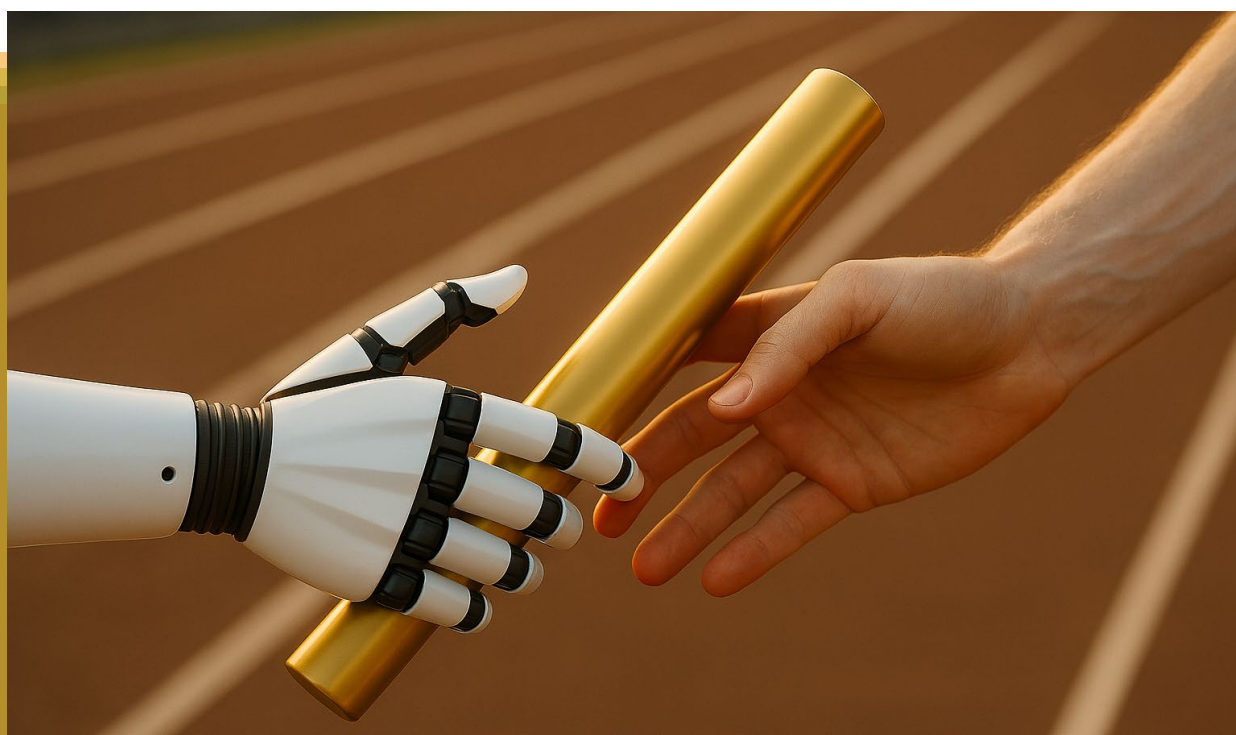


EXPLORING AND APPLYING ARTIFICIAL INTELLIGENCE TOOLS FOR GREY LITERATURE SEARCHES WITHIN THE NEED INITIATIVE



CÉDRIC SCHOONEN, LAURENCE PIRENNE, ANOUK BERGEZ, CAROLINE OBYN, IRINA CLEEMPUT,
CHARLINE MAERTENS DE NOORDHOUT, VINCENT BOUCHER

Title:	Exploring and applying artificial intelligence tools for grey literature searches within the NEED initiative
Authors:	Cédric Schoonen (B12 Consulting), Laurence Pirenne (B12 Consulting), Anouk Bergez (B12 Consulting), Caroline Obyn (KCE), Irina Cleemput (KCE), Charline Maertens de Noordhout (KCE), Vincent Boucher (B12 Consulting)
Senior supervisor:	Irina Cleemput (KCE)
Project manager:	Sabine Corachan (KCE)
External validators:	Benoit Macq (UCLouvain), Hubert Stoop (MyData-Trust), Alexandre Templier (Quinten)
Acknowledgements:	Rani Claerman (Sciensano), Robby De Pauw (KCE), Claudia Schoenborn (KCE), Julie Paradis (KCE), Laurence Kohn (KCE), Nicolas Fairon (KCE), Patrice Chalon (KCE), Stephan Devriese (KCE), Elena Costa (KCE), Céline Pouppez (KCE), Bertrand Chenal (B12 Consulting)
Reported interests KCE members:	KCE is a federal institution funded by INAMI/RIZIV, by the Federal Public Service of Health, Food Chain Safety and Environment, and by the Federal Public Service of Social Security. KCE's mission is to advise policymakers on decisions relating to health care and health insurance on the basis of scientific and objective research. It is expected to identify and shed light on the best possible solutions, in the context of an accessible, high-quality health care system with due regard for growing demand and budgetary constraints. KCE has no interest in companies (commercial or non-commercial, i.e. hospitals and universities), associations (e.g. professional associations, unions), individuals or organisations (e.g. lobby groups) that could be positively or negatively affected (financially or in any other way) by the implementation of the recommendations. All experts involved in the writing of the report or the peer-review process completed a declaration of interest form. Information on potential conflicts of interest is published in the colophon of this report. All members of the KCE Expert Team make yearly declarations of interest and further details of these are available upon request.
Reported interests:	None declared
Layout:	Ine Verhulst
Financial support:	The NEED project is financially supported by Belspo, under grant number RT/23/NEED
AI Disclaimer:	In the development of this work, the author(s) utilised Copilot, ChatGPT and Gemini Deep Search to assist with the wording and linguistic flow and with the research process. Following its use, the content was carefully reviewed and revised by the author(s), who accept full responsibility for the final version of the manuscript.

Disclaimer:	• The external experts were consulted about a (preliminary) version of the scientific report. Their comments were discussed during meetings. They did not co-author the scientific report and did not necessarily agree with its content. • Subsequently, a (final) version was submitted to the validators. The validation of the report results from a consensus or a voting process between the validators. The validators did not co-author the scientific report and did not necessarily all three agree with its content. • Finally, this report has been approved by common assent by the Executive Board.
--------------------	--



- Only the KCE is responsible for errors or omissions that could persist. The policy recommendations are also under the full responsibility of the KCE.

Publication date:	2 April 2026
Project number:	2023-80-2_WP2
Domain:	Needs Examination, Evaluation and Dissemination (NEED)
MeSH:	Unmet needs, needs assessment, prioritisation
NLM Classification:	W84.3
Language:	English
Format:	Adobe® PDF™ (A4)
Legal depot:	D/2026/10.273/18
ISSN:	2466-6459
Copyright:	KCE reports are published under a “by/nc/nd” Creative Commons Licence http://kce.fgov.be/content/about-copyrights-for-kce-publications .



How to refer to this document?

Schoonen C, Pirenne L, Bergez A, Obyn C, Cleemput I, Maertens De Noordhout C, Boucher V. Exploring and Applying Artificial Intelligence Tools for Grey Literature. Brussels: Belgian Health Care Knowledge Centre (KCE). 2025. KCE Reports 416. <https://doi.org/10.57598/R416C>.

This document is available on the website of the Belgian Health Care Knowledge Centre.



■ TABLE OF CONTENTS

■	SCIENTIFIC REPORT	4
1	INTRODUCTION	4
2	OBJECTIVES	5
3	BACKGROUND AND CONTEXT	5
3.1	THE CHALLENGE OF FILLING THE NEED DATABASE WITH GREY LITERATURE	5
3.2	FORMALIZING A WORKFLOW FOR REVIEWING GREY LITERATURE	6
4	METHODOLOGY	7
4.1	TOOL IDENTIFICATION AND ASSESSMENT PROCESS	7
4.1.1	Step A: Tool identification	8
4.1.2	Step B: Tool categorization	9
4.1.3	Step C: Filtering out inadequate tools	10
4.1.4	Step D: Assessment criteria determination	10
4.1.5	Step E: Examination of remaining tools	11
4.1.6	Step F: Synthesis of results and strategic interpretation	13
5	OBJECTIVE 1: TOOL LANDSCAPE REVIEW AND ANALYSIS	13
5.1	CATEGORY 1: AGGREGATED REPOSITORIES	14
5.2	CATEGORY 2: AI SEARCH TOOLS	14
5.3	CATEGORY 3: DOCUMENT RETRIEVAL TOOLS	15
5.4	CATEGORY 4: TRADITIONAL REVIEW ACCELERATORS	16
5.5	CATEGORY 5: MISCELLANEOUS TOOLS	17
5.6	RECOMMENDATIONS AND RETAINED CANDIDATE TOOLS FOR MODULAR INTEGRATION	18
5.6.1	Candidate tools retained for Step 1: Discovery module	18
5.6.2	Candidate tools retained for Step 2 and 3: Exploration and extraction module	22
6	OBJECTIVE 2: STRATEGY FOR APPLYING AI TOOLS WITHIN THE NEED WORKFLOW	28
6.1	OVERVIEW OF THE INSTRUMENTALIZATION FRAMEWORK	28
6.1.1	Core components of the framework	29
6.1.2	Key concepts for understanding the workflow	30
6.2	STAGE 1: PROTOCOL DEFINITION AND SOURCE DISCOVERY	30
6.2.1	Prompt engineering strategy for Source discovery	32
6.2.2	Technical specifications of Stage 1	32
6.3	STAGE 2: SOURCE EXPLORATION AND DOCUMENT EXTRACTION	33
6.3.1	Defining the extraction protocol and configuration	34
6.3.2	Technical specifications of Stage 2	35
6.4	STAGE 3: ANALYSIS AND SYNTHESIS	35
6.4.1	Human expert review and system feedback	37



6.4.2	Technical specifications of Stage 3.....	37
6.5	CONSIDERATIONS AND CONSTRAINTS FOR THE INSTRUMENTALIZATION FRAMEWORK.....	37
6.5.1	Legal and ethical considerations for grey literature retrieval.....	37
6.5.2	Operational and security considerations.....	39
6.5.3	External tools integration and modularity.....	40
6.5.4	Controlled agency: A deterministic orchestration model.....	42
7	DISCUSSION AND CONCLUSION	44
7.1	OBJECTIVE 1: TOOL LANDSCAPE REVIEW AND METHODOLOGICAL REFLECTIONS.....	44
7.2	OBJECTIVE 2: STRATEGY FOR APPLYING AI TOOLS WITHIN THE NEED WORKFLOW.....	45
7.3	CONCLUSION: WHAT AI CAN AND CANNOT DO FOR GREY LITERATURE REVIEW.....	46
8	RECOMMENDATIONS AND NEXT STEPS	46
8.1	IMPLEMENTATION ROADMAP.....	46
8.1.1	Level 0: Empirical tool validation and feasibility benchmarking.....	46
8.1.2	Level 1: Scripted automation and technical specification building.....	47
8.1.3	Level 2: Integrated instrumentalization framework platform.....	47
8.1.4	Level 3: Future enhancements and extended capabilities.....	48
■	REFERENCES	49
■	APPENDICES	51
APPENDIX 1.	LIST OF TOOLS BY CATEGORY	51
APPENDIX 2.	TOOLS DESCRIPTION	53
APPENDIX 3.	TOOLS EXCLUSION MOTIVES	60

LIST OF FIGURES

Figure 1 – Systematic review workflow for grey literature.....	6
Figure 2 – Methodology for the screening and analysis of candidate tools.....	8
Figure 3 – High-level overview of the instrumentalization framework.....	29
Figure 4 – Stage 1 workflow: protocol definition and source discovery.....	31
Figure 5 – Stage 2 workflow: Source exploration and document extraction.....	33
Figure 6 – Stage 3 workflow: Analysis and synthesis.....	36



LIST OF TABLES

Table 1 – Classification of potential AI tools for grey literature review, ordered by appearance in the workflow.	9
Table 2 – Weighted assessment criteria for examining the remaining tools.....	11
Table 3 – Summary of the 5 tool categories for grey literature review	18
Table 4 – Comparison of the three retained AI search tools.....	19
Table 5 – Comparison of the five retained crawling and scraping tools	23
Table 6 – Components of the Source discovery prompt.....	32

LIST OF ABBREVIATIONS

ABBREVIATION	DEFINITION
AI	Artificial Intelligence
API	Application Programming Interface
BeBOD	Belgian national Burden of Disease study
DOI	Digital Object Identifiers
HTA	Health Technology Assessment
LLM	Large Language Models
MVP	Minimum Viable Product
NEED	Needs Examination, Evaluation and Dissemination
NIHDI	National Institute for Health and Disability Insurance
OECD	Organisation for Economic Co-operation and Development
PoC	Proof-of-Concept
RAG	Retrieval-Augmented Generation
R&D	Research & Development
UI	User Interface
WHO	World Health Organization



■ SCIENTIFIC REPORT

1 INTRODUCTION

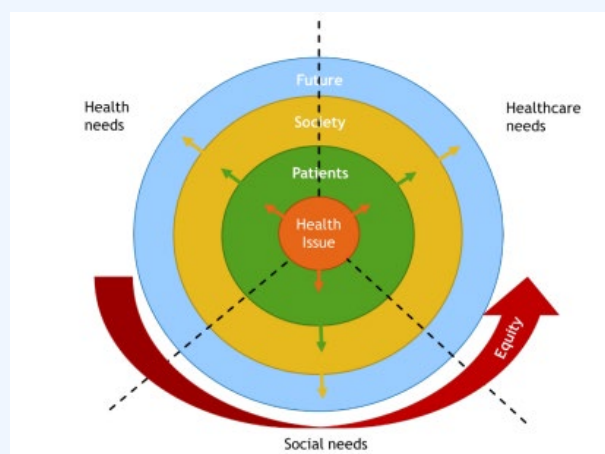
The NEED project was launched in the context of the Belgian Presidency of the Council of the European Union during the first semester of 2024.¹ While there is general agreement that healthcare policy (from approval to reimbursement) and Research & Development (R&D) in healthcare should be needs-driven, they currently remain highly supply-driven. Consequently, several (high) health-related needs remain unmet. Addressing patients' and/or societal unmet health-related needs should be a determining factor in prioritisation exercises at all levels in the healthcare system (from decisions regarding the development of new treatments up to pricing and reimbursement decisions and clinical applications).

The creation of a needs-driven health system requires a life-cycle approach, involving (1) the identification of the most important unmet needs, both on the individual and on the societal level, (2) the creation of incentives for health technology developers to steer R&D activities towards the highest needs and (3) approval and reimbursement processes that take the performance of new products on these unmet needs into account.

Patient and societal needs, as defined in the NEED framework^{2, 3} (see Box 1), are independent of potential future products or treatments and should be assessed on their own. Unmet needs must first be identified as such, while the types of solutions to address them should be evaluated in a subsequent step. It might be, for instance, that in some cases the highest needs of patients might be solved by a pharmaceutical product, while in other cases a surgical or behavioural intervention might be more appropriate to solve a high unmet need. For the efficient allocation of healthcare resources (obtaining the highest value for a given amount of resources), it is important to know where value, in terms of meeting patients' highest needs, can be obtained.

Box 1 – Key components of the NEED framework

To identify health-related needs, the NEED framework is structured along dimensions, domains, criteria and indicators.



- Dimensions: relate to the perspectives taken to identify and measure needs: patient, societal, or future (with equity as a transversal dimension)
- Domains: relate to the type of health-related need: health, healthcare, or social.
- Criteria: represent the different needs by domain and within each dimension.
- Indicators: assess to what extent health-related needs identified by the criteria are met or unmet.

In 2024, the NEED database⁴ was launched, which currently contains information on unmet needs related to Crohn's disease, malignant melanoma, long Covid and 80 additional health conditions from [the Belgian national Burden of Disease](#) (BeBOD) study. The objective of this report is to accelerate the population of the [NEED database](#) through secondary data sources, following the approach used

for incorporating BeBOD data. To achieve this, we explore the use of artificial intelligence (AI) tools to systematically identify, retrieve, and extract relevant unmet needs estimates from secondary data sources. In particular, we want to investigate how AI can support the systematic identification and extraction of grey literature to accelerate the population of the NEED database. Grey literature is defined in this report as documents produced by governments, universities, NGOs, agencies, and other organizations, but not published by commercial publishers. It encompasses, for instance, scientific reports, abstracts or databases. Importantly, the population of the NEED database must comply with binding legal, ethical, data governance, operational, and security requirements, which frame the range of acceptable technical solutions.

This study is part of the NEED initiative (health-need.eu). It builds on the methodological work, such as the NEED standardised literature search strategy developed as part of the NEED protocol.⁵

2 OBJECTIVES

1. To investigate how AI-based tools can facilitate the discovery, identification, exploration, and retrieval of grey literature across a variety of sources and formats; and
2. To develop a practical strategy for applying these tools within the NEED workflow, or, where no sufficient solution exists, to design and pilot a new AI tool that addresses identified gaps while respecting data protection, privacy, and ethical requirements.

3 BACKGROUND AND CONTEXT

3.1 The challenge of filling the NEED database with grey literature

While a large part of the methodology in traditional scientific reviews relies on centralized, standardized databases of peer-reviewed literature, a substantial portion of evidence relevant to health policy is found in the so-called grey literature.^{6, 7} These materials are often not easily discoverable via traditional academic or journal publishing channels.⁸ In the context of health policy, this includes critical resources such as reports from Health Technology Assessment (HTA) agencies, white papers, government documents, fact sheets, conference proceedings, technical notes, and policy papers from international organizations such as World Health Organization (WHO) or Organisation for Economic Co-operation and Development (OECD). Unlike scientific publications, these resources are highly decentralized and dispersed across numerous agency websites and repositories. They lack the standardized indexing and metadata that enable systematic retrieval, and often appear in varied, unstructured formats ranging from PDF reports to dynamic web pages.⁹⁻¹¹

This fragmentation and lack of standardized retrieval mechanisms pose a substantial challenge for research projects that have (at least partly) to rely on evidence from grey literature, such as NEED. When testing the NEED approach on Crohn's disease and malignant melanoma as proof-of-concept (PoC) studies¹², the research team had to manually identify grey-literature sources through pragmatic, non-systematic web searches. This process involved defining specific keywords and searching online databases, published scientific literature (PubMed and Google Scholar), and the general web (Google). For example, specific unmet health-related needs indicators were searched using keywords on Google (e.g., "economic burden", or "environmental impact") or manually extracted from government websites and registries (e.g., the [Belgian Cancer Registry](#) or the National Institute for Health and Disability Insurance ([NIHDI](#)) website). This process was not only labour-intensive and time-consuming, but because there was no automated mechanism for discovering new sources, searches targeted only a fixed and limited set of websites. Important information from unknown or less visible repositories was likely missed. Therefore, the challenge is to scale this data-collection process to cover the more than 90 health conditions planned for the [NEED database](#).

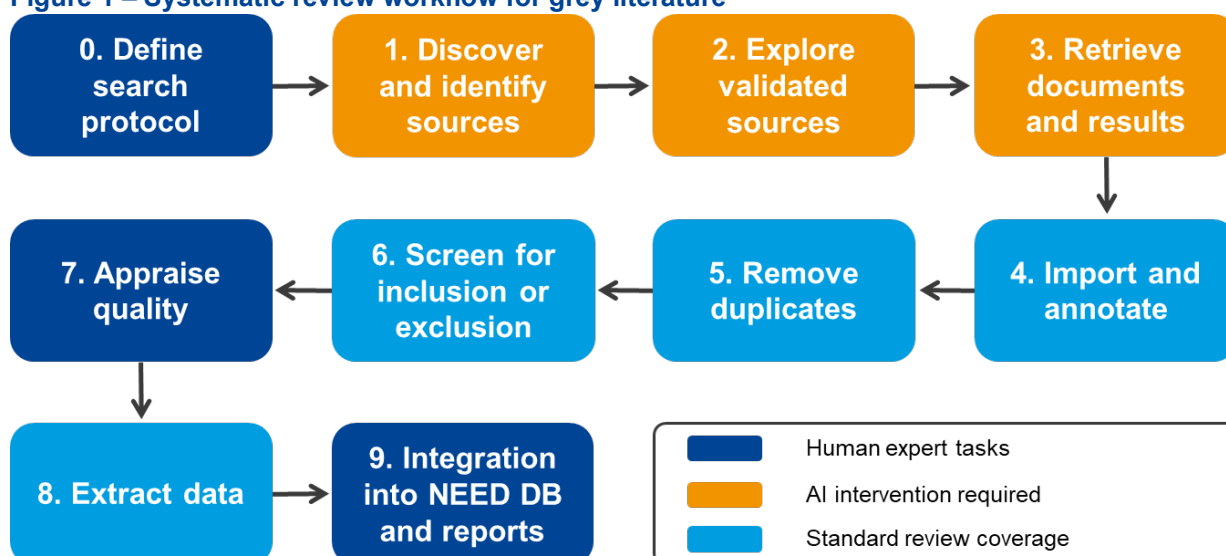


Beyond the technical and methodological challenges described above, the collection and processing of grey literature for the NEED database must operate within a defined legal, ethical, and security perimeter. Grey literature retrieval often involves automated access to heterogeneous web sources, document downloads, and text processing activities that are subject to copyright rules, terms of use, and responsible access practices. In addition, as a federal research institution, KCE must ensure that data governance, auditability, and information security requirements are upheld throughout the workflow, including when interacting with external AI services under controlled and policy-governed conditions. These constraints are not just implementation details. They shape the design of the workflow itself, including which technical solutions are acceptable, how sources can be accessed, how documents are managed, and how outputs are shared. The challenge of scaling grey literature collection is therefore not only methodological, but also regulatory and operational, requiring solutions that are compatible with institutional obligations as well as scientific standards. A detailed discussion of these legal, operational, and security considerations is provided in Section 6.5.

3.2 Formalizing a workflow for reviewing grey literature

To move from the pragmatic approach used in the proof of concept (PoC) studies to a systematic and scalable process, we first formalized the end-to-end workflow for grey-literature review (Figure 1).

Figure 1 – Systematic review workflow for grey literature



This diagram presents the workflow established for collecting grey literature for the NEED database. The steps are color-coded to indicate project priorities. Dark blue: human expert tasks requiring domain knowledge and manual decision-making. Orange: steps where AI intervention has been prioritized for this mission. Light blue: activities typically supported by commonly available literature-review software. DB: Database

The workflow begins with a human expert defining the research protocol (Step 0) and establishing the scope, disease condition, as well as target unmet health-related needs indicators.

Based on this input, the first operational challenge is the discovery and identification of relevant grey-literature sources (Step 1). This encompasses diverse locations, such as websites of Health Technology Assessment (HTA) agencies, national health departments, patient associations, and international organization repositories, that might contain the data required for the NEED database. Ideally, sources identified during this phase are validated by experts and documented in a dedicated database for future scalability, including metadata describing the source type, scope, required update frequency and other relevant information for the NEED researchers.

Once sources are validated, the workflow proceeds to exploration (Step 2), which involves navigating through the validated websites to locate relevant specific pages or entries. This is followed by retrieval

(Step 3), where the associated content and documents are downloaded and extracted, requiring the handling of diverse and unstructured formats.

The retrieved materials then enter the processing phase, supported by existing literature-management tools. Materials are imported and annotated with metadata (source, URL, date, etc.) (Step 4), and duplicate records are removed (Step 5) to clean the corpus. These tools facilitate the screening of content against inclusion and exclusion criteria (Step 6) defined in the initial research protocol. As grey literature might lack standard titles or abstracts (e.g., website sections or database entries), this screening often requires reviewing executive summaries or performing full-text searches to assess relevance.

The process concludes with human experts analysing and synthesizing the retrieved data. Experts appraise the methodological quality or risk of bias (Step 7) and extract relevant data (e.g., PICO description, outcomes, numerical results) (Step 8). Finally, they include estimates into the NEED database and synthesize the evidence into a detailed report on the unmet needs associated with the specific disease (Step 9), completing the review cycle.

For this study, the three initial steps (Steps 1–3) were identified as the most relevant for implementing AI assistance. While the subsequent processing steps (Steps 4–9) are already well-supported by established platforms (like Rayyan or DistillerSR), the initial identification (Step 1) and retrieval of these decentralized sources represent a significant bottleneck in the current manual research workflow. Therefore, this project focuses on automating these early stages to ensure reproducible data collection, while maintaining essential human oversight throughout the workflow.

4 METHODOLOGY

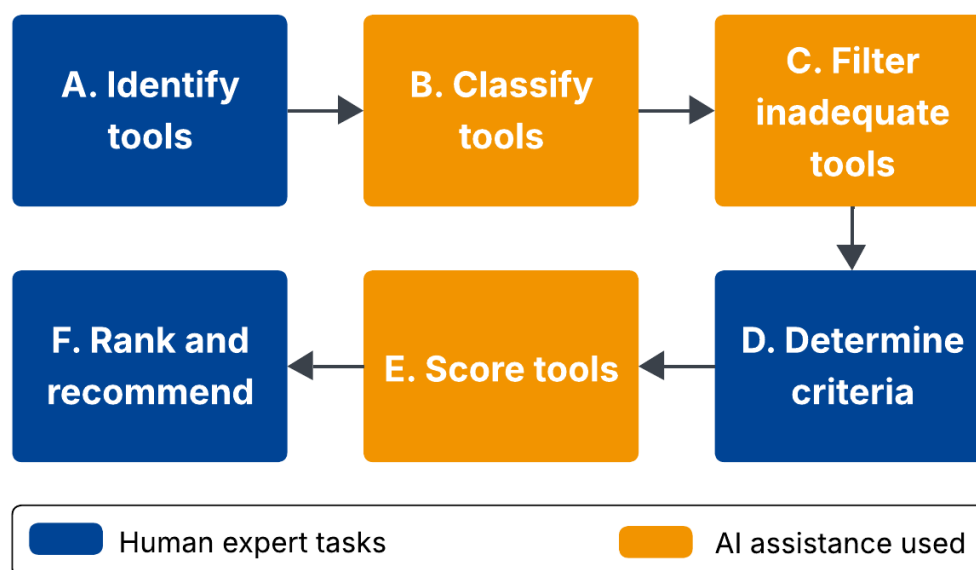
4.1 Tool identification and assessment process

The first objective of this work was to assess whether the NEED research team could rely on an existing off-the-shelf tool to support the upstream stages of grey-literature review. More specifically, this involved determining whether a single off-the-shelf solution could support the first three steps of the formalized workflow (Figure 1): Discovery, Exploration, and Retrieval. If no such end-to-end solution could be identified in the current tool landscape, then a custom solution would need to be developed to automate the workflow.

To support this initial assessment, we established a multi-phased methodology enhanced by AI-powered deep searches to analyse candidate tools. This evaluation was conceived as a structured market screening to inform architectural decision-making rather than as a formal benchmarking or statistically reproducible ranking study. Our approach consisted of the following six-step procedure (see Figure 2).



Figure 2 – Methodology for the screening and analysis of candidate tools



This diagram summarizes the six-step procedure used to screen and analyse candidate tools and to determine whether an existing solution could support the NEED workflow (Steps A-C). The color-coding reflects the primary contribution to each Step. Dark Blue: steps requiring domain knowledge and manual decision-making. Orange: steps benefiting from targeted AI assistance (e.g., automated filtering and specialized scoring using Gemini Deep Search).

The process began with the identification of a broad collection of potentially relevant tools (Step A). These were then categorized (Step B) based on their nature (database or software) and the steps of the workflow they appeared capable of supporting. Next, we employed Gemini Deep Search, an AI capability that performs autonomous, multi-step web searches to gather and synthesize up-to-date information, to support the preliminary screening of candidate tools (Step C) and exclude those that clearly did not meet the project's requirements. We then defined assessment criteria (Step D) to examine the extent to which the remaining tools could support the NEED workflow end to end. The tools retained after filtering were then individually assessed against these criteria (Step E) with AI-assisted documentation and human review. Finally, the results were synthesized (Step F) to determine whether an existing end-to-end solution could support the full NEED workflow, or whether the development of a custom solution would be required.

Each step is detailed in the following sections.

4.1.1 Step A: Tool identification

An initial set of software tools and databases/repositories was gathered through background research on the NEED project and on grey literature in general, and was expanded using several methods:

- Following the citations network of initial key reports¹³ and articles on the topic⁵.
- Conducting manual web searches for additional information on grey literature, as well as on the way systematic reviews are conducted and the tools used to assist researchers in this process.
- Consulting guides available on university library websites¹⁴, which were particularly useful for identifying commonly used grey literature resources.
- Broadening the search with Google's Deep Search feature in Gemini. Note that at this early stage, the Gemini searches were not documented, and no refined or reusable prompt had yet been developed.

This exploratory process resulted in a list of 62 resources, including databases, search engines, AI assistants, and web-scraping tools. The diversity of these tools and their functionalities highlighted the need for an initial categorization to structure the subsequent screening and analysis.

4.1.2 Step B: Tool categorization

A key challenge posed by this heterogeneous list was the need to organize and classify the identified tools. This step was conducted with support from Gemini, which was prompted to research, group, and cluster the 62 tools based on shared functionalities and characteristics (see supplementary material).

The initial classification was performed twice. The two outputs were then compared to identify discrepancies between classifications. The final categories were derived from this comparison and refined by the authors to optimize the overall structure.

The resulting categorized list contains the 62 tools distributed across five functional areas (Table 1).

Table 1 – Classification of potential AI tools for grey literature review, ordered by appearance in the workflow.

Category	Description	Number of tools
Aggregated repositories	Databases/repositories which articles, documents, etc. (e.g., BASE, which serves as a "place" rather than a computational "tool").	18
AI search tools	Tools that can autonomously identify which websites/databases should be investigated.	10
Document retrieval tools	Tools that can be used to download documents/data from a website (e.g., scraping tools).	7
Traditional review accelerators	Tools designed to accelerate traditional systematic review tasks (e.g., deduplication, screening, synthesis).	20
Miscellaneous tools	Various search-related tools which poorly fit in other categories.	7

The table presents the categorization of the 62 tools identified in Step A (Figure 2), based on their core functionality and nature (e.g., database vs. software). The classification illustrates the diversity of the landscape and supports the need for subsequent filtering and more detailed examination of the remaining candidates.

Note that the category “Aggregated repositories” does not represent AI tools *per se*. Rather, these are information sources that emerged during the identification of tools for the grey literature review and may be relevant inputs for the NEED database. As such, they will not be assessed within the present analysis.

As part of the documentation, a set of concise tool descriptions (approximately 50 words each) is provided in the appendix “list-of-tool-descriptions” (see section Appendix 2). These descriptions were generated to assist the reader in reading this document, which refers to many tools by name only. They also contain one or two URLs to the tool website or to an address where key information can be found. The descriptions were generated semi-automatically using Gemini’s Deep Search feature to gather information on tools within each category. Because Deep Search produces reports in a standard narrative structure (e.g., introduction, body, conclusion), a secondary prompt was used to reformat the content into concise 40–60-word summaries suitable for the appendix.



4.1.3 Step C: Filtering out inadequate tools

The identification process produced a total of 62 tools requiring examination. Assessing this volume manually, or even semi-automatically, on a tool-by-tool basis would be time-consuming and resource-intensive. In addition, when Gemini's underlying large language model is asked to evaluate many tools or criteria simultaneously, increasing task complexity can reduce the quality and reliability of the generated analysis.¹⁵ Experiments grading LLM-generated answers based on information dispersed across large prompts show that answer quality tends to decrease as prompt size increases.¹⁶ For this reason, asking Gemini to analyse all tools simultaneously was considered impractical. Therefore, clearly unsuitable tools were eliminated early in the process to allow a more focused assessment of the remaining tools.

This filtering step was implemented by instructing Gemini to conduct quick, focused searches to identify potential limitations or fatal flaws associated with each tool (the prompts used here are provided in the supplementary material). It is important to distinguish these exclusion motives from the formal assessment criteria defined in the next step of the methodology. Rather than evaluating tools against a fixed standardized scale, the AI was tasked with proactively searching for specific reasons why a tool might be unsuitable for the NEED project. These included critical technical or operational issues, such as a tool being discontinued, lacking a functional API, or explicitly prohibiting automated access, which would serve as immediate deal-breakers.

For each tool, the model generated a list of three such potential negative attributes. These served as red flags for human review: a human expert verified these AI-generated alerts to determine whether the tool should be retained or excluded. This step was efficient because, unlike the detailed assessment phase, the exclusion queries could be executed across all tools within a category simultaneously. The complete list of exclusion motives is provided in the appendix "list-of-exclusion-motives" (see Appendix 3), and the final set of tools retained for further assessment is presented in chapter 5.

4.1.4 Step D: Assessment criteria determination

To guide the examination of the identified tools, a set of assessment criteria was defined based on the NEED project requirements. The criteria were established through a multi-stage process combining internal expertise and collaborative discussions. An initial list was developed drawing on B12 Consulting's domain expertise, the requirements outlined in the public tender offer, and insights from the project's kick-off meeting. This process produced an initial list of 28 potential criteria, which was presented to the NEED team for validation and refinement. Following this review, the list was consolidated into a final set of 22 criteria. To reflect the priorities of the NEED project, each criterion was assigned a weight indicating its relative importance:

- High priority (weight 5): Criteria considered most important for the project (e.g., those relating to legal compliance, data provenance, and core AI functionality).
- Lower priority (weight 2): Criteria relating to ancillary functions or features (e.g., standard screening or deduplication capabilities).

The final set of 22 weighted criteria is grouped into four categories: Feature, Usability, Availability, and Project Adequacy (Table 2).

Table 2 – Weighted assessment criteria for examining the remaining tools

Criteria	Description	Weight
<i>Feature criteria</i>		
Source discovery	Is the tool able to discover new sources (websites/databases) or does it rely on a fixed list?	5
Source types	What types of sources does the tool search (e.g., HTA agencies, registries, collections of reports)?	5
Document retrieval	Can the tool retrieve or scrape documents (e.g., PDFs, web pages) from a given source?	2
Metadata extraction	Is the tool able to extract structured metadata (e.g., authors, publication date, DOI, source URL)?	5
Full-text extraction	Can the tool parse and extract full text from documents, including formats like HTML, DOCX, or OCR?	2
Data extraction	Is the tool able to extract specific data from the document body (e.g., PICO elements, outcomes, numbers)?	2
Deduplication	Can the tool identify and remove duplicate sources or documents?	2
Screening	Can the tool assist with or automate screening to discard irrelevant documents?	2
<i>Usability criteria</i>		
Configurability	Are searches/projects configurable, and can these configurations be saved as templates for reuse?	2
Scripting	Does the tool offer scripting capabilities (or an API) to automate repetitive tasks?	2
Integration	Can the tool integrate with other tools in a pipeline?	2
User interface	Does the tool have a user-friendly graphical interface for non-technical users?	2
Documentation	Does the tool have clear documentation, tutorials, or responsive user support?	2
<i>Availability criteria</i>		
Up-to-date	Is the tool still actively maintained and up-to-date?	5
Pricing model	Is the tool free? If not, what are the pricing options and feature limitations?	5
Open-source	Is the tool open-source?	2
Hosting	Can the tool be installed locally, or is it a web-based (SaaS) application?	2
<i>Project adequacy criteria</i>		
Transparency	Is the tool transparent and auditable (e.g., data provenance, logging search strategies)?	5
Validation	Is the tool well-established, validated, or cited in other methodological papers?	2
Health focus	Is the tool adequate for finding data in the health sector?	5
European focus	Is the tool adequate for finding data in the European Union?	5
Grey focus	Is the tool specifically designed for grey literature?	5

This table presents the criteria used to examine how the identified tools align with the NEED project requirements, together with their assigned weights.

4.1.5 Step E: Examination of remaining tools

The tools that passed the filtering phase (Step C) were analysed individually to examine their alignment with the 22 assessment criteria defined in the previous section (Table 2).

Internal testing revealed that asking the model to perform deep research and criterion-based scoring in a single step often led to hallucinations or inconsistent recommendations. Therefore, to structure the analysis and reduce these issues, the process was divided into three distinct operations:



information gathering, score extraction and formatting, and verification. All full prompts used in this step are available in the supplementary material.

First, Gemini was instructed to conduct a comprehensive Deep Search for each tool. The objective of this prompt was to gather factual evidence regarding the tool's capabilities, using the 22 assessment criteria as a research guide (see Example Prompt 1). The prompt also requested an indicative score for each criterion, together with a short justification (20–40 words), allowing the reasoning behind the generated value to remain visible.

Example Prompt 1 (Deep Search): "I want you to gather information on the following tool: [Tool Name]. I want you to assess the relevance of this tool based on the following list of criteria:

- Feature criteria: 1. SOURCE DISCOVERY /5; 2. SOURCE TYPES /5; [...]
- Usability criteria: 9. CONFIGURABILITY /2; [...]
- Availability criteria: 14. UP-TO-DATE /5; [...]
- Project criteria: 18. TRANSPARENCY /5; [...]

I want you to score the tool against every criterion. Report a list of paragraphs with the criterion keyword, score/maximum, and a 20-40 word comment on the tool's performance."

Second, using the detailed report generated in the previous step as context, a second prompt was submitted (see Example Prompt 2). This instruction asked the model to convert the qualitative findings into numerical values and a structured format (CSV) suitable for subsequent analysis.

Example Prompt 2 (Extraction): "Please use the results of your deep search to score the tool against every criterion of the list. [...] Conclude with the following: A one-row table (I insist, even if very wide) that includes the following columns: [Tool name, Total score, Feature score, ...]. A csv string with the tool name and all these scores, such as: 'Tool name, 56, 12, 5, ...' in the same order as in the table."

Third, a verification step was performed to ensure the model respected the weighting logic (e.g., ensuring a criterion weighted /2 was not scored higher than 2) (see Example Prompt 3).

Example Prompt 3 (Verification): "Please check the scores you assigned to each criterion: Did you respect the maximal score I gave for each criterion? If not, re-evaluate these scores."

The results were exported as structured CSV files and imported into Excel to generate tables summarizing the criterion scores assigned to each tool (see supplementary material 1 "table-all-scores.xlsx").

Large Language Models (LLMs) can exhibit sensitivity to context and prompt structure. Internal testing on the same tool revealed score variations of approximately 5 to 10 points across different runs (see supplementary material 2 "table-variability-of-scores"). The accompanying score justifications were therefore useful for interpreting the generated outputs and were used to manually review the results where necessary. The outcomes of this analysis are presented in chapter 5.

Human arbitration was applied when the AI-generated output showed clear inconsistencies with the supporting evidence collected during the Deep Search phase, internal incoherence across criteria, or obvious technical errors. In such cases, the human reviewer checked the score justification against the documented evidence and corrected the output where necessary to restore consistency. These interventions were targeted corrections of unsupported outputs rather than a full independent re-scoring of each tool.

It should also be noted that Gemini Deep Search was used to support the screening of candidate tools and the generation of the indicative scores. As Gemini is a commercial product and some evaluated tools belong to the same ecosystem, the possibility of unintended bias cannot be entirely excluded. AI-generated outputs were therefore reviewed and, where necessary, corrected through human

arbitration. The analysis was designed to explore capability coverage across tools rather than to provide a definitive comparative benchmark.

Prompt formulation followed a consistent template to maintain a similar structure across analyses; however, systematic robustness testing under alternative prompt formulations was not conducted. Given the known sensitivity of AI-powered tools to prompt design, such testing would be appropriate in a subsequent implementation phase focused on real-world deployment.

4.1.6 Step F: Synthesis of results and strategic interpretation

The final step of the methodology consisted of synthesizing the information generated during the previous phases in order to derive strategic insights relevant to the NEED project.

First, the criterion scores produced during the tool examination phase (Step E) were consolidated into summary tables providing an overview of how the remaining tools aligned with the assessment criteria and the core tasks of the NEED workflow (Steps 1–3 of the workflow). These numerical outputs were then interpreted in light of B12 Consulting's domain expertise and qualitative review. This interpretative step enabled the transition from the analytical objective of examining available tools (Objective 1) to the formulation of architectural considerations for the NEED workflow (Objective 2, see chapter 2).

Importantly, this analytical step was not intended to identify a single “best” tool. Rather, it aimed to determine whether an existing solution could support the full NEED workflow end to end. The numerical outputs generated during the analysis were therefore used primarily to support the interpretation of capability coverage across tools. Accordingly, these outputs should be interpreted as structured decision-support indicators rather than as definitive or strictly reproducible benchmarks. Given the exploratory nature of this market screening, the use of AI-assisted analysis, and the absence of fully anchored score levels, these outputs should be understood as indicative of capability coverage across tools at the time of analysis rather than as an absolute or statistically robust comparative evaluation.

No formal sensitivity analysis was conducted to assess the impact of alternative weight configurations on the resulting numerical outputs. As the central analytical objective was to determine whether a single all-in-one solution existed — rather than to establish fine-grained distinctions among closely performing tools — the robustness of small score variations under alternative weighting schemes was not considered critical at this stage. A stage-specific tool selection during implementation would warrant a more formalized weighting framework and sensitivity analysis.

5 OBJECTIVE 1: TOOL LANDSCAPE REVIEW AND ANALYSIS

This chapter presents a landscape review of different tools, classified **into five categories** to examine their potential utility across the discovery, exploration, and extraction phases.

- Category 1: Aggregated Repositories: Good for finding sources, but static and incomplete.
- Category 2: AI Search Tools: Semantic search can be effective (Exa.ai, Perplexity.ai), but workflow integration is challenging; many tools provide limited coverage of grey literature.
- Category 3: Document Retrieval Tools: Crawlers (Stormcrawler) are effective for scalable retrieval, but require engineering effort.
- Category 4: Traditional Review Accelerators: Can support systematic review tasks, but are not suitable for grey literature discovery.
- Category 5: Miscellaneous Tools: Some provide access to grey literature, but offer limited automation and scalability.



This section presents the results of the tooling landscape analysis, in which identified tools were classified into five functional categories to examine how they could support the “discovery, exploration, and retrieval” phases of the research workflow. In the following subsections, we detail the characteristics of each category, provide the rationale for specific tool exclusions, and summarize the results of the analysis for the retained tools.

The numerical outputs presented in this chapter should be interpreted as indicative decision-support elements. Given the exploratory nature of the analysis and the variability observed in AI-assisted scoring, small differences between tools should be read qualitatively rather than as meaningful performance gaps.

5.1 Category 1: Aggregated repositories

The first category, initially composed of 18 identified resources, consists of information repositories and searchable indices, such as databases, registries, or directories. These function as centralized locations for potential grey literature sources or documents, rather than as software tools capable of supporting or automating the review workflow.

During the filtering phase (Step C), two resources were explicitly flagged for exclusion due to operational limitations:

- [Grey Literature Reporter](#). This resource, produced by the New York Academy of Medicine (NYAM), was a curated service consisting of a searchable database and a periodic publication that indexed public-health grey literature from 1999. However, it was discontinued in 2017 and is therefore not up to date anymore, which is the reason it was filtered out of the list.
- [re3data](#). This resource is a repository of raw research datasets (e.g., genomic data). It was excluded from further analysis as it does not contain the textual reports and policy documents required for the NEED database.

The remaining 16 resources can be divided into two distinct sub-groups relevant to the scope of the NEED project:

- **Directories of sources.** This sub-category comprises resources such as [Grey Matters](#), [OpenDOAR](#) (Directory of Open Access Repositories), and the [GreyNet International](#) databases ([PUBGREY](#), the [GreyGuide repository](#), and the [GreySource index](#)). These act as catalogs for identifying and locating other grey literature repositories. While valuable for Discovery (Step 1, see Figure 1), they are primarily static lists that are updated and maintained manually.
- **Document databases.** This sub-category includes large-scale repositories that host or index documents directly. This ranges from broad search engines like [Google Scholar](#) to highly specialized registries like [CTIS](#) (the European Clinical Trials Information System). These are valuable target locations for the search and retrieval steps (Steps 2 and 3, see Figure 1), but they do not constitute independent tools for automating the review process.

As these resources serve as potential inputs for the search workflow rather than the engines that drive it, we did not score them against the functional assessment criteria used for the other categories. Instead, they were considered as potential source environments to be explored, validated, and, where relevant, integrated into the broader workflow.

5.2 Category 2: AI search tools

This category, initially comprising 10 tools, represents the main AI-based candidates capable of supporting source discovery (Step 1, Figure 1). Unlike traditional search engines that rely primarily on exact keyword matching, these tools use semantic search and LLMs to interpret the intent behind a query. In principle, they are designed to identify relevant websites or documents across the web based on natural-language descriptions (e.g., “Find recent HTA reports on melanoma in Europe”).

During the filtering phase (Step C, Figure 2), seven tools from this category were excluded. Two main types of exclusion motives were identified:

- [Undermind.ai](#), [Keenious](#), [Consensus](#), [Scispace](#), [Elicit](#), and [Sourcely](#) were excluded because of their limited relevance for grey literature retrieval. Additional review showed that, although these tools present themselves as "AI research assistants", they do not meaningfully crawl the open web or agency websites. Instead, they function primarily as advanced interfaces built on centralized academic databases such as [Semantic Scholar](#) or [OpenAlex](#). As a result, they are well suited to identifying peer-reviewed articles, preprints, and theses, but provide limited coverage of the decentralized government reports and policy documents required for this study.
- [Amplifyfi](#) was also excluded because of its scope. Its relevance algorithms appear to prioritize commercial market intelligence rather than the types of information required for NEED. Furthermore, it operates largely as a black box, meaning that its proprietary crawling logic cannot easily be audited for search sensitivity or specificity, which is not well aligned with the transparency requirements of a systematic review process.

The remaining three tools, Exa.ai, Perplexity.ai, and Tavily, are genuine web-search agents capable of indexing the broader internet. Among them, Exa.ai and Perplexity.ai appeared to offer the broadest coverage of capabilities relevant to the NEED workflow, particularly for semantic search tasks related to source discovery. Tavily appeared more limited in this context. This likely reflects its orientation toward general real-time web retrieval, such as news, blogs, and general web content, for use in generic AI assistant applications. It does not appear optimized to prioritize European institutional sources or complex health-policy documents over mainstream web content. Furthermore, it is primarily designed to parse HTML content and does not natively support the reading or retrieval of PDF reports, which constitute a substantial share of grey literature.

Exa.ai and Perplexity.ai appear to be promising tools for supporting the discovery of grey literature sources. However, integrating them into an automated workflow would still require substantial custom development, as neither tool natively supports the full sequence of discovery, exploration, and retrieval in a way that would allow straightforward export of documents into downstream review tools such as citation managers. In practice, this means that additional components or custom connectors would still be needed to build a cohesive end-to-end solution.

General-purpose AI tools like Gemini or ChatGPT were not included in the initial pool of candidates because the initial search focused on more specialized, niche tools explicitly positioned as supporting grey-literature review workflows. Nevertheless, these general-purpose tools fit conceptually within this category and may be considered as alternative options to the more specialized tools discussed here.

5.3 Category 3: Document retrieval tools

This category, initially comprises seven tools, groups software designed to support the retrieval phase (Step 3, Figure 1). Unlike source-discovery tools, which help identify where to search, these tools are designed to operate on a predefined, validated list of websites to systematically retrieve content.

During the filtering phase (Step C), two tools were excluded due to operational constraints:

- [Zenodo_get / OSFclient](#). This was excluded because it is hard-coded to search specific repositories (Zenodo and OSF) and cannot be configured to retrieve documents from external agency websites.
- [Unpaywall](#). This tool was excluded because it relies specifically on Digital Object Identifiers (DOIs) to locate content. Unpaywall operates by scanning webpages for a DOI and matching it against the [Crossref database](#), which links DOIs to open-access versions of papers. Consequently, if a document does not have a DOI registered in Crossref, Unpaywall effectively behaves as if it does not exist. Unlike journal articles, which routinely receive DOIs from commercial publishers, grey literature is produced by a diverse range of organizations that often do not register DOIs or deposit their documents in DOI-enabled repositories. Although recent initiatives have sought to promote



more systematic DOI assignment for reports and non-traditional outputs, the majority of historical and current grey literature (e.g., agency reports, government PDFs, policy briefs) still relies solely on standard URLs.¹⁷ Therefore, this tool would likely return very limited results for the NEED database.

The remaining tools (including [Crawl4AI](#), [StormCrawler](#), [Scrapy](#), [ScrapeGraphAI](#), and [Firecrawl](#)) are generic web crawlers and scrapers. A web crawler (or "spider") is a program that navigates the web by following links to identify URLs (e.g., mapping all pages within a website), whereas a web scraper is a script designed to extract specific data (text, PDFs, tables) from those pages once found. In practice, most modern tools combine both functions. These tools were examined against the assessment criteria and the corresponding numerical outputs are reported in appendix "table-all-scores".

Overall, these tools showed broad capability coverage across many dimensions. However, the analysis also revealed a discrepancy in the "Health Focus" criterion that illustrates one limitation of the AI-assisted scoring approach. Indeed, Crawl4AI received a low score (1/5) while StormCrawler was assigned the maximal score of 5 for this criterion. Gemini's justification indicated that its Deep Search identified a specific published study in which StormCrawler had been used to index health portals.¹⁸ On that basis, StormCrawler was treated as highly relevant to the health sector, while Crawl4AI, for which no comparable example was identified, was treated as less relevant. In practice, however, both tools function in a broadly similar way: they do not interpret or understand content but simply retrieve it. Therefore, project-specific criteria such as "Health Focus" are of limited relevance for evaluating generic retrieval frameworks, and the crawlers and scrapers included here should be considered broadly comparable on that dimension.

Interestingly, these tools also appeared to perform well on the "Source Discovery" criterion. They can expand an initial list of sources by recursively extracting and following hyperlinks. For example, StormCrawler was used in one study to expand a seed list of 22,000 URLs into a dataset of more than 230,000 unique health-related websites.¹⁸ However, this capability depends heavily on the structure of the link graph: a crawler can only discover new sources that are connected via hyperlinks to the initial seed list. If the seed list is biased (e.g., geographically limited), the crawler may fail to reach relevant but isolated sources in other regions. Nonetheless, this remains a valuable capability for expanding an initial set of identified sources.

It is important to note that these tools are developer frameworks rather than off-the-shelf solutions. Their integration into an automated workflow for research-data retrieval would therefore require substantial programming expertise and technical implementation effort.

5.4 Category 4: Traditional review accelerators

This category, comprising 20 identified tools (such as Covidence, Rayyan, and DistillerSR), groups software designed to support the standard downstream phases of a systematic review: deduplication, screening, data extraction, and synthesis (Steps 4–9, Figure 1).

Unlike the tools in the previous categories, these software solutions were not examined individually against the assessment criteria. As defined in the research objectives, the main bottleneck for the NEED database lies in the initial discovery and retrieval of decentralized grey literature (Steps 1–3, Figure 1). Traditional review tools typically operate after this phase, as they require a corpus of documents to be imported before they can function and offer limited or no capabilities for autonomously identifying new sources on the web.

Although these tools were not examined further, a preliminary review was conducted to generate concise descriptions and identify key limitations, particularly their lack of web-crawling or source-discovery features. This information is reported in the appendices "list-of-tool-descriptions" and "list-of-tool-exclusion-motives" (Appendix 3) and is included to provide a broader overview of the review-management software landscape.

5.5 Category 5: Miscellaneous tools

This category, initially comprising seven tools, groups specialized utilities that support specific aspects of the search workflow but do not fit the standard definition of a repository or a full-stack search engine. These tools typically address niche tasks such as keyword generation, query translation, or specific metasearch functions.

During the filtering phase (Step C), four tools were excluded or set aside:

- [gdelt doc](#) / [Mediacloud](#). These tools were excluded due to a mismatch in scope. They are designed primarily for monitoring journalism and news ecosystems rather than scientific or policy literature. Furthermore, they rely on sentiment-analysis algorithms, which automatically classify text as positive or negative in order to track public opinion. This creates a potential source of distortion for the present study, as objective health reports (e.g., on mortality rates or disease burden) could be algorithmically down-ranked or misinterpreted as negative content.
- [Litsearchr](#) and [Polyglot](#). These were excluded from the criterion-based examination because they are not search tools in themselves. However, they remain methodologically useful for specific tasks. Litsearchr assists researchers in discovering and generating keywords based on word frequencies in a reference set of papers, while Polyglot translates search strings into the appropriate syntax for different databases.

The remaining tools were examined against the assessment criteria (see Table 2). The corresponding numerical outputs are reported in the supplementary material “table-all-scores”.

The analysis revealed that both Greylitsearcher and Grey Literature Reporter extension are effectively no longer maintained. They were originally developed for an evidence-synthesis hackathon but do not appear to have been updated since the event, making them unstable and unsuitable for a long-term institutional workflow.

This left two functional tools in this category:

- [MedNar](#). This is a deep web search engine designed to cover some of the blind spots of standard search engines. It can search multiple grey-literature sources simultaneously and export results in formats compatible with traditional review software. However, it appeared limited on the scalability criterion because it is primarily designed for manual, interactive use, and its internal search logic is relatively opaque (“black box”), limiting the level of transparency required for the NEED protocol.
- [Google Programmable Search Engine \(PSE\)](#). Unlike MedNar, this is better understood as a configurable search platform rather than a ready-to-use tool. It allows developers to build custom search engines, typically by providing a predefined list of websites to index. This can be useful for retrieving certain types of documents (e.g. PDFs from a known source), as the search can be restricted to specific file types and domains. However, the tool can only retrieve content that has been indexed by Google crawlers and cannot access parts of the deep-web, such as content hidden behind searchable interfaces, user interactions, or login barriers. Furthermore, relevant search terms must still be provided as input in the search query. Some manual effort is therefore required to define and refine these queries, unless an additional tool is used to support keyword generation. Another limitation is that Google PSE relies on a more restricted index than standard Google Search¹⁹, which makes it more likely to miss niche content.



5.6 Recommendations and retained candidate tools for modular integration

The category-based analysis (Table 3) indicates that no single tool can reliably support the full NEED grey-literature workflow end-to-end (Step 1: Discovery → Step 2: Exploration → Step 3: Retrieval) while meeting expectations for scalability, transparency, and integration. As a result, **the recommended approach is to retain a set of complementary tools that can be combined in a modular workflow, with a clear separation between (i) source discovery and (ii) retrieval/extraction.**

Table 3 – Summary of the 5 tool categories for grey literature review

Category	Definition / Purpose	Included Tools / Sub-categories
Aggregated repositories	Places or platforms where users can find grey literature sources or documents.	Lists of grey sources: Grey Matters; OpenDOAR; PUBGREY; GreyGuide; GreySource. Databases containing grey literature: Google Scholar; CTIS.
AI search tools	Tools designed (in theory) to autonomously identify relevant websites/databases for grey literature using semantic search.	Exa.ai; Perplexity.ai; Tavily.
Document retrieval tools	Crawlers and scrapers that retrieve documents from a pre-defined list of websites.	Stormcrawler; Crawl4AI; other crawlers/scrapers.
Traditional review accelerators	Tools that support screening and extraction in classic systematic reviews.	E.g., Covidence, Rayyan, DistillerSR. (Multiple tools not individually analysed.)
Miscellaneous tools	Diverse tools that support specific aspects of the search workflow (e.g., semantic search, metasearch, or keyword generation).	Evaluated Tools: MedNar; Google Programmable Search Engine (PSE). (Methodologically useful, not evaluated: Litsearchr; Polyglot.)

5.6.1 Candidate tools retained for Step 1: Discovery module

To support the identification of new potential sources (websites, portals, repositories) and rapidly scope grey literature beyond static lists, three AI search tools were retained for deeper analysis:

- **Exa.ai:** prioritized for its semantic “agent-first” discovery and structured outputs that can be integrated downstream via API.
- **Perplexity.ai:** retained for fast, real-time web retrieval and citation-backed narrative synthesis that can be valuable for expert scoping and triangulation.
- **Tavily:** retained as a pragmatic “retrieval companion” for producing clean extracted webpage text optimized for LLM/RAG pipelines, complementing discovery-oriented outputs.

These three tools represent distinct trade-offs (structured vs. narrative outputs, reproducibility limits, metadata depth, automation fit). Their respective strengths and constraints are discussed below.

This discovery module focuses on identifying grey-literature sources through AI-powered web searching. The following section describes the capabilities, limitations, and operational requirements of the three retained tools: **Exa.ai**, **Perplexity.ai** and **Tavily**. A detailed review of the tools examined is provided in the appendix. A summary table of the retained tools is also available in Table 4.

Table 4 – Comparison of the three retained AI search tools

Feature/Constraint	Exa.ai	Perplexity.ai	Tavily
Main strength	Semantic depth: finds relevant grey literature by meaning, even when standard terminology is not used	Narrative synthesis: Provides immediate, cited answers for rapid scoping and human-in-the-loop verification	Process automation: Consolidates searching, scraping, and content cleaning.
Main limitation	Reproducibility: shifting AI results make searches hard to repeat exactly.	Reproducibility: dynamic nature makes it difficult to replicate exact search histories	Metadata richness: basic bibliographic data is often missing
Search mechanism	Semantic discovery based on embeddings and meaning	Dynamic real-time web retrieval with narrative synthesis	Real-time web search designed to support RAG workflows
Primary output	Structured list of URLs and metadata	Citation-backed narrative responses	Clean extracted webpage text optimized for AI analysis
Source and document scope	Heterogeneous web sources including organisational websites and other online content	Web-based sources surfaced through search retrieval	Web page (HTML) content retrieved from the live web
Metadata extraction	Supports structured metadata (e.g. source information, publication date)	Limited, with emphasis on narrative output and citations	Minimal, typically limited to title, URL and extracted text
Reproducibility & auditability	Limited due to undisclosed index scope and ranking logic	Limited due to lack of exportable audit trails and search histories	Limited due to limited public documentation of indexing and ranking processes
Health or regional filtering	No dedicated health filters; Regional filtering can be set by a parameter	No dedicated health filters; Regional filtering can be set by a parameter	No dedicated health filters; Regional filtering can be set by a parameter
Operational & licencing	Usage-based API; Subscription tiers for advanced features (Websets).	Freemium/Pro subscription; Enterprise plans for higher throughput.	Credit-based API; Free tier available for small-scale testing.

While all three tools utilize LLMs to enable their semantic search capabilities, they leverage this technology differently:

- Generative and conversational AI tools like **Perplexity.ai** function as conversational "answer engines". They use LLMs to not only find information but also to synthesize it into cited, narrative responses, making them ideal for rapid scoping and human-in-the-loop verification.
- Agentic search tools such as **Exa.ai** and **Tavily** are optimized for "agentic" research. They use LLMs to understand the deeper meaning behind a query, moving beyond mere keyword matching, to retrieve structured lists of URLs and clean webpage text for further analysis and integration into automated pipelines.



5.6.1.1 *Exa.ai*

Exa.ai is a web search engine designed for use within AI driven workflows. The tool facilitates semantic retrieval of web-based content and provides structured results that support automated research, analysis, and the generation of summaries.

A. Strengths

The tool's primary strength lies in its semantic discovery capabilities, which allow it to search for the meaning and intent of a query rather than just matching specific words. This is particularly valuable for identifying diverse grey literature when terminology varies across documents, as it can find relevant reports even if they use different language to describe the same topic. Exa.ai supports discovery across heterogeneous web sources, including organisational websites, policy documents, and reports, offering a broader discovery approach than traditional keyword-based bibliographic databases. Additionally, the tool can prioritize results from specific countries, making it easier to find local government reports and national health data.

Exa.ai supports structured data extraction through its API, including metadata such as publication dates and source information. This supports traceability and transparent reporting of search outputs. Its Websets functionality allows users to group and monitor collections of sources over time, supporting ongoing discovery as new material becomes available.

B. Limitations

The primary limitations of Exa.ai relate to transparency and domain specificity. The tool does not publicly disclose the scope or composition of its indexed sources. As a result, it is difficult to fully assess coverage completeness or to replicate searches with confidence.

Exa.ai is not designed specifically for health research. It does not employ controlled vocabularies such as MeSH or Emtree and does not provide study design filters. Precision for health-focused searches therefore depends on expert judgment during screening and review.

C. Operational and licensing considerations

Exa.ai is delivered as a cloud-hosted service accessed through a usage-based API, where search requests, content retrieval, and structured extraction are billed per operation. Subscription tiers provide monthly credit allocations to support high-volume use and grant access to advanced features such as Websets for organizing persistent source collections. From an operational perspective, these API-driven workflows require some technical expertise and may limit accessibility for users without programming experience but offer a standard interface which simplifies the integration with a custom solution. A network access outside the local institution network is essential to use the service, which may present a challenge for institutions operating within restrictive firewall or data-governance environments.

5.6.1.2 *Perplexity.ai*

Perplexity.ai is a conversational "answer engine" that provides direct, written responses to questions instead of just showing a long list of website links. It works like a combination of a traditional search engine and a chatbot, scanning the live internet in real time to summarize the most relevant information into a single, easy-to-read answer.

A. Strengths

The primary strength of Perplexity.ai is its dynamic source discovery. It performs near real-time web retrieval and constructs search queries programmatically, enabling the identification of recent and

emerging web-based literature beyond static databases. In addition, the tool can prioritize results from specific countries, making it easier to find local government reports and national health data.

Perplexity.ai produces citation-backed narrative synthesis, delivering responses with visible source citations. This supports rapid scoping while allowing experts to verify underlying sources. The platform offers multiple search modes that can broaden discovery, including options designed to emphasize academic or in-depth retrieval.

B. Limitations

The primary limitations of Perplexity.ai relate to source transparency and process reproducibility. The platform does not publicly specify the full set of websites or repositories included in its search coverage, limiting the ability to assess retrieval scope.

The search and retrieval process are not fully reproducible in a systematic review sense. Perplexity.ai does not provide exportable audit trails, full search histories, or documented multi-step search strategies that would allow searches to be comprehensively recorded and repeated. Outputs are primarily narrative rather than structured, which limits consistent metadata extraction and complicates large-scale deduplication.

While the API exposes some filtering and retrieval parameters, these do not equate to the controlled vocabularies, study design filters, or fully transparent search strategies used in traditional bibliographic databases. As a result, expert oversight remains necessary to evaluate relevance and methodological suitability. In addition, the platform operates within an evolving legal and technical environment related to web access and content availability, which may affect long-term stability of retrieved sources.

C. Operational and licensing considerations

Perplexity.ai utilizes a freemium and subscription-based model where paid tiers provide higher usage limits and access to advanced features. Programmatic integration is supported through a usage-based API with costs determined by the selected model and token volume. This API offers granular controls, such as result limits and domain-level filtering, to facilitate integration into automated or agentic workflows. Because it is a cloud-hosted service, Perplexity requires outbound network connectivity. This requirement may necessitate coordination with institutional IT departments to address potential challenges related to restrictive firewalls or data-governance policies.

5.6.1.3 Tavily

Tavily is a search-as-a-service API designed to handle the information-gathering needs of AI-powered research. It automates the process of searching, reading, and cleaning web data in a single step, returning information in a format that AI models can immediately understand. It prioritizes factual sources and real-time data, functioning as a bridge between static AI models and the live web.

A. Strengths

The primary strength of Tavily lies in its dynamic web discovery. The tool performs real-time web search and can surface a wide range of online sources, including organisational websites, government pages, and news outlets that may not be captured by static databases.

Tavily provides clean extraction of webpage text, returning readable content with extraneous page elements removed. This supports efficient downstream analysis and integration into LLM-based workflows. Additionally, Tavily now supports geographic prioritization, allowing users to boost results from specific countries to better identify local government reports.



B. Limitations

The primary limitations of Tavily relate to transparency and metadata richness. While the tool can discover various file types, the metadata provided is often limited to basic fields such as the title and URL. Critical bibliographic details, such as specific publication dates, formal authorship, or peer-review status, are not consistently available.

Furthermore, Tavily does not publicly document the full scope of its indexed sources or its internal ranking mechanisms, which can limit the transparency and reproducibility required for some formal reviews. The tool also does not provide traditional controlled vocabularies (like medical subject headings) or study design filters, meaning expert oversight is still required to evaluate the methodological quality of the results.

C. Operational and licensing considerations

Tavily operates under a credit-based API model where a free tier provides limited monthly access, and paid plans increase operational capacity for high-volume retrieval. The API is designed for automation-first use and integrates into agentic and RAG pipelines. Because the platform is primarily accessed through code, its use requires a certain level of technical capacity. Additionally, as a cloud-based service, Tavily requires outbound network access to external servers. This may result in implementation constraints for institutions governed by strict data-governance requirements or restrictive firewalls.

5.6.2 Candidate tools retained for Step 2 and 3: Exploration and extraction module

For systematic exploration of validated sources and automated retrieval of documents (HTML/PDF and, where supported, Office formats), **five frameworks** were retained for deeper analysis because they span the main deployment and capability profiles needed for the NEED context:

- **StormCrawler**: well suited for environments where auditability (the ability to see exactly what was crawled, when, and how through detailed logs), provenance, and large-scale crawling are priorities (self-hosted, engineering-heavy).
- **Crawl4AI**: strong for modern/dynamic crawling and clean text extraction (removing menus, ads, and other page elements that are not part of the main text), with flexible deployment (open source).
- **Firecrawl**: attractive for quicker operationalization (API-first, interface options), with trade-offs depending on managed vs. self-hosted use.
- **ScrapeGraphAI**: included for AI-assisted extraction of specific information into structured fields from known websites (helps reliably extract data even when websites change their layout or phrasing).
- **Scrapy**: retained as a robust, transparent baseline framework for rule-based crawling where maximum control is required (but requires engineering and add-ons for document text extraction).

These five tools are discussed as a set in the report and reviewed in detail below.

This exploration and retrieval module is responsible for systematically navigating a predefined set of sources (Exploration) and downloading associated content (Extraction/retrieval). This section describes the capabilities, limitations, and operational requirements of five retained frameworks. A detailed review of the tools examined is provided in the Appendix 2.

Table 5 – Comparison of the five retained crawling and scraping tools

Feature/constraint	Stormcrawler	Crawl4AI	Firecrawl	ScrapeGraphAI	Scrapy
Primary role	Large-scale systematic web exploration and document retrieval	Dynamic web crawling and text extraction	Automated crawling and extraction for RAG workflows	AI-assisted extraction from predefined sources	Rule-based web crawling and data extraction
Exploration approach	Link following and sitemap-based discovery	Adaptive exploration with rule based prioritisation	Search-based discovery and deep site crawling	Exploration of predefined sources	Rule-based link following within defined domains
Supported content types	HTML pages; PDFs; Office documents	HTML pages; limited PDF support	HTML pages; PDFs; Word documents	HTML pages; PDFs; Word documents	HTML pages; file download support (e.g. PDFs)
Text extraction	Full-text extraction with document conversion	Clean text extraction optimised for analysis	Structured text with preserved document layout	Semantic extraction into structured fields	Rule-based extraction from HTML (external tools required for PDFs)
Metadata	Structured metadata (e.g. source, publication date)	Crawl-time and document-level metadata	Structured metadata where available	AI-derived metadata and extracted fields	Explicitly defined metadata fields via rules
Deduplication & filtering	Built-in deduplication and relevance filtering	Built-in deduplication and relevance filtering	Built-in deduplication and relevance filtering	Relevance filtering and deduplication	Built-in deduplication and relevance filtering
Reproducibility & auditability	High; detailed logging and archive support	Moderate; dependent on configuration	Moderate; varies by deployment mode	Moderate; dependent on validation of AI extraction	High; explicit crawl logic and detailed logs

5.6.2.1 StormCrawler

StormCrawler is a self-hosted web crawling framework designed for large-scale, systematic exploration of web sources and associated documents. It continuously discovers new sources by following links and sitemaps and is particularly well suited for comprehensive and auditable collection of web-based grey literature.

A. Strengths

StormCrawler provides comprehensive source discovery by systematically identifying new web sources through link following and sitemap parsing. It supports a wide range of source types commonly encountered in grey literature, including standard web pages as well as document formats such as PDF and Microsoft Office files. This enables consistent inclusion of reports, guidelines, and policy documents alongside HTML content.

The tool offers robust document retrieval, full-text extraction, and metadata capture. Retrieved content is converted into searchable text, and key metadata such as publication dates and source information



are recorded to support structured datasets and downstream screening. StormCrawler also includes mechanisms for deduplication and relevance filtering, allowing duplicate sources to be managed and scope to be controlled during exploration and extraction. A key strength is its high level of transparency and auditability, with detailed logging and support for standard web archive formats enabling full traceability of retrieved content and reproducibility of search strategies.

B. Limitations

StormCrawler requires substantial technical expertise to operate. It is designed as a developer-oriented framework and does not provide a graphical interface for non-technical users. Configuration, monitoring, and maintenance typically require dedicated technical support.

The tool is domain agnostic and does not include built-in health-specific vocabularies or predefined screening criteria. Domain relevance must therefore be implemented through custom rules or downstream analysis.

C. Operational and licensing considerations

StormCrawler is open source and free to use under the Apache License, Version 2.0. It must be self-hosted on local or institutional infrastructure, which increases deployment and maintenance requirements but allows full control over data handling. Its self-hosted nature and strong support for provenance tracking make it suitable for research environments with strict requirements for transparency, governance, and reproducibility.

5.6.2.2 Crawl4AI

Crawl4AI is an open-source web crawling and extraction tool designed to collect and structure content from websites, including those with dynamic or interactive elements. It focuses on transforming web-based sources into clean, analysis-ready text formats suitable for downstream processing in automated research and data pipelines.

A. Strengths

Crawl4AI supports dynamic and adaptive source exploration. It can discover new sources during crawling based on relevance rather than fixed source lists, allowing efficient coverage of large and evolving web spaces. The tool is well suited to modern web environments and can retrieve content from websites that rely on dynamic loading, interactive elements, or continuous scrolling. Retrieved content is converted into clean, structured text that reduces non-essential elements and supports efficient downstream analysis.

The framework supports structured metadata and data extraction, capturing crawl-time and document-level metadata and enabling extraction of specific data elements into structured formats. It includes rule-based filtering and deduplication mechanisms to manage duplicate content and focus collection on pertinent material. A key strength is its flexible deployment model, which allows the tool to be deployed locally or on institutional infrastructure, supporting governance- and privacy-aligned research workflows.

B. Limitations

Crawl4AI requires technical expertise to operate. It is primarily used as a programming library and does not provide a graphical interface for non-technical users. Configuration, execution, and monitoring are performed through code, which may limit accessibility in non-technical research teams.

Crawl4AI is primarily HTML-first. PDFs are supported depending on configuration, but Office formats are not native. While PDF content can be retrieved and processed, other common grey literature formats such as Word or PowerPoint documents require additional tooling and are not handled through

a single, unified pipeline. In addition, large-scale use can introduce operational overhead related to infrastructure and resource management.

The framework is domain agnostic and does not include health-specific vocabularies, controlled terminology, or predefined screening criteria. It has not yet been widely validated in formal health evidence synthesis workflows, and its use in such contexts requires careful oversight.

C. Operational and licensing considerations

Crawl4AI is open source and released under the Apache License, Version 2.0. It is free to use and can be deployed locally or on institutional servers, providing full control over data and infrastructure. Effective use in large-scale or automated research pipelines typically requires technical expertise and ongoing maintenance, particularly when handling dynamic websites or high crawl volumes.

5.6.2.3 Firecrawl

Firecrawl is a web crawling and extraction tool designed to systematically collect and structure content from websites and associated documents. It supports both automated source discovery and deep crawling of websites, with outputs optimised for downstream analysis and use in retrieval-augmented generation workflows.

A. Strengths

Firecrawl supports dynamic and automated source discovery. It can identify new sources through search-based queries and systematically enumerate content within websites, enabling broad exploration beyond predefined source lists. The tool is well suited to modern web environments and can retrieve content from dynamic websites and single-page applications that are not accessible to static crawlers.

The framework converts retrieved content into clean, structured text formats, preserving document structure while removing non-content elements. It supports common grey literature formats such as PDF and Word documents and enables extraction of structured metadata and specific data elements from unstructured text. Firecrawl includes configurable crawling and filtering options, as well as built-in deduplication, supporting focused and efficient collection of relevant material. A further strength is the availability of a user-friendly web interface, which allows non-technical users to explore sources, test extraction settings, and review outputs without writing code.

B. Limitations

The primary limitations of Firecrawl relate to dependency and control. While the core codebase is open source under the AGPL 3.0 license, some advanced features and optimisations, including enhanced handling of blocking and large-scale crawling, are primarily available through the managed cloud service. As a result, self-hosted deployments may offer more limited functionality without additional configuration.

Use of the managed service is subject to a usage-based pricing model, which may introduce cost constraints for high-volume or long-running projects. Self-hosting enables full data control but increases operational responsibility, as users must manage deployment, maintenance, and supporting services to ensure reliable crawling.

C. Operational and licensing considerations

Firecrawl is available both as a managed cloud service and as a self-hosted deployment. The core codebase is released under the AGPL 3.0 license, which requires sharing of modifications when deployed as a network service. Self-hosting is free to use but requires technical expertise and ongoing maintenance.

The managed cloud service operates under a freemium and subscription-based pricing model and provides an API-first interface with official client libraries, supporting integration into automated research pipelines. The self-hosted option is particularly relevant for organisations with strict data

protection, security, or compliance requirements, as it allows data processing to remain within institutional infrastructure.

5.6.2.4 *ScrapeGraphAI*

ScrapeGraphAI is an open-source tool designed to support AI-assisted exploration and extraction of content from predefined web-based and document-based sources. It focuses on transforming unstructured webpages and documents into structured, analysis-ready data through semantic extraction workflows.

A. Strengths

ScrapeGraphAI supports flexible exploration of predefined sources, including websites and common document formats such as PDFs and Word documents. It is capable of navigating complex and heterogeneous content, enabling systematic examination of web-based grey literature and document repositories once relevant sources have been identified.

A key strength of the tool lies in its semantic extraction capabilities. ScrapeGraphAI uses AI-based reasoning to extract metadata and specific data elements from unstructured text and map them into predefined, structured formats. This enables detailed extraction of complex information and supports consistent downstream analysis. The framework also supports relevance-based filtering and deduplication within its workflows, helping to focus outputs on information aligned with the research objective.

ScrapeGraphAI can be deployed locally or on institutional infrastructure, allowing data processing to remain within controlled environments. This supports research workflows with strict requirements for data governance, privacy, or data sovereignty.

B. Limitations

The primary limitations of ScrapeGraphAI relate to its reliance on AI-based extraction. Use of LLMs may introduce operational costs when commercial services are used and requires careful validation of extracted outputs to mitigate the risk of incorrect or incomplete information.

Although the tool reduces reliance on rigid extraction rules, effective use still requires technical expertise to configure and manage extraction workflows. Performance may be lower than traditional rule-based tools for very large-scale crawling tasks, and support for some document formats may require additional preprocessing rather than being handled through a single, unified pipeline.

C. Operational and licensing considerations

ScrapeGraphAI is open source and released under the MIT License. It is free to use and can be deployed locally or on institutional infrastructure, providing full control over data handling and configuration. Advanced use cases depend on access to AI models, either through commercial APIs with usage-based costs or through locally hosted models that require additional infrastructure.

The tool does not natively manage large-scale network or access constraints, which may need to be addressed through external tooling. Despite these requirements, its ability to operate entirely within local environments makes ScrapeGraphAI suitable for research settings with strict privacy, compliance, or data sovereignty requirements.

5.6.2.5 *Scrapy*

Scrapy is an open-source web scraping framework designed for systematic exploration and extraction of content from web-based sources. It is widely used for large-scale collection of structured information from websites and is particularly suited for harvesting content from complex or fragmented web domains, including those commonly associated with grey literature.

A. Strengths

Scrapy supports comprehensive exploration of web sources through rule-based navigation. It can systematically follow links and website structures to identify and retrieve content across entire domains, enabling thorough coverage beyond manually specified pages. This makes it effective for exploring institutional, governmental, or organisational websites where relevant material is distributed across multiple sections.

The framework supports extraction of structured data from a wide range of web sources, including standard webpages and web-based data services. It allows precise capture of metadata and content fields using predefined extraction rules, supporting consistent datasets suitable for downstream screening and analysis. Scrapy also includes built-in mechanisms for deduplication, ensuring that duplicate pages or documents are not repeatedly processed.

Scrapy is highly transparent and configurable. Crawling logic is explicitly defined through configuration and extraction rules, enabling traceable and repeatable execution under stable web conditions. Crawling logic is explicitly defined through configuration and extraction rules, and detailed logging of requests and responses enables full traceability of retrieved content. This supports auditable data collection workflows and reproducibility of search strategies, which are essential in research contexts.

B. Limitations

Scrapy requires substantial technical expertise to configure and operate. It is designed as a developer-oriented framework and does not provide a graphical user interface for non-technical users. All crawling and extraction logic must be implemented through code, which may limit accessibility for research teams without programming support.

While Scrapy can download document files such as PDFs and Word documents, it does not natively extract full text from these formats. Additional tools or libraries are required to process the contents of binary documents. The framework is also domain agnostic and does not include built-in health-specific vocabularies or predefined screening criteria, meaning domain relevance must be addressed through custom rules or downstream processing.

C. Operational and licensing considerations

Scrapy is open source and released under the BSD 3-Clause License. It is free to use and can be run locally or deployed on institutional servers. Its self-hosted nature provides full control over data handling and supports compliance with data governance and privacy requirements.

While Scrapy can be scaled for large collection tasks, doing so typically requires additional infrastructure and operational support. Commercial hosting options exist but are not required. Overall, Scrapy is well suited for research environments that prioritise transparency, flexibility, and reproducibility, provided that sufficient technical expertise is available.



6 OBJECTIVE 2: STRATEGY FOR APPLYING AI TOOLS WITHIN THE NEED WORKFLOW

Section 6 describes **the instrumentalization framework**: a hybrid strategy in which a custom orchestration layer coordinates specialized external tools as interchangeable modules while retaining institutional control over protocols, validation decisions, provenance, and the evidence record.

- **Modular orchestration strategy**: Since no single off-the-shelf tool covers the full workflow, the framework coordinates specialized AI modules through a custom orchestration layer to ensure institutional autonomy, traceability, and controlled re-execution of workflow steps.
- **Three-stage technical workflow**: The system aims to reduce manual bottlenecks through: (1) semantic discovery of sources, (2) systematic extraction of documents, and (3) standardized handoff to review software.
- **System memory**: Centralized repositories store search protocols and validation decisions to prevent redundant effort and enable traceable evidence updates across future searches.
- **Human-in-the-loop**: To maintain scientific rigor, AI outputs require mandatory human expert approval before being integrated into the formal evidence base.

The analysis of the current AI tooling landscape (chapter 5) indicates that no single, off-the-shelf solution can manage the full grey-literature workflow while also meeting the NEED project's requirements in terms of scope, scalability, and transparency. Instead, covering the workflow would require combining several existing independent tools. However, relying on human experts to manually transfer information between these tools would undermine efficiency and introduce significant risks to standardization and repeatability. Conversely, building a fully custom end-to-end system could address these requirements, but would involve longer development timelines and delayed operational value.

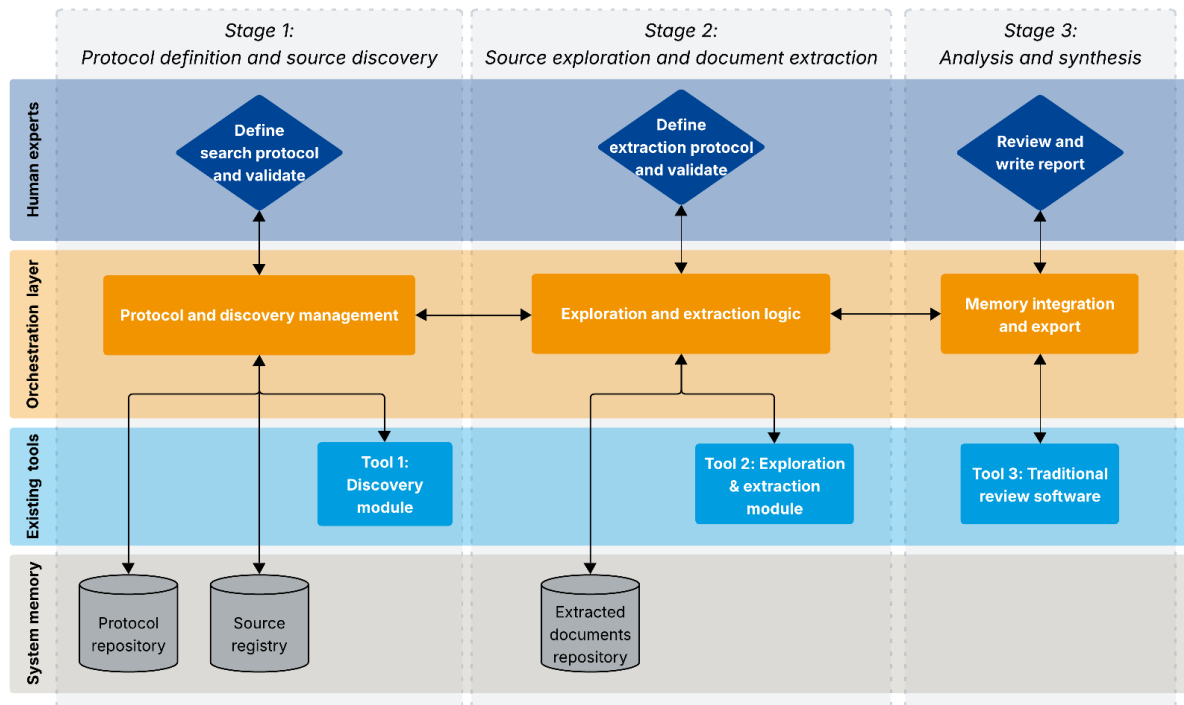
To bridge this gap, we recommend the adoption of an **instrumentalization framework**. This hybrid strategy combines the strengths of existing specialized tools with a custom-developed orchestration layer, allowing researchers to leverage proven components while preserving strategic flexibility and long-term autonomy.

To avoid confusion, the framework distinguishes between the **scientific steps** (the methodological requirement as outlined in Figure 1) and the **technical stages** (the instrumentalization of the scientific steps as outlined in Figure 3). The instrumentalization framework is therefore organized into three technical stages (Figure 3) that focus on the technical orchestration of Scientific Step 1 (Discover and identify sources), Step 2 (Explore validated sources), and Step 3 (Retrieve documents and results) of the formalized systematic review workflow (see Figure 1). These three steps were identified as the primary bottlenecks in the current manual process and established as the high-priority targets for automation and AI integration.

6.1 Overview of the instrumentalization framework

The instrumentalization framework is a modular architecture that treats AI technologies as interchangeable "bricks" coordinated by a central system (Figure 3). This approach is intended to support scientific traceability and methodological transparency by ensuring that every automated step is logged and guided by human oversight.

Figure 3 – High-level overview of the instrumentalization framework



6.1.1 Core components of the framework

In order to automate the grey literature workflow, the framework is built around four foundational layers designed to ensure that the transition to AI-augmented retrieval remains robust and auditable:

- The **orchestration layer** acts as the central coordinator that links human expertise with the system's functional modules. It assists researchers in building search protocols, configures the technical parameters for data extraction, and ensures information flows seamlessly between user inputs, external APIs, and the system's memory.
- The **existing tools layer** represents the functional bricks of the system, using three specialized modules for discrete tasks:
 - Tool 1 (Discovery module): AI search agents used to identify decentralized grey literature sources based on conceptual relevance.
 - Tool 2 (Exploration & extraction module): Robust web crawlers and scrapers that systematically navigate validated sites to retrieve raw documents and text.
 - Tool 3 (Traditional review software): Established platforms (e.g., Rayyan, DistillerSR) where the final scientific analysis is performed (out of scope of this report).
- The **system memory layer** ensures the traceability of the methodology by maintaining key information in three repositories:
 - The **Protocol repository** stores search keywords, refined prompts, and methodological strategies to ensure reproducibility.
 - The **Source registry** tracks identified websites by metadata on their credibility, thematic scope, and validation status.
 - The **Extracted documents repository** manages retrieved documents, raw text, and metadata for easy export to reference management software.



- The **human experts layer** ensures that scientific validation remains at the heart of the process through a "human-in-the-loop" design. Experts are responsible for defining the research scope, validating AI-generated configurations, and performing the final qualitative screening and data synthesis to ensure scientific rigor.

6.1.2 Key concepts for understanding the workflow

To navigate this framework, it is helpful to understand four core AI concepts that drive the system's efficiency (see Box 2).

Box 2 – AI Terminology for grey literature review

Large language models (LLMs)

AI systems trained on massive datasets to learn patterns in human language. By analysing how words and sentences commonly appear together, they learn to predict and generate text. This enables LLMs to summarize information and generate coherent text across a wide range of tasks.²⁰

Embeddings

Numerical representations of text that group related meanings together, allowing for comparison based on concept rather than exact wording.²¹

Generative AI

Generative AI refers to deep-learning models that can synthesize novel content (such as text or images) from a simple prompt and patterns learned on massive datasets. These models identify the underlying distribution of their training data to generate statistically plausible samples.

Machine Learning

Machine learning is a branch of AI and computer science that uses statistics and optimization techniques to build systems capable of learning patterns in data to make predictions and decisions. These systems do not require instruction-based programming and can adapt to unseen data.

Semantic search

A retrieval method that uses embeddings to identify results conceptually related to a query, even when different terminology is used. This allows more natural and flexible retrieval compared to traditional keyword search.²²

Retrieval-Augmented Generation (RAG)

A technique that combines LLM capabilities with targeted retrieval from trusted data sources to ensure more reliable, citation-backed responses.²³

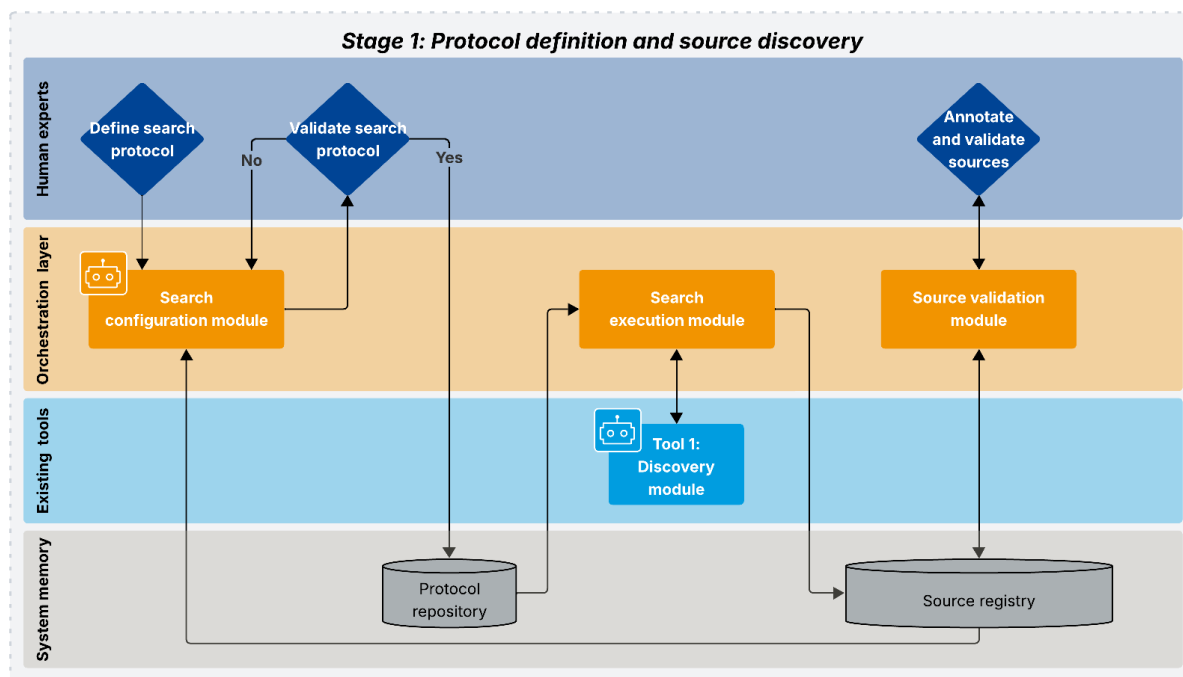
Application Programming Interface (API)

A standard design of interface that allows for automated communication between different applications and web services.

6.2 Stage 1: Protocol definition and source discovery

This first technical stage addresses Scientific Step 1 (Discovery) and the critical challenge of identifying where relevant grey literature resides (Figure 4). Unlike traditional searches that require exact keyword matches, the instrumentalization framework utilizes **semantic AI search** to discover sources based on conceptual relevance. By utilizing embeddings and LLMs, the framework can identify results that are conceptually related even when different organizations use varied terminology to describe the same health indicators (for instance, identifying data on "mobility impact" even if that specific phrase is absent from the source text).

Figure 4 – Stage 1 workflow: protocol definition and source discovery



'Robot' icon represents the steps where AI is used

The discovery process is a **collaborative cycle between the researcher and the orchestration layer**, where the system is designed to promote visibility and transparency throughout this phase.

It begins with the human expert **defining the research question and scope**, including the health condition, geographic focus, specific indicators of interest, and the relevant patient or societal perspectives. The orchestration layer, specifically through the **search configuration module**, then assists the expert in building a formal search protocol.

First, using semantic search methodologies, the AI module proposes additional keywords, refined prompts, or alternative search terms derived from the initial research question. Second, to maximize efficiency and avoid redundant effort across different disease areas, this module also queries the centralized Source registry to identify websites that have already been validated, excluded, or flagged as untrustworthy in previous cycles. The module combines these elements into a structured input to generate natural-language prompts specialized for the downstream Discovery Module (Tool 1). Before any automated execution occurs, the orchestration layer presents the new search protocol to the human expert for review and refinement. Once approved, the protocol is saved in the **Protocol repository** to ensure full transparency, verifiability, and scientific reproducibility.

The **Search execution module** then loads the validated protocol into the **AI-driven discovery module (Tool 1)** to perform a targeted search for new potential sources by calling external tools via their API services. The results returned by Tool 1, including URLs and timestamps, are saved into the **Source registry** where they are organized by domain name to facilitate a streamlined validation process. The orchestration layer provides the human expert with a summary of search performance, documenting the total number of sources found and those discarded due to prior exclusion.

Through the **Source validation module**, the researcher then performs a final qualitative review, marking newly identified sources as validated, irrelevant, or untrustworthy. These annotations update the Source registry, refining the system's memory and informing future search configurations to prevent the repeated gathering of known or poor-quality sources. Notably, this process does not constitute the "training" of an AI model, as training a model to recognize relevance in this context would be prohibitively expensive and highly dependent on the specific search context. In case of a



very high number of sources, a filtering or ranking system may be necessary to suggest the most relevant sources first for validation by human experts. However, such a system is not recommended in a proof-of-concept implementation and should be considered only if it appears to be necessary.

6.2.1 Prompt engineering strategy for Source discovery

The Discovery Module tools are primarily **prompt-driven**: researchers specify the scope, constraints, and expected outputs of a search through structured natural-language instructions. In this framework, prompts serve as the primary interface to tools such as **Exa.ai**, **Perplexity.ai**, and **Tavily** (see section 5.6.1).

This matters because these tools use LLMs to interpret the intent of a query rather than relying on exact keyword matches. As a result, the discovery process can surface relevant grey literature even when organizations use different terminology for the same health indicators or societal needs. In practice, this supports the identification of decentralized sources (such as policy papers, institutional reports, and other non-journal publications) across the live web.

To support clarity, transparency, and a consistent structure across tools, the framework decomposes each search instruction into four conceptual components (Table 6). Together, these elements make the search intent explicit, document constraints, and encourage consistent outputs across platforms.

Table 6 – Components of the Source discovery prompt

Component	Purpose	Example
Context	Communicates the overall intent and analytical framing of the discovery task.	Conduct a structured grey-literature search aligned with a systematic review methodology.
Research focus	Defines the subject matter, scope, and boundaries of the search.	Focus on [health condition X], addressing [specific indicators] within [geographic area].
Source preferences	Specifies preferred source types to guide retrieval and prioritization.	Prioritize publications from governments, HTA agencies, and universities.
Output structure	Requests a consistent and structured presentation of results.	List the URLs of discovered websites.

This prompting structure is intentionally **tool-agnostic**: it standardizes how search intent and output requirements are expressed without relying on any single tool's internal implementation.

In addition, effective use of these tools typically relies on an **iterative prompting process**. Discovery often begins with a broad prompt for a new health condition to map the available landscape. Based on initial results, the researcher refines the prompt (e.g., by adding constraints to target missing source types or narrowing the focus to specific regions or organizational levels). By documenting each prompt iteration in the Protocol repository, the framework ensures that the evolution of the search strategy remains transparent and scientifically auditable. Consequently, the performance and reproducibility of the Discovery Module depend on translating the human-validated search protocol into prompts that are both high-quality and standardized.

6.2.2 Technical specifications of Stage 1

The technical execution of source discovery relies on the following infrastructure within the orchestration layer:

- **Search configuration module:** This sub-component translates the human-defined research question into specialized prompts for AI-driven tools (and optionally, traditional search strings for keyword engines). It coordinates with the Source registry to automatically filter out previously validated or untrustworthy sites, ensuring search relevance.

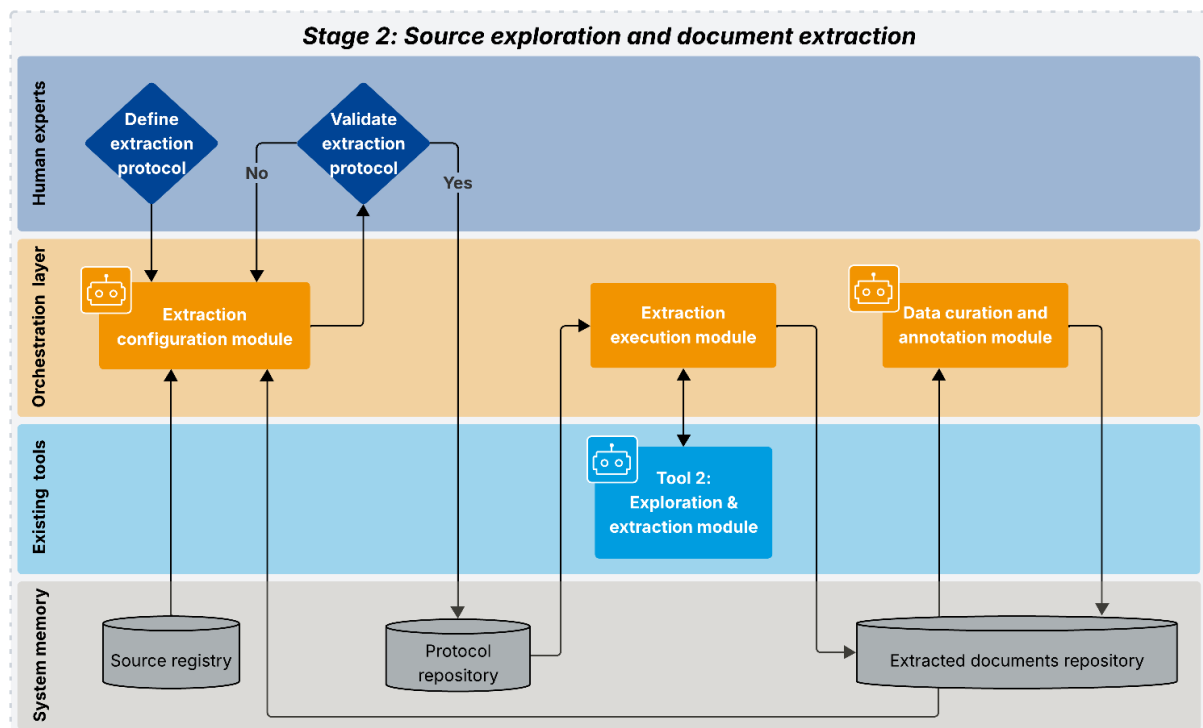
- Search execution module: This module triggers the discovery module (Tool 1) via API and performs the initial data processing. It saves the returned URLs and timestamps into the Source registry, respecting the data organization logic to prepare them for review.
- Source validation module: This component serves as the primary interface for human oversight. It retrieves the organized data from the registry, provides the expert with a summary of search performance (e.g., total sources found vs. excluded), and records the final qualitative annotations to update the system's memory.
- Memory integration: The layer maintains a persistent connection to the Protocol repository and Source registry to inform and refine future search configurations.

6.3 Stage 2: Source exploration and document extraction

This second stage consolidates the technical execution of Scientific Step 2 (Explore validated sources) and Scientific Step 3 (Retrieve documents and results), see Figure 1. While the previous stage identified "where" the information resides, this stage focuses on the systematic navigation of those sources to locate relevant pages and extract underlying evidence.

The goal of this stage is therefore to systematically navigate the list of validated sources to locate relevant pages and extract the underlying data into a structured format. Because grey literature often resides in unstructured environments like dynamic web pages or decentralized PDF repositories, this phase relies on a combination of robust web crawling and AI-driven curation (Figure 5).

Figure 5 – Stage 2 workflow: Source exploration and document extraction



Robot' icon represents the steps where AI is used

The process begins with the researcher **defining an extraction protocol**, meaning a set of rules and parameters directing the system to specific types of content (such as PDF reports, Word documents, or HTML tables) that should be retrieved. The **Extraction configuration module** assists the expert in this task by presenting a list of validated websites and proposing tailored scraping strategies. This module allows the expert to refine the scope of the extraction, ensuring that the system focuses only

on the most relevant sections of a source. Once validated, these settings are recorded in the **Protocol repository** to maintain a transparent and auditable trail of the retrieval methodology.

The **Extraction execution module** then triggers the functional module (**Tool 2**) to explore the targeted sites. Depending on the specific tool used, such as StormCrawler for large-scale systematic exploration or Crawl4AI for dynamic web content (see section 5.6.2), the system navigates the site's hierarchy to identify and retrieve relevant materials. The system is designed to handle three distinct types of output:

- **Internal URLs** are returned when the system is still navigating. If a page functions as a hub, index, or landing page (e.g. category pages, search results, document listings), the crawler extracts and returns internal links for further exploration. No content is stored yet; these URLs are queued for subsequent crawling.
- **Full documents** are returned when the crawler encounters a downloadable file. If a link points directly to a binary resource (PDF, DOCX, PPTX, etc.), the system downloads the file in full.
- **Raw extracts** are returned when relevant information exists only as page content. If the target information is embedded in a webpage rather than provided as a downloadable file (common in dynamic websites, dashboards, or registries), the system extracts and cleans the visible text from the page body. This text is returned as a raw extract instead of a full document.

These outputs are saved into the **Extracted documents repository** along with their source URLs and timestamps. To maximize efficiency, the framework implements a **feedback loop** where the system automatically checks the Extracted documents repository and the Source registry to identify "known locations" that have already been processed. By comparing URLs and previous actions, the system can bypass redundant links and focus resources only on new or updated content.

The raw output of a web scrape is often messy and lacks consistent metadata. To address this, the **Data curation and annotation module** employs LLMs to analyse and clean the retrieved content. This AI-driven module identifies missing metadata, such as publication dates or authorship, and categorizes the documents based on the NEED indicators. Crucially, this module can also assist with generating citations in a standardized RIS format, facilitating the seamless transfer of these results into traditional reference management software such as Zotero or Endnote.

6.3.1 *Defining the extraction protocol and configuration*

A central requirement of this phase is the definition of an extraction protocol. Unlike Step 1, where the user drafts a natural-language prompt to "talk" to the search tool, Step 2 is **configuration-driven**. The researcher interacts with the Extraction configuration module to set technical parameters that govern the exploration and extraction behaviour. This approach shifts the focus from creative prompting to a declarative, rule-based setup that ensures high precision and maintainability.

Instead of defining "what to look for" (as in the discovery phase), the expert validates how the system should "behave" on the target websites:

- **Crawl depth:** Determines how many links deep the system should follow from the starting page.
- **Path routing:** Targets specific sub-directories (e.g., /publications/ or /hta-reports/) to ensure the tool stays within relevant sections of a site.
- **File-type filters:** Prioritizes specific formats like PDFs or Word documents over generic webpage content.
- **Extraction rules:** These act like digital "bookmarks" that tell the system exactly where to look on a page for specific data, such as a publication date or a document title.

The orchestration layer then takes these human-validated parameters and translates them into technical instructions for Tool 2. Depending on the structure of the target website, the framework utilizes two distinct exploration methodologies:

- Rule-based exploration: For institutional websites with predictable, well-organized layouts, the system follows strict paths and explicit rules to systematically retrieve documents from known locations. This method is significantly faster and more cost-effective for standardized sites.
- Semantic extraction: For disorganized or complex websites where pages differ significantly, the tool uses an LLM to "read" the content like a human would, evaluating context and relevance in real-time. In this mode, the Extraction Configuration Module employs a static, internal system prompt built into the software. This internal prompt provides standardized instructions that tell the AI how to interpret page elements and identify data points aligned with the NEED research protocol.

Importantly, the researcher does not necessarily have to manually select between these two methodologies for every source. The Extraction configuration module is designed to provide an intelligent recommendation or automatically detect the most appropriate method based on the website's structure. This "behind-the-scenes" automation removes the need for the researcher to write individual instructions for every new website, ensuring the process remains scalable and reproducible.

6.3.2 *Technical specifications of Stage 2*

The technical infrastructure for source exploration and extraction includes:

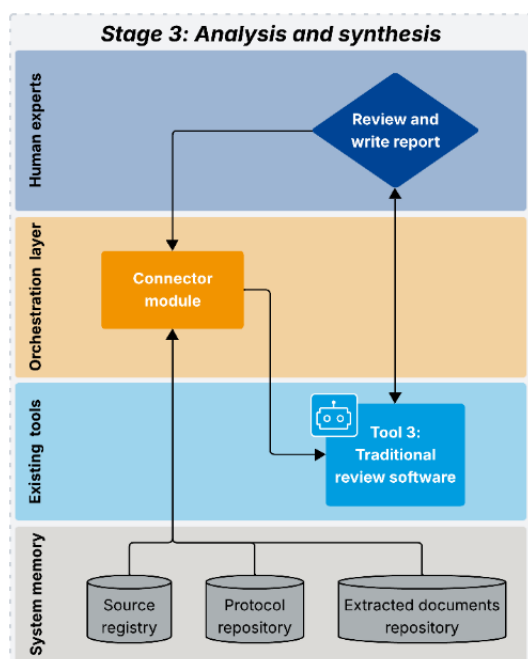
- Extraction configuration module: Translates human requirements into scraping configurations, managing the logic for "what" to download from "where".
- Extraction execution module: Manages the communication with Tool 2 (e.g., StormCrawler, Crawl4AI, Firecrawl, ScrapeGraphAI, or Scrapy) via API or local orchestration to retrieve raw data.
- Data curation module: An AI-powered layer that performs text cleaning, metadata extraction, and RIS citation generation.
- Memory integration: Persistent connections to the Protocol repository and Extracted documents repository ensure that the provenance of every document is preserved for scientific rigor. In practice, the Extracted documents repository may be an existing reference management tool such as Zotero or EndNote, which researchers already use to store full-text PDFs and bibliographic metadata.

6.4 Stage 3: Analysis and synthesis

The final technical stage serves as a bridge to scientific steps 4 through 9 (Appraisal, Synthesis, and Reporting, see Figure 1), which are considered out-of-scope for the automation pipeline and remain within the domain of human expert review. The primary goal here is the seamless "handoff" of retrieved evidence into traditional review software (Figure 6).



Figure 6 – Stage 3 workflow: Analysis and synthesis



Robot' icon represents the steps where AI is used

The **Connector module** is responsible for transitioning the retrieved evidence from the system's automated environment into the researcher's analytical environment. This module interfaces directly with the system memory, retrieving processed documents from the Extracted documents repository, along with supporting data from the Source registry and Protocol repository.

The primary responsibility of the Connector Module is to package this information into standardized formats compatible with Tool 3 (traditional review software). This involves:

- RIS export: Generating comprehensive citation files including the AI-enriched metadata.
- Document linking: Ensuring that exported citations are correctly linked to their corresponding downloaded PDF or text files.
- Categorization tags: Including the "NEED Indicators" as searchable tags or keywords within the review software to facilitate initial screening.

Modern review software such as [Rayyan](#) or [Covidence](#) supports the direct loading of extracted documents into their environments:

- Rayyan: Supports bulk uploading of full-text PDFs and uses a PDF Automatch feature to automatically link documents to their corresponding citations.
- Covidence: Allows for bulk PDF imports, often facilitated by exporting a structured EndNote XML or RIS file from the reference manager to ensure all attachments and metadata are preserved during the transfer.

6.4.1 Human expert review and system feedback

At this stage, the scientific work (reviewing and writing the report) is conducted externally by human experts in their traditional review software. However, the framework maintains an active connection to support transparency and avoid redundant work in future search cycles. As researchers perform screening and selection within their chosen tool, the workflow incorporates a feedback loop:

- Inclusion/exclusion decisions: Final decisions made by experts (e.g., "Excluded: Irrelevant population") can be exported back to the orchestration layer.
- Memory update: These decisions are logged in the Source registry with a timestamp. This ensures that in future updates of the NEED database, the system automatically recognizes these documents and avoids re-processing them.
- Audit trail: By capturing these external decisions, the orchestration layer provides a complete, auditable record of the evidence selection process, from initial discovery to final inclusion.

6.4.2 Technical specifications of Stage 3

The technical components supporting the analysis and synthesis transition include:

- Connector module: The interface that manages the retrieval of curated data from internal storage, converts it into exportable formats (like RIS), and ensures the secure transfer of documents and metadata to external review software (Tool 3).

6.5 Considerations and constraints for the instrumentalization framework

Implementing the instrumentalization framework in a governmental setting requires aligning automated grey literature retrieval with three constraint families: (i) **legal and ethical requirements** governing access, copyright, and responsible use of web resources; (ii) **operational safeguards** that preserve the integrity, traceability, and auditability of the evidence workflow; and (iii) **security and digital-sovereignty requirements** that ensure resilient infrastructure, controlled external dependencies, and institutional independence in an AI-assisted environment. This section summarizes these constraints and explains how they shape the design and deployment of the framework.

6.5.1 Legal and ethical considerations for grey literature retrieval

The automated strategy for identifying and retrieving grey literature raises important legal and ethical considerations, particularly in relation to copyright, website access conditions, and responsible use of web resources. Federal research institutions operate within defined European and national legal frameworks that both enable scientific research and impose clear obligations regarding data handling and responsible conduct.

6.5.1.1 Copyright and text/data mining

At the European level, [Directive \(EU\) 2019/790](#) on copyright and related rights in the Digital Single Market introduced **specific exceptions allowing text and data mining (TDM)**. These exceptions were designed to support scientific research while preserving the legitimate interests of rightsholders.

In particular, Article 3 of the Directive establishes a mandatory exception allowing research organisations to reproduce and extract content for TDM **for the purposes of scientific research**, provided they have lawful access to the content and that copies are stored securely, in line with the Directive's conditions. This exception has been transposed into [Belgian law in the Code of Economic Law \(Article XI.191/1\)](#) and applies irrespective of contractual clauses that would otherwise seek to prohibit such activities. As a public research organisation conducting scientific studies, KCE falls within the scope of this exception when performing text and data mining on lawfully accessible material.



A second, broader exception (Article 4 of the Directive) allows **reproductions/extractions of lawfully accessible works for any purpose**, only where rightsholders have not expressly reserved their rights “in an appropriate manner”, explicitly including machine-readable means for content made publicly available online. In practical terms, this means that “publicly accessible” does not necessarily imply “available for unrestricted automated mining” outside the scientific research perimeter of Article 3.

6.5.1.2 *Storage, retention, and publication boundaries for the NEED workflow*

For the NEED project, the above legal framework supports the internal reproduction and analysis of grey literature that KCE can lawfully access (e.g., openly available materials, or content accessed via institutional subscriptions), where the activity serves a scientific research purpose under the Article 3 exception. However, this exception primarily concerns mining and analysis, **not automatic redistribution of the original works**. Accordingly, the NEED database and associated reporting outputs should be designed around a clear separation between:

- **Internal research copies** (controlled): Full-text copies and machine-readable extracts may be stored within the secure research environment with an appropriate level of security, access controls, and auditability, consistent with Article 3’s security requirement and the need to retain material for verification.
- **Public-facing outputs** (restricted): Public deliverables should prioritise extracted indicators, aggregated results, and synthesised findings, and should avoid re-hosting full documents by default. Where helpful, the system can publish stable references (e.g., source URLs, citations, provenance metadata) rather than the full text.

Where redistribution is desirable (e.g., to support transparency or reproducibility), it should be limited to cases where the legal basis is clear, for example, when the document is distributed under an open licence that permits reuse/redistribution, or where explicit permission is obtained. To operationalise this boundary, the instrumentalization framework should systematically capture **licensing and rights metadata** at ingestion (licence type where declared, source terms, access mode), so that publication rules can be applied consistently and defensibly.

6.5.1.3 *Access restrictions and responsible automated retrieval*

Websites hosting grey literature may implement **technical measures to regulate automated access**, such as request rate limits, CAPTCHA systems, dynamic content rendering, session-based access controls, or blocking mechanisms that respond to high-frequency or non-standard traffic. Modern browser automation tools can address many technical challenges related to dynamic or JavaScript-rendered content. However, mechanisms such as CAPTCHA systems, login walls, or explicit access controls are often intentionally designed to restrict automated access. While copyright law permits text and data mining for scientific research under defined conditions, this does not imply a right to bypass such safeguards. Actively circumventing access controls may raise legal and ethical concerns beyond copyright law and is inconsistent with responsible research practices.

For this reason, the NEED workflow is designed to respect both technical and ethical standards of web access. The automated exploration and extraction module should clearly identify itself through an **institutional user agent** and apply conservative rate limits between requests to avoid unnecessary load on external servers. Where available, robots.txt directives should be consulted and respected. In addition, when websites set contractual access conditions (terms of use), these should be considered alongside robots.txt to reduce legal and reputational risk. Although such directives do not create legal obligations under copyright law, they represent a widely accepted mechanism through which website administrators communicate preferred crawling behaviour.

When automated retrieval encounters access restrictions (e.g. repeated HTTP errors, rate limiting, or blocking), the system should not attempt to bypass these measures. Instead, the orchestration layer should flag the affected source for **manual handling**. A human expert can then access the website through a standard browser, download the relevant document, and upload it to the extraction

database. This “manual fallback” approach ensures that no critical data is lost due to technical blocking, while avoiding technical circumvention and preserving constructive relationships with external data providers. It aligns with the broader human-in-the-loop design of the NEED workflow and supports responsible, transparent, and sustainable grey literature collection.

6.5.2 *Operational and security considerations*

To operationalize the framework in a federal-sector environment, the implementation must preserve the integrity, reliability, and institutional accountability of the evidence workflow while complying with stringent information-security and procurement constraints.

6.5.2.1 *Technical requirements and constraints for federal institutions*

The deployment of an automated evidence-gathering workflow within a governmental organization is shaped by a set of institutional requirements that extend beyond technical performance. The NEED methodological framework is publicly documented and designed for transparency, traceability, and methodological consistency. Consequently, the primary concern is not the confidentiality of search strategies, but the integrity and faithful execution of the methodology in an AI-assisted setting. The system must ensure that prompts, tool configurations, and retrieval logic reflect the documented framework and cannot be altered, injected, or redirected through unauthorized configuration changes or unverified dynamic prompt modifications that would distort the analysis. The authoritative record of configurations, screening decisions, annotations, and validation steps is maintained within agency-managed systems to guarantee traceability and auditability.

Second, the framework must ensure **integrity, traceability, and auditability** across the full pipeline. The system should maintain an end-to-end audit trail that records what was queried, what was retrieved, when and how it was processed, which transformations were applied, and which expert validated or rejected the outputs at each step. This traceability is essential both for scientific defensibility (verification and documented reconstruction of prior states) and for accountability in a public-sector setting.

Third, the architecture must support **human-in-the-loop reliability**. AI-assisted outputs should be treated as preliminary proposals rather than authoritative records. The workflow must therefore make uncertainty visible, preserve intermediate results, and provide clear mechanisms for experts to correct, override, and document decisions before any item becomes part of the validated evidence base.

Finally, the framework must comply with stringent **information security and supplier risk-management constraints**. Tools and models must be selected, deployed, and operated in line with institutional security policies, including appropriate review processes, logging and monitoring, vulnerability management, and contractual safeguards. Where external services are used, the system should apply data-minimization by default (sharing only what is necessary to perform a task) and route outbound requests through a controlled gateway that enforces authentication, logging, and policy-based restrictions. This design ensures that external tools support the documented methodology without compromising institutional oversight or traceability.

Taken together, these requirements imply that the framework should be implemented as a controlled environment in which automation accelerates collection and processing, while institutional governance remains the final authority over validation, inclusion, interpretation, and publication of results.

6.5.2.2 *Infrastructure architecture and trust boundaries*

The infrastructure is implemented as an end-to-end, containerized environment designed for scalability, reliability, and controlled deployment. The system is partitioned into three layers: **(i)** a frontend user interface for expert interaction and validation, **(ii)** a backend orchestration layer that manages processing pipelines and authenticated calls to external tools, and **(iii)** a centralized database that records results and supports traceability, auditability, and controlled reconstruction of prior workflow states. To enable flexible deployment while maintaining operational separation, the



frontend and orchestration components are hosted in **separate containers** (e.g., Docker) and can be deployed on **agency-controlled infrastructure**, either on premises or within an approved cloud tenant (e.g., Azure or AWS), with network segmentation and access controls enforcing the institutional trust boundary.

At the core of this design, the **orchestration layer** runs within the protected environment and acts as the authoritative workflow controller. It schedules tasks, maintains state across pipeline stages, and applies governance rules (e.g., which sources are eligible for automated retrieval, which outputs require manual validation, and which items may be exported to downstream tools). The orchestration layer also enforces a separation between shareable query content and internal records. Structured prompts may be transmitted to external discovery services for semantic retrieval, while expert annotations, screening decisions, evaluation logs, and curated registries remain stored within the institutional environment. The LLM component used for prompt refinement can be deployed locally when required, ensuring that intermediate context remains within the secure perimeter in configurations where institutional policy so requires.

Data persistence is handled through a centralized database (cloud-hosted within the approved environment or hosted internally, depending on institutional policy) that stores, at minimum: **search configurations, validated URLs, retrieval timestamps, document fingerprints/deduplication keys, provenance metadata, and expert annotations**. This persistent record enables end-to-end traceability (what was searched, retrieved, processed, and validated), supports auditability, and allows controlled reuse of prior work across review cycles without re-running high-cost or high-risk external retrieval steps.

To support compliance-by-design, the architecture includes a dedicated **metadata and rights-management component**. This component records, where available, the access mode (open web vs subscription), declared licence information, and source-level constraints, and links each retrieved item to its provenance in the Source Registry. These metadata allow the workflow to implement enforceable guardrails, such as ensuring that automated processing proceeds only where lawful access is documented and ensuring that public-facing outputs default to extracted indicators and synthesised findings rather than redistribution of full texts unless permitted by an open licence or explicit authorisation. Embedding these checks directly into the orchestration logic reduces reliance on ad hoc manual controls and increases consistency across teams and over time.

Finally, the trust boundary is reinforced through infrastructure controls appropriate to the public sector context: strict network egress rules from internal components, role-based access for users and service accounts, centralized logging and monitoring, and clear separation between the internal database/logging domain and any external tool interfaces. This combination of containerized modularity and explicit trust boundaries ensures that the framework remains scalable and maintainable while preserving institutional control over execution, validation decisions, and evidence records.

6.5.3 *External tools integration and modularity*

In the recommended architecture, the orchestration layer relies on external services for two functions that are deliberately kept outside the secure core: (i) discovery, where web-search services are used to identify candidate grey-literature sources, and (ii) retrieval/extraction, where dedicated crawling or scraping services are used to collect documents. To preserve long-term maintainability and reduce vendor lock-in, these external capabilities should be integrated through a small number of stable, well-defined interfaces rather than tightly coupling the workflow to any single provider.

This modular separation increases the number of viable technical alternatives for each task — including open-source or self-hosted solutions — and allows components to be replaced without redesigning the surrounding system, thereby strengthening long-term sustainability, and compliance in a public-sector research environment. In addition, avoiding tight coupling to a single provider reduces the risk that model behaviour, ranking heuristics, or service-level changes beyond institutional

control could inadvertently influence retrieval outcomes. The ability to substitute tools or compare outputs across multiple services provides an additional safeguard against undetected bias and strengthens the robustness of the evidence-gathering process.

A practical way to achieve this is to prefer tools that expose a simple REST API (a standard design of interface for the communication between applications and web services, see <https://www.ibm.com/think/topics/rest-apis>) over HTTP (typically JSON in/JSON out), because REST interfaces are broadly interoperable and make it easier to replace a tool later without redesigning the surrounding system. Within this pattern, the orchestration layer should call external tools only through the controlled gateway described in the trust-boundary architecture, where outbound requests can be authenticated, logged, throttled, and constrained by allow-lists. Depending on institutional policy, discovery prompts may be limited to task-specific formulations derived from publicly documented methodology, avoiding transmission of internal evaluation logs or expert annotations. This ensures that modularity does not come at the expense of governance: the institution retains control over what leaves the perimeter, what enters it, and how external dependencies are used.

The search tools previously identified for Stage 1 (Protocol definition and source discovery) fit this “replaceable REST module” approach. Providers such as Exa²⁴, Perplexity²⁵, and Tavily²⁶ offer APIs specifically positioned for programmatic web discovery. In addition, general AI platforms now expose web-connected capabilities that can be integrated in a similar way. OpenAI provides a built-in web search tool in its API²⁷, and its pricing model explicitly distinguishes between tool calls and the search-content tokens used to ground responses²⁸. Google’s Gemini API offers “Grounding with Google Search,” which connects model outputs to real-time web content²⁹ and has its own billing schedule³⁰. Mistral and Anthropic (via the Claude API platform) similarly document a built-in “Websearch” tool in their agents framework³¹. If an organisation prefers deeper integration within a cloud ecosystem, Google’s [Vertex AI Search/Agent Builder](#) family can also be considered; it supports website indexing modes and operational requirements such as domain verification for advanced indexing, but it generally implies a more platform-centric setup than lightweight search APIs.

Although not included in the original list of tools, which focused on solutions explicitly positioned for literature-review or grey-literature workflows, general-purpose agent platforms are valid candidates for specific tasks within the modular solution presented here. These platforms were not evaluated under the same grid, as the initial objective was to identify domain-oriented tools rather than full-stack agent architectures. However, they are fully compatible with the modular contracts described above and can be assessed during a stage-specific selection phase. In this regard, additional search tools such as You.com may also be considered for integration into the framework. The pricing models of these APIs are complex and subject to frequent change. The references below point to the official, up-to-date pricing documentation for each selected API.

- <https://exa.ai/pricing>
- <https://docs.perplexity.ai/getting-started/pricing>
- <https://www.tavily.com/#pricing>
- <https://ai.google.dev/gemini-api/docs/pricing>
- <https://openai.com/api/pricing/>
- <https://mistral.ai/pricing#api-pricing>
- <https://claude.com/pricing#api>
- <https://you.com/pricing>

These pricing models differ substantially: some are request-based, others token- or credit-based, and many combine multiple schemes. Establishing a common basis for comparison is difficult not only because of these structural differences, but also because token consumption and usable results per request are inherently task-dependent. The cost of a single API-based “research” task can therefore



vary widely depending on the number of operations and tokens consumed. For example, Tavily's research task ranges from 4 to 250 credits (0.032\$ to 2\$ per call; see <https://docs.tavily.com/documentation/api-credits>).

At the current blueprint stage — where expected query volumes and invocation depth are not yet defined — any aggregated cost estimate would necessarily rely on speculative workload assumptions and could therefore be misleading. A similar uncertainty would apply to a monolithic LLM-agent alternative, whose cost would likewise depend on task complexity and token usage. Meaningful cost comparison therefore requires empirical benchmarking under defined usage scenarios rather than static pricing snapshots. From a public-sector governance perspective, committing to an architectural model based solely on indicative API price lists, without workload modeling, may expose the institution to unforeseen operational costs.

For Stage 2 (Source exploration and document extraction), the same modular logic applies: the orchestration layer should treat crawlers/scrapers as interchangeable “workers” that accept a URL (plus a small configuration payload) and return internal URLs, full documents, and raw text extracts (and, where applicable, structured fields), together with provenance metadata. Within the identified tools of this category, ScrapeGraphAI³², Firecrawl³³, and Crawl4AI³⁴ provide API-first approaches aligned with this model and document rate limiting and operational features as part of their interfaces. Emerging AI-powered browser automation agents, including agent-driven browser control solutions capable of interacting with dynamic or JavaScript-rendered websites, may similarly be integrated as interchangeable Stage 2 workers when interactive navigation is required. Frameworks such as Scrapy remain relevant when the institution needs full control over the crawling stack, but they shift responsibility for hosting, scaling, and security hardening to the organisation. However, Zyte (a cloud service built upon Scrapy) offers a managed web-scraping API with a detailed HTTP reference and explicit pricing documentation, which can serve as an enterprise-oriented alternative.³⁵

Finally, tools that require substantial local installation and topology management like StormCrawler can still be used, but they fit less naturally into a “swap-in/swap-out” service module unless the organisation commits to operating them as an internal platform.

To make modularity effective in practice (not just in principle), the orchestration layer should define two internal contracts: one for Stage 1 (query → ranked candidate URLs + minimal metadata) and one for Stage 2 (URL + policy settings → document payload + provenance + extraction output). A third contract governs the handoff package to Tool 3 (e.g., RIS + linked attachments + tags), implemented by the Connector module. Each external provider is then implemented as an adapter to these contracts. This allows the institution to change providers in response to the evolution of tool capabilities, security reviews, procurement constraints, performance issues, or cost changes without rewriting the overall workflow. It also supports conservative data-sharing by design: discovery calls can be limited to high-level “search intents,” and retrieval calls can be constrained to approved domains, throttled access patterns, and strict logging, while keeping internal screening logic, expert annotations, and validation history inside the protected environment.

6.5.4 *Controlled agency: A deterministic orchestration model*

The field of artificial intelligence is evolving rapidly, and agent-based models—in which a LLM autonomously uses tools—are becoming a prominent approach to workflow automation. Agent-based search, crawl, and scraping tools provided by OpenAI, Claude, Tavily, Firecrawl, and others are strong candidates to automate specific steps in the architecture presented. Frameworks such as LangGraph, AutoGen, and CrewAI further extend this paradigm by enabling agentic coordination at the orchestration level itself.

We acknowledge that the proposed architecture represents a specific class of deterministic orchestration agents. While an autonomous agent orchestrator (e.g., an LLM with tool-use and a dynamic execution graph) is a viable alternative to coordinate discovery and extraction, the choice of a fixed-graph agentic design in the present framework is deliberate. This approach is grounded in the

institutional requirements of traceability, auditability, and procedural control described in section 6.5. Rather than attempting to eliminate variability, the design makes explicit where it is permitted and where it is constrained: adaptive behavior may occur within well-scoped, "agentic" modules (e.g., semantic search or extraction services), while the high-level orchestration logic, validation checkpoints, and re-execution boundaries remain fixed and versioned.

While the architecture includes elements commonly found in agent systems (tool invocation, memory components, human-in-the-loop checkpoints), the design favors structural agency over autonomous planning. The orchestration layer executes a predefined, versioned workflow graph and does not autonomously generate or modify its own control flow. In this sense, it functions as a controlled workflow engine that provides the functional benefits of an agent without the unpredictability of self-directed execution.

The NEED workflow is structured and stage-based, requiring explicit human validation points that are often incompatible with unconstrained autonomous loops. Sources must be approved before crawling, validated repositories must remain under researcher control, and specific segments of the workflow must be re-executable in isolation. A deterministic, centralized orchestrator reflects the operational properties of the research process more directly than a distributed multi-agent coordination layer.

In the proposed architecture, agentic solutions are not excluded but are positioned as modular services invoked by the orchestration layer. An autonomous agent may operate within a defined module (e.g., complex website navigation) without controlling the high-level workflow. This "governed agency" preserves modular substitution and institutional governance while remaining compatible with advances in agentic capabilities.

The proposed design therefore prioritizes transparency, controlled re-execution, and modular replaceability, ensuring that the researcher—rather than an autonomous planning loop—remains the primary orchestrator of the evidence synthesis process.

Finally, it should be noted that full computational reproducibility in web-based, AI-assisted retrieval systems cannot be guaranteed. External search APIs perform semantic expansion, reranking, and continuous index updates that may introduce variability in returned results over time. The objective of the proposed architecture is therefore not to eliminate variability, but to govern and document it. By versioning prompts, logging exact queries and tool parameters, storing returned candidate lists, and preserving document fingerprints and timestamps, the system enables traceable re-execution and defensible "freezing" of evidence states at defined decision points. Variability is thus contained within bounded modules, while the orchestration logic and validation checkpoints remain stable, auditable, and firmly under human direction.



7 DISCUSSION AND CONCLUSION

The NEED project was launched to support the transition toward a needs-driven health system by improving how patient and societal unmet needs are identified, documented, and translated into evidence for policy decisions. A major practical barrier to scaling the NEED database is the systematic collection of unmet-need estimates from grey literature, which is decentralized, heterogeneous in format, and inconsistently indexed. As demonstrated during the initial proof-of-concept studies, manual web searches can identify valuable evidence but are labour-intensive, difficult to reproduce, and unlikely to scale across the full set of health conditions targeted by NEED.

Against this backdrop, the overarching aim of this study was to examine whether, and under what conditions, AI technologies can meaningfully support the systematic review of grey literature in a way that is scalable, traceable, and suitable for the NEED context. Two operational objectives were pursued. First, the study aimed to identify and examine existing AI tools capable of searching, retrieving, and classifying grey literature across diverse formats and sources. Second, it aimed to translate these findings into a practical strategy for applying such tools within the NEED workflow, or, where no sufficient solution exists, to design and pilot a targeted solution that addresses remaining gaps while respecting the ethical, legal, and public-sector constraints described earlier in the report.

7.1 Objective 1: Tool landscape review and methodological reflections

To address Objective 1, this study applied a multi-phased analytical procedure combining broad tool identification, structured categorization, early filtering, and criterion-based examination. AI-powered deep search was used to accelerate the documentation of candidate tools and to scale the screening process across a landscape that is both heterogeneous and fast-evolving. This approach made it feasible to examine a broad range of tools in a consistent manner and to systematically document evidence relevant to the NEED requirements. However, it also required careful methodological safeguards. Publicly available tool documentation is uneven in quality and completeness, and AI-assisted synthesis is sensitive to prompt framing and source availability. Internal testing further showed that AI-supported scoring can vary between runs, meaning that the value of the approach lies in standardizing the analytical procedure and preserving the justifications used to interpret numerical outputs, rather than treating outputs as mechanically reproducible measurements. For this reason, the workflow separated information gathering from score generation and incorporated human verification at key decision points, treating AI as an accelerator for structured research rather than as an autonomous evaluator.

Within these methodological boundaries, two conclusions emerge as particularly robust. First, the tooling ecosystem remains highly segmented along functional lines. **Candidate solutions typically specialize in one part of the grey literature workflow** (semantic discovery, crawling/scraping, or downstream review acceleration) **rather than providing an end-to-end solution that supports the full NEED evidence pipeline.** This fragmentation reflects the intrinsic complexity of grey literature retrieval, where decentralized sources, unstructured formats, and inconsistent metadata create challenges that differ fundamentally from journal-based systematic searches.

Second, feasibility for automation is determined less by how “intelligent” a tool appears and more by pragmatic constraints that function as hard gates for integration and sustained use. Tools without stable programmatic access cannot be reliably integrated into an orchestrated workflow and therefore cannot support scalable update cycles across many conditions. Tools that are discontinued or weakly maintained introduce operational and security risks that are difficult to justify in a public-sector setting. Finally, grey literature retrieval is shaped by external access conditions at the source level (terms of use, rate limits, bot detection, CAPTCHAs), which can reduce the reliability of automated collection even when content remains accessible to human users. Together, these constraints explain why early filtering was essential and why the most viable candidates tend to be API-first services and well-maintained platforms that can operate within controlled governance boundaries.

These findings support the report's central empirical conclusion: **the current market does not offer a single, off-the-shelf solution that can cover the full NEED requirements for scale, traceability, and institutional constraints.** The numerical outputs generated during the analysis were not intended to identify a competitive “winner,” but to test the structural hypothesis that a single tool could cover the full NEED workflow. Their principal outcome is therefore structural rather than comparative: the absence of an all-in-one solution constitutes the main empirical result of Objective 1. The value of the landscape analysis lies not in fine-grained score differentials between closely performing tools — which remain within an expected margin of methodological variability — but in identifying which specialized capabilities can be combined safely and effectively within an architecture that preserves reproducibility, governance, and long-term autonomy.

The analytical grid was initially constructed to test the hypothesis that a single tool could cover the full NEED workflow. Once no such universal solution was identified, the analytical focus shifted toward combining specialized tools across workflow stages. The original grid was retained during the screening phase to preserve internal comparability; however, stage-specific criteria aligned with the functional role of each tool would provide a more precise basis for implementation decisions. Accordingly, before any future development of the proposed architecture is undertaken, a new market analysis should be conducted at the start of the implementation phase to examine candidate tools against criteria tailored to their intended role in the pipeline. Agent-based solutions offered by large AI platforms (such as OpenAI, Claude, Mistral, and Gemini), as well as additional tools not included in the initial inventory, can be evaluated at that stage according to the architectural role they are intended to fulfil.

7.2 Objective 2: Strategy for applying AI tools within the NEED workflow

Objective 2 translates the tool landscape findings into an actionable strategy for the NEED context. The central implication of Objective 1 is that the most robust path forward is architectural rather than tool-specific. Instead of relying on a single platform and adapting the workflow to its limitations, the recommended approach is the **instrumentalization framework**: a hybrid strategy in which a custom orchestration layer coordinates specialized external tools as interchangeable modules while retaining institutional control over protocols, validation decisions, provenance, and the evidence record.

This strategy fits the NEED context for two reasons. First, it targets the true bottlenecks identified in the formalized workflow, namely the upstream steps of discovering grey literature sources and reliably retrieving documents and relevant content across heterogeneous web environments. These steps are the main drivers of manual workload and variability, whereas the downstream phases of screening and synthesis are already supported by mature review platforms and remain grounded in expert interpretation. Second, the strategy aligns with the conceptual requirement of the NEED framework that unmet needs should be assessed independently of potential future products or treatments. By keeping protocol definition, source validation, and evidence inclusion decisions under human oversight and full traceability, the framework reduces the risk that opaque tool behaviour or ranking mechanisms implicitly shape the evidence base.

Where the tool landscape is insufficient, the recommended “new development” is not a bespoke AI model but the orchestration and integration capability itself. The principal gap is the absence of an end-to-end, governed workflow that can persist institutional memory across cycles while remaining flexible enough to swap external tools as the market evolves. A controlled orchestration layer creates value even if individual tools change: it standardizes how search protocols are documented, how validated sources are tracked, how retrieval outputs are stored with provenance, and how curated results are handed off into traditional review environments.

This approach also has several practical strengths for a federal research context. It supports human-in-the-loop integration by treating AI outputs as proposals that become part of the evidence base only after expert validation. It promotes robustness and repeatability by turning protocols, source registries, and extraction configurations into reusable assets that reduce redundant work and enable traceable updates over time. It preserves flexibility and modularity by decoupling the core workflow from any

single vendor, reducing lock-in and allowing components to be replaced as security, procurement, or performance conditions change. Finally, it accelerates value creation by enabling incremental deployment: the orchestration layer can be piloted and improved while leveraging existing best-in-class tools, rather than requiring a long development cycle for a fully custom end-to-end system.

7.3 Conclusion: What AI can and cannot do for grey literature review

The findings of this report indicate that AI can meaningfully support grey literature review for NEED, but only when applied within a governed workflow that preserves traceability and expert validation. The most immediate and scalable contribution is upstream: AI-enabled semantic discovery can broaden coverage beyond fixed lists of known websites and reduce dependence on exact keyword matches, which is particularly valuable when organizations use different terminology for similar indicators. AI can also add value downstream of retrieval by assisting with content cleaning, metadata enrichment, and initial categorization, especially where grey literature lacks standardized bibliographic fields.

At the same time, AI does not eliminate the need for expert judgment or governance. Discovery outputs still require qualitative source validation to manage credibility and relevance, and extraction outputs require checks for provenance, completeness, and appropriate interpretation. In practice, AI improves speed and breadth, but scientific defensibility continues to depend on transparent protocols, auditable logs, and human oversight. This reinforces the report's central recommendation: the most sustainable way to benefit from AI is not to seek a single autonomous tool, but to implement a modular instrumentalization framework in which automation accelerates early-stage bottlenecks while institutional processes remain the final authority over evidence inclusion and interpretation.

8 RECOMMENDATIONS AND NEXT STEPS

8.1 Implementation roadmap

The recommended approach is to implement the instrumentalization framework progressively. Rather than treating deployment as a binary choice between a manual process and a fully integrated platform, the roadmap is structured as a sequence of maturity levels. Each level delivers incremental value while reducing uncertainty about tool suitability, technical feasibility, and operational constraints. This staged approach is particularly appropriate in a fast-evolving AI tooling landscape and in a public-sector environment where security, procurement, and governance requirements must be addressed explicitly.

8.1.1 *Level 0: Empirical tool validation and feasibility benchmarking*

Level 0 is a mandatory empirical validation phase prior to any infrastructure investment. While this report provides a structured landscape review and capability screening, architectural decisions should not rely solely on feature comparison or indicative scoring from the present report. Given the rapid evolution of AI tools — including frequent changes in models, APIs, and pricing — shortlisted solutions must be benchmarked under current conditions before further commitment.

The objective at this level is practical feasibility. Retained discovery and retrieval tools should be tested on real NEED use cases (e.g., one previously studied condition and one new topic) to assess their ability to identify relevant grey literature sources and successfully retrieve and extract usable documents. This phase should remain lightweight: search protocols, prompts, and tool settings can be documented in a structured workspace, and discovered sources recorded provisionally, without assuming a final storage architecture.

Unlike the exploratory landscape review, Level 0 involves real-case execution under controlled conditions. Benchmarking should include comparison against a manual baseline and assess, at minimum: recall (percentage of known relevant sources retrieved), precision (proportion of retrieved

sources deemed relevant), time to completion, cost per condition, and incidence of false positives or hallucinated outputs.

The evaluation design should integrate the methodological learnings of this report and the peer-review feedback. In particular, it should define criteria per functional category, apply criterion-by-criterion comparisons across tools, use clearly anchored scoring rules, test robustness under prompt variation, and record execution logs to support traceability and defensible interpretation. Progression from Level 0 to Level 1 should require documented evidence that automated or semi-automated approaches achieve acceptable recall and precision relative to the manual baseline, together with demonstrable time-efficiency gains and manageable operational cost. Absent such evidence, further architectural investment should not proceed.

8.1.2 Level 1: Scripted automation and technical specification building

Level 1 introduces basic orchestration through scripts (e.g., Python) to automate repeatable interactions between modules and generate structured outputs. This stage reduces manual workload for repetitive tasks — particularly in Stage 1 discovery (API calls, URL storage, deduplication) — while forcing explicit definition of module inputs, outputs, error handling, and logging rules.

The storage model remains lightweight but becomes structured. Scripts should produce standardized artifacts (e.g., CSV/JSON) that can be versioned and compared across runs. A simple Source registry (e.g., structured table or lightweight database) should record discovery results, human validation decisions, tool identifiers, query IDs, and timestamps. Extraction outputs should enforce systematic provenance capture (source URL, retrieval timestamp, file hash/fingerprint), even if documents remain stored in shared systems.

Level 1 formalizes the operational “contracts” between modules (discovery and extraction) and defines logging and metadata requirements necessary for traceability and auditability, without yet investing in a full user interface or enterprise-grade infrastructure.

Progression from Level 1 to Level 2 should require documented evidence that scripted workflows execute reliably across repeated runs, that logging and provenance capture are complete and consistent, and that governance requirements (traceability, versioning, controlled re-execution) are operationally satisfied.

Finally, before committing to the development of the presented architecture, an alternative implementation should be tested to assess the capabilities and usefulness of a more autonomous orchestrator. Although the deterministic path presented in this report seems preferable for this application, the alternative based on autonomous agents also warrants empirical testing. A lightweight prototype could be developed using agentic frameworks such as Claude or LangGraph to inform the strategy to be followed in the development of the next levels.

8.1.3 Level 2: Integrated instrumentalization framework platform

Level 2 corresponds to the integrated, containerized platform described in the core architecture sections. The objective is to transition from script-based automation to a stable operational workflow that researchers can use routinely, with centralized governance, controlled integration of external tools, and a consistent user experience. The focus is on delivering a coherent end-to-end implementation of prioritized workflow stages (source discovery, validation, extraction, and structured handoff to review software) within a secure institutional environment.

At this stage, the memory model becomes a first-class component. The Protocol repository, Source registry, and Extracted documents repository are implemented as centralized services within an approved hosting environment (on-premises or approved cloud tenant). The platform records not only outputs but also the methodological context necessary for traceable re-execution and defensible updates: protocol versions, tool configurations, prompts where relevant, timestamps, and validation decisions. Extracted documents and raw text are stored with provenance metadata and deduplication

fingerprints, enabling controlled updates while avoiding redundant processing. Standardized exports (e.g., RIS with linked attachments and NEED indicator tags) operationalize consistent handoff to downstream review tools.

Progression to Level 3 should require evidence that the integrated platform operates reliably across multiple conditions, that governance and logging mechanisms remain effective at scale, and that performance indicators (recall, precision, time-efficiency, and cost) remain within acceptable thresholds without structural degradation.

8.1.4 Level 3: Future enhancements and extended capabilities

Once the integrated platform is stable, future iterations can extend functionality to improve scalability, coverage, and user experience without altering the core governance model. Enhancements at this level may include parallelized discovery across multiple providers, advanced monitoring dashboards (e.g., retrieval success rates, update frequency by source type), analytics on corpus evolution, and automated assistance for configuring extraction strategies based on observed website structures. Additional capabilities may include scheduled periodic updates, automated change detection for updated reports, and expanded support for heterogeneous outputs such as structured tables embedded in web pages or online registries.

At this stage, the memory model evolves from structured traceable storage to operational intelligence. The Source registry may incorporate lifecycle metadata (e.g., update frequency, volatility indicators, retrieval success history), while the Protocol repository may support reusable templates by condition or indicator family. The objective is cumulative efficiency: each update cycle should reduce marginal effort while preserving transparency, auditability, and expert oversight. Augmented retrieval generation (RAG) or variants such as Graph-RAG, may be considered for querying the corpus of documents retrieved. These techniques may be particularly useful in later stages of the workflow (such as screening or synthesis) and in the management of the NEED database.

Overall, the staged roadmap enables progressive value capture while reducing technical and operational uncertainty. Empirical validation (Level 0) establishes performance feasibility; scripted orchestration (Level 1) stabilizes logging and governance; and the integrated platform (Level 2) institutionalizes a reusable, controlled workflow. Level 3 builds on these foundations to enhance scalability and long-term maintainability. By treating protocols, validated sources, and extracted evidence as cumulative institutional assets, each stage strengthens the system's capacity to scale across conditions and update cycles without compromising transparency, accountability, or scientific defensibility.

■ REFERENCES

1. NEED: Needs Examination, Evaluation and Dissemination [Web page]. Brussels [cited 21/03/2024]. Available from: <https://health-needs.eu/index.php/en/>
2. Maertens de Noordhout C, Levy M, Claerman R, De Jaeger M, De Pauw R, Kohn L, et al. Needs Examination, Evaluation and Dissemination (NEED): assessment framework. Health Services Research (HSR). Brussels: Belgian Health Care Knowledge Center (KCE); 2024 03/2024. KCE Reports 377C1 (377C1) Available from: <https://doi.org/10.57598/R377C1>
3. Claerman R, Kohn L, Schönborn C, Obyn C, De Pauw R, Cleemput I, et al. NEED assessment framework 2025: EU expert consensus and core version. Needs report. Brussels: Belgian Health Care Knowledge Center (KCE); 2025 14/05/2025. KCE Reports 401 Available from: <https://doi.org/10.57598/R401C>
4. NEED collaborators. Unmet health-related needs for three selected health conditions in Belgium (v2024-10-10) [Data set]. In. Brussels; 2024.
5. Schönborn C, Obyn C, De Pauw R, Claerman R, Kohn L, Parardis J, et al. Protocol for NEED (needs examination, evaluation and dissemination) research projects. 10.57598/R406C. Brussels: Belgian Health Care Knowledge Center (KCE); 2025. Needs Available from: <https://doi.org/10.57598/R406C>
6. Farrah K, Mierzwinski-Urban M. Almost half of references in reports on new and emerging nondrug health technologies are grey literature. Journal of the Medical Library Association: JMLA. 2019;107(1):43.
7. Myohanen L, Taylor E, Keith L. Accessing grey literature in public health: New York Academy of Medicine's Grey Literature report. Publishing research quarterly. 2005;21(1):44-52.
8. Farace DJ, Schöpfel J. Grey literature in library and information studies. KG Saur; 2010.
9. Adams J, Hillier-Brown FC, Moore HJ, Lake AA, Araujo-Soares V, White M, et al. Searching and synthesising 'grey literature' and 'grey information' in public health: critical reflections on three case studies. Systematic reviews. 2016;5(1):164.
10. Landerdahl Stridsberg S, Richardson MX, Redekop K, Ehn M, Wamala Andersson S. Gray literature in evaluating effectiveness in digital health and health and welfare technology: a source worth considering. Journal of Medical Internet Research. 2022;24(3):e29307.
11. Mahood Q, Van Eerd D, Irvin E. Searching for grey literature for systematic reviews: challenges and benefits. Research synthesis methods. 2014;5(3):221-34.
12. De Jaeger M, Van Goethem R, Dobbelaere J, Léonard U, Zhi Tan Y, Van Montfort T, et al. Needs Examination, Evaluation and Dissemination (NEED): Identification of unmet health-related needs associated with Crohn's disease. Health Services Research (HSR). Brussels: Belgian Health Care Knowledge Center (KCE); 2024 03/2024. KCE Reports 377C3 (377C3) Available from: <https://doi.org/10.57598/R377C3>
13. Schönborn C, Léonard U, Van Goethem R, Dobbelaere J, Zhi Tan Y, Van Montfort T, et al. Needs Examination, Evaluation and Dissemination (NEED): Identification of unmet health-related needs associated with malignant melanoma. Health Services Research (HSR). Brussels: Belgian Health Care Knowledge Center (KCE); 2024 03/2024. KCE Reports 377C2 (377C2) Available from: <https://doi.org/10.57598/R377C2>
14. Affengruber L, van der Maten MM, Spiero I, Nussbaumer-Streit B, Mahmić-Kaknjo M, Ellen ME, et al. An exploration of available methods and tools to improve the efficiency of systematic review production: a scoping review. BMC Medical Research Methodology. 2024;24(1):210.
15. Liu NF, Lin K, Hewitt J, Paranjape A, Bevilacqua M, Petroni F, et al. Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics. 2024;12:157-73.



16. Levy M, Jacoby A, Goldberg Y. Same task, more tokens: the impact of input length on the reasoning performance of large language models. arXiv preprint arXiv:2402.14848. 2024.
17. Farace D, Biagioni S, Carlesi C, Baars C. Grey Literature and Persistent Identifiers: GreyNet's Use Case. In: Proceedings of GL2021 CONFERENCE PROCEEDINGS: Twenty-Third International Conference on Grey Literature'Digital Transformation of Grey Literature: Exploring Next Generation Grey'; 2022: TextRelease; p. 45-53.
18. Zowalla R, Wetter T, Pfeifer D. Crawling the german health web: Exploratory study and graph analysis. Journal of medical Internet research. 2020;22(7):e17853.
19. Programmable Search Engine vs Google.com [Web page].2026 [cited 26 January 2026]. Available from: <https://support.google.com/programmable-search/answer/70392?hl=en>
20. Stryker C. What are large language models (LLMs)? [Web page]. IBM;2026 [cited 26 January 2026]. Available from: <https://www.ibm.com/think/topics/large-language-models>
21. Barnard J. What is embedding? [Web page]. IBM;2026 [cited 26 January 2026]. Available from: <https://www.ibm.com/think/topics/embedding>
22. Google. Qu'est-ce que la recherche sémantique ? [Web page]. Google Cloud;2026 [cited 26 January 2026]. Available from: <https://cloud.google.com/discover/what-is-semantic-search>
23. IBM. What is retrieval augmented generation (RAG)? [Web page].2026 [cited 26 January 2026]. Available from: <https://www.ibm.com/think/topics/retrieval-augmented-generation>
24. Exa [Web page].2026 [cited 19 January 2026]. Available from: <https://exa.ai/docs/reference/getting-started>
25. Perplexity API Platform [Web page].2026 [cited 19 January 2026]. Available from: <https://docs.perplexity.ai/api-reference/search-post>
26. Tavily [Web page].2026 [cited 19 January 2026]. Available from: <https://docs.tavily.com/documentation/api-reference/introduction>
27. OpenAI Platform [Web page].2026 [cited 19 January 2026]. Available from: <https://platform.openai.com/docs/guides/tools-web-search>
28. OpenAI API Pricing [Web page].2026 [cited 19 January 2026]. Available from: <https://openai.com/api/pricing/>
29. Gemini API [Web page].2026 [cited 19 January 2026]. Available from: <https://ai.google.dev/gemini-api/docs>
30. Gemini Developer API pricing [Web page].2026 [cited 19 January 2026]. Available from: <https://ai.google.dev/gemini-api/docs/pricing>
31. Mistral Docs&API - Websearch [Web page].2026 [cited 19 January 2026]. Available from: <https://docs.mistral.ai/agents/tools/built-in/websearch>
32. ScrapeGraphAI [Web page].2026 [cited 19 January 2026]. Available from: <https://docs.scrapegraphai.com/api-reference/introduction>
33. Firecrawl v2 [Web page].2026 [cited 19 January 2026]. Available from: <https://docs.firecrawl.dev/api-reference/v2-introduction>
34. Crawl4ai [Web page].2026 [cited 19 January 2026]. Available from: <https://docs.crawl4ai.com/>
35. Zyte API reference documentation [Web page].2026 [cited 19 January 2026]. Available from: <https://docs.zyte.com/zyte-api/usage/reference.html#zapi-reference>

■ APPENDICES

APPENDIX 1. LIST OF TOOLS BY CATEGORY

CATEGORY 1: DATABASES/REPOSITORIES

These are the “places” where data lives—indexes, registries, and aggregators that contain the actual records or documents.

- Grey Matters
- OpenDOAR
- re3data
- GreyNet International Resources
- Grey Literature Reporter (NYAM)
- Policy Commons
- Dimensions
- OpenAlex
- Lens.org
- Semantic Scholar
- Google Scholar
- BASE
- CORE
- OAlster
- Europe PMC
- TRIP
- Clinical Trials Information System
- Overton

CATEGORY 2: MISC. TOOLS

Various search-related tools which poorly fit in other categories. Many of these tools require the user to provide a specific target (a list of websites) or are designed to search a specific, pre-defined set of deep-web databases.

- Greylitsearcher
- MedNar
- Google Programmable Search Engine
- gdeltdoc / Mediacloud
- Litsearchr
- Polyglot
- Grey Literature Reporter (Extension)



CATEGORY 3: AI SEARCH TOOLS

Tools which either recommend specific repositories for you to visit or software used traversing the web/citation networks to surface sources hard to find through standard keyword searches.

- Undermind.ai
- Keenious
- Amplyfi
- Sourcely
- Exa.ai
- Tavily
- Perplexity.ai
- Scispace
- Consensus
- Elicit

CATEGORY 4: DOCUMENT RETRIEVAL TOOLS

These are technical utilities for scraping, crawling, or bulk-downloading files and data.

- Unpaywall
- Zenodo_get / OSFclient
- Scrapy
- Crawl4AI
- ScrapeGraphAI
- Firecrawl
- StormCrawler

CATEGORY 5: SYSTEMATIC REVIEW TOOLS

These tools handle workflow management, screening, deduplication, snowballing (citation chasing), and data extraction.

- Citation Chaser
- SpiderCite
- synthesisr
- LatteReview
- DistillerSR
- Covidence
- Rayyan
- EPPI-Reviewer
- RevMan
- ASReview

- SRA / Tera Tools
- Picoportal
- SysRev
- Laser AI
- SWIFT Review & Active Screener
- Robot Analyst
- Robot Reviewer
- CADIMA
- Catchii
- LiteRev

APPENDIX 2. TOOLS DESCRIPTION

CATEGORY 1: DATABASES/REPOSITORIES

These are the “places” where data lives—indexes, registries, and aggregators that contain the actual records or documents.

1. Grey Matters – This tool is a curated checklist designed to standardize grey literature searching for systematic reviews and health technology assessments. Recently transitioned to Canada’s Drug Agency, it organizes international HTA agencies, clinical trial registries, and regulatory bodies into a structured protocol, ensuring that searches for non-commercial evidence are comprehensive, transparent, and reproducible. (<https://greymatters.cda-amc.ca/>)
2. OpenDOAR – As the definitive global directory of quality-assured open access repositories, OpenDOAR relies on manual curation to verify that each entry provides free, unrestricted access to academic outputs. It serves as a critical lookup service for researchers and administrators to identify compliant repositories, benchmark infrastructure, and analyse trends in open scholarship distribution worldwide. (<https://opendoar.ac.uk/>)
3. re3data – The Registry of Research Data Repositories is the global standard for identifying where to store and find research datasets across all disciplines. It indexes over 3,000 repositories, providing detailed metadata on persistent identifiers, certification, and licensing to help researchers comply with open data mandates and operationalize FAIR data principles. (<https://www.re3data.org/>)
4. GreyNet – Operating as a human and organizational network, GreyNet facilitates the study and preservation of grey literature. Its key resource, the GreyGuide Repository, aggregates conference proceedings, best practices, and metadata standards. It collaborates with national archives to preserve non-commercial research outputs, ensuring the long-term accessibility of the field’s intellectual history. (<http://www.greynet.org/>)
5. Grey Literature Reporter (NYAM) – Once the gold standard for manually curated public health reports, this tool has ceased active publication but remains a vital historical archive. Hosted by the New York Academy of Medicine, its collection preserves decades of non-commercial documents, offering researchers access to “fugitive” literature essential for understanding the history of medical and urban policy. (<https://www.nyam.org/library/collections-and-resources/grey-literature-report/>)
6. Policy Commons – This platform offers industrial-scale access to millions of policy reports from IGOs, NGOs, and think tanks, actively archiving content to prevent “link rot.” Its advanced technology extracts and indexes individual tables and figures from PDFs, allowing researchers to



pinpoint specific data visualizations within the massive corpus of global policy literature. (<https://policycommons.net/>)

7. Dimensions – Dimensions goes beyond standard citation indexing by creating a linked knowledge graph that connects publications to grants, patents, clinical trials, and policy documents. This structure allows funders and institutions to trace the full lifecycle of research impact, from initial funding investment through to commercial application and societal implementation. (<https://www.dimensions.ai/>)
8. OpenAlex – An open-source alternative to proprietary indices, OpenAlex provides a comprehensive catalog of the global research system under a CC0 public domain license. It utilizes a massive directed graph architecture to connect works, authors, and institutions, offering bulk data access and APIs that democratize bibliometric analysis and support reproducibility in science of science research. (<https://openalex.org/>)
9. Lens.org – Bridging the gap between academic science and industrial innovation, The Lens integrates massive patent and scholarly datasets. It features unique tools like PatSeq for searching biological sequences in patents and offers extensive citation mapping, allowing users to trace the influence of basic research on commercial products and global problem-solving. (<https://www.lens.org/>)
10. Semantic Scholar – Developed by the Allen Institute for AI, this tool uses natural language processing to combat information overload. It provides AI-generated “TL;DR” summaries, context-aware citation analysis, and a Semantic Reader that overlays definitions and references directly onto text, helping researchers comprehend complex scientific content faster and more effectively. (<https://www.semanticscholar.org/>)
11. Google Scholar – As the most comprehensive academic search engine, Google Scholar indexes a vast array of articles, theses, and case law. While it excels at surfacing obscure grey literature and maximizing reach, it operates as a “black box” with limited search customization and no public API, making it better for discovery than systematic analysis. (<https://scholar.google.com/>)
12. BASE – Managed by Bielefeld University Library, BASE aggregates metadata from thousands of OAI-PMH compliant repositories. It distinguishes itself with robust multilingual support and precise filtering options, such as Dewey Decimal Classification and license types, ensuring high-quality results by harvesting only from structured, academic sources rather than performing general web crawls. (<https://www.base-search.net/>)
13. CORE – Focusing on the aggregation of full-text open access content, CORE provides a massive corpus for text and data mining. It offers APIs and bulk data downloads that allow researchers to analyze millions of papers at a semantic level, while also providing dashboard services to help repositories manage and enrich their content. (<https://core.ac.uk/>)
14. OAIster – A massive union catalog of digital resources managed by OCLC, OAIster integrates open access repository records directly into the global WorldCat system. This strategic integration ensures that digital repository content is discoverable alongside physical library collections, normalizing open access materials for a broader audience of students and public library users. (<https://oaister.worldcat.org/>)
15. Europe PMC – As a designated Global Core Biodata Resource, Europe PMC provides open access to life sciences literature. It innovates by indexing preprints alongside peer-reviewed articles and using text-mining algorithms to link biological entities—like genes and proteins—directly to external datasets, transforming static articles into dynamic interfaces for biological discovery. (<https://europepmc.org/>)
16. TRIP – Designed specifically for evidence-based medicine, TRIP prioritizes high-level evidence like systematic reviews and clinical guidelines over primary studies. It supports PICO-formatted queries and features an AI-assistant, AskTrip, which synthesizes answers from its curated

database, helping clinicians find reliable answers to patient care questions rapidly at the point of need. (<https://www.tripdatabase.com/>)

17. Clinical Trials Information System (CTIS) – CTIS serves as the centralized portal for clinical trial authorization and oversight across the European Union. It harmonizes the submission process into a single digital application for up to 30 countries and enforces strict transparency mandates, ensuring public access to trial protocols, inspection reports, and results to benefit patient safety. (<https://euclinicaltrials.eu/>)
18. Overton – Overton tracks the impact of research on government policy by indexing millions of documents from parliaments, IGOs, and think tanks globally. By mapping citations between academic papers and policy outputs, as well as between policy documents themselves, it allows institutions to measure societal influence and trace the genealogy of legislative ideas. (<https://www.overton.io/>)

CATEGORY 2: MISC. TOOLS

- Greylitsearcher – Greylitsearcher is an R package and web-based Shiny application designed to facilitate systematic, transparent searches of organizational websites for grey literature. It leverages Google’s Site Search functionality to perform structured queries across specific URLs, allowing researchers to bypass manual browsing. The tool generates reproducible records of search strings and results, addressing the critical need for transparency in evidence synthesis workflows. (<https://www.eshackathon.org/software/greylitsearcher.html>)
- MedNar – MedNar is a free, federated search engine specifically engineered for the medical domain to access the “deep web.” Unlike standard crawlers, it broadcasts real-time queries to disparate, siloed databases—including government repositories and professional societies—aggregating and ranking the results to retrieve grey literature and data often missed by commercial publishers. (<https://mednar.com/>)
- Google Programmable Search Engine – This platform allows researchers to architect custom search engines restricted to specific “slices” of the web, such as trusted public health domains. It supports complex ranking algorithms and structured data filtering via Schema.org markup. Its JSON API enables automated, high-throughput querying for systematic review workflows, often serving as the backend for tools like Greylitsearcher. (<https://programmablesearchengine.google.com/about/>)
- gdeltdoc / Mediacloud – gdeltdoc is a Python wrapper for the GDELT Project, enabling quantitative analysis of global news events and tone across 65 languages. Media Cloud focuses on narrative ecosystems, utilizing network maps to analyze how stories propagate through media sources. Together, they facilitate high-level “culturomics” and framing analysis of global information flows. (<https://pypi.org/project/gdeltdoc/> | <https://www.mediacloud.org/>)
- Litsearchr – Litsearchr is an R package that automates the formulation of search strategies using text mining and keyword co-occurrence networks. By analyzing a “naïve search,” it identifies key terms based on network centrality and programmatically generates complex Boolean search strings in up to 53 languages, reducing subjective bias in systematic reviews. (<https://elizagrames.github.io/litsearchr/>)
- Polyglot – The Polyglot Search Translator is an automated tool designed to translate search string syntax across multiple bibliographic databases, such as converting PubMed queries to Embase or Web of Science formats. Validated in randomized trials, it significantly reduces translation time and syntax errors, though it requires manual verification for controlled vocabulary mapping. (<https://sr-accelerator.com/#/polyglot>)
- Grey Literature Reporter (Extension) – This Chrome extension augments manual browsing by functioning as a programmable, client-side scraper. It allows users to “train” the tool on specific website layouts to extract metadata from unstructured results. It exports findings into structured



formats like RIS, allowing grey literature to be directly integrated into reference management software. (<https://www.eshackathon.org/software/grey-lit-reporter.html>)

CATEGORY 3: AI SEARCH TOOLS

Tools which either recommend specific repositories for you to visit or software used traversing the web/citation networks to surface sources hard to find through standard keyword searches.

- Undermind.ai – Undermind is an agentic discovery engine designed for complex research tasks. Unlike standard search, it uses a “successive search” architecture where an AI agent iteratively queries, reads, and refines its strategy over several minutes. It navigates citation graphs in the Semantic Scholar database to find highly relevant, hard-to-find papers that keyword searches often miss. (<https://www.undermind.ai/>)
- Keenious – Keenious operates as a context-aware recommendation engine that integrates directly with word processors like Microsoft Word and Google Docs. By analyzing the text of your active draft, it uses dense vector retrieval to suggest relevant academic literature from the OpenAlex database. It emphasizes serendipitous discovery, connecting users to interdisciplinary research without requiring specific keywords. (<https://keenious.com/>)
- Amplyfi – Amplyfi is a strategic intelligence platform that mines the “Deep Web,” accessing unindexed content like patents, grey literature, and subscription databases. Using federated search technology, it aggregates structured and unstructured data to detect weak signals and emerging trends. It is primarily used by enterprises for competitive intelligence and risk monitoring rather than general academic search. (<https://amplyfi.com/>)
- Sourcely – Sourcely is an AI-powered citation assistant targeting undergraduates. Users paste their essays into the tool, and it identifies claims requiring support, suggesting relevant papers from a database of over 200 million sources. It focuses on efficiency, offering features like text-to-citation matching and automated formatting to streamline the reference generation process for students. (<https://www.sourcely.net/>)
- Exa.ai – Exa.ai is a neural search engine built specifically for AI agents rather than human users. Instead of keyword matching, it uses embeddings to search by meaning, allowing for “promptable search” where queries can be examples of desired content. It provides developers with clean, structured data and “Websets” to power RAG pipelines and autonomous systems. (<https://exa.ai/>)
- Tavily – Tavily is a search-as-a-service API designed to handle the logistics of Retrieval-Augmented Generation (RAG) for AI agents. It consolidates searching, scraping, and cleaning into a single API call, returning text chunks optimized for LLM context windows. It prioritizes factual sources and real-time data, functioning as the bridge between static models and the dynamic web. (<https://tavily.com/>)
- Perplexity.ai – Perplexity is a conversational answer engine that synthesizes information into narrative responses rather than lists of links. It combines proprietary indexing with models like GPT-4 and Claude 3 to provide grounded answers with inline citations. Its “Pro Search” feature acts as an agent, asking clarifying questions to refine results for complex queries. (<https://www.perplexity.ai/>)
- Scispace – Scispace functions as an interactive research workstation, combining search with a “Chat with PDF” copilot. It allows users to highlight complex text, math, or tables in academic papers and receive simple explanations. With support for multi-language queries and a database of over 200 million papers, it bridges the gap between discovery and comprehension. (<https://scispace.com/>)
- Consensus – Consensus is a search engine for scientific evidence that grounds answers strictly in peer-reviewed literature. Its “Consensus Meter” aggregates findings from multiple studies to answer binary questions (e.g., “Yes/No/Possibly”), providing a snapshot of scientific agreement.

It filters out non-academic noise to ensure high reliability for researchers and evidence-based decision-makers. (<https://consensus.app/>)

- Elicit – Elicit is an AI research analyst that automates the extraction of data for systematic reviews. It finds relevant papers and extracts specific variables—like sample size or dosage—into a structured matrix. By using sentence-level citations to link every claim back to the source PDF, it ensures accuracy while drastically reducing the time required for literature screening. (<https://elicit.com/>)

CATEGORY 4: DOCUMENT RETRIEVAL TOOLS

These are technical utilities for scraping, crawling, or bulk-downloading files and data.

- Unpaywall – Unpaywall is an open-access discovery engine that aggregates metadata from over 50,000 publishers and repositories. It leverages a massive database of Digital Object Identifiers (DOIs) to legally link users to free, peer-reviewed versions of paywalled articles. By integrating with library systems and providing a browser extension, it seamlessly unlocks scientific literature for researchers and the public. (<https://unpaywall.org/products/extension>)
- Zenodo_get / OSFclient – These command-line utilities facilitate the management of scientific data logistics. Zenodo_get automates the bulk downloading and integrity checking of large datasets from CERN's Zenodo repository. OSFclient acts as a bridge to the Open Science Framework, enabling researchers to programmatically clone projects, manage files, and ensure reproducible workflows across collaborative research environments. (https://github.com/dvolgyes/zenodo_get, <https://github.com/osfclient/osfclient>)
- Scrapy – Scrapy is an industrial-strength web scraping framework for Python, built on the Twisted event-driven architecture. Designed for high-throughput vertical crawling, it manages asynchronous requests efficiently to extract structured data from massive websites. Its robust middleware ecosystem enables complex customizations, from handling cookies and sessions to rotating proxies and bypassing anti-bot measures. (<https://docs.scrapy.org/>)
- Crawl4AI – Crawl4AI is an open-source, AI-friendly web crawler designed to convert web content into clean, LLM-ready Markdown. It features adaptive crawling algorithms that analyze page content to optimize extraction and remove noise like navigation bars. With support for structured data extraction and browser automation via Playwright, it bridges the gap between raw HTML and generative AI pipelines. (<https://docs.crawl4ai.com/>)
- ScrapeGraphAI – ScrapeGraphAI redefines scraping by using Large Language Models and graph logic to build modular extraction pipelines. Instead of brittle selectors, users define declarative graphs where nodes handle navigation and parsing. It supports local models via Ollama and cloud APIs, enabling semantic data extraction from websites and documents while minimizing the need for manual coding. (<https://scrapegraphai.com/>)
- Firecrawl – Firecrawl is a specialized scraping tool that turns entire websites into clean, well-structured Markdown or structured JSON for LLM ingestion. It handles complex challenges like JavaScript rendering, caching, and anti-bot mechanisms automatically. With features to crawl subpages and generate `llms.txt` files, it is specifically optimized to feed Retrieval-Augmented Generation (RAG) systems with high-fidelity data. (<https://www.firecrawl.dev/>)
- StormCrawler – StormCrawler is a Java-based SDK for building low-latency, distributed web crawlers on top of Apache Storm. Ideally suited for “web-scale” tasks, it treats URLs as continuous streams rather than static batches. It provides robust mechanisms for politeness, diverse fetchers, and sitemap parsing, making it the standard for building massive, resilient search indexes. (<https://stormcrawler.apache.org/>)



CATEGORY 5: SYSTEMATIC REVIEW TOOLS

These tools handle workflow management, screening, deduplication, snowballing (citation chasing), and data extraction.

- **Citation Chaser** – Citation Chaser is an R package and web-based Shiny application that automates the process of forward and backward citation chasing. By interfacing directly with the Lens.org database, it allows researchers to quickly retrieve lists of references and citing articles from a set of seed papers, visualizing the citation network to identify relevant literature efficiently. (<https://estech.shinyapps.io/citationchaser/>)
- **SpiderCite** – SpiderCite is a user-friendly citation searching tool integrated into the TERA (formerly SRA) suite. Designed for speed and simplicity, it utilizes the Lens.org corpus to generate forward and backward citations from an input library of included studies. It is particularly effective for identifying connected literature in complex search topics where keyword searches may fail. (<https://tera-tools.com/spidercite>)
- **synthesizr** – synthesizr is an R package designed to streamline the import, assembly, and deduplication of bibliographic datasets. It parses disparate file formats (like RIS and BibTeX) from various databases into a standardized structure, enabling reproducible data cleaning. This scriptable approach ensures that every step of the data wrangling process is transparent and auditable. (<https://cran.r-project.org/package=synthesizr>)
- **LatteReview** – LatteReview is a Python-based framework that utilizes Multi-Agent Systems and Large Language Models (LLMs) to automate systematic reviews. It supports multimodal analysis, allowing agents to review both text and images, and employs Retrieval-Augmented Generation (RAG) to ground decisions in external context. It allows for the creation of sequential or parallel agent workflows. (<https://github.com/PouriaRouzrokh/LatteReview>)
- **DistillerSR** – DistillerSR is an enterprise-grade platform designed for regulatory and high-stakes reviews. It features infinite configurability, rigorous audit trails, and AI-powered workflow optimization, including AI Reprioritization to bubble up relevant records and AI Error Checking to act as a safety net against human exclusion errors. It is widely used by pharmaceutical companies and government agencies. (<https://www.distillersr.com/>)
- **Covidence** – Covidence is a web-based platform that streamlines the standard systematic review production process, widely adopted by the Cochrane Collaboration. It features an ML-powered RCT Classifier that automatically identifies randomized controlled trials and uses active learning to sort records by relevance. Its intuitive interface focuses on seamless collaboration and workflow linearity. (<https://www.covidence.org/>)
- **Rayyan** – Rayyan is a popular free and freemium tool known for its mobile-first approach, allowing researchers to screen abstracts via a smartphone app. It utilizes an AI-powered 5-star rating system to learn user preferences and prioritize relevant studies. It facilitates blind screening and collaboration, making it a favorite among students and global research teams. (<https://rayyan.ai/>)
- **EPPI-Reviewer** – EPPI-Reviewer is a sophisticated web-based software for managing all stages of a review, with a focus on complex data and text mining. It integrates with OpenAlex to support “living” evidence maps and recently incorporated GPT-4o to automate screening and characterization of research. It is particularly strong in clustering and semantic analysis. (<https://eppi.ioe.ac.uk/cms/er4/>)
- **RevMan** – RevMan (Review Manager) is the specific software used for preparing and maintaining Cochrane Reviews. It enforces a protocol-driven workflow to ensure methodological rigor and includes built-in statistical tools for performing meta-analyses and generating Forest plots. The web-based version supports dynamic analysis reporting for living reviews. (<https://revman.cochrane.org/info>)

- **ASReview** – ASReview is an open-source, Python-based tool that utilizes active learning to accelerate the screening process. Uniquely, it offers a “Simulation Mode” that allows researchers to replay previous reviews to validate different machine learning models. It prioritizes transparency, allowing users to fully inspect and control the algorithms used for prioritization. (<https://asreview.nl/>)
- **SRA / Tera Tools** – The Evidence Research Accelerator (TERA), formerly the Systematic Review Accelerator (SRA), is a suite of modular tools designed to speed up specific review tasks. It includes the Deduplicator, Screenatron for rapid screening, and the Polyglot Search Translator. These tools are designed to be integrated into flexible workflows to remove manual bottlenecks. (<https://tera-tools.com/>)
- **Picoportal** – Picoportal is a web-based tool that leverages the PICO framework to streamline screening. It features “Ask AI,” a Generative AI assistant that allows users to query full-text PDFs directly to extract data. The interface automatically highlights Population, Intervention, Comparison, and Outcome elements in abstracts to accelerate human decision-making. (<https://picoportal.org/>)
- **SysRev** – SysRev is a platform built on FAIR data principles, hosting thousands of public, open-access review projects. It integrates active learning and an “Auto-labeler” that uses generative AI prompts to automatically classify documents. It is highly interoperable, offering an API that allows data scientists to pull curated datasets for machine learning research. (<https://sysrev.com/>)
- **Laser AI** – Laser AI is a next-generation tool architected specifically for Living Systematic Reviews (LSRs). It handles “delta updates” natively, presenting only new evidence when searches are updated while preserving previous data. It also utilizes NLP to automate data extraction into custom templates, ensuring reviews remain current with minimal manual effort. (<https://www.laser.ai/>)
- **SWIFT Review & Active Screener** – SWIFT comprises desktop and web tools focused on text mining and prioritization. Its standout feature is “Recall Estimation,” which provides a statistical estimate of the remaining relevant articles, giving researchers a data-driven justification to stop screening early. It also offers powerful topic modeling visualizations to scope the literature landscape. (<https://www.sciome.com/swift-activescreener/>)
- **Robot Analyst** – Robot Analyst is a text-mining tool developed by the National Centre for Text Mining (NaCTeM) to support public health reviews. It uses machine learning to cluster documents by topic and prioritize screening references. It is designed to minimize human workload in the study identification phase through visual analytics and active learning. (<https://research.manchester.ac.uk/en/impacts/saving-time-and-costs-for-evidence-based-public-health-interventi>)
- **Robot Reviewer** – Robot Reviewer is a specialized machine learning tool that automates the assessment of Risk of Bias in Randomized Controlled Trials (RCTs). It analyzes full-text PDFs to predict bias judgments across standard domains and extracts the specific sentences that support its decision, offering a transparent “white box” approach to automation. (<https://www.robotreviewer.net/>)
- **CADIMA** – CADIMA is a free web tool established to increase the rigor of evidence synthesis in agricultural and environmental sciences. It supports the entire review process with a focus on comprehensive documentation and reproducibility, ensuring that reviews can withstand critical scrutiny in policy-relevant fields. (<https://www.cadima.info/>)
- **Catchii** – Catchii is a free, web-based screening tool designed for accessibility and speed. It offers offline capabilities for researchers with unstable internet and includes an “Auto Reviewer” AI that provides text-based justifications for its inclusion/exclusion suggestions. It also features built-in duplicate detection, a feature often reserved for paid tools. (<https://catchii.org>)



- LiteRev – LiteRev is an automation tool developed to handle the entire review pipeline, from search to synthesis. It performs automated searches across open databases and uses vector space modeling and clustering to identify relevant papers. Validation studies have shown it can identify the vast majority of relevant papers while significantly reducing workload. (<https://literev.unige.ch/>)

APPENDIX 3. TOOLS EXCLUSION MOTIVES

Legend

- Strong exclusion motive
- Weak point which could motivate an exclusion
- Weak point, but not enough on its own for discarding the tool

CATEGORY 1: DATABASES/REPOSITORIES

- Grey Matters
 - **Static Format:** Distributed as a Word/HTML checklist, requiring brittle screen-scraping rather than programmatic database queries.
 - **No Federated Search:** Acts as a directory of links; you must build hundreds of separate scrapers for each target agency.
 - **Link Maintenance:** Pointers to external agency websites rot rapidly, requiring complex validation logic before every automated run.
- OpenDOAR
 - **Granularity Mismatch:** The API retrieves metadata about repositories (containers), not the individual articles or reports held within them.
 - **Integration Overhead:** Automation requires interacting with thousands of distinct repository interfaces, creating an unsustainable maintenance burden.
 - **Metadata Chaos:** Repositories use divergent schemas, making it impossible to construct a unified, reliable search strategy across all sources.
- re3data
 - **Object Mismatch:** Designed for indexing raw datasets (CSV, genomics), not the textual grey literature reports required for your reviews.
 - **Pointer System:** Operates as a registry of repositories without an internal search index for the actual content or documents.
 - **Decentralized Access:** Finding data requires querying the registry and then separately querying each external repository's unique API.
- GreyNet
 - **Low Recall:** The collection is niche (Library Science focus) and too small to support comprehensive searches for clinical health conditions.
 - **Manual Ingestion:** Relies on voluntary author submission, resulting in significant coverage gaps compared to active harvesting systems.
 - **Technical Maturity:** Lacks a robust, high-throughput REST API suitable for integrating into a Python/R automated pipeline.
- Grey Literature Reporter (NYAM)

- **Discontinued Service:** The database ceased adding new records in 2017, rendering it methodologically unsound for modern systematic reviews.
- **Link Rot:** As a dormant archive, a significant portion of the external pointers are broken and will fail during retrieval.
- **Legacy Infrastructure:** Does not support modern interoperability standards or APIs, forcing the use of unstable web scraping methods.
- Policy Commons
 - **Prohibitive Cost:** Deep API access is typically reserved for enterprise licenses and is cost-prohibitive for individual researchers.
 - **Anti-Scraping:** Commercial bot detection measures will likely flag and block an automated pipeline running 90 sequential searches.
 - **Opaque Coverage:** Proprietary crawler logic makes it difficult to verify search exhaustiveness for PRISMA reporting compliance.
- Dimensions
 - **Paywalled API:** The Analytics API required for complex queries and bulk extraction is a premium enterprise product, not free.
 - **Usage Restrictions:** Free access is strictly for non-commercial personal use; automated harvesting for a team pipeline violates terms.
 - **Structure Bias:** Excellent for structured data (grants/patents) but captures less unstructured PDF grey literature than Google Scholar.
- OpenAlex
 - **Classification Ambiguity:** Often misclassifies grey literature as “journals” or “other,” making precise filtering for reports difficult.
 - **DOI Reliance:** Heavily dependent on DOIs, meaning government reports and white papers without DOIs are frequently excluded.
 - **No Full-Text:** Provides metadata only; your pipeline must implement complex link-resolving logic to fetch the actual PDFs.
- Lens.org
 - **Commercial Licensing:** Free API access is revoked if the project has commercial funding or intent, requiring an expensive license.
 - **Rate Limiting:** Free tier rate limits may bottleneck a pipeline attempting to process 90 complex boolean queries concurrently.
 - **Domain Skew:** While excellent for patents, coverage of clinical guidelines and policy reports is less comprehensive than specialized tools.
- Semantic Scholar
 - **Content Bias:** Prioritizes academic papers and preprints, systematically under-indexing government reports and NGO white papers.
 - **Indirect Access:** API provides links to publisher landing pages, not PDFs, requiring a custom “resolver” to navigate paywalls.
 - **Mining Restrictions:** License agreements often restrict the bulk harvesting of data for the creation of derivative datasets.
- Google Scholar



- **Bot Blocking:** Aggressively bans IPs detecting automation; running 90 searches requires expensive proxy rotation and CAPTCHA solving.
- **No API:** Lacks an official API, forcing reliance on third-party scrapers that break whenever the HTML structure changes.
- **Metadata Truncation:** Search results often display incomplete metadata, forcing the scraper to visit every source URL for validation.
- BASE
 - **IP Blocking:** Strict rate limiting will interpret 90 concurrent search strategies as a Denial of Service attack.
 - **Metadata Heterogeneity:** Inconsistent tagging across source repositories makes filtering by “publication type” (e.g., reports) unreliable.
 - **Protocol Complexity:** Deep integration often requires OAI-PMH handling, which is more verbose and complex than modern REST APIs.
- CORE
 - **API Throttling:** The free API has strict quotas (e.g., 5 requests/10s), creating a massive bottleneck for high-volume batch processing.
 - **Version Duplication:** Often indexes multiple versions (preprint/postprint) of the same paper, inflating screening volume with duplicates.
 - **Usage Policy:** Terms distinguish between “mining” and “reading”; bulk reproduction for human screening may violate acceptable use.
- OAIster
 - **Legacy Interface:** Lacks a modern lightweight API, relying on subscription-based library discovery layers (OCLC/WorldCat) for access.
 - **Link Rot:** As a union catalog of metadata, it does not guarantee the target resource exists, leading to high 404 errors.
 - **Duplicate Saturation:** Aggregates records from other aggregators, resulting in massive duplication that is difficult to purge automatically.
- Europe PMC
 - **Domain Scope:** Strictly biomedical; will miss relevant grey literature on the economic or psychosocial aspects of health conditions.
 - **Preprint Noise:** Aggregation of multiple preprint servers floods results with unreviewed versions that require robust deduplication.
 - **Shallow Grey Lit:** Lacks systematic indexing of “deep” grey literature like government agency reports or NGO white papers.
- TRIP
 - **Paywall Barrier:** Advanced grey literature filters and regulatory content are often gated behind the “Pro” subscription.
 - **No Public API:** Lacks a documented public API for individuals; automated access is usually a bespoke institutional arrangement.
 - **Export Limits:** Batch export functionality is often capped, requiring complex pagination logic to retrieve large result sets.

- Clinical Trials Information System
 - **No Public API:** The system explicitly lacks a public machine-readable interface, forcing the use of fragile “headless browser” scrapers.
 - **Data Fragmentation:** Historic data is split between EudraCT and CTIS during the transition, requiring searches across two disparate systems.
 - **Download Restrictions:** The portal does not support bulk downloads, forcing the pipeline to fetch documents one by one.
- Overton
 - **Premium Add-on:** API access is not standard and is sold as an expensive add-on to institutional subscriptions.
 - **Volume Limits:** Licensing terms often cap data extraction volume, creating legal risks for a 90-condition bulk review.
 - **Proprietary Taxonomy:** Mapping 90 specific health conditions to Overton’s internal topic classification requires significant manual validation.

CATEGORY 2: MISC. TOOLS

- Greylitsearcher
 - **Abandonware Status:** Last updated circa 2022; unmaintained code risks immediate breakage if Google changes frontend.
 - **Blocking Risk:** High-volume scraping triggers Google’s anti-bot measures, leading to likely IP bans.
 - **Retrieval Gap:** Returns only URLs or metadata, not full-text PDFs, requiring a separate fetching pipeline.
- MedNar
 - **No Automation API:** Lacks a public batch API; designed for manual, real-time human interaction only.
 - **Export Restrictions:** Strict caps on exportable records force fragmented, manual data extraction sessions.
 - **Black Box:** “Intelligent” query translation obscures syntax interpretation, preventing validation of search strategy.
- Google Programmable Search Engine
 - **Recall Cap:** Hard limit of 100 results per query systematically excludes relevant “long tail” evidence.
 - **Reproducibility Crisis:** Dynamic, proprietary ranking algorithms change results over time, violating systematic review standards.
 - **Syntax Weakness:** Incompatible with complex boolean logic and proximity operators required for precision searching.
- gdeltdoc / Mediacloud
 - **Wrong Scope:** Indexes journalism and news, not primary clinical guidelines or technical health reports.
 - **Copyright Barriers:** API provides metadata/URLs but blocks full-text access, preventing automated text mining.



- **Algorithmic Bias:** Sentiment analysis algorithms introduce sociological bias irrelevant or harmful to clinical evidence.
- Litsearchr
 - **Not a Searcher:** Designed for keyword discovery, not document retrieval; cannot execute searches or fetch files.
 - **Dependency Fragility:** Non-CRAN package with unstable dependencies creates high risk of breakage in long-term pipelines.
 - **Manual Bottleneck:** Requires human judgment to define keyword cutoffs, preventing fully automated batch processing.
- Polyglot
 - **Translation Errors:** Rule-based translation frequently causes “silent failures” in complex boolean logic and field tags.
 - **Web Incompatibility:** Designed for structured bibliographic databases, not the simple search interfaces of grey literature sites.
 - **No Execution:** Generates query strings but lacks capabilities to execute searches or retrieve documents.
- Grey Literature Reporter (Extension)
 - **Manual Training:** Requires manual point-and-click training for each site; unscalable for 90 conditions.
 - **Layout Fragility:** Scrapers break instantly if target website layout or CSS changes, requiring constant maintenance.
 - **Hardware Scaling:** Browser-based extension cannot run in headless cloud environments, limiting processing throughput.

CATEGORY 3: AI SEARCH TOOLS

- Undermind.ai
 - **High Latency:** Recursive agent loops take minutes per query, making synchronous batch processing impossible.
 - **Unverifiable Strategy:** Dynamic, non-deterministic query generation complicates the structured reporting and traceable documentation required for systematic reviews.
 - **Academic Only:** Relies on Semantic Scholar, missing the government websites needed for grey literature.
- Keenious
 - **Workflow Mismatch:** Designed for analyzing existing text drafts, not for initial keyword-based document retrieval.
 - **Limited Scope:** Indexes only academic metadata (OpenAlex), missing policy documents and agency reports.
 - **No Bulk API:** Lacks a public API for high-volume, automated query execution.
- Amplyfi
 - **Enterprise Barriers:** Sales-led model blocks self-serve API access needed for rapid prototyping.



- **Business Bias:** Relevance algorithms prioritize market intelligence over the clinical data needed for HTA.
- **Black Box:** Proprietary crawling logic prevents auditing of search sensitivity or specificity.
- Sourcely
 - **Hallucination Risk:** Prioritizes finding plausible-looking sources over accurate ones, risking citation fabrication.
 - **Student Focus:** Tuned for essay writing, lacking the technical depth required for systematic reviews.
 - **TOS Violation:** Terms explicitly prohibit automated access and scraping of their services.
- Exa.ai
 - **No Binary Downloads:** Returns parsed HTML text but cannot natively download PDF files.
 - **Firewall Blocks:** Crawlers often fail against secured government domains, leaving gaps in evidence.
 - **Restrictive Terms:** Terms limit caching data for derivative datasets, creating liability risks.
- Tavily
 - **Context Truncation:** Returns snippets optimized for LLMs, losing methodological details from full texts.
 - **High Latency:** Extracting raw content significantly slows down throughput for large batch operations.
 - **Generalist Bias:** Prioritizes mainstream web content over obscure technical government reports.
- Perplexity.ai
 - **Citation Fabrication:** Known to hallucinate sources or misattribute details, compromising review integrity.
 - **Platform Risk:** Active copyright litigation creates risk of sudden API shutdown or restriction.
 - **Summarization Bias:** Optimizes for concise answers, filtering out “long tail” documents you need.
- Scispace
 - **Walled Garden:** Analyze-only tool; requires manual upload and cannot crawl the web.
 - **Upload Limits:** File size and quantity caps create bottlenecks for processing thousands of documents.
 - **Opaque Ranking:** Relevance sorting prioritizes precision over the recall needed for systematic reviews.
- Consensus
 - **Academic Lock-in:** Indexes only scientific papers, missing government reports and grey literature.
 - **No File Fetching:** Extracts claims but does not provide direct file download paths.
 - **Cost Scaling:** Pricing models based on credits are expensive for high-volume extraction.
- Elicit
 - **No Web Crawling:** Processes only uploaded files or internal library; cannot spider websites.



- **High Cost:** “Systematic Review” features are expensive compared to custom Python scripts.
- **Verification Needed:** ~90% accuracy requires human oversight, reducing automation benefits.

CATEGORY 4: DOCUMENT RETRIEVAL TOOLS

- Unpaywall
 - **DOI Reliance:** Returns zero results for government reports and grey literature that lack Digital Object Identifiers.
 - **No Search:** Acts as a metadata resolver, not a search engine; cannot accept keyword queries to find new sources.
 - **Academic Bias:** Heavily prioritizes peer-reviewed journals, underrepresenting the agency reports and clinical guidelines needed for your review.
- Zenodo_get / OSFclient
 - **Domain Restricted:** Hard-coded to specific repositories (Zenodo/OSF) and cannot crawl the external agency websites where most reports reside.
 - **No Discovery:** Requires prior knowledge of specific Record IDs; cannot perform keyword searches to discover new relevant documents.
 - **Siloed Navigation:** Cannot follow hyperlinks or citation trees that lead outside the specific repository to other web sources.
- Scrapy
 - **Maintenance Burden:** Requires writing and updating rigid extraction rules for every single target site, creating unsustainable technical debt.
 - **JavaScript Blindness:** Cannot render dynamic content or modern web apps, missing grey literature hosted on JavaScript-heavy platforms.
 - **Steep Curve:** Configuring asynchronous architecture and middleware requires advanced engineering skills, creating a high barrier to entry.
- Crawl4AI
 - **Resource Intensive:** Running headless browsers for every request consumes massive RAM/CPU, causing bottlenecks when scaling to thousands of pages.
 - **Weak PDF Handling:** Optimized for HTML-to-Markdown; extracting structured data from binary PDFs requires complex, fragile secondary pipelines.
 - **Opaque Logic:** Reliance on AI/heuristic extraction creates a “black box,” making it difficult to debug why specific links were missed.
- ScrapeGraphAI
 - **High Costs:** Sending page content to LLMs for every navigation decision incurs astronomical API token costs at scale.
 - **Hallucination Risk:** LLM agents may invent links or misinterpret navigation, undermining traceability and auditability requirements in systematic review contexts..
 - **High Latency:** The “stop-and-think” processing time for agents makes it too slow for high-throughput discovery workflows.



- Firecrawl
 - **Formatting Loss:** Converting complex PDF reports to Markdown often destroys table structures and layout data essential for health reviews.
 - **Binary Handling:** Default behavior favors text extraction over file retrieval, complicating the archival of original binary assets (PDFs).
 - **SaaS Limits:** Reliance on the hosted API subjects the workflow to rate limits that may interrupt large-scale crawls.
- StormCrawler
 - **Infrastructure Complexity:** Requires deploying and managing a distributed cluster (Apache Storm/Zookeeper), creating excessive DevOps overhead.
 - **Hardware Demands:** High memory and CPU requirements make it cost-inefficient and unsuitable for local execution or small cloud instances.
 - **Customization Barrier:** Configuring extraction topology requires complex Java programming, lacking the accessibility of Python-based tools.

CATEGORY 5: SYSTEMATIC REVIEW TOOLS

- Citation Chaser
 - **Index Dependency:** Relies on Lens.org; misses unindexed grey literature lacking DOIs.
 - **No Retrieval:** Returns bibliographic metadata only; cannot download full-text documents.
 - **Update Latency:** Aggregator lag risks missing very recent grey literature reports.
- SpiderCite
 - **Circular Citations:** Academic seeds yield academic results; fails to bridge to grey literature.
 - **Manual Seed Need:** Requires identifying high-quality seed papers for all 90 conditions manually.
 - **Interface Bound:** Designed for interactive sessions; lacks headless batch processing capabilities.
- synthesizr
 - **No Fetching:** Library for cleaning and importing only; possesses no web crawling capabilities.
 - **Format Rigidity:** Requires structured RIS/BibTeX; fails on raw, unstructured web scraps.
 - **Downstream Tool:** Solves data merging; does not address discovery or downloading.
- LatteReview
 - **High Cost:** Multi-agent LLM approach is prohibitively expensive for 90 high-volume reviews.
 - **Prototype Instability:** Experimental framework; lacks production-grade robustness and documentation.
 - **Input Bound:** Processes provided text streams; cannot autonomously navigate web sources.
- DistillerSR
 - **Manual Ingestion:** Optimized for wizard uploads; offers no native web crawler.
 - **Prohibitive Cost:** Enterprise licensing is expensive for single-researcher scale.



- **GUI Centric:** Designed for human interaction; resists headless programmatic automation.
- Covidence
 - **Closed Ecosystem:** No public API for creating reviews or ingesting data programmatically.
 - **Zero Discovery:** Strictly for screening; offers no search or fetch tools.
 - **Rigid Deduplication:** Fails on unstructured grey literature that lacks standard identifiers.
- Rayyan
 - **No Ingestion API:** Requires manual file uploads for every single review project.
 - **Abstract Dependent:** AI ranking fails on sparse grey literature metadata lacking abstracts.
 - **Screening Only:** Binary inclusion/exclusion tool; offers no automated data extraction.
- EPPI-Reviewer
 - **Complex Configuration:** Steep learning curve for setting up text mining on 90 reviews.
 - **No Web Crawler:** Text mining tools operate only on data already imported manually.
 - **Performance Issues:** Complex web application can be slow with massive, unstructured datasets.
- RevMan
 - **Category Mismatch:** Writing and analysis tool; not designed for searching or screening.
 - **Manual Entry:** Designed for manual entry of final included studies only.
 - **Rigid Templates:** Enforces Cochrane schemas that are unsuitable for diverse grey reviews.
- ASReview
 - **Screening Only:** Active learning ranker; cannot find or download papers from the web.
 - **Structured Input:** Requires clean CSV/RIS; fails on raw unstructured text inputs.
 - **Pre-process Burden:** Needs OCR and text extraction before it can function effectively.
- SRA / Tera Tools
 - **Platform Instability:** Transitioning from SRA to TERA; features are currently in flux.
 - **Fragmented Suite:** Disparate micro-tools; no unified pipeline for automated batch processing.
 - **Manual Interface:** Browser-based tools require manual file movement between steps.
- Picoportal
 - **No Search:** Import-to-screen workflow; contains no discovery or fetching capabilities.
 - **Import Dependency:** Relies entirely on user-provided structured files for every project.
 - **Freemium Limits:** Citation and project caps are likely exceeded by 90 reviews.
- SysRev
 - **Discovery Gap:** Python client manages projects but does not crawl the web.
 - **Openness Risks:** Public project defaults risk exposing proprietary search scopes.
 - **Limited Extraction:** Focuses on collaborative screening rather than complex data scraping.
- Laser AI
 - **High Cost:** Enterprise solution for living reviews; likely over budget.



- **Structured Bias:** Monitors structured databases like PubMed, not the unstructured grey web.
- **Feature Overkill:** Living review management features add unnecessary overhead for this task.
- SWIFT Review & Active Screener
 - **Desktop Constraints:** SWIFT Review is local; difficult to automate in a cloud pipeline.
 - **Vocabulary Mismatch:** Tuned for MeSH/Biomedical terms; struggles with policy language.
 - **No Fetching:** Requires an imported corpus; cannot autonomously browse repositories.
- Robot Analyst
 - **Research Prototype:** Academic tool; lacks commercial support and reliability guarantees.
 - **No Automation API:** Interactive web tool; cannot be driven programmatically for batch tasks.
 - **Data Privacy:** Public server usage may expose proprietary research data.
- Robot Reviewer
 - **RCT Exclusivity:** Trained on Randomized Controlled Trials; fails on grey literature reports.
 - **Extraction Only:** Analyzes PDFs for bias; cannot find or download them.
 - **Schema Mismatch:** Risk of Bias fields are irrelevant for non-trial grey literature.
- CADIMA
 - **Documentation Focus:** Built for reporting and audit trails, not for execution.
 - **Legacy Interface:** Slow, manual navigation makes managing 90 projects painful.
 - **Data Silo:** No connection to external web sources; requires manual uploading.
- Catchii
 - **Browser Limits:** Local storage relies on browser cache; crashes with large datasets.
 - **Screening Only:** Mobile-first screening tool; offers no search or fetch functionality.
 - **Manual Handling:** Relies on CSV uploads; lacks API for automated pipelines.
- LiteRev
 - **Manual Intervention:** User must select topics manually; prevents fully headless automation.
 - **Metadata Dependency:** Needs abstracts for clustering; fails on sparse grey literature data.
 - **Narrow Validation:** Tested on specific domains; risky to generalize to 90 topics