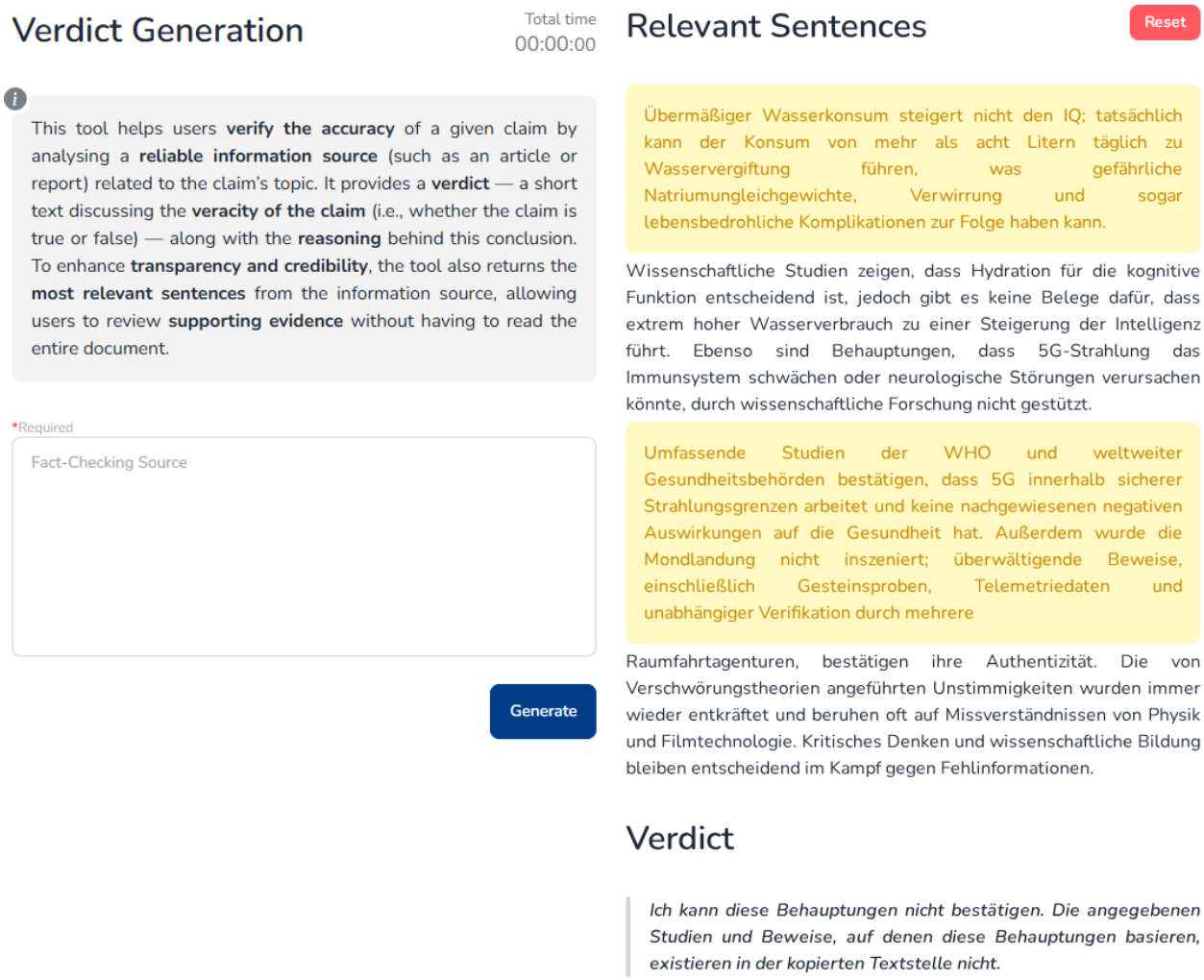

Figure 10: *Verdict generation in AI4TRUST Platform v2*

2.1.6. Fact-checked Claim Retrieval

The Fact-checked Claim Retrieval tool is a new component introduced in AI4TRUST Platform v2. This tool compares the given claim with a database of fact-checked content and provides a list of the most similar verified claims, along with a similarity score and a link to the relevant fact-checking article. The UI displays the top five most similar verified claims based on the provided textual input. Users can hover over a specific claim and click on it to access the associated fact-checking article, as illustrated in Figure 11.

Figure 11: *Fact-checked Claim Retrieval in AI4TRUST Platform v2*

2.2. Accessing automatically flagged contents using AI4TRUST Monitoring and Human Validation Dashboard

In addition to the enhanced Toolbox, AI4TRUST Platform v2 introduces a **new Monitoring and Human Validation Dashboard**, which serves as the central hub for monitoring automatically collected and analysed content (see Section 3 for further details). As shown in Figure 12, the Dashboard consists of two main sections:

- **Filter and Sorting Bar** – Allows users to refine and organise the displayed content based on specific criteria.
- **Content List** – Displays the automatically collected content in an intuitive, structured list, enabling users to easily explore and access the information.

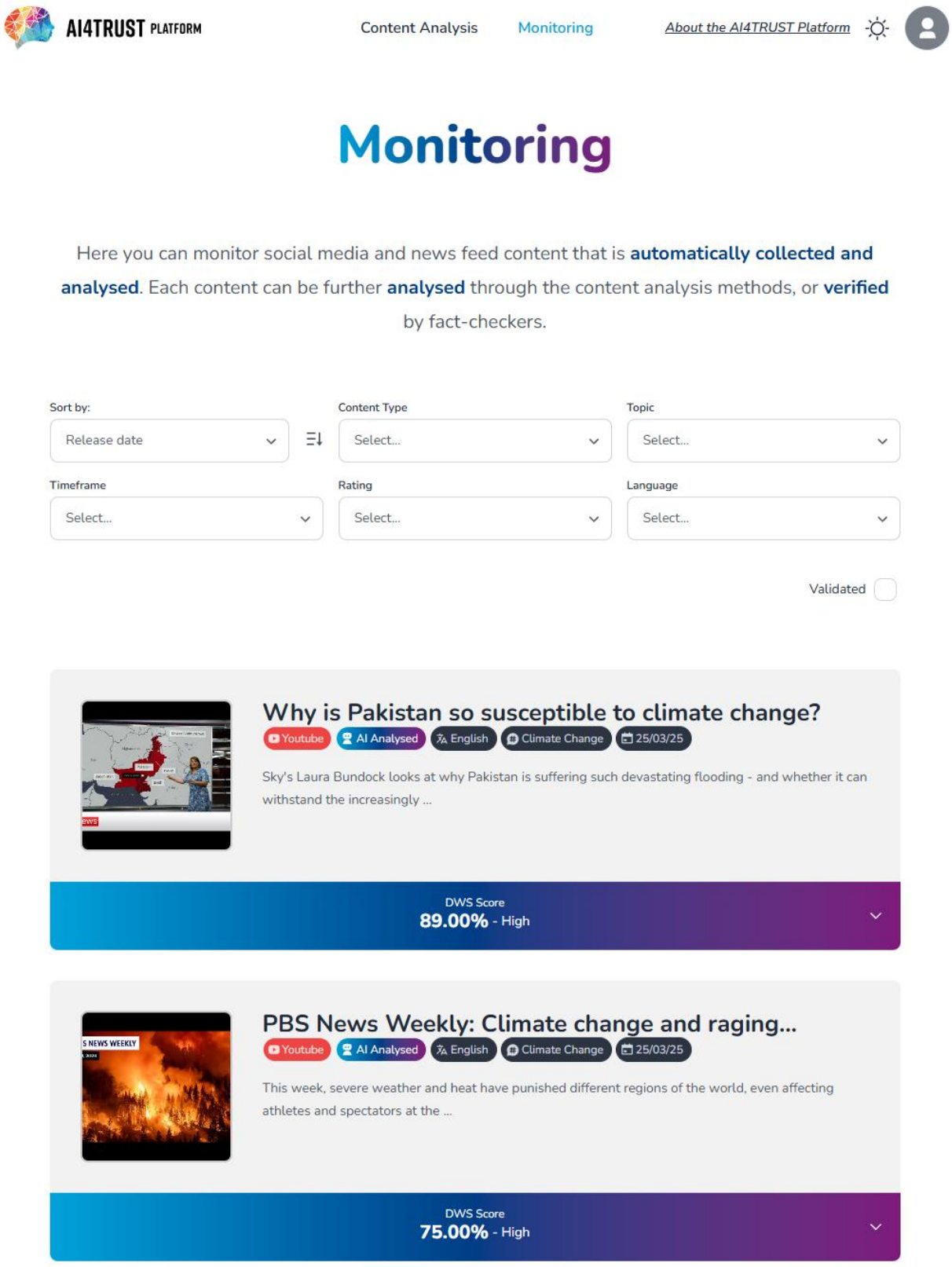

Figure 12: *Monitoring UI*

The **Filter and Sorting Bar** provides an interactive way to navigate through the collected data, ensuring that users can efficiently focus on content relevant to their needs. It offers the following sorting/filtering options:

- **Sort by:** Allows users to sort the content by Release Date, or by DWS Score. The sorting can be in ascending/descending order;
- **Content Type**: Allows users to filter content by source (e.g., YouTube videos, news articles, etc.);
- **Topic**: Filters content according to the three project-defined topics: Climate Change, Migration, and Public Health;
- **Timeframe**: Enables filtering based on date intervals, with three predefined options:
    - **Today**: Displays only content collected within the current day;
    - **Last Week**: Shows content gathered from today up to seven days prior;
    - **Last Month**: Lists content collected within the past 30 days;
- **Fact-Checking Rating**: Filters content based on its validation status;
- **Language**: Narrows results based on languages supported by the project;
- **Validation Status**: A checkbox option allows users to filter between Human-validated content only (i.e., content that has been validated by a fact-checker through the Human Validation functionality), non-Human-validated content only (i.e., content that was not validated by a fact-checker through the Human Validation functionality), or both.

An example of filtering is illustrated in Figure 13.

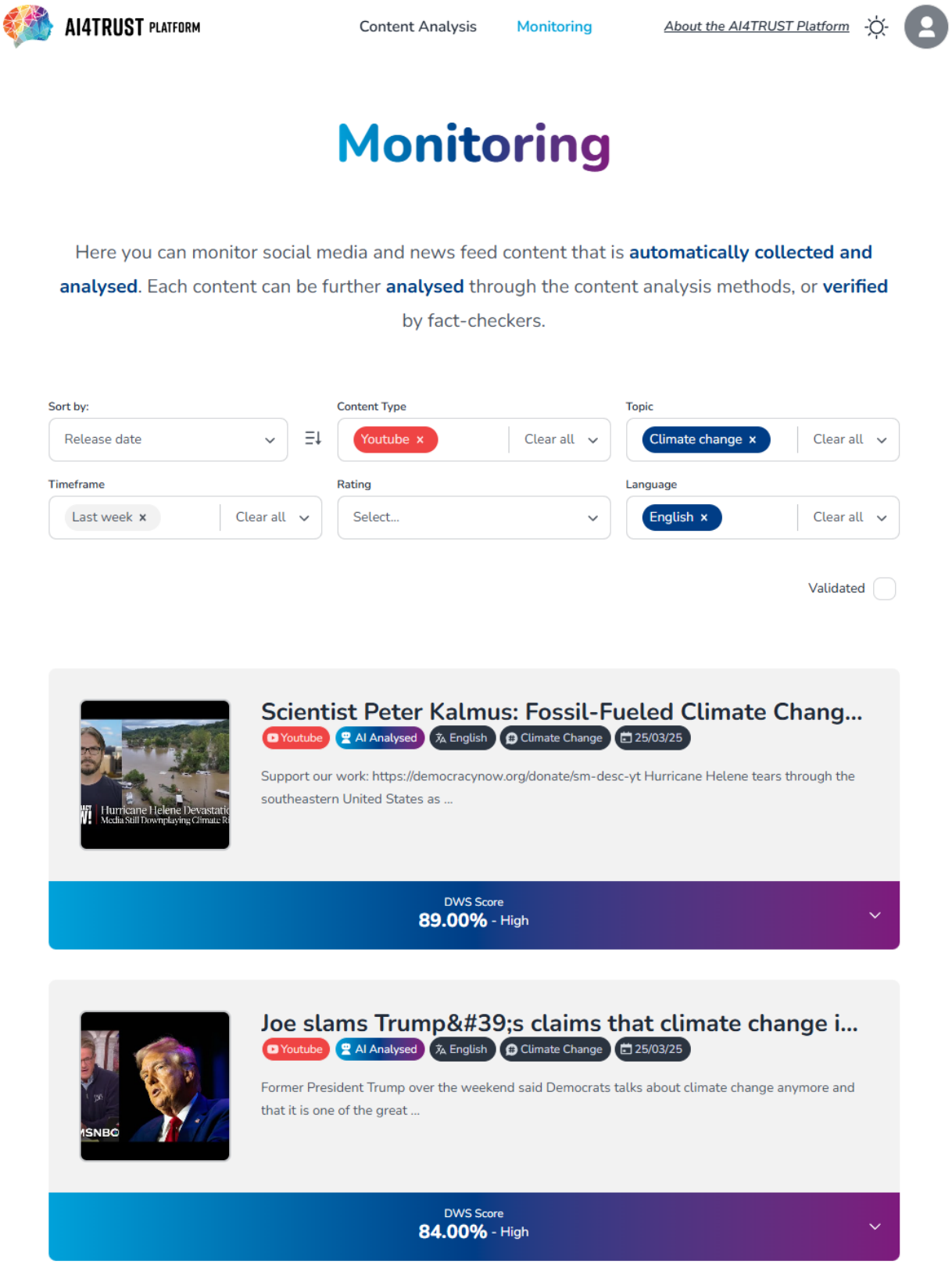

Figure 13: *Filtering example*

The **Content List** provides a structured and interactive way to explore content automatically collected by the Data Platform. Each content (see Figure 14) is displayed in a structured way, and

includes an **image preview,** a **title**, a **description**, a set of **badges**, and an expandable **analysis section** (represented specifically in the Figure 14 by the blue and purple bottom section).

Figure 14: *Monitoring content example*

The **title** and **description** varies by content type (e.g., for a YouTube video, the title and description correspond to the video's title and its provided description; for a news article, the title reflects the headline of the article, while the description consists of its main content; etc.), and the same applies also for the **image preview** (e.g., for YouTube videos, the preview displays the video thumbnail, which is the default image representing the video on the platform; for news articles, the preview features the landing image associated with the article; etc.). In some cases, a content may not provide an associated image. When this occurs, a placeholder image is displayed instead to maintain a consistent layout within the dashboard (see Figure 15), ensuring a visually-coherent user experience.

Figure 15: *Example of content without an image*

Each piece of content includes **badges** summarising key metadata, such as content type, language, topic, collection date, and whether it has been analysed by the Disinformation Warning System (DWS) or validated by a fact-checker. The analysis section adapts depending on whether the content has been **analysed by AI (DWS) or validated by a human fact-checker**.

AI Analysis (DWS): If the content has been analysed by the DWS (see Figure 16), the section displays the calculated DWS score along with a corresponding label (High, Medium, or Low). Expanding this section reveals the list of tools used for analysis, the weights indicating each tool's influence on the final DWS score, and a confidence indicator, which can be “High”, “Medium”, or “Low”.

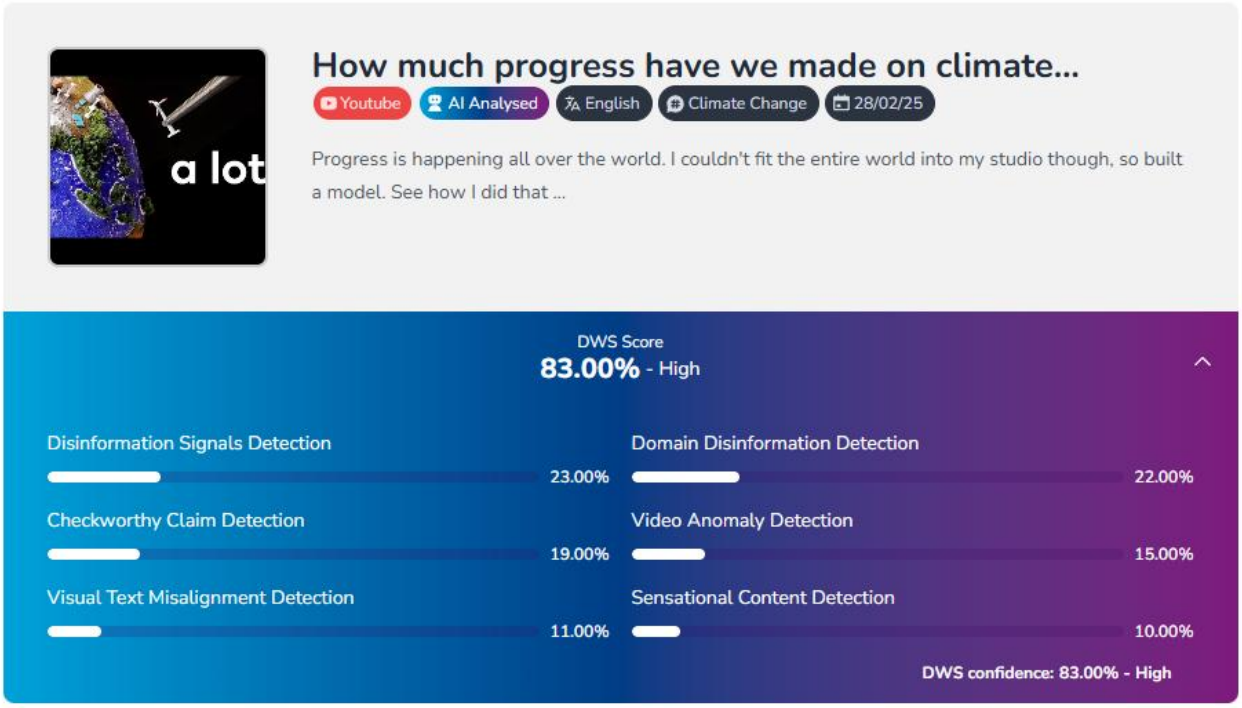

Figure 16: *DWS Analysis section*

Human Validation: When content is fact-checked by a human through the Human Validation functionality (detailed below), the section instead presents the final rating assigned by the fact-checker (see Figure 17). Expanding it provides details such as the claim associated with the content, the debunking article title and link, the validation date, and the fact-checking organisation responsible for the assessment through the platform. Based on the feedback from fact-checkers, it was agreed to display only the organisation associated with the account that performed the fact-checking within the AI4TRUST Platform.

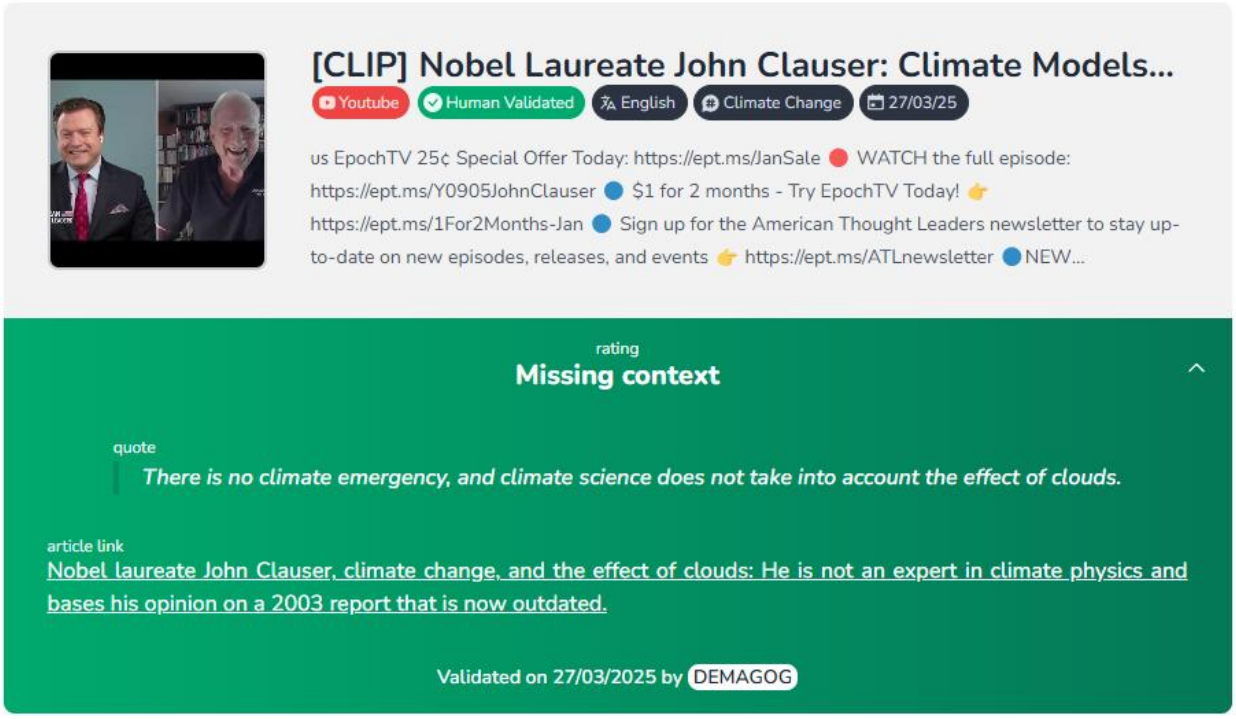

Figure 17: *Fact-checking analysis section*

Each content can be selected, as shown in Figure 18. Once selected, users can perform a series of actions on the content:

- **Analyse**: This feature allows users to **manually** analyse various parts of the content through the AI-driven data analysis methods provided in the first version of the platform. For YouTube videos, the analysis can be applied to the video itself, the audio, the title and description (text), and the thumbnail (image); for news articles, the analysis can be performed on the landing image or the title and article content (text);
- **Validate**: This feature is exclusive to registered fact-checkers within the platform, allowing them to select one or more pieces of content (see Figure 19) (ideal for analysing similar or related content) for Human Validation functionality.
- **Go to source:** Allows users to access the original content (YouTube video or news article).

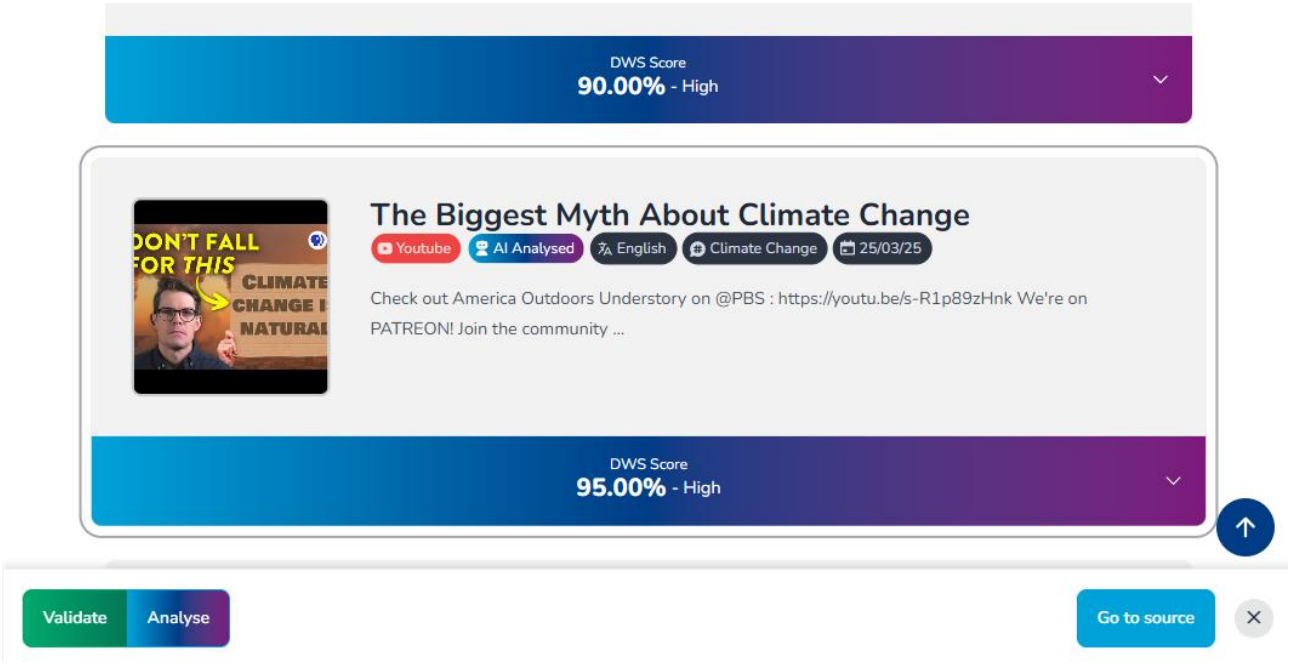

Figure 18: *Content selection example*

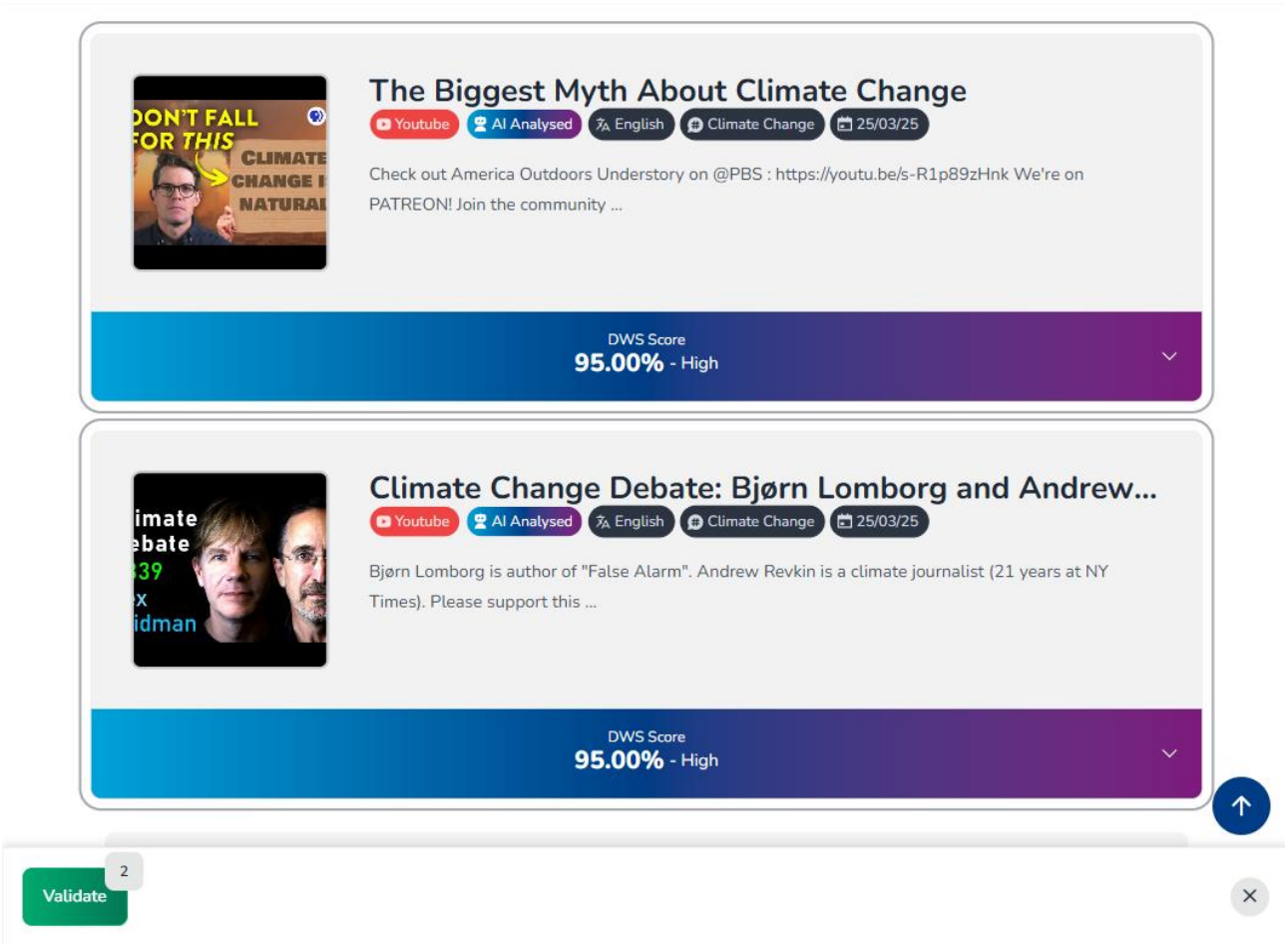

Figure 19: *Multi-content selection example*

To perform a **Human Validation**, one or more contents can be selected, and a Validate button is shown at the bottom of the page (see Figure 18). Upon clicking the button, a form is displayed for a fact-checker to complete. The Human Validation functionality was co-designed in collaboration with the fact-checkers within the consortium, providing a robust tool that enables fact-checkers to manually validate content directly within the Monitoring and Human Validation Dashboard. The Validation form is shown in Figure 20.

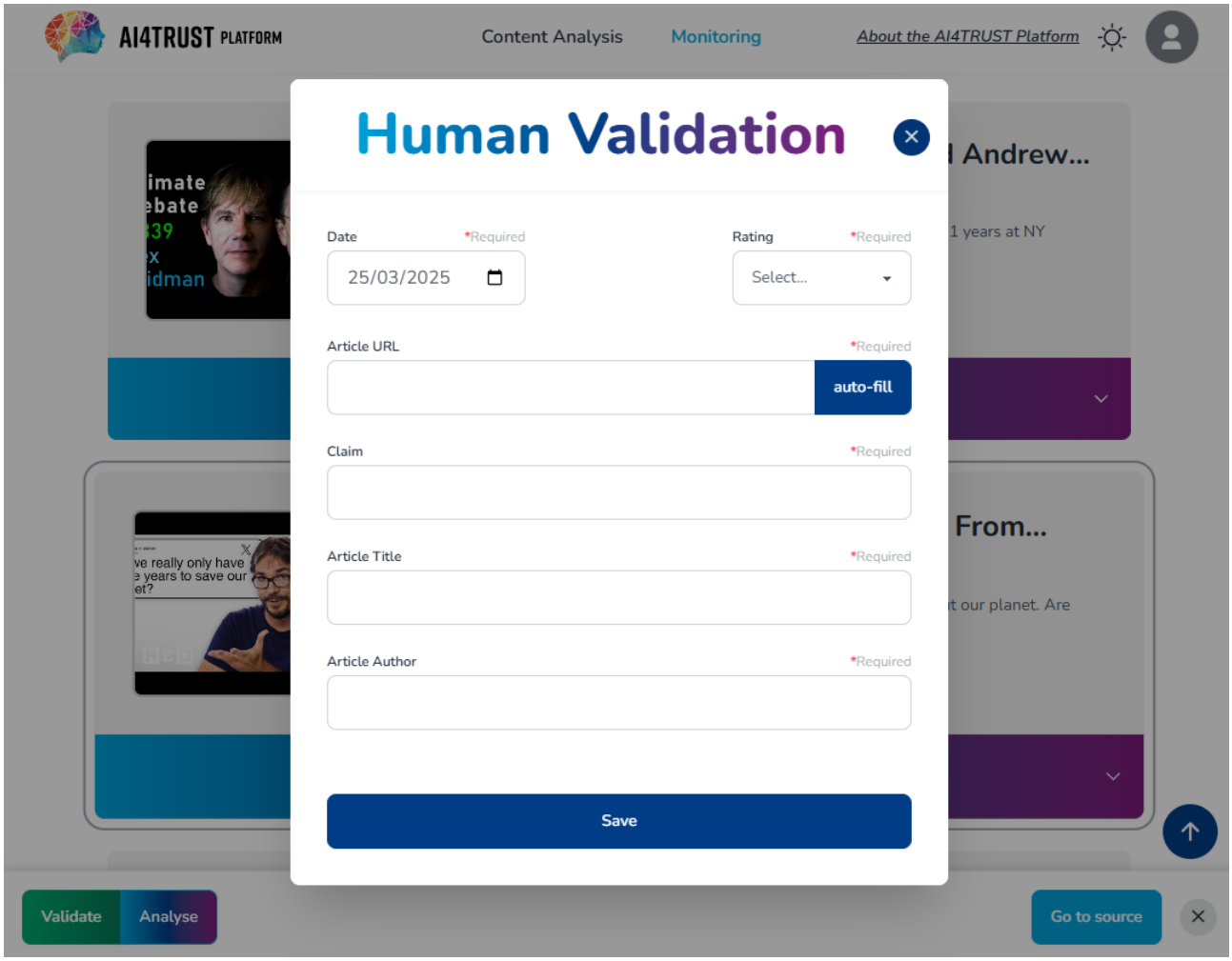

Figure 20: *Validation form*

The validation form is structured with fields that are directly aligned with the **ClaimReview**[11] **Markup schema**, adhering to the latest standards for reviewing claims. Additionally, to standardise the process, a uniform rating system was implemented. Recognising that each fact-checker organisation utilises its own rating system, the AI4TRUST Platform adopted the **Meta rating**[12] **standard**. To facilitate this, an equivalence table was created in collaboration with the fact-checkers within the consortium (see Table 2), mapping the organisation's ratings directly to the platform's standardised rating system.

[11] https://www.claimreviewproject.com/
[12] https://transparency.meta.com/features/content-ratings-fact-checkers-use/

Table 2: *Human Validation Rating Equivalence Table*

RATING	MALDITA[13]	ELLINIKA HOAXES	DEMAGOG
FALSE	Hoax; False context / False information, Hoax; false quote, Hoax; Invent, Hoax; False alarm	Ψευδές/ Fake News	Fałsz
ALTERED	Hoax; manipulated content	Τροποποιημένο Βίντεο/ Τροποποιημένη Εικόνα	Przerobione zdjęcie / Przerobiony film (Manipulated photo / Manipulated video)
PARTLY FALSE	No evidence	Μίξη γεγονότων και παραποιήσεων/ Παραπληροφόρηση	Częściowy fałsz
MISSING CONTEXT	What do we know	Λείπει Θεματικό Περιεχόμενο	Brakujący kontekst
SATIRE	Hoax; Satire	Σάτιρα	Satyra
TRUE	/	Αληθές	Prawda

The fields available for the fact-checkers to complete in the validation form are as follows:

- **Date Picker:** This field allows the fact-checker to select the publication date of the fact-checking article, ensuring that the content's timeline is accurately captured.
- **Rating:** The fact-checker assigns a rating to the content to be validated, reflecting the veracity of the claim based on their findings.
- **Article URL:** The direct link to the fact-checking article that validates/debunks the selected content, providing easy access for reference and review.
- **Claim:** This field identifies the specific claim associated with the content to be validated, helping to contextualise the content being reviewed.
- **Article Title** and **Author:** The title and the author of the fact-checking article are included to provide clear attribution and reference to the original content.

Note that the article's author could differ from the fact-checking organisation conducting the validation. In accordance with the fact-checkers, the platform's structure and UI defines as **article author** the organisation that owns the fact-checking article provided, and **validator** the organisation that validates the content within the platform. For example, a Demagog fact-checker

[13] Maldita employs two complementary taxonomies to determine the rating: "Internal Status" and "Disinformation Rating." The final result is derived from their combination, ensuring alignment with the META rating.

could validate the content within the platform while linking an article from Maldita[14], which would be associated with the article's author field. The UI then presents **the user with the validation date and the fact-checking author**, e.g., the organisation who performed the validation within the platform, which may differ from the article's author that can be retrieved from the article link. An indicative example can be seen in Figure 21:

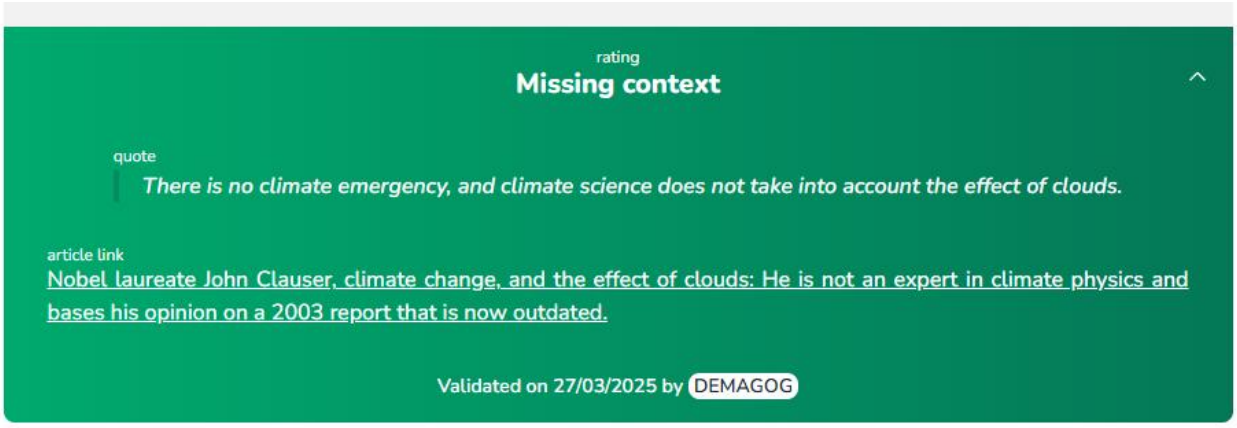

Figure 21: *Validation author example*

These fields are designed to capture **essential information for a standardised fact-checking process**, aligning with ClaimReview fields[15]. The validation form includes an autofill feature from Maldita's "Claim Review Extractor", which processes fact-checking article URLs to extract and populate metadata such as date, claim, rating, title, and author, reducing manual effort. Once saved, the form updates the selected content with the Human Validation results in the analysis section. For technical implementation details, please see Section 3.

[14] https://maldita.es/clima/20241230/premio-nobel-clauser-cambio-climatico/
[15] https://schema.org/ClaimReview

3. Technical Implementation

The scenarios outlined in the previous section were implemented through **backend business logic**, achieved **by significant integration efforts to ensure effective** communication between the components shown in Figure 22 and described below:

- **Data Collectors:** Components responsible for retrieving raw data from external news and social media sources.
- **Data Platform:** A platform that manages data collection, processing, and storage, logically divided into three main blocks: Streaming Platform, Serverless Platform, and Data Lakehouse.
- **Streaming Platform:** A data platform handling the normalisation, routing, and persistence of data during processing.
- **Serverless Platform:** A computing platform dedicated to executing processing steps during data collection, such as transformations, anonymisation, and auditing.
- **Data Lakehouse:** A unified data storage solution capable of storing both structured and unstructured data, offering advanced querying capabilities.
- **Dispatcher:** A component responsible for routing data to automatic processing.
- **API Gateway:** A centralised component for controlling access to the platform's components and storage.
- **AI-Driven Data Analysis Methods:** AI-based algorithms that perform textual, visual, audio, and multimodal analysis of news and social media content.
- **Disinformation Warning System (DWS):** A component that evaluates the risk of disinformation in media items, such as online articles or social media posts with associated images/videos, by aggregating the outputs of various analysis methods.
- **Database:** A storage solution that extracts a subset of information from the Data Lake in a structured format optimised for the Web Application, ensuring low-latency access and efficient data retrieval.
- **Web Application:** A component enabling end-users to interact with AI4TRUST Platform functionalities, including the Toolbox (see Section 2.1) and Monitoring and Human Validation (see Section 2.2).
- **Identity and Access Management:** A component ensuring secure and scalable user authentication and authorisation.

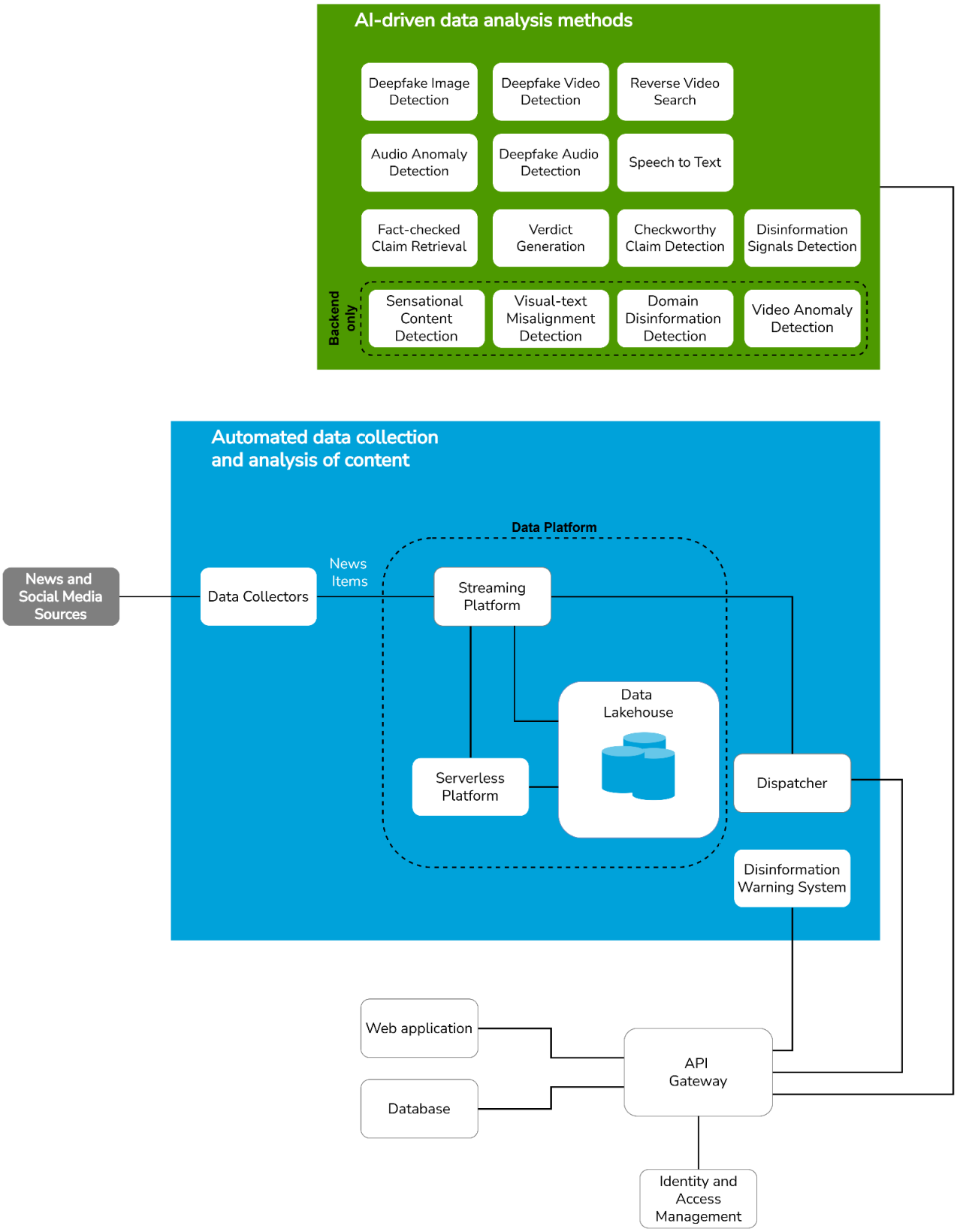

Figure 22: *Overview of the components*

The orchestration carried out in the backend by the system integrator is detailed in the following data flows. In the **Toolbox data flow**, the user provides the audio-video (see Figure 23), image

(see Figure 24) or text (see Figure 25) content through the **Web Application**, which sends it to the **API Gateway** that enriches it with the needed settings and forward it to the **appropriate AI-Driven Data Analysis Methods**. The outcomes are then returned to the Web Application that displays them with interactive dashboards as described in Section 2.1.

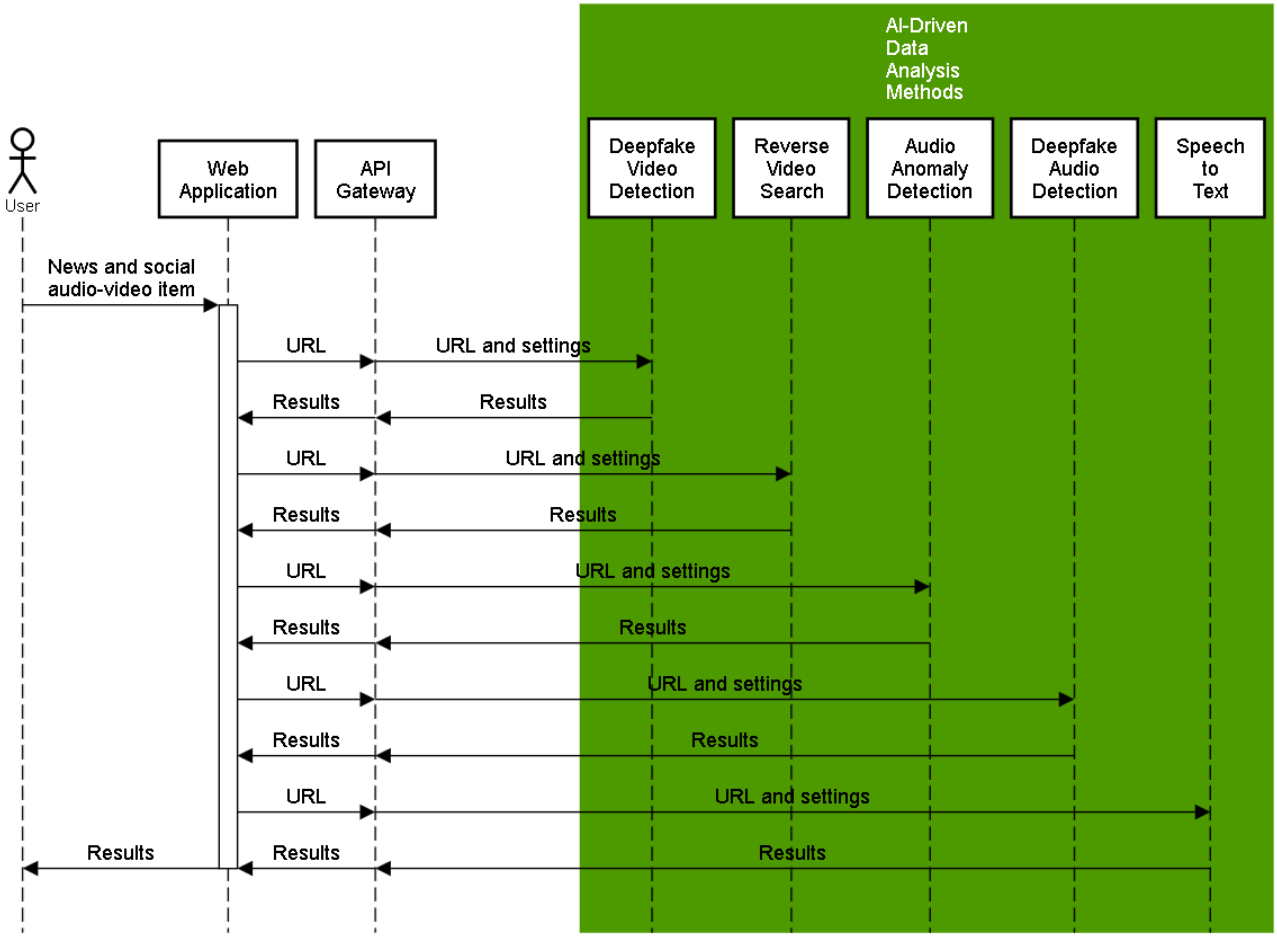

Figure 23: *The Toolbox data flow for an audio-video item*

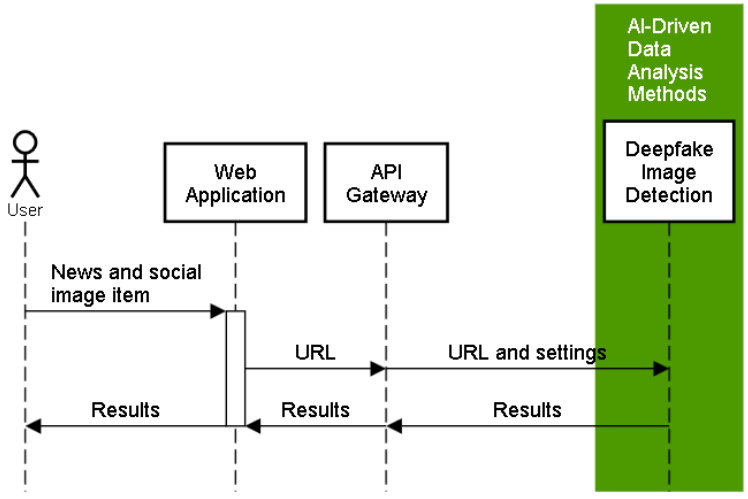

Figure 24: *The Toolbox data flow for an image item*

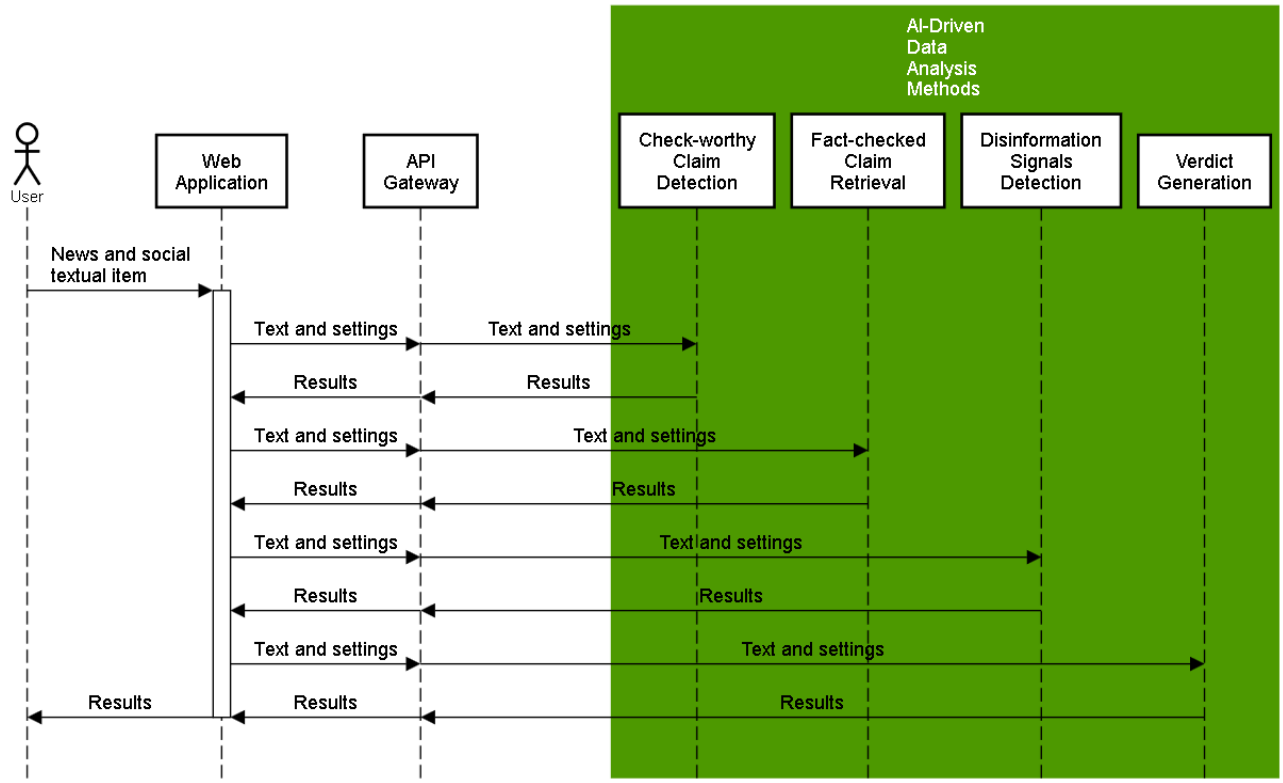

Figure 25: *The Toolbox data flow for a textual item*

In the **Monitoring and Human Validation data flow**, the data collected by the Data Collectors is sent to the Streaming Platform that formats the data and sends them to the Data Lakehouse for storing and to the Serverless Platform for preprocessing. The Serverless Platform processes the data, and returns the processed data to the Streaming Platform, which archives them in the Data Lakehouse and sends them to the Dispatcher (see Figure 26). For further details please refer to Section 3.2.

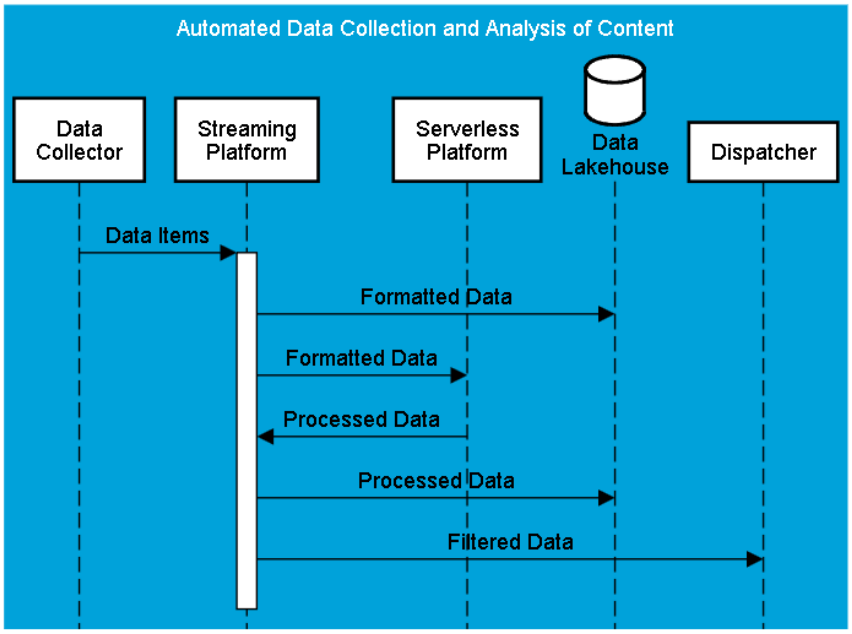

Figure 26: *Monitoring and Human Validation data flow for the data collection and preprocessing*

At this point the **Dispatcher component** handles the routing of the data, as shown in Figure 27. First, via the API Gateway, the Dispatcher sends the data to a subset of the AI-Driven Data Analysis Methods (for further details about the Dispatcher logics see Section 3.11). Then, once the data are processed by the AI-Driven Data Analysis Methods and the results are returned through the API Gateway, the Dispatcher forwards the results to the Disinformation Warning System (DWS). The DWS generates an aggregated result that is returned to the Dispatcher through the API Gateway. The Dispatcher sends all the outputs received to the Database and to the Streaming Platform, which in turn stores them in the Data Lakehouse.

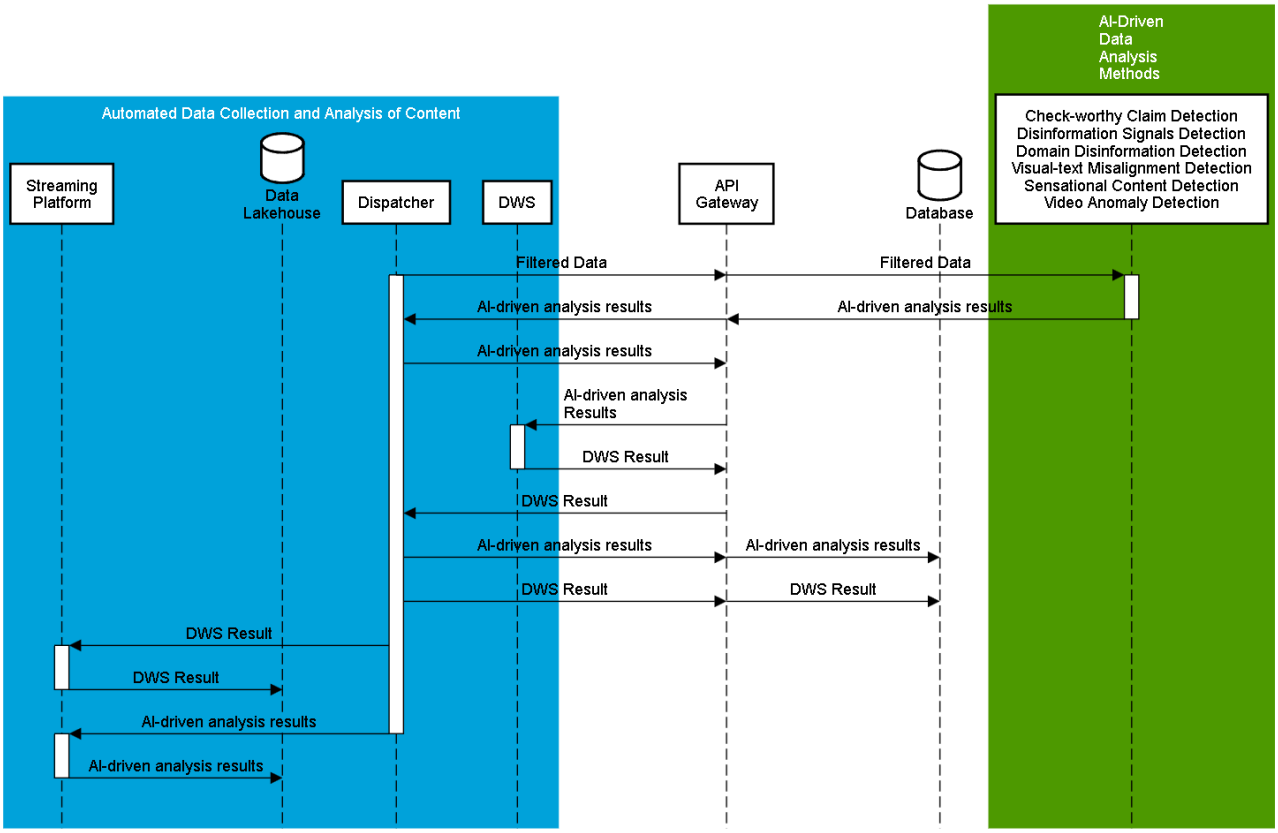

Figure 27: *Monitoring and Human Validation data flow for the processing*

As shown in Figure 28, when the user opens the **Monitoring and Human Validation Dashboard**, the Web Application retrieves the information stored in the Database via the API Gateway and displays it. In case the user is a fact-checker, the content retrieved can be fact-checked (Human Validation). Via the API Gateway, the Web Application sends the outcome of the Human Validation to the Database and the Streaming Platform, which in turn forwards it to the Data Lakehouse. This ensures that the outcomes are stored and available for display in the Monitoring Dashboard, as described in Section 2.2.

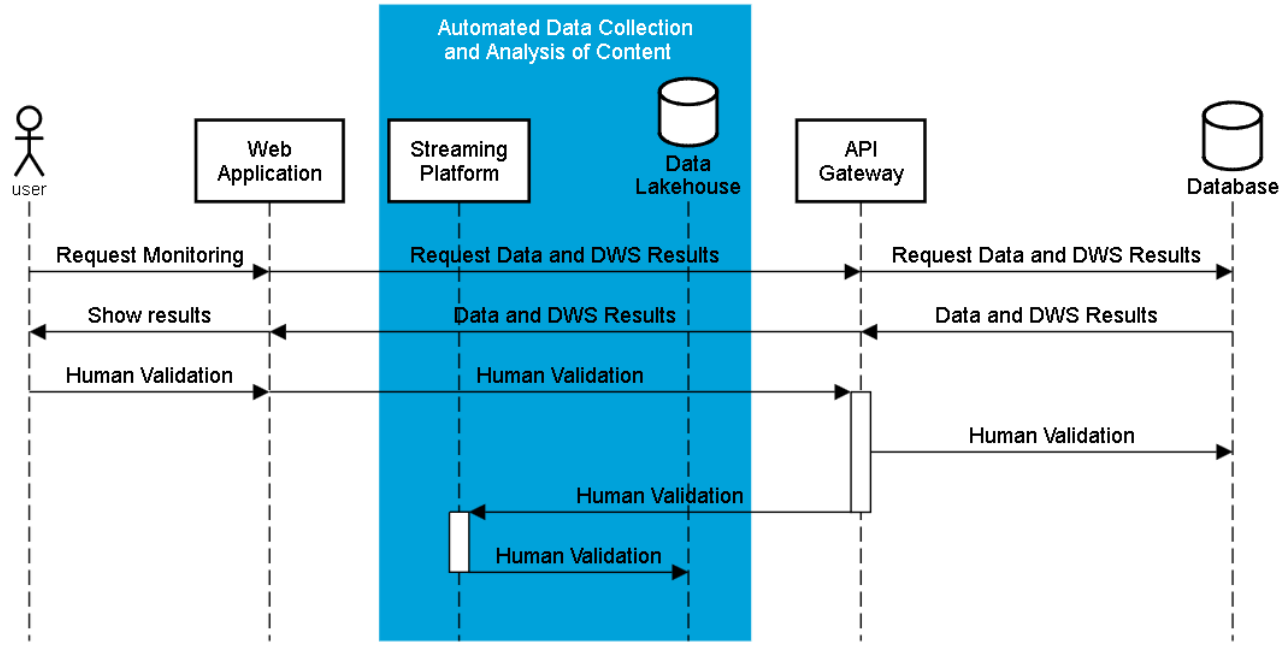

Figure 28: *Monitoring and Human Validation data flow for the dashboard*

In the following sections, each of the mentioned components are described in detail, providing information about their implementation and the embedded logics.

3.1. AI-Driven Data Analysis Methods

The AI4TRUST Platform enrols a number of **AI-based algorithms** responsible for the **textual, visual, audio and multimodal analysis of news and social media items**. As shown in Table 3, some of them are directly shown to the user via the **Toolbox** ("Frontend usage") and some of them power the Monitoring and Human Validation process behind the scenes. ("Backend usage").

Table 3*: AI-Driven Data Analysis Methods usage.*

AI-Driven Data Analysis Methods	Frontend usage (user-facing)	Backend usage
Deepfake Video Detection	YES	NO
Reverse Video Search	YES	NO
Video Anomaly Detection	NO	YES
Deepfake Image Detection	YES	NO

AI-Driven Data Analysis Methods	Frontend usage (user-facing)	Backend usage
Sensational Content Detection	NO	YES
Audio Anomaly Detection	YES	NO
Deepfake Audio Detection	YES	NO
Speech to Text	YES	NO
Check-worthy Claim Detection	YES	YES
Disinformation Signals Detection	YES	YES
Verdict Generation	YES	NO
Fact-checked Claim Retrieval	YES	NO
Domain Disinformation Detection	YES	YES
Visual-text Misalignment Detection	NO	YES

Below is provided a **short recap of the AI-Driven Data Analysis Methods** (for further details please refer to D3.1[16]); the asterisk ("*") indicates the ones that were introduced for the first time in the AI4TRUST Platform v2.

- **Deepfake Video Detection**: identifies AI-manipulated videos (commonly known as deepfakes) using multiple detection models, generating a descriptive analysis and a probability score.
- **Reverse Video Search**: performs a reverse video search on the Web, using a set of selected representative video keyframes and the relevant functionality of online search engines (e.g., Google Lens). It automatically fetches the retrieved results and provides the user with links

[16] D3.1 - First release of AI tools for disinformation detection (https://ai4trust.eu/public-deliverables/)

to a) the retrieved near-duplicate videos, and b) online sources containing similar visual content.

- **Video Anomaly Detection*:** analyses a given video and outputs a score in [0,1] indicating the presence of abnormal events in the input video, where 0 means the video is normal and 1 means it contains abnormal events.
- **Deepfake Image Detection**: detects AI-generated or manipulated images, providing a descriptive analysis and a probability score.
- **Sensational Content Detection*:** analyses the visual content of a given image or video and returns the most relevant action/event from a predefined list of sensational actions/events (typically found in disinformation items) along with a sensational score; the score represents the similarity of the media item (video, image, or both) to the identified action/event, with higher scores indicating greater similarity.
- **Audio Anomaly Detection**: analyses the audio segment by segment and identifies anomalous segments (segments that might comprise splicing points or might have been generated using AI)
- **Deepfake Audio Detection**: analyses the audio as a whole and indicates whether the audio is real or was fully generated using AI
- **Speech to Text**: transcribes into text the spoken content in the input audio or video file.
- **Check-worthy Claim Detection:** detects check-worthy claims (i.e., factual and verifiable text that appears to be false, may be of public interest or of impact to the public, or may cause harm to the society, entities, groups, or individuals), labelling the given input text (e.g., a post on a social media platform) as “check-worthy”, and provides a confidence score.
- **Disinformation Signals Detection:** analyses input text to identify a variety of disinformation signals. It detects and categorises signals such as hate speech, offensive language, and clickbait. The system also identifies unique signals associated with specific manipulation tactics, grouping them into six prevalent tactics: conspiracy theories, discrediting, trolling, pseudoscience, science denialism, and polarisation. Additionally, it detects common disinformation signals that do not fall under any specific tactic, such as emotional manipulation. The system annotates relevant text segments with appropriate labels and assigns confidence scores to each labelled segment. The output is a structured set of annotated text segments, containing the original text, assigned label(s), and confidence score(s).
- **Verdict Generation:** Generates reliable and professional responses (verdicts) about a claim, using a trusted textual source (i.e., fact-checking articles), and an indication about the number of relevant sentences from the trusted textual source to be displayed as evidence.
- **Fact-checked Claim Retrieval*:** checks whether the claim has already been verified or is similar to past claims contained in the AI4TRUST database. It provides the 5 most similar, already-verified claims, with a similarity score of at least 0.8.

- **Domain Disinformation Detection*:** classifies a domain as present (1) or not present (0) by extracting it from a URL using a system that indexes and labels content, leveraging models and expert validation to assess disinformation across languages.
- **Visual-text Misalignment Detection*:** analyses the visual content of a given image or video and the associated textual description, and outputs a score indicating the level of contextual misalignment between them (the higher the score the greater the misalignment between the image/video and the associated text); based on this score, it also provides a predicted label: 0 if the items are aligned and 1 if the items are misaligned;

The following subsections provide more details about the integrated AI-Driven Data Analysis Methods in AI4TRUST Platform v2, describing the newly added ones and reporting on the conducted extensions and improvements on methods that were part of AI4TRUST Platform v1.

3.1.1. Deepfake Video Detection

The Deepfake Video Detection tool has been updated after: i) replacing the previously used visual-based model with an advanced model, and ii) integrating a new model that detects and temporally localises deepfakes using multimodal signals. Specifically, after a comprehensive comparative analysis involving several state-of-the-art methods and 6 benchmarking datasets, we identified the most effective unimodal (visual-only) deepfake detector that has been then deployed through CERTH's deepfake service and is now available through AI4TRUST Platform v2. In addition, we have developed a novel methodology for multimodal deepfake detection that leverages inconsistency between visual and audio extracted speech features, which is also deployed through CERTH's deepfake service and is available through AI4TRUST Platform v2.

3.1.2. Reverse Video Search

The tool for Reverse Video Search on the Web has been updated to tackle one of the main user requirements from the first pilot that relates to its time efficiency. In particular, to reduce the time needed for processing the video we replaced the executable file for video segmentation (which was the main source of errors in the first pilot testing) with a faster python-based implementation. Moreover, we preloaded and ran the different models (i.e., the TransNetV2 for video shot segmentation, the Google Net for feature extraction, the VGG16 for aesthetic quality assessment and the RL-DiVTS for thumbnail selection) on GPU. To decrease the time needed for retrieving the search results (i.e., the near-duplicate videos and the web sources with visually-similar content), we restricted the number of used thumbnails for search (8 instead of 10) and the number of retrieved results (8 instead of 10). In this way, we resulted in a smaller number of searches and of subsequent requests for downloading the thumbnail of the retrieved videos and web sources (64 instead of 100) that was proven to be quite time-demanding, without sacrificing the retrieval performance. Furthermore, we applied asynchronous downloading of the thumbnails of the retrieved videos and web sources, enabling the simultaneous downloading of multiple thumbnails

(instead of doing this in a serial manner). The applied updates resolved a main source of errors during the analysis and resulted in a significant reduction of the overall time needed for providing the user with the analysis results for a given video. More specifically, the processing time in the first version of the tool was approximately equal to 3.5 times the video duration (average time for a set of test videos with various durations). The new version of the tool allows faster than real-time analysis as it needs approx. 0.7 times the video duration, thus reducing the waiting time for the user by approx. 5 times.

3.1.3. Video Anomaly Detection

The Video Anomaly Detection tool is a new service that feeds the Disinformation Warning System. This tool provides endpoints to determine whether a video is normal or anomalous. Anomalies may include visual artifacts, objects appearing in unexpected contexts, or events occurring in environments where they would not typically be observed. Unlike traditional approaches, the tool leverages a training-free model, meaning it is not trained on a predefined set of anomalous classes. This allows it to generalise effectively to various anomalies, including previously unseen or unexpected ones.

3.1.4. Speech to Text

The Speech to Text Transcription tool has been enhanced in the platform's second release to include support for German, Italian, and Greek. As a result, the service now covers all eight languages used by the consortium: en, fr, de, it, pl, ro, es, and el. Additionally, most transcription models have been upgraded. Conformer-based models have replaced the older HMM-TDNN hybrid models, with one exception—Greek still uses HMM-TDNN models. From an implementation perspective, the data ingestion utility responsible for fetching multimedia content from social media has been updated. We have replaced *youtube-dl*[17] with *yt-dlp*[18], a more frequently updated tool that helps prevent download errors caused by API changes on social media platforms. Finally, the speech diarisation tool, which identifies and separates different speakers before transcribing their speech, has also been improved. The updated system now employs a noise-robust hybrid diarisation model consisting of: (i) a fine-tuned Pyannote segmentation model for voice activity detection and overlapped speech detection, (ii) a Titanet-L-based feature extractor for multi-scale embeddings, and (iii) Normalised Maximum Eigengap (NME) Spectral Clustering for speaker grouping.

3.1.5. Checkworthy Claim Detection

The Checkworthy Claim Detection tool has been updated in the second release of the platform to also include Spanish in addition to Italian and English among the supported languages. Extensive

[17] https://github.com/ytdl-org/youtube-dl
[18] https://github.com/yt-dlp/yt-dlp

experiments have been performed to identify the best models to be used in the AI4TRUST Platform. Moreover, the exposed API for check-worthy claim detection has been further extended to work on long inputs, to accept inputs in English, Italian, and Spanish, and to label check-worthy segments within long input texts.

3.1.6. Disinformation Signals Detection

The Disinformation Signals Detection tool has been expanded from 3 signals to a total of 41 signals, retaining the original 3 signals from the AI4TRUST Platform v1. An open-source, state-of-the-art LLM, specifically Llama 3.1 with 8 billion parameters, has been utilised with 7 prompts to detect and annotate text segments that display 4 signs of common disinformation, as well as signals that reveal the use of the 6 most prevalent disinformation tactics: conspiracy theories, trolling, discrediting, pseudoscience, science denialism, and polarisation. The signals corresponding to each tactic were grouped into specific prompts. Additionally, 2 prompts were created to assess whether the LLM is more effective than the fine-tuned models from AI4TRUST Platform v1 in detecting both hate speech and offensive language (combined in one prompt) as well as clickbait. For comparison, the fine-tuned models output labels in all capital letters, while the LLM uses initial capital letters only. In total, 9 prompts were developed for the LLM, hence 9 LLM disinformation signals classifiers for detecting 38 disinformation signals in total.

3.1.7. Verdict Generator

The Verdict Generator tool enables the automated generation of concise fact-checking verdicts, supporting fact-checkers in evaluating claims across multiple languages. Compared to the previous version, which supported only English, the new version of the tool introduces a multilingual framework covering eight languages (English, German, Greek, Spanish, French, Italian, Polish, and Romanian), along with API key authentication for secure access. Accessible via a REST API, the tool takes as input a claim and one or more fact-checking articles and produces a short verdict (up to three sentences) explaining the claim's veracity. It is powered by LLaMA 3.1 8B, fine-tuned on the EuroVerdict dataset, specifically curated for multilingual verdict generation. To ensure language accuracy, it uses an "English prompt with language instruction" strategy (e.g., "Respond in Italian"). The API accepts the parameters claim, article, lang, and sentences, and returns top_sentences and the generated verdict, enhancing the AI4TRUST Toolbox with real-time multilingual fact-checking.

3.1.8. Fact-checked Claim Retrieval

The Fact-checked Claim Retrieval tool has been implemented for the AI4TRUST Platform v2. It supports both input and output claims in English, Spanish, French, Italian, and German and it is based on recent multilingual text embedding models. Extensive experiments have been conducted to select the most promising method. The API for fact-checked claim retrieval returns a rich output consisting of the top-10 most similar claims in any of the supported languages to the one expressed

in the input. They are provided in a decreasing order by similarity score, and each of them has attached information including the score, the original source, the URL of the fact-check, and its language.

3.1.9. Domain Disinformation Detection

The Domain Disinformation Detection tool is a new service that provides an endpoint to classify a domain as Present (1) or Not Present (0) by extracting it from a submitted URL. It uses indexing, machine learning models, and expert validation to assess disinformation across multiple languages. The service exclusively analyses open web URLs and does not process social media links. Domains with a history of disinformation are more likely to host new URLs that also contain disinformation. To access the API, users:

1. Request credentials, specifying their desired username.
2. Receive login details via a 1Password link upon approval.
3. Authenticate with /login using their username and password to obtain an access token.

The API returns a JSON object with the classification result (1/0) and the submitted URL.

3.1.10. Visual-Text Misalignment Detection

The tool for Visual-Text Misalignment Detection is a new API. It integrates an AI-model that consists of a visual and a text encoder and was trained using a triplet loss function (presented in Section 5.3 of D3.1). The visual encoder employs a vision transformer (ViT) architecture to extract high-level visual features from the input visual data. The text encoder utilises a transformer-based architecture to encode textual descriptions into embedding vectors. Both encoders are pre-trained on the CLIP's objective to align images with their corresponding captions, thus allowing the creation of a joint multimodal embedding space. The triplet loss function is designed to train the network architecture by minimising the distance between the images (anchor) and the corresponding real captions while maximising the distance between real and misleading captions. Training was based on a dataset containing approximately 2 Million image-text pairs (called VisualNewsDC in D3.1) that was created by using the Phi-21 LLM to generate misaligned (and thus misleading) versions of real captions accompanying the images of the VisualNews dataset. This tool can process large amounts of image/video-text pairs (such as the ones that are automatically collected by the AI4TRUST platform) and provide the DWS with evidence about the check-worthiness of each corresponding media item.

3.1.11. Sensational Content Detection

The tool for Sensational Content Detection is a new API. It integrates knowledge about (>170) sensational actions/events that appear in disinformation items (specified by the fact-checking

partners of AI4TRUST), using natural language descriptions of them. So, in terms of functionality, it significantly extends the detection capacity of our previous tool that was focusing on the detection of visually-disturbing content (reported in Section 4.4 of D3.1), to support the detection of several types of sensational content. For a given image or video, this tool returns the most relevant sensational action/event from a predefined list of sensational actions/events (typically found in disinformation items), along with a sensational score that represents the similarity between the identified action/event and the input image/video, with higher scores indicating greater similarity. This tool is able to analyse large volumes of visual data (such as the ones that are automatically collected by the AI4TRUST platform) and provide the DWS with evidence about the check-worthiness of each image/video.

3.2. Disinformation Warning System

Before detailing the infrastructure of the AI4TRUST Platform v2, a summary of all its components has been provided. Among these, the Disinformation Warning System (DWS) is a key service designed to flag potentially misleading and check-worthy content. It processes structured payloads containing precomputed feature scores from multiple detection models and employs a probability-based ensemble classifier to generate a risk score.

Submissions are ingested adhering to a predefined schema with content type (YouTube, Telegram, News), metadata fields (language, summary, title, and URL), raw textual content, media URLs, and feature scores extracted from models. DWS aggregates the output of the following AI-based classification methods (see Figure 29):

- **Check-worthy Claim Detection, Video Anomaly Detection, and Visual-Text Misalignment Detection** produce binary classification outputs (0/1) with confidence scores [0,1].
- **Sensational Content Detection** produces one or more categorical labels with associated confidence scores [0,1].
- **Disinformation Signals Detection** produces multiple disinformation signals with confidence scores.
- **Domain Disinformation Detection** produces binary classification outputs (0/1).

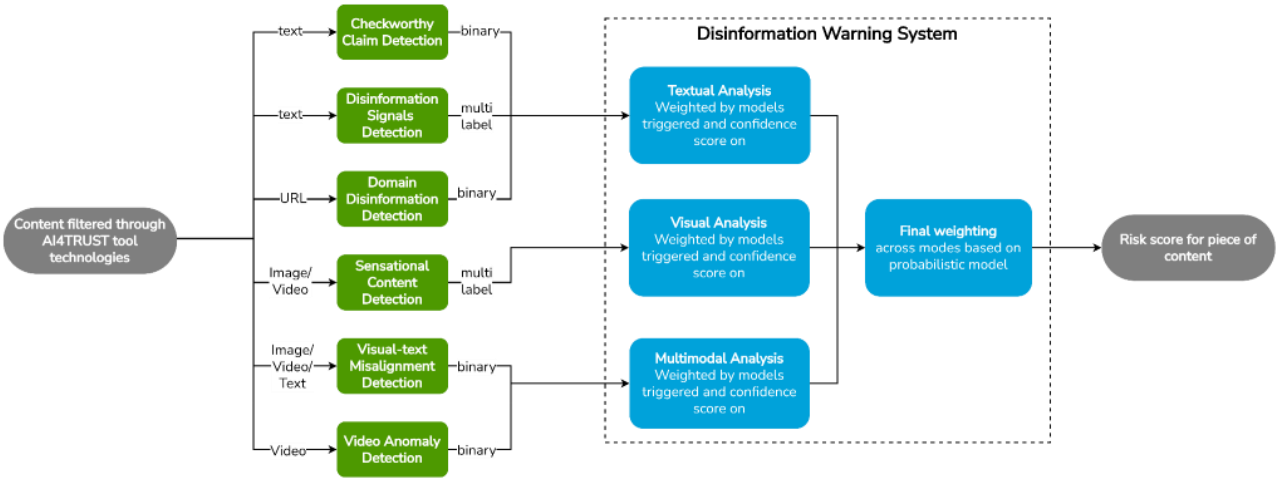

***Figure 29**: Disinformation Warning System Architecture from D3.1*

Processing is asynchronous, and submission status can be retrieved, providing information about the processing states such as pending, in progress, or completed. Once processing is finalised, it returns the aggregated risk score [0,1], the confidence score [0,1], and feature attribution values, which quantify the relative influence of each detection model on the final decision enabling explainability.

DWS leverages a **probabilistic ensemble classifier** that integrates feature scores using weighted contributions optimised through supervised learning on expert-labelled datasets. This model is trained on multimodal datasets, where multimedia content has been manually annotated with ground-truth labels. This enables the model to learn optimal weighting strategies across diverse detection tools, improving classification accuracy. This architecture enables external models (whether high or low latency) to generate structured feature representations, which DWS then processes to ensure low-latency risk assessment.

3.3. Data Platform

The Data Platform is a key component in the AI4TRUST infrastructure. It is designed to address the requirements of data collection, processing and management in a scalable, consistent and compliant manner. The data platform is logically divided into 3 macro blocks, as depicted in Figure 30:

- The **Data Lakehouse**, which acts as heterogeneous data repository;
- The **Streaming Platform** as (pre)processing and transport layer;
- The **Serverless Platform** as main processing layer for data collectors and transformations.

At the core, the Data Platform is based on a **modern data lake-house design**, where a traditional flat data lake is augmented with modern and powerful capabilities aimed at managing data schemas, tables, transactions and data versioning, all in a highly integrated environment. The data

layer is completed by the adoption of a powerful and easy-to-use query engine, which enables both data scientists and operators to quickly discover, explore, catalogue and interact with the data collected from external sources.

At the front of the Data Platform, we encounter the **Data Ingestion Layer**, which is built on a flexible **Serverless Platform** that handles the lifecycle of data collectors, offering modern operational capabilities such as deployment management, horizontal scaling, routing, ingress and egress networking and continuous deployment.

The data collected by the Ingestion Layer is pushed inside the **Streaming Platform**, which constitutes the core of the preprocessing layer. The tasks of receiving, transforming and adapting raw data to properly modelled features are achieved through a series of transformations deployed inside a platform that handles all the routing, persistence, delivery and scaling requirements in a transparent manner. Eventually, all the processed data assets persisted as proper data products inside the lakehouse, by a series of sink processors that act as terminators for the data flow.

While the design and implementation of the Data Platform is geared towards proper data management, the Serverless Platform is fully capable of executing custom data processors for transformations, classification, topic extraction, text analysis etc. All the business logic relevant for the collection pipelines is implemented as serverless functions, connected to the Streaming Platform, which handles data in a streaming fashion, ensuring high performance, scalability and resilience in the data processing. The usage of queues and the consistency of consuming marks, guaranteed by the Streaming Platform, lets developers implement processors focusing on the business logic, leaving all the consistency and scalability issues to the underlying systems.

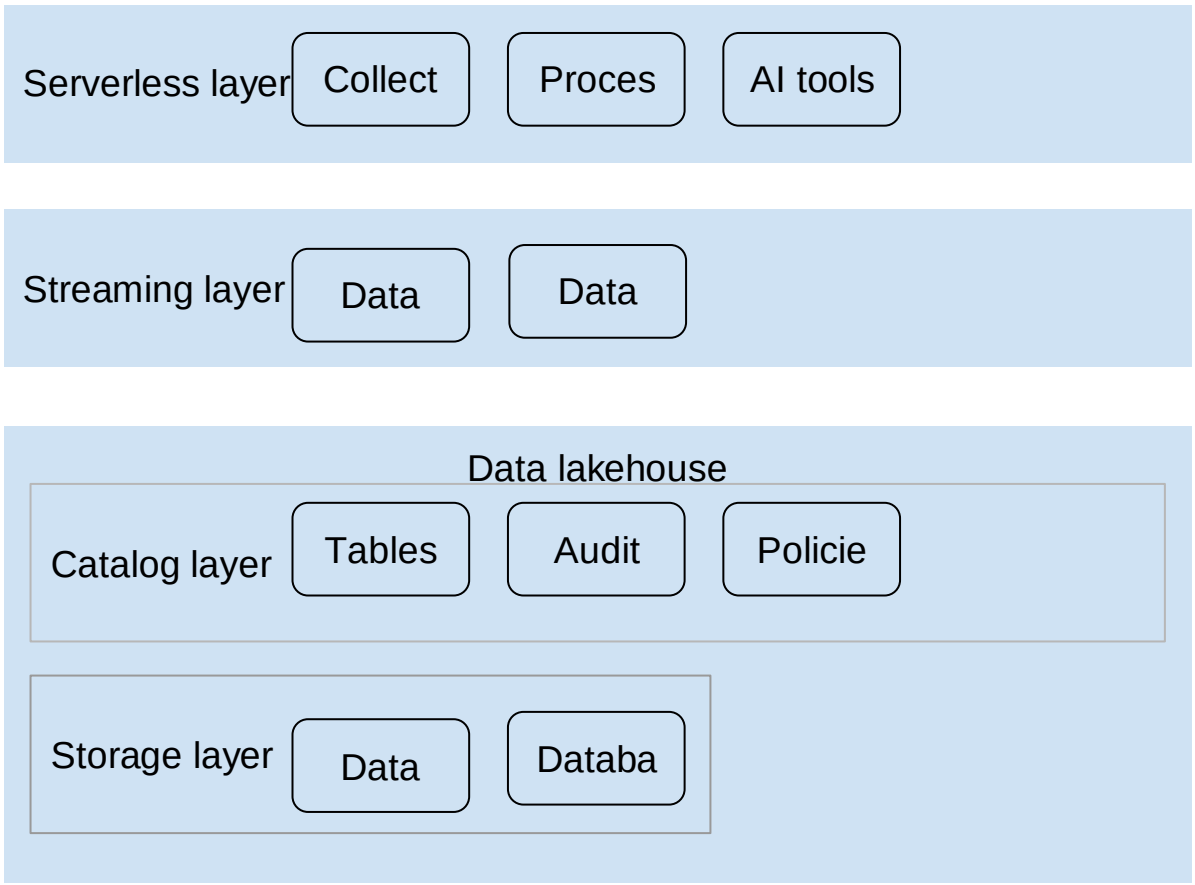

Figure 30: *Data platform components*

3.4. Data Collector

Data collectors are the entry-point of the data flow: they ingest content from remote sources such as websites or social media and push the results into the streaming platform. One of the key aspects of the AI4TRUST Platform is the ability to monitor content in an automated manner, by scanning relevant and/or interesting data sources and then automatically ingest them for research and analysis.

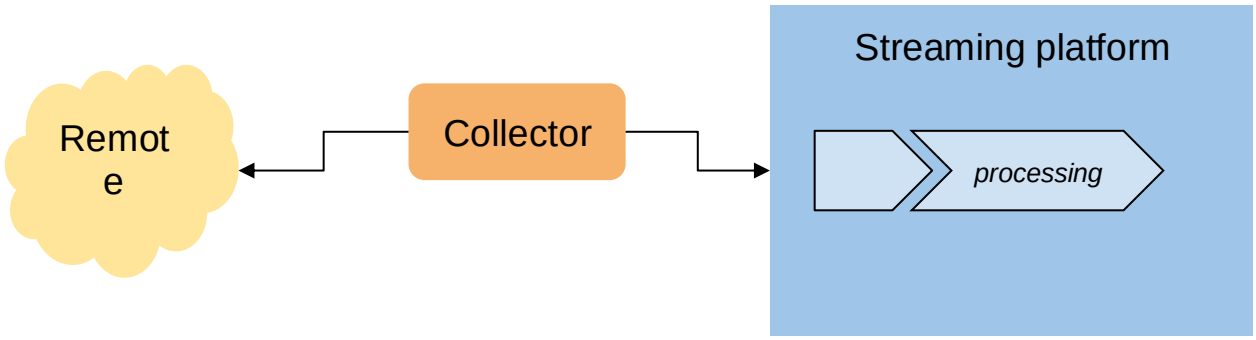

Figure 31: *Data collection flow*

Collectors are the **starting point of the data ingestion process**: they monitor external (remote) sources by implementing either a polling, scraping or consuming logic in order to discover new data to be processed. They are purposely built components that can handle various types of data such as text, video, images and audio, supporting all the languages selected for the project.

Furthermore, as ingress points in the pipeline, they solve not only the collection of entries, but also the identification of content and their destination, along with details about the discovery process such as relevant keywords, timestamp, sources etc. All these metadata are collected along with the actual data content, stored and then preserved through the entire processing to ensure proper lineage of results is kept. For every piece of content collected and persisted, the platform automatically adds and tracks:

- **A global addressable identifier**, which uniquely identifies a single content along with its source and lineage.
- **The data owner**, i.e., the partner in charge of the content and its handling.
- **The creation data**, automatically set to the instant the content entered the data platform.
- **The relevant language**, either extracted from the source or inferred from the content itself.
- **The keywords and topic(s) related to the content**, in case the source was an API search.
- **The type of the content** (text, video, image, audio, multimodal).

All data, raw or processed, is aligned with the metadata described above in order to guarantee proper ownership and management of the datasets. Currently, the platform hosts collectors for:

- **Online news data**
- **YouTube data**
- **Telegram data**

See Section 3.7 for details about the data collection.

3.5. Streaming Platform

The Streaming Platform is a core component of the Data Platform: it solves the problems of storing, transporting and dispatching data in a streaming fashion, by connecting data producers, such as collectors, with consumers, such as processors and AI tools. The platform is built on top of the following components:

- **Apache Kafka[19];**
- **Kafka Connect.**

Producers write (raw/processed) data on topics, used as data queues, with a proper content, as defined by data schemas, and proper addressing metadata, such as source, ownership and ids. Consumers read (one-by-one or via micro-batching) from the various queues and then process the content to:

[19] https://kafka.apache.org/

- **Transform values or structure** (for example to apply anonymisation);
- **Derive new data** (for example to produce a score, or download assets such as thumbnails);
- **Persist the content** (for example to keep a relevance score for channels).

The sequence of producers, topics and consumers depicts a data pipeline represented as a DAG, which is used to perform a series of operations on data in an ordered and reproducible manner. At the very end of every branch of the graph, a sink collects all the content and properly persists it to the Data Lakehouse, ensuring that content is both persisted in the storage layer and registered in the data catalogue.

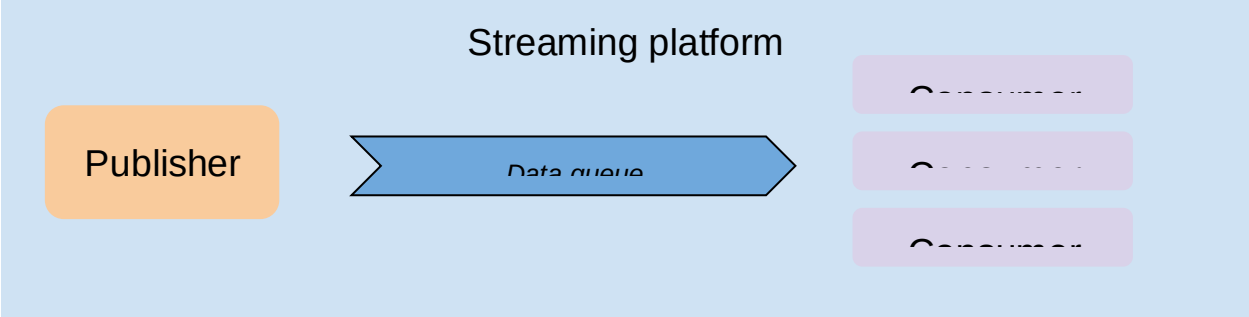

Figure 32: *Pub/sub data flow*

3.6. Serverless Platform

The Serverless Platform represents the compute part of the Data Platform, where processors handling data collection, transformation and production can execute. The Data Platform leverages a serverless layer as the execution platform to offer developers and researchers the ability to write their business logic to be executed in the data pipelines without addressing all the operational requirements, such as credentials handling, deployment management, scaling, routing and networking (Figure 33). The components used are:

- **Kubernetes**[20], as the compute platform;
- **Nuclio**[21], as the serverless platform.

In order to provide an easy to use, extensible and versatile platform we choose to support only **Python** as programming language, ensuring that all processors deployed in the Serverless Platform are built with the same software stack. This reduces the complexity of the data pipelines and reduces the operational overhead versus supporting multiple programming languages and stacks.

Another important feature driving the adoption of the serverless platform in AI4TRUST is the **push towards reproducible and consistent data collection**, which is a fundamental block in building a reliable and trustable data platform for research. By adopting the functional paradigm, we push for reproducible, idempotent processors to be built, which ensure transparency, reproducibility and

[20] https://kubernetes.io/
[21] https://nuclio.io/

explainability in the data handling process. Nevertheless, some transformations are inherently disruptive or irreproducible: for example, one-way anonymisation or random scoring produce *by design* different results at every iteration, even with the same input.

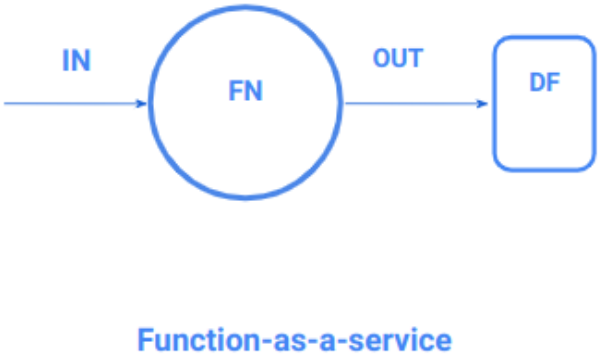

Figure 42: *Data processor as serverless function (FN) producing a data frame (DF) as output*

3.7. Data Lakehouse

The Data Lakehouse is a modern data architecture that combines the best features of both data lakes and data warehouses into a single, unified solution. Its main advantages are:

- **Flexibility**: the ability to store structured, semi-structured and unstructured data together;
- **Reliability**: datasets are kept consistent even with multiple writes;
- **Scalability**: the lakehouse can scale to multiple Terabytes with ease;
- **Performance**: data is stored compressed, in a highly optimised format that ensures fast lookups with billions of data points in a single table.

We built our data platform leveraging:

- **Minio**[22], as the open-source object store;
- **Apache Parquet**[23] **and Apache Iceberg**[24], as data and table format;
- **Project Nessie**[25], as data catalogue;
- **Trino**[26], as the query engine;
- **PostgreSQL**[27], as operational storage.

Every piece of data stored in the platform is globally identifiable and addressable, and we actively enforce the presence of **metadata** about ownership and tracking: content that lacks proper

[22] https://min.io/
[23] https://parquet.apache.org/
[24] https://iceberg.apache.org/
[25] https://projectnessie.org/
[26] https://trino.io/
[27] https://www.postgresql.org/

information will not be stored in the lakehouse. By keeping a fast, low latency, structured database bound to the object storage we can support complex operations, which require persistent states, to be performed in the data processing stages. For example, the YouTube processor can keep track of channels already visited and adjust a global score for relevance, keeping track of changes over time without resorting to a slower lakehouse lookup. Nevertheless, all data is captured and stored long term inside the lakehouse in an individual Iceberg table. Lastly, the **Data Lakehouse** manages **permissions and policies for data storage and access**, according to relevant guidelines and regulations, to ensure that only properly authorised operators can read, write and consume datasets according to their individual access rights.

3.8. Data Pipelines

The Data Platform manages various pipelines, which are batch processing Directed Acyclic Graphs (DAGs) dedicated to the ingestion, collection and processing of content from a given data source. We have designed the data collection to be producer oriented, where a shared set of components are used to implement the Extract-Transform-Load (ETL) process respecting the peculiarities of the external source. **The pipelines leverage the various components** as depicted in Figure 35.

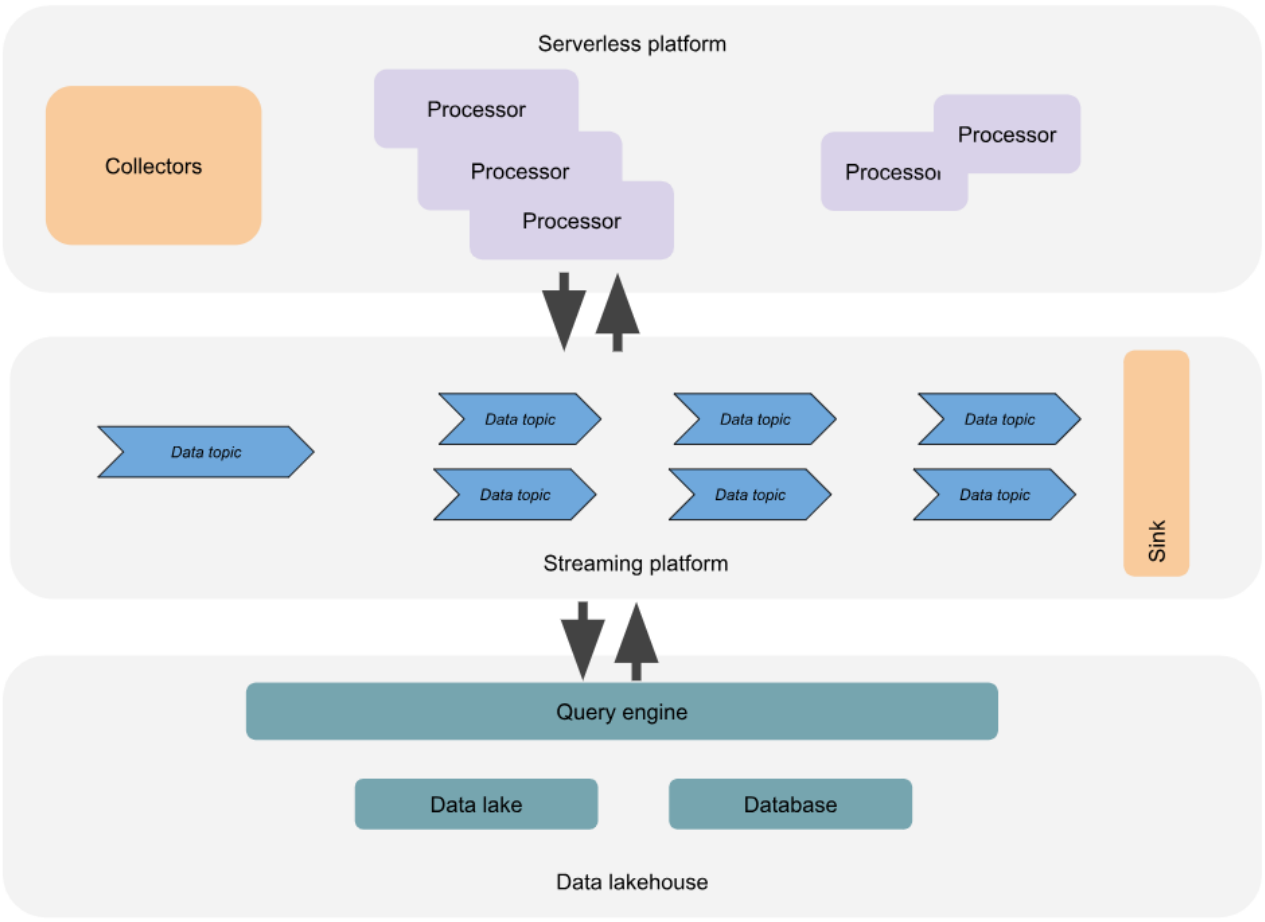

Figure 35: *Data platform components*

3.8.1. News Pipeline

The News Collector actively monitors the articles published by source outlets worldwide by leveraging the GDELT Project[28] for the individuation of the relevant content, and then by actively scraping the source website via a custom crawler. The process starts from a list of keywords of interest, grouped by language and topics, which are used as a source for interrogating the GDELT API. The results are then evaluated, by checking the source and evaluating their policy for scraping. When the check is positive, the content is scraped and then sent to the processing pipeline within the streaming platform.

[28] https://www.gdeltproject.org/

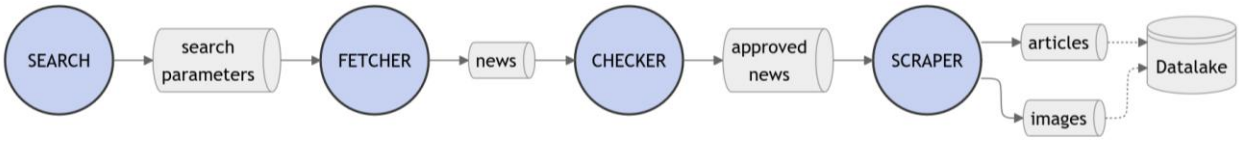

Figure 36: *News pipeline*

3.8.2. YouTube Pipeline

The YouTube Collector monitors videos published on YouTube for relevant content. It leverages a similar set of keywords and topics as the news collector to extract videos of interest from the YouTube API and then proceeds to forward the information to the processing pipeline that downloads all the metadata and the content (when allowed).

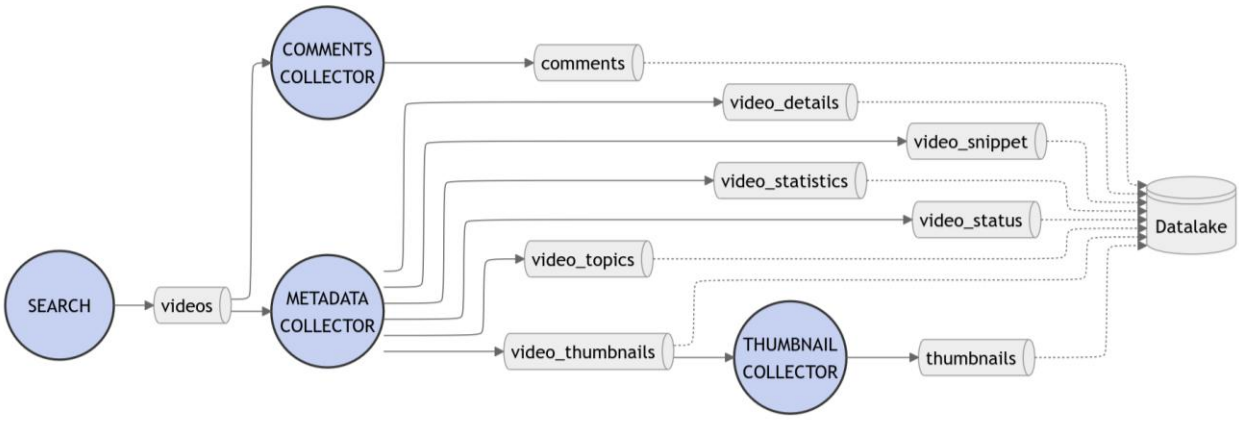

Figure 37: *YouTube pipeline*

3.8.3. Telegram Pipeline

The Telegram Collector is structured into two different stages, due to the lack of an official API able to select content based on keywords. As such, we implemented a split approach, where we start from keywords search to discover channels, and then crawl the social network by exploring neighbours and relatives' relationships between channels. At a second stage, we explore channels and collect messages, by requesting the chat history and ingesting all the content.

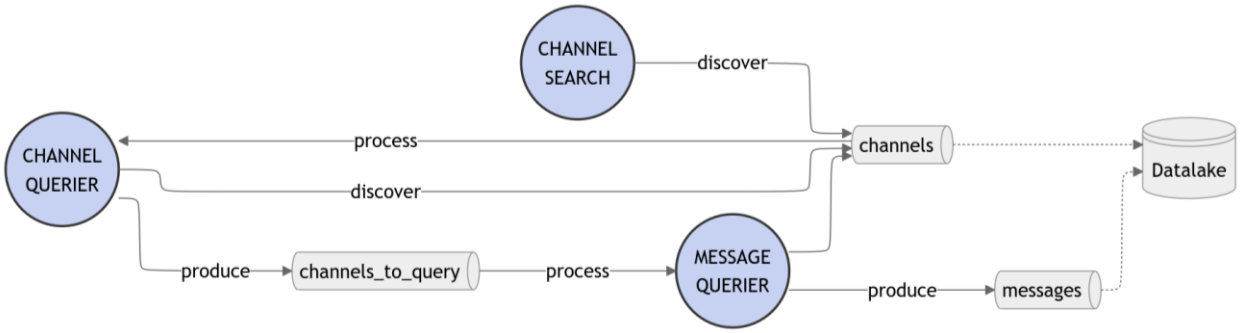

Figure 38: *Telegram pipeline*

3.9. Web Application

The Web Application is the primary interaction point for users of the AI4TRUST Platform, ensuring a seamless and intuitive experience when accessing its suite of AI-powered features. With the latest version, significant improvements have been introduced to enhance usability, clarity, and efficiency, incorporating valuable insights gathered from piloting sessions.

The AI4TRUST web application is based on React[29], a JavaScript library that allows building dynamic and reusable UI components. It is build using Vite[30], which significantly speeds up local development and reduces build times, ensuring a fast and efficient front-end development experience. The project's package management is handled by Yarn[31], which helps mitigate dependency-related issues and ensures consistency across different environments. The AI4TRUST web application uses Zustand[32] for efficient state management, providing a simple and scalable solution for centralised state access without prop drilling, enhancing flexibility and performance. For styling, the project uses Tailwind CSS[33], which implies a utility-first approach to styling, making it easy to apply responsive designs without writing custom CSS. To enhance the user interface, DaisyUI[34] is used, which extends Tailwind by providing pre-built, design-ready components for a more user-friendly and visually appealing experience.

[29] https://react.dev/
[30] https://vite.dev/
[31] https://yarnpkg.com/
[32] https://github.com/pmndrs/zustand
[33] https://v3.tailwindcss.com/
[34] https://v4.daisyui.com/

The introduction of the **Monitoring Dashboard** involved complex filtering and user interactions, which is efficiently managed using React Select[35], a library that simplifies the implementation of customisable dropdowns and multi-select components. This improved the user experience by providing an intuitive way to filter and select options dynamically. For additional information regarding the Monitoring functionality, please refer to Section 2.2. These enhancements collectively improve the user experience, making the AI4TRUST Platform more **efficient, accessible, and aligned** with real-world mis/disinformation detection and fact-checking needs.

A visual representation of the website map is provided in Figure 39, illustrating **how users navigate through the platform's interface and how each page is interconnected**. The grey sections denote sections of the user interface that are accessible without authentication (i.e., the homepage, the login page), while the coloured sections indicate areas requiring user credentials (i.e., the Toolbox and the Monitoring and Human Validation Dashboard). The green colour corresponds to the Toolbox dashboard for the AI-driven data analysis methods functionality. The light blue colour represents the Monitoring and Human Validation Dashboard for the automated data collection and content analysis functionality. The dark blue colour signifies informative content, such as the "About" section of the platform. The **routing system** in the AI4TRUST web application is managed using React Router V6[36], which enables seamless navigation between different pages and components.

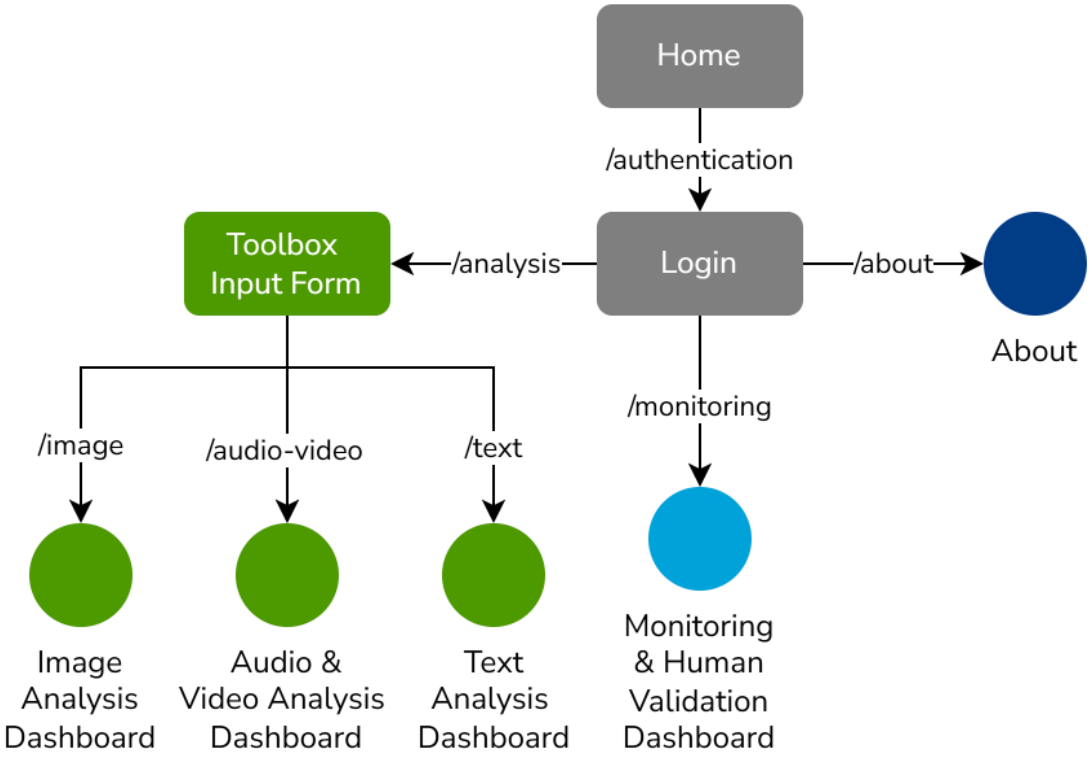

***Figure 39**: Homepage*

[35] https://react-select.com/home
[36] https://reactrouter.com/6.30.0

The user first arrives at the **homepage**, accessible via the / path, which provides a brief welcome message along with an overview of the platform's purpose. This page serves as the entry point, offering users a clear understanding of the platform's objectives and functionalities. To access personalised features, the user can click on the **Login** button, which redirects them to the authentication page. Here, they can **securely log in to their account**, gaining access to the platform's full suite of AI-driven tools. For additional information about the authentication phase, please refer to Section 3.13. A visual representation of these pages is provided in Figure 40.

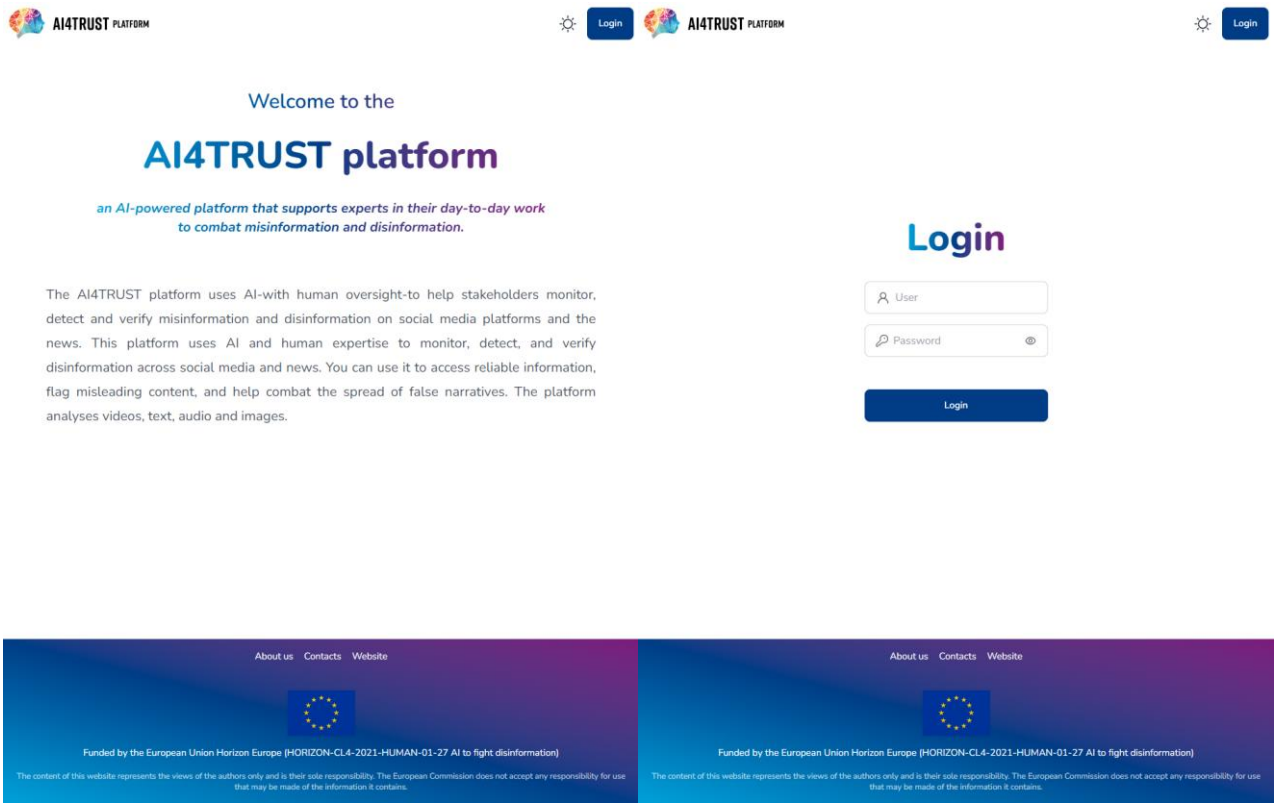

***Figure 40**: Homepage and Login page*

The **authentication process** takes place on the */authentication* path, where users enter their username and password to log in securely. Upon successful authentication, they gain access to the Toolbox input form, the central hub for submitting content for analysis. Once logged, access is granted also to the Monitoring and Human Validation Dashboard via */monitoring* and the About page via */about*.

The **Web Application development project** continues to follow the "grouping by file types and features" organisational approach. This structure, which categorises files based on their types (e.g., components, utilities, assets) and their respective functionalities, has been useful to maintain a well-structured codebase. For the AI4TRUST Platform v2, no changes were made to the core implementation structure, as it has consistently facilitated code maintainability, modular development, and efficient collaboration.

By keeping this approach, the project retains its **clarity and organisation**, ensuring that new features integrate seamlessly without compromising the existing architecture. This consistency

also helps streamline **further enhancements**, allowing the modification of relevant parts of the code while preserving **overall stability**.

3.10. API Gateway

In AI4TRUST Platform v1, the back-end architecture included only an **API Gateway**. With the introduction of AI4TRUST Platform v2, **new back-end components have been added**, and the API Gateway's role has been explicitly defined. It now not only manages the APIs of the WP3 services, handling authentication, input/output mapping, and error harmonisation, but also serves as an interface for the Database supporting the Web Application. This enhancement facilitates API interactions, reducing the complexity for clients while ensuring consistency across services.

Built with Spring Boot, the API Gateway provides a lightweight yet robust solution for API traffic management. It utilises **Open API** to standardise API definitions, providing seamless integration between the front-end and WP3 services while maintaining compatibility with automatic code generation tools. This architectural evolution enhances the platform's modularity and efficiency while preserving backward compatibility. For further details on its architecture, implementation, and security measures, please refer to D5.5[37].

3.10.1. Implementation

The API Gateway has been extended to include a database interfacing layer, allowing it to handle interactions with the **Database**. This integration is implemented using **JDBC**[38] for direct database connectivity, ensuring efficient query execution and optimised performance. Additionally, **Spring Data JPA**[39] has been defined for each table, streamlining data access by providing a high-level abstraction over database operations. This allows seamless integration between the API Gateway and the database while reducing boilerplate code. For more complex queries or performance-critical operations, **JDBC** is used directly. This approach enables fine-grained control over SQL execution, ensuring optimal efficiency when handling large datasets or executing custom queries.

3.10.2. Open API Specifications

The AI4TRUST Platform v2 introduces **a series of new tools** and in the following sections are reported the related new endpoints. For the details about the previously implemented endpoints, please refer to D5.5[40].

[37] D5.5 - AI4TRUST Platform v1 (https://ai4trust.eu/public-deliverables/)
[38] https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/
[39] https://spring.io/projects/spring-data-jpa
[40] D5.5 - AI4TRUST Platform v1 (https://ai4trust.eu/public-deliverables/)

3.10.2.1. Platform

- GET /formats/audio

  This endpoint returns the list of compatible audio file formats of the AI4TRUST platform

- GET /formats/video

  This endpoint returns the list of compatible video file formats of the AI4TRUST platform

3.10.2.2. Video

- POST /video/anomaly-detection

  The video anomaly detection endpoint takes a **video URL** as input (through a request body), and forwards it to the Video Anomaly Detection service that starts a video anomaly detection job. To keep track of the status of the job and to retrieve the finished result, the Video Anomaly Detection service returns a **request id**, that it is forwarded back.

- GET /video/anomaly-detection/status/{req_id}

  The video anomaly detection status end-point takes a **request id** as input (through path) and returns the status of the request. The status can be "PROCESSING" if the job is still running, "COMPLETED" otherwise.

- GET /video/anomaly-detection/report/{req_id}

  The video anomaly detection report end-point takes a **request id** as input (through path) and returns the video anomaly detection results.

3.10.2.3. Text

POST /text/previously-fact-checked-claim-retrieval

The previously fact checked claim retrieval endpoint takes a **text** and a **language** as input (through a request body), and returns a fact checked claim retrieval result

3.10.2.4. Multimodal

- POST /multimodal/sensational-content-detection

  The sensational content detection endpoint takes a **video URL** as input (through a request body), and forwards it to the Sensational Content Detection service that starts a sensational content detection job. To keep track of the status of the job and to retrieve the finished result, it returns a **request id**.

- GET /multimodal/sensational-content-detection/status/{req_id}

  The sensational content detection status end-point takes a **request id** as input (through path) and returns the status of the request. The status can be "PROCESSING" if the job is still running, "COMPLETED" otherwise.

- GET /multimodal/sensational-content-detection/report/{req_id}

  The sensational content detection report end-point takes a **request id** as input (through path) and returns the sensational content detection results.

- POST /multimodal/visual-text-misalignment-detection

  The visual text misalignment detection endpoint takes a **video URL** as input (through a request body), and forwards it to the Visual-Text Misalignment Detection service that starts a visual text misalignment detection job. To keep track of the status of the job and to retrieve the finished result, it returns a **request id**.

- GET /multimodal/visual-text-misalignment-detection/status/{req_id}

  The visual text misalignment detection status end-point takes a **request id** as input (through path) and returns the status of the request. The status can be "PROCESSING" if the job is still running, "COMPLETED" otherwise.

- GET /multimodal/visual-text-misalignment-detection/report/{req_id}

  The visual text misalignment detection report end-point takes a **request id** as input (through path) and returns the visual text misalignment detection results.

- POST /multimodal/disinformation-warning-system

  The disinformation warning system endpoint takes a **video URL** as input (through a request body), and forwards it to the Disinformation Warning System service that starts a disinformation warning system job. To keep track of the status of the job and to retrieve the finished result, it returns a **request id**.

- GET /multimodal/disinformation-warning-system/status/{req_id}

  The disinformation warning system status end-point takes a **request id** as input (through path) and returns the status of the request. The status can be "PROCESSING" if the job is still running, "COMPLETED" otherwise.

- GET /multimodal/disinformation-warning-system/report/{req_id}

  The disinformation warning system report end-point takes a **request id** as input (through path) and returns the disinformation warning system results.

- POST /multimodal/data-platform

  The data platform endpoint takes a **video URL** as input (through a request body) and returns a Domain Disinformation Detection result.

3.10.2.5. Monitoring

- POST /monitoring/content

The post content endpoint takes a **content object** as input (through a request body) and saves it into the monitoring database. For a detailed reference on the content object structure, please refer to Annex I.

- GET /monitoring/content

  The get monitoring content endpoint retrieves a **list of content items** stored in the platform's database. It accepts multiple optional query parameters for filtering, sorting, and pagination.

- GET /monitoring/content/filters

  The get monitoring content filters endpoint returns all the **available filtering values** for the get monitoring content endpoint

- GET /monitoring/content/{content_id}

  The get monitoring content endpoint takes a **request id** as input (through path) and returns the given content.

- POST /monitoring/dws-output

  The post dws output endpoint takes a **DWS result object** as input (through a request body) and saves it into the monitoring database. For a detailed reference on the dws result object structure, please refer to Annex I.

- GET /monitoring/dws-output/{content_id}

  The get dws output endpoint takes a **request id** as input (through path) and returns the given dws result.

- POST /monitoring/human-validation

  The post human validation endpoint takes a human validation object as input (through a request body), saves it into the monitoring database, and forwards the data to the data platform. This ensures synchronisation of the fact-checking results between the internal Database and the Datalake. For a detailed reference on the human validation object structure, please refer to Annex I.

- GET /monitoring/human-validation/{content_id}

  The get human validation endpoint takes a **request id** as input (through path) and returns the given human validation result.

- POST /monitoring/tool-output

  The post tool output endpoint takes a **tool output object** as input (through a request body) and saves it into the monitoring database. For a detailed reference on the tool output object structure, please refer to Annex I.

- GET /monitoring/tool-output/{content_id}

  The get tool output endpoint takes a **request id** as input (through path) and returns the given tool result.

- POST /monitoring/claim-review-extractor

  The Claim Review Extractor endpoint accepts a **fact-checking article URL** in the request body and extracts the ClaimReview markup. For additional information, please refer to Section 2.2.

3.11. Dispatcher

This section provides a detailed explanation of the implementation and purpose for **the Dispatcher component**. The Dispatcher is responsible for processing incoming Kafka messages from the Streaming Platform and handling the routing of the data for the automatic processing across different modules.

3.11.1. Implementation

The Dispatcher is built using Spring Boot[41], a robust Java-based framework, and Maven[42] for project management, ensuring maintainability and dependency management. It follows the Open API specification, leveraging Open API Generator[43] to standardise API definitions and minimise manual implementation errors. The Dispatcher follows a structured architecture to promote maintainability. Its implementation is organised into **three primary layers**:

- **Model Layer**: Defines the data structures required for processing messages and validation data, with some structures being auto-generated based on Open API specifications.
- **Controller Layer**: Manages incoming requests and routes them to the appropriate services for processing.
- **Service Layer**: Implements core business logic, including message transformation, API communication, and Kafka message handling.

The **Dispatcher** acts as a central intermediary, handling a high volume of messages from the **Streaming Platform** via a Kafka topic. Functioning as both a **Kafka Consumer and Producer**, it ingests, processes, and routes messages to various **AI-Driven Data Analysis Methods**.

[41] https://spring.io/projects/spring-boot
[42] https://maven.apache.org/
[43] https://github.com/OpenAPITools/openapi-generator

Within the **data processing pipeline**, the Dispatcher receives structured Kafka messages categorised as **YouTube, News, and Telegram**, orchestrating their analysis across multiple modules. Each incoming message follows a predefined **JSON schema**, ensuring seamless integration with downstream components. Upon ingestion, the Dispatcher processes and forwards these messages to the **AI-Driven Data Analysis Methods** via the **API Gateway**. These methods include:

- **Check-Worthy Claim Detection**
- **Video Anomaly Detection**
- **Visual Text Misalignment**
- **Sensational Content Detection**
- **Disinformation Signals Detection**
- **Domain Disinformation Detection**

Each method produces structured outputs, which are then merged and relayed into the **Disinformation Warning System (DWS)**. The DWS consolidates the results to generate a final assessment of potential disinformation risks.

In detail, in the **YouTube data flow** (see Figure 41), the field *relevance_language* representing the language of *title*, *description* and video itself of the received YouTube Kafka message is passed by the Dispatcher as *language* to multiple AI-Driven Data Analysis Methods, including Check Worthy Claim Detection, Disinformation Signals Detection, and Visual Text Misalignment Detection. The *title* field of the YouTube Kafka message, which serves as a brief but informative summary of the video content, is passed by the Dispatcher as the *short_text* field to Visual Text Misalignment Detection. The *description*, is a more detailed textual representation of the video, is passed by the Dispatcher as *text* to the Disinformation Signals Detection, allowing this tool to analyse the content's credibility and potential disinformation signals. The *video_url*, which provides a direct link to the video, is passed by the Dispatcher to Sensational Content Detection, Video Anomaly Detection, and Visual Text Misalignment Detection as *video_url*, enabling them to assess visual and behavioural anomalies. Similarly, the *image_url*, which corresponds to the caption image of the YouTube video, is passed by the Dispatcher to the Sensational Content Detection as *image_url*, where it is analysed together with the video. The *content_type* field, which identifies the type of message analysed; the *media_url* field, which is mapped from *video_url* referring to the Youtube video's link; the *missing features*, which is a list of AI-Drive Data Analysis methods that were not available for analysis; the *raw_content* containing a sample description of the content the DWS is going to analyse; the *metadata* object, which includes multiple fields such as *data_owner* (set to 'FBK'); *publish_time* (timestamp of video publication) and *keyword* (representing the video's main topic) are sent to the Disinformation Warning System together with the outputs generated by the AI-Driven Data Analysis Methods.

As previously mentioned, the outcomes of the AI-Driven Data Analysis Methods called by the Dispatcher and the DWS result are also sent to the Streaming Platform and to the Database for later access through the Web Application (Monitoring and Human Validation Dashboard).

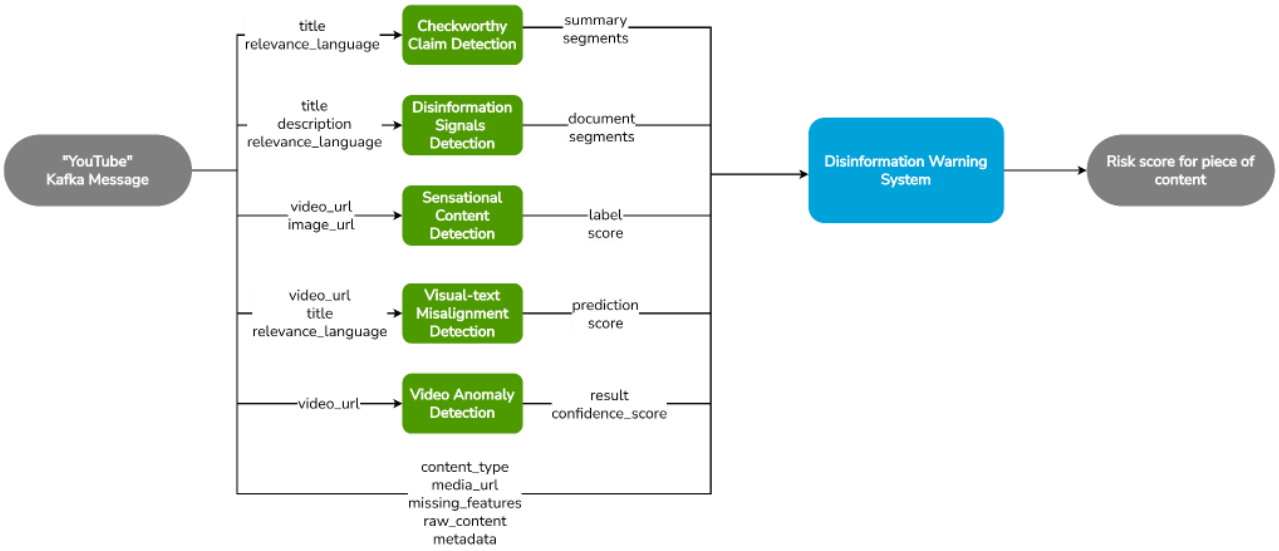

Figure 41: *YouTube data flow*

In the **News data flow** (detailed in Figure 42), the *language* field of the received Kafka message identifies the language of the article's title and text, ensuring proper linguistic processing by tools such as Check Worthy Claim Detection, Disinformation Signals Detection, and Visual Text Misalignment Detection. The *title*, which provides a concise overview of the news article, is used as *short_text* in Visual Text Misalignment Detection and is also a key field in Disinformation Signals Detection and Checkworthy Claim Detection, where it helps evaluate the credibility of the news content. The *text* field, which contains the full content of the news article, is mapped to the *text* input of the Disinformation Signals Detection, supporting in-depth textual analysis. The *url*, which links to the full news article, is crucial for Domain Disinformation Detection and Visual Text Misalignment Detection, facilitating the verification of sources. The *image_url*, which points to the main visual associated with the article, is analysed in Sensational Content Detection. The *content_type* field, which identifies the type of message analysed; the *media_url* field, which is mapped from *video_url* referring to the news article's link; the *missing features*, which is a list of AI-Drive Data Analysis methods that were not available for analysis; the *raw_content* containing a sample description of the content the DWS is going to analyse; the *metadata* object, which includes multiple fields such as *data_owner* (set to 'FBK'); *publish_time* (timestamp of video publication) and *keyword* (representing the video's main topic) are sent to the Disinformation Warning System together with the outputs generated by the AI-Driven Data Analysis Methods.

As previously mentioned, the outcomes of the AI-Driven Data Analysis Methods called by the Dispatcher and the DWS result are also sent to the Streaming Platform and to the Database for later access through the Web Application (Monitoring and Human Validation Dashboard).

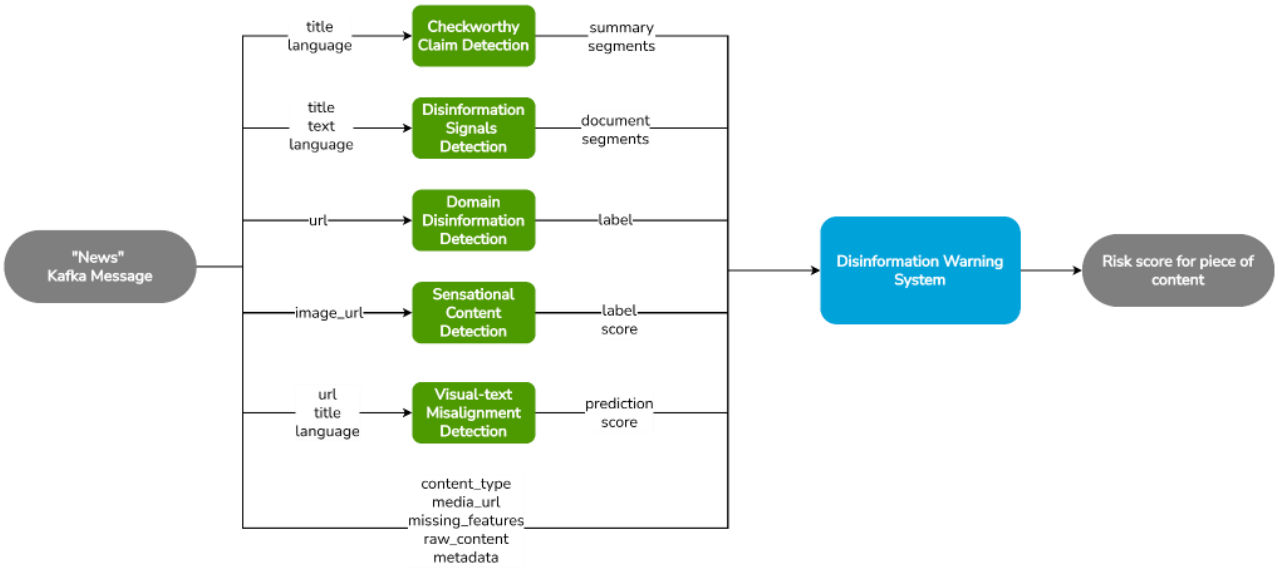

Figure 42: *News data flow*

In the **Telegram data flow**, the received Kafka message contains text that can be analysed by the Check-worthy Claim Detection and the Disinformation Signals Detection. As previously mentioned, the outcomes of the AI-Driven Data Analysis Methods called by the Dispatcher are sent by the Dispatcher to the Streaming Platform, to the Database and to the DWS, which in turn can report to the Dispatcher its result. The Dispatcher can then send the DWS result back to the Streaming Platform and store it in the Database for later access through the Web Application (Monitoring and Human Validation Dashboard).

3.12. Database

To support both the monitoring and human validation functionalities, **persistent data storage** is required. The first version of the platform integrated an off-the-shelf PostgreSQL database, which was configured and optimised to better meet the platform's needs. Initially, the primary use of the database was for authentication and user management via Keycloak[44], an open-source identity and access management solution that provides authentication, authorisation, and user federation services. A more detailed explanation of its usage within the platform is provided in Chapter 3.12.

With the evolution of the platform, **the PostgreSQL database was further extended** to support additional features such as storing and managing platform data, including monitoring and validation results, ensuring a unified and efficient persistence layer for the platform's expanded capabilities. A dedicated **database for Monitoring** was introduced, operating alongside the **Keycloak-generated database**. This separation ensures structured and efficient data persistence

[44] https://www.keycloak.org/

while maintaining a clear distinction between authentication data and monitoring-related information.

For the **Monitoring** and **Human Validation** functionality, three key types of information are stored:

- **Content** retrieved from the data platform pipeline.
- **DWS inputs and outputs** for processing and analysis.
- **Human validation information**, capturing fact-checking assessments.

The relationships between the tables in the Database are designed to maintain a clear structure and ensure consistency across the different types of data stored. For further details please refer to Annex I. The **Content** table acts as the central entity in the system, linking various data types together. Each content item is initially stored in this table, and over time, additional records may be associated with it. The **DWS Output** table maintains a one-to-one relationship with the **Content** table, meaning that for each piece of content, a corresponding DWS output is generated, as long as the processing completes successfully. The **Human Validation** table also has a one-to-one relationship with **Content**, but unlike the DWS output, a human validation record is only created if a user actively chooses to perform validation. This structure ensures that content is the foundation, with other data layers being added as processing and validation occur.

Additionally, the **Content** table has a one-to-many relationship with the **Tool Output** table. This means that a single content item can be processed by multiple WP3 Lite Tools, each generating a separate output. Each record in the **Tool Output** table corresponds to a specific tool and its raw output data for the given content, allowing the platform to capture and process various results from different tools applied to the same content.

For each **DWS Output**, there is a list of **Tool Weights** associated with it, creating a one-to-many relationship between the two tables. These weights represent the contribution of each tool's output to the final DWS score, reflecting how much each individual tool influences the overall result. Each **Tool Weight** entry is associated with a specific **Tool Output** through a one-to-one relationship, as each weight corresponds to a particular tool's output and its impact on the DWS output. This relationship allows for precise tracking of the individual tool contributions to the final DWS analysis.

Each table uses **UUIDs (Universally Unique Identifiers)** as primary keys to ensure global uniqueness, particularly in distributed systems. UUIDs provide several benefits over sequential indexes:

- **Global Uniqueness**: UUIDs can be generated independently across different systems without collisions, making them ideal for distributed environments.
- **Security**: UUIDs are pseudo-randomly generated, making it difficult to infer meaningful information about the associated data.

- **Concurrency**: UUIDs avoid conflicts in high-concurrency environments, where sequential IDs may require locking mechanisms.

To generate UUIDs efficiently, **version 1 UUIDs**[45] were used. UUIDv1 was chosen for its time-based component, which ensures chronological ordering, optimises indexing by reducing fragmentation, and improves query performance. Unlike UUIDv4, which is fully random and less efficient for indexing, UUIDv1 allows for better event tracing and debugging. To mitigate security concerns related to MAC address exposure, a randomly generated multicast MAC address was used. While UUIDv7 offers similar advantages, it was not selected due to limited support in existing database systems. The database structure can be summarised in Figure 43:

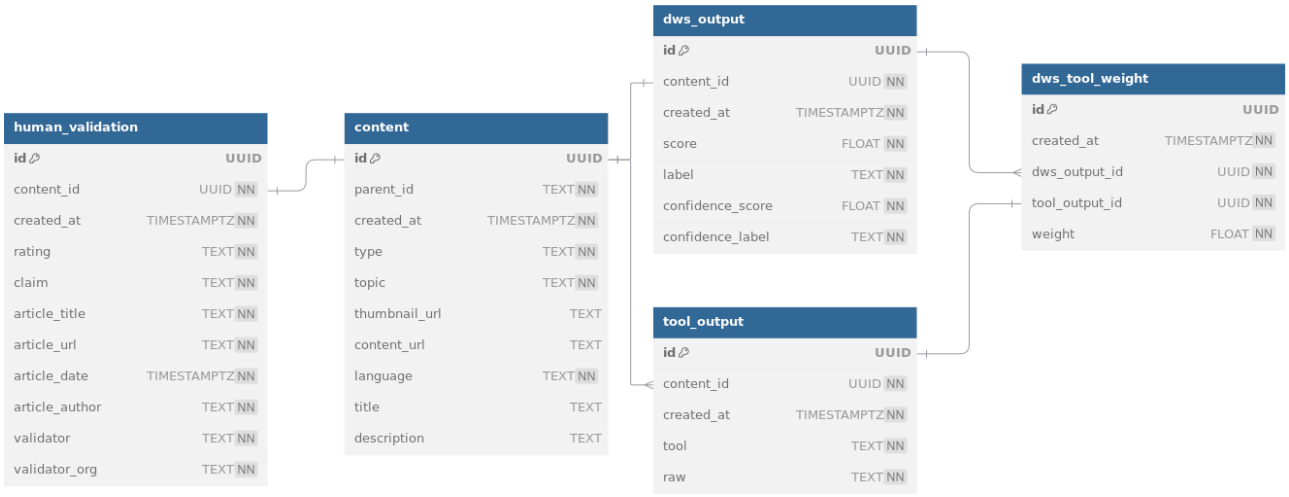

Figure 43: *Monitoring database structure*

3.13. Identity and access management

In AI4TRUST Platform v1, **access management** was primarily handled by a pre-configured instance of Keycloak[46], which was personalised to meet the platform's needs. This provided authentication through JWT (JSON Web Tokens), ensuring secure and scalable user authentication and authorisation. Keycloak's robust identity and access management features allowed for efficient user role management and secure access control across the platform.

For AI4TRUST Platform v2, an enhancement was introduced to better accommodate the interactions between the Dispatcher and the API Gateway. To facilitate simpler and more secure machine-to-machine communication, **API key authentication was implemented**. This new authentication method enables secure integration between various internal services, ensuring that only authorised machines can interact with the system. By using **API keys**, the platform achieves fine-grained access control, ensuring that communication between components is both efficient and secure, while maintaining the **platform's security standards**. To support API key

[45] https://www.postgresql.org/docs/current/uuid-ossp.html
[46] https://www.keycloak.org/

authentication, **a Spring Boot security filter**[47] has been implemented in the **API Gateway component**. This filter intercepts incoming requests, checking for a valid API key in the request headers. If the key matches the predefined static API key, the request is authenticated, enabling secure interaction between services such as the Dispatcher and the API gateway. An overview of the update security management of the AI4TRUST Platform is shown in Figure 44.

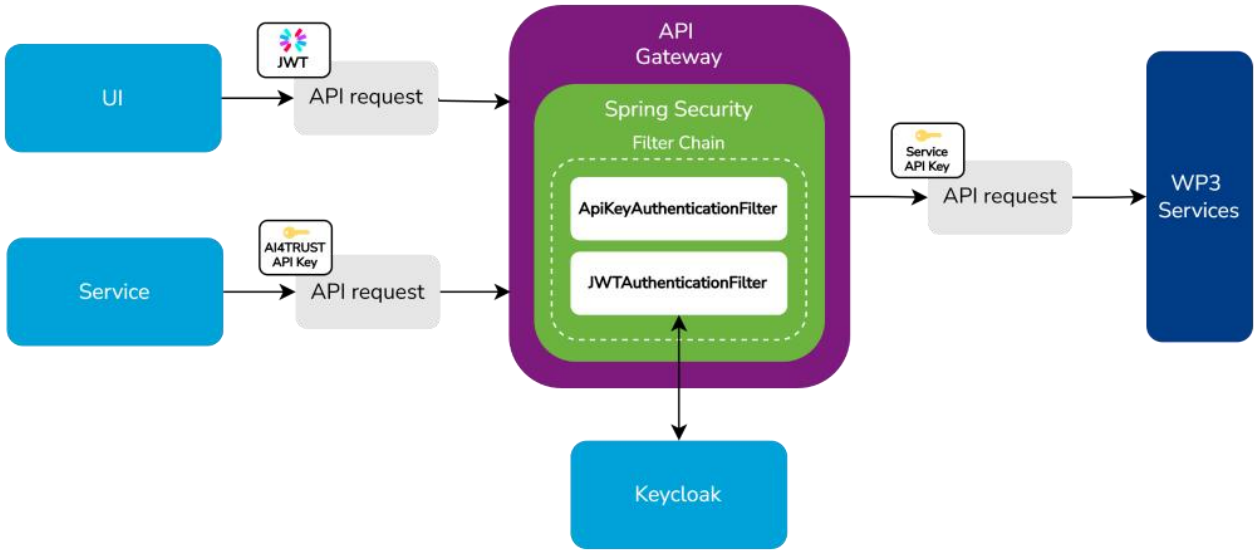

Figure 45: *Updated security structure*

The **identity management** functionality in Keycloak has been leveraged to manage access within both the back-end and front-end. **Keycloak** roles are used to define various levels of access within the platform. Each user can be assigned one or more roles, which correspond to different privileges or permissions. These roles are centrally managed in Keycloak and can be used to restrict or grant access to specific features or endpoints within the system. By configuring Keycloak roles, it is possible to fine-tune access control across the platform, ensuring that only the right users have access to specific functionalities, while others are appropriately restricted. This **role-based access control (RBAC) model** ensures that the platform remains secure and efficient at providing a tailored user experience.

To ensure that **content validation is restricted** to fact-checkers, the back-end endpoints related to the Human Validation functionality are limited to users who hold the fact-checker. This ensures that **only authorised individuals can perform these sensitive actions**. Similarly, in the front-end, access to the human validation form and related features is limited to fact-checker users, while non-fact-checker users have certain functionalities hidden. This separation of access ensures both the security and relevance of the platform's user interface for different user roles.

[47] https://docs.spring.io/spring-security/reference/servlet/architecture.html

3.13.1. Attributes

The application defines two primary macro types of roles:

- **Admin:** have full access to all platform functionalities, enabling them to manage and control every aspect of the system.
- **User:** have more limited access based on the specific permissions granted to them.

Additionally, a sub-role, the **fact-checker**, has been introduced to differentiate between standard users (journalists, researchers, policy makers, etc.) and users involved in validation-related tasks (fact-checkers). Users with the *fact-checker* sub-role are granted access to validation-specific operations defined in the Human Validation functionality (see Subchapter #). This tiered role structure allows for clear and secure access control, providing the necessary privileges to different user groups based on their responsibilities. A visual structure of the roles can be seen in Figure 46.

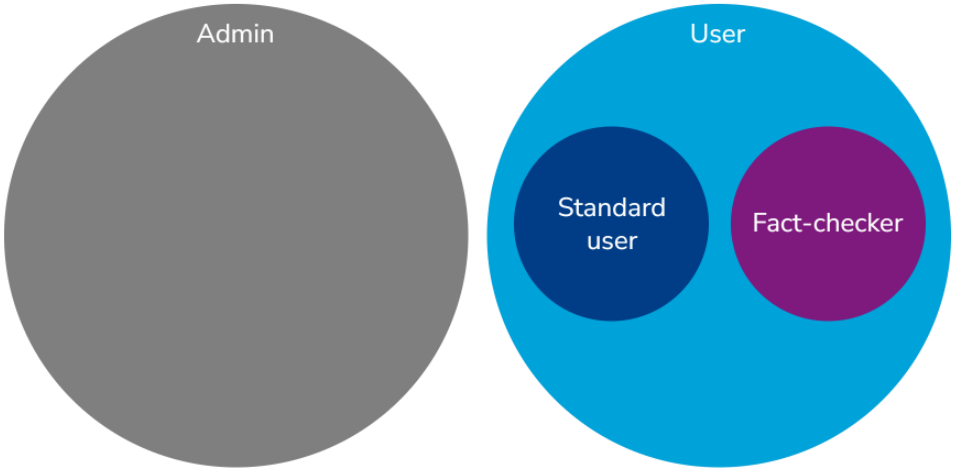

Figure 46: *Roles visual structure*

4. Integration and Deployment

This section provides an **overview of the integration and deployment processes for each component of the AI4TRUST Platform v2**. It outlines the deployment procedures, the integration strategies, and the standards adopted to ensure a seamless and consistent implementation. By following these structured approaches, the platform ensures efficient interoperability between its various elements. For more details on the infrastructure where these components have been deployed, please refer to D5.5[48].

4.1. Integration

To facilitate the integration process, **an API-first[49] approach was adopted**, ensuring that interfaces for interacting with the WP3 Lite Tools and the Disinformation Warning System (DWS) were defined upfront. Despite the complexity of aligning inputs and outputs across multiple components and coordinating with various partners, this method ultimately enabled a seamless integration of all system elements. To support this effort, dedicated repositories were created following predefined guidelines (for further details, refer to D5.5[50]). The same methodology was applied to interactions with the Streaming Platform, where Kafka Messages were established in advance to standardise data exchange, ensuring consistency and efficiency.

To further enhance the integration process, **a pipeline was implemented using GitHub Actions**[51] to facilitate the build and deployment workflow. When a new tag is pushed to GitHub (e.g., "v3.0.1"), the system automatically builds the project, packages it as a Docker image, and pushes it to GitHub Packages. This setup allows the latest version to be easily retrieved from the container registry and manually deployed on the hosting system as needed, ensuring a fast and reliable deployment process.

4.2. Deployment

To implement the AI4TRUST Platform v2, **the following services were deployed on the hosting system** (for further details about the hosting system please refer to D5.5[52]):

- **Web Application**: Manages the user interface, providing users with access to platform functionalities and interacting with the back-end via secure API calls;

[48] D5.5 - AI4TRUST Platform v1 (https://ai4trust.eu/public-deliverables/)
[49] https://www.postman.com/api-first/
[50] D5.5 - AI4TRUST Platform v1 (https://ai4trust.eu/public-deliverables/)
[51] https://github.com/features/actions
[52] D5.5 - AI4TRUST Platform v1 (https://ai4trust.eu/public-deliverables/)

- **API Gateway**: Handles business logic, processes data, and serves as an interface for the WP3 tools, integrating with services such as Keycloak and PostgreSQL to ensure seamless operation across the platform;
- **Keycloak**: Manages authentication and user roles, providing secure access to the platform's features and ensuring users are granted appropriate privileges;
- **PostgreSQL**: Acts as the relational database, responsible for storing and managing both Keycloak-related data and platform data;
- **Dispatcher**: Manages data routing between components, functioning as the backbone for AI analysis in the Monitoring functionality;
- **Disinformation Warning System**: A service developed by GDI, responsible for processing and analysing output from the WP3 lite tools, ensuring relevant analysis is conducted on incoming content;
- **Swagger UI**: Offers an interactive interface for exposing and testing the platform's API endpoints, simplifying interactions for users and developers working with the platform's APIs;
- **Reverse Proxy:** Acts as an intermediary between the client and the various deployed platform components, routing incoming requests to the appropriate service;
- **Data Platform**: A platform that addresses the requirements of data collection, data processing and data management.

The services are orchestrated and managed through **a set of Docker Compose files**, ensuring efficient service management and deployment on the hosting system.

The **reverse proxy** in the AI4TRUST Platform is implemented using Nginx[53], which is configured to route incoming requests to the appropriate component based on the URL. Through the reverse proxy, multiple services are securely exposed, ensuring **structured access to the platform's functionalities:**

- **UI service**: Accessible through the root path (/), providing users with direct interaction with the platform.
- **Platform API**: Exposed under the */api* path, serving as the main entry point for back-end services and business logic.
- **Documentation**: Available at the */docs* path, offering an interactive interface for exploring and testing API endpoints.

This configuration ensures a **clear separation of concerns** while maintaining **secure and efficient access** to the platform's services.

[53] https://nginx.org/

In addition to routing, the reverse proxy handles **SSL encryption using Let's Encrypt**[54], ensuring secure HTTPS connections for the platform's frontend. Nginx server uses TLS 1.0 through 1.3 protocols for secure communication. Let's Encrypt automatically issues and renews SSL certificates, providing an automated and reliable way to maintain secure connections, enhancing both the security and performance of the platform. The **reverse proxy** is also responsible for redirecting all incoming HTTP requests to HTTPS (except for requests to the project URL). This ensures that any non-secure traffic is automatically upgraded to a secure connection, reinforcing the platform's overall security. Moreover, it guarantees that all components, including the user interface, platform API, and documentation, are accessed securely and efficiently, with proper routing to the relevant services and ensuring that communication is encrypted. The reverse proxy serves as the backbone for managing all traffic and securing the platform's connections. The **updated deployment structure** is shown in Figure 47.

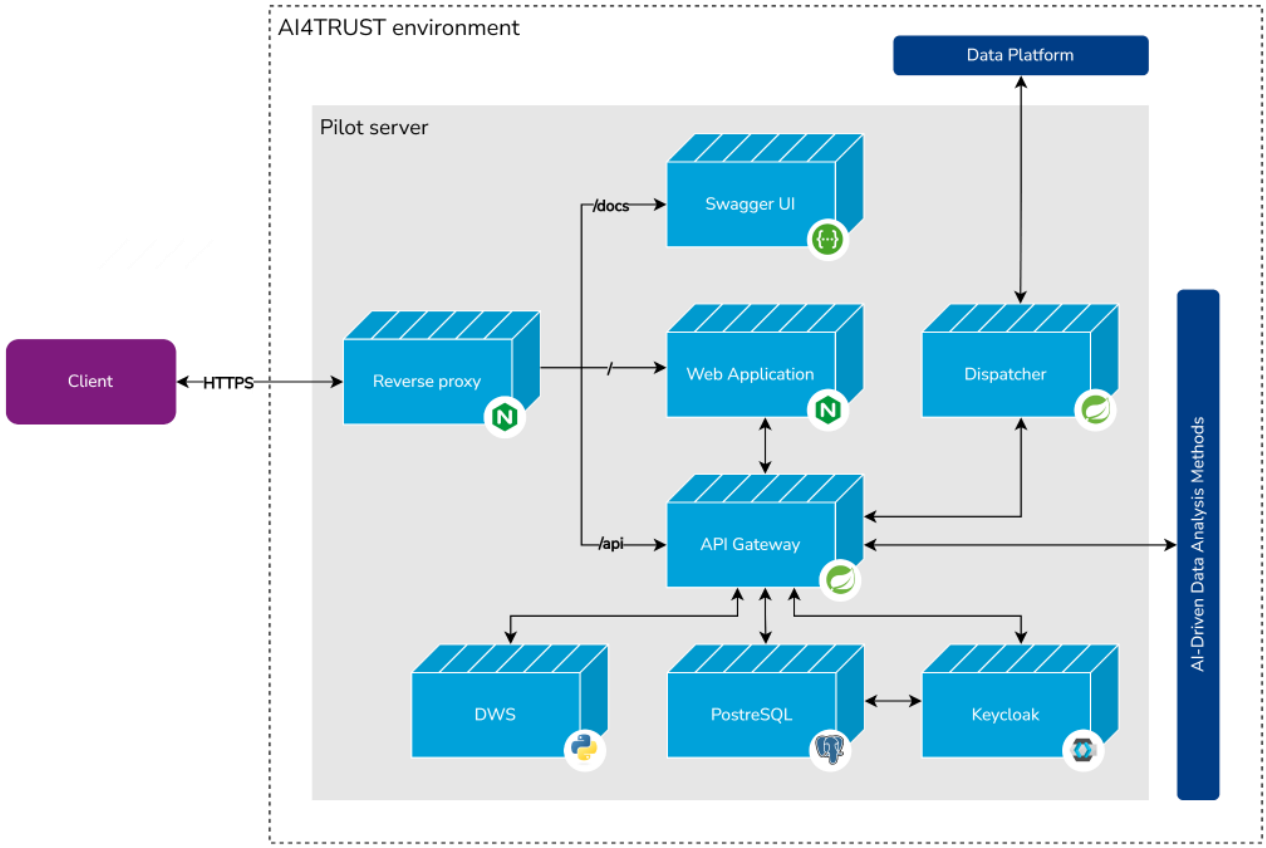

Figure 47: *Updated deployment structure*

[54] https://letsencrypt.org/

5. Conclusions and Next Steps

This deliverable **"D5.6 - AI4TRUST Platform v2"** details the advancements obtained since the last AI4TRUST Platform v1 described in D5.5[55]. The recommendations outlined in the "General Project Review Consolidated Report (HE)", dated 28 June 2024, following the project's first Review Meeting, have been instrumental in shaping this document. In particular, this deliverable immediately clarifies the **Platform Roadmap** and **how AI4TRUST Platform v2 updates and extends the functionalities of AI4TRUST Platform v1** from the user's perspective. Only after this **user-centric explanation**, it digs into the **technical aspects**, starting with an overview that explains the **data flow**, showcasing the **integration and communication** among the various parts, and proceeding with a detailed description of the **implementation and updates** of the mentioned components. As a result, this report demonstrates that the AI4TRUST Platform v2 is **fully implemented, operational, and prepared for the second pilot evaluation (WP6)**, scheduled to take place between April and May 2025.

Furthermore, the detailed Platform Roadmap clarifies **how this AI4TRUST Platform v2 relates with the forthcoming AI4TRUST Platform v3** (i.e., described in D5.7[56]), which will heavily leverage on the improved "AI-driven data analysis methods" and the newly introduced "automated data collection and analysis" to provide end-users (such as fact-checkers, journalists, media practitioners, and policymakers) with visualisation dashboards for displaying advanced analytics based on an "analysis at-scale of collected data" performed by the methods and tools that will be developed in **WP4 - Human-Centred Explainability, Interpretation and Policy**.

In sum, the **second version of the platform**, described in this deliverable, will undergo **testing, piloting, and validation by the project's end-users in April 2025 (WP6)**. The first release of **AI4TRUST Platform v3** is scheduled for **August 2025**, followed by a **third piloting phase** involving the same end-users, as well as external stakeholders, including policy-makers — as outlined in the Grant Agreement — and other potentially interested parties (e.g., additional EDMO fact-checkers from *United Against Disinformation*). AI4TRUST Platform v3 will introduce **large-scale analysis of collected data**, integrating **AI-driven data analysis methods with automated data collection and content evaluation**. It will also incorporate **feedback from the above-mentioned end-users**, gathered through an internal report prepared by the WP6 Leader, NCSR-D, aligned with a dedicated project milestone. This third piloting phase will culminate in Deliverable **D6.3 – Piloting Sessions Report v2**, scheduled for submission on 31 October 2025. This **step-by-step progression** ensures that each version of the platform builds upon the previous one, continuously enhancing its capabilities to effectively tackle digital disinformation. The **full integration** process will be comprehensively documented **in Deliverable D5.7 – AI4TRUST Platform v3**, due in M38 (February 2026).

[55] D5.5 - AI4TRUST Platform v1 (https://ai4trust.eu/public-deliverables/)
[56] D5.7 - AI4TRUST Platform v3, due by M38

6. Annex I

This annex provides deep technical and implementation details about some of the components described in Section 3.

6.1. Database: Tables

To support the **Monitoring and Human Validation Dashboard**, the following database tables were designed:

- **content**: Stores metadata related to content items obtained from the data platform.
  - **id**: Unique identifier for each content item (UUID).
  - **parent_id**: Identifier linking the content item to its parent entity (provided by the Data Platform).
  - **created_at**: Timestamp indicating when the content was created.
  - **type**: Type of content (e.g., News, Youtube, etc.).
  - **topic**: Topic category assigned to the content.
  - **thumbnail_url**: URL to a thumbnail or image associated with the content .
  - **content_url**: URL to the full content.
  - **language**: Language of the content.
  - **title**: Title of the content (if available).
  - **description**: Description of the content.
- **tool_output**: Stores raw results generated by the WP3 Lite Tools for a given content item.
  - **id**: Unique identifier for each tool output (UUID).
  - **created_at**: Timestamp indicating when the tool output was generated.
  - **content_id**: Reference to the associated content.
  - **tool**: Name of the WP3 tool that generated the output.
  - **raw**: Raw output data from the tool.
- **DWS Output**: Stores the structured results produced by the DWS for the given content item.
  - **id**: Unique identifier for each DWS output (UUID).
  - **created_at**: Timestamp indicating when the output was generated.
  - **content_id**: Reference to the associated content.
  - **score**: DWS-generated score evaluating the content.
  - **label**: Classification label assigned to the content by the DWS.
  - **confidence_score**: Confidence score for the classification.
  - **confidence_label**: Label associated with the confidence score.

- **DWS Tool Weight**: Establishes relationships between tool outputs and DWS results, assigning weights to each tool's contribution. Each tool output is linked one-to-one with a weight, while multiple tool outputs can be associated with a single DWS result.
    - **id**: Unique identifier for each weight entry (UUID).
    - **created_at**: Timestamp indicating when the weight was recorded.
    - **dws_output_id**: Reference to the associated DWS output.
    - **tool_output_id**: Reference to the associated tool output.
    - **weight**: Weight assigned to the tool's contribution to the DWS output.
- **Human Validation**: Stores fact-checking results, including claims, article references, validator details, and ratings.
    - **id**: Unique identifier for each validation entry (UUID).
    - **created_at**: Timestamp indicating when the validation was recorded.
    - **content_id**: Reference to the associated content.
    - **rating**: Fact-checker's rating of the content's credibility.
    - **claim**: The claim being fact-checked.
    - **article_title**: Title of the fact-checking article used for validation.
    - **article_url**: URL to the fact-checking article used for validation.
    - **article_date**: Publication date of the fact-checking article used for validation.
    - **article_author**: Author of the fact-checking article.
    - **validator**: Name or identifier of the fact-checker.
    - **validator_org**: Organisation associated with the fact-checker.

6.2. Dispatcher: Kafka Messages Structures

As mentioned in the Section 3.11, **the Dispatcher** integrates communication with the Data Platform through Kafka, a distributed event streaming platform used for high-throughput, fault-tolerant data processing. **Kafka** operates based on the publish-subscribe model, where producers send messages to topics, and consumers subscribe to topics to process the messages asynchronously.

Kafka topics act as logical channels that store and distribute messages, ensuring reliable communication between system components. The Dispatcher, acting as both a Kafka Consumer and Producer, ingests messages from specific topics, processes them, and then publishes results back to designated topics.

Kafka was chosen over direct HTTP-based communication due to its ability to efficiently handle high volumes of real-time data while ensuring fault tolerance. While HTTP calls can be designed with fault tolerance, Kafka provides a distributed, asynchronous messaging system that helps decouple components, reducing bottlenecks and improving overall system responsiveness. Unlike traditional request-response mechanisms, Kafka allows message processing to be fully

asynchronous without requiring clients to continuously poll for updates. Additionally, the Kafka instance used for this purpose is the same as the one described earlier for the Dispatcher component, ensuring consistency in data flow management across the system. The following sections list the various types of messages exchanged through Kafka topics, detailing their structure and purpose within the AI4TRUST Platform.

6.2.1.1. Source data

The **Kafka messages** containing the source data are **sent from Data Platform to Dispatcher**. The message varies according to the content type (e.g., Youtube videos, News articles). Below are some examples of the structures of the messages.

- **YouTube Message**

The YouTube Message contains the following fields:

  - *data_owner:* Owner of this message data, usually set to 'FBK'.
  - *relevance_language:* Language in which the title and description are written.
  - *id:* UUID identifying this message data in the Data Platform collection.
  - *title:* Title of the YouTube message data.
  - *description:* Description of the youtube message data.
  - *image_url:* URL for the caption image of the YouTube video.
  - *video_url:* URL for the YouTube video.
  - *publish_time:* Time when a YouTube video has been published.
  - *keyword:* Word describing the main topic of the video.

- **News Message**

The News Message contains the following fields:

  - *data_owner:* Owner of this message data, usually set to 'FBK'.
  - *language:* Language in which the title and description are written.
  - *id:* UUID identifying this message data in the Data Platform collection.
  - *url:* URL for the news article.
  - *title:* Title of the youtube message data.
  - *text:* Text of the news article.
  - *image_url:* URL for the caption image of the news article.
  - *publish_time:* Time when a news article has been published.
  - *keyword:* Word describing the main topic of the video.

6.2.1.2. Input data for YouTube messages

This section outlines **how the fields from an input YouTube message are mapped** to invoke different AI-Driven Data Analysis Methods and the Disinformation Warning System. Each method processes specific attributes extracted from the structured JSON message received via Kafka.

- **Check Worthy Claim Detection Input Message**:
  - *text:* This field is mapped from the *title* of the YouTube message, which provides contextual information about the video content.
  - *language:* This field is mapped from *relevance_language*, indicating the language in which the tile is written.
- **Disinformation Signals Detection Input Message:**
  - *title:* Mapped from the *title* of the YouTube message, which summarises the video's content.
  - *text:* Mapped from the *description*, *offering additional details regarding the video.*
  - *language:* Mapped from *relevance_language*, denoting the language used in the title and description.
- **Sensational Content Detection Input Message:**
  - *video_url:* Directly mapped from *video_url*, representing the link to the YouTube video.
  - *image_url:* Directly mapped from *image_url*, which provides the URL of the video's caption image.
- **Visual Text Misalignment Detection Input Message:**
  - *video_url:* Directly mapped from *video_url*, referencing the YouTube video link.
  - *short_text:* Mapped from *tile*, as it provides a concise textual representation of the video's subject.
  - *language:* Mapped from *relevance_language*, indicating the language of the textual content.
- **Video Anomaly Detection Input Message:**
  - *video_url:* Directly mapped from *video_url*, pointing to the YouTube video under analysis.
- **Domain Disinformation Detection Input Message:**
  - *url:* This field is derived from *video_url* as it represents the source of the video content.
- **Disinformation Warning System Input Message:**
  - *content_type:* Mapped from the message type, identified as YOUTUBE.
  - *feature_scores:* Populated based on the outputs generated by the WP3 Lite Tools.
  - *media_url:* Mapped from *video_url*, referring to the YouTube video's link.
  - *metadata:* Includes multiple fields such as *data_owner* (set to 'FBK'), *publish_time* (timestamp of video publication) and *keyword* (representing the video's main topic).

- *missing_features:* List computed based on unavailable AI-Driven Data Analysis Methods
- *raw_content:* Field containing a sample description of the content the DWS is going to analyse.

- **Input data for News messages**

This section outlines **how the fields from an input News message are mapped** to invoke different AI-Driven Data Analysis Methods and the Disinformation Warning System. Each method processes specific attributes extracted from the structured JSON message received via Kafka.

- **Check Worthy Claim Detection Input Message**:
  - *text:* Mapped from *title, which contains the main content of the next article*.
  - *language:* Mapped from *language*, indicating the language in which the article is written.
- **Disinformation Signals Detection Input Message:**
  - *title:* Mapped from *title*, representing the headline of the news article.
  - *text*:. Mapped from *text*, providing the full content of the news article.
  - *language:* Mapped from *language*, specifying the language of the article.
- **Sensational Content Detection Input Message:**
  - *image_url:* Mapped from *image_url*, which provides the URL of the caption image associated with the news article.
- **Visual Text Misalignment Detection Input Message:**
  - *image_url:* Mapped from *image_url*, representing the news's article associated image.
  - *short_text:* Mapped from *title*, as it provides a brief textual representation of the news article.
  - *language:* Mapped from *language*, indicating the language of the textual content.
- **Domain Disinformation Detection Input Message:**
  - *url:* Mapped from *url*, referencing the news article's source.
- **Disinformation Warning System Input Message:**
  - *content_type:* Mapped as *NEWS*, identifying the message type
  - *feature_scores:* Populated based on the outputs generated by the AI-Driven Data Analysis Methods.
  - *media_url:* Mapped from *url*, representing the source of the news article.
  - *metadata:* Includes multiple fields such as *data_owner* (set to 'FBK'), *publish_time* (timestamp of video publication) and *keyword* (representing the video's main topic).
  - *missing_features:* List computed based on unavailable AI-Driven Data Analysis Methods
  - *raw_content:* Field containing a sample description of the content the DWS is going to analyse.

6.2.1.3. Results data

The results from AI-Driven Data Analysis Methods first return to the API Gateway, which then forwards them to the Dispatcher via an API call. Once received, the Dispatcher processes and sends the results to the Data Platform. The Kafka messages containing the results data (AI-Driven Data Analysis Methods results) are sent from the Dispatcher to the Streaming Platform through a Kafka topic, Below are the structures of the messages:

- **Check Worthy Claim Detection Result Message**:
  - *summary:* Object describing overall the level of worthiness.
  - *segments:* List of objects, specifying for each segment span and level of worthiness.
- **Disinformation Signals Detection Result Message:**
  - *document:* List of text describing the document given in input
  - *segments:* List of objects describing for each segment values such as tactic, span and confidence score.
- **Sensational Content Detection Result Message:**
  - *label:* Text indicating the sensational content detected.
  - *score:* Score between 0 and 1 describing the confidence.
- **Visual Text Misalignment Detection Result Message:**
  - *prediction:* Value indicating if Video and Text are not aligned.
  - *score:* Score between 0 and 1 describing the confidence.
- **Video Anomaly Detection Result Message:**
  - *result:* Word describing if video given in input is anomalous or not.
  - *confidence_score:* Score between 0 and 1 indicating the confidence on the result
- **Domain Disinformation Detection Result Message:**
  - *label:* Status label indicating whether the input is classified as present or not.
- **Disinformation Warning System Result Message:**
  - *confidence:* Value describing the level of confidence, it can be 'High', 'Medium' or 'Low'.
  - *confidence_score:* Score describing the level of confidence, it can range from 0 to 1.
  - *explainability_coef:* List of objects containing a summary of each result returned from WP3 Lite Tools.
  - *n_features_in:* Number of features.
  - *risk:* Value describing the level of risk, it can be 'High', 'Medium', 'Low'.
  - *risk_score:* Score describing the level of confidence, it can range from 0 to 1.