Research-AI

Software Project Proposal

David Grote Beverborg

1. Introduction client

The client is the Utrecht University – ITS – research and data management (RDMS). This department manages research applications that are supportive for research. Besides managing applications, RDMS also develops application that have a substantial user base across the Netherlands. Examples are for Yoda and Ricgraph.

2.Goal of the project

1.1 Goal and overview/Doel en overzicht

The goal of this project is to develop a Research AI Portal: a reusable, open-source web application that provides AI-assisted access to institutional research information in a controlled, transparent, and policy-aware manner.

Research AI will be used by hundreds of employees at the Utrecht University at first, and later (when successful) by thousands of employees at other Dutch and international universities.

The system builds upon an existing Research AI stack developed by the client, including integrations with Pure (CRIS), Ricgraph, OpenAlex, and Retrieval-Augmented Generation (RAG) pipelines. The primary objective of this Software Project is not to create experimental AI components, but to integrate, consolidate, extend, and productize these existing building blocks into a coherent and maintainable system suitable for use within a university environment.

The main function of the system is to allow authenticated users to query and explore research information at the level of persons, organizations, and collaborations, using both conversational interaction and predefined, high-quality prompts for common research and policy use cases (e.g. researcher profiles, funding CVs, and organizational narratives).

The intended users of the system are:

- researchers,
- research managers,
- policy advisors,
- research support staff,
- communication advisors,
- students.

Possible use cases for Research AI are listed below:

- A researcher wants to create a CV for the UU profile page, based on the actual research that the researcher has done. **Result:** Research AI creates a narrative CV.
- A policy advisor explores connections via co-authored publications or shared researchers across faculties. **Result:** Research AI presents concrete publication-based links, including a summary.
- A researcher asks for other publications similar to theirs, based on abstract similarity. **Result:** Research AI returns relevant internal publications and authors, supporting awareness of related work.
- A research manager of molecular biology wants to know what the top 5 external organizations (other institutes or universities) his department has worked with, and asks for a summary of these collaborations. **Result:** Research AI delivers the top 5 external organizations.
- A research support officer wants to encourage researchers to collaborate outside their faculty, and wants to create a list of possible external collaborations. **Result:** Research AI delivers a list of researchers with overlapping research subjects.

As can be see, the software is used in institutional settings such as research evaluation, funding preparation, strategic analysis, and internal reporting. Access to information is governed by authentication via SURFconext and fine-grained authorization rules (for example, allowing organization-level insights while restricting access to personal data).

The system is bilingual by design, supporting English and Dutch, and is configurable to support additional languages in the future.

A key requirement of the project is that the resulting software is fully open source and portable, allowing other universities to deploy the system with minimal configuration by connecting their own CRIS (Current Research Information System) and identity infrastructure refinements.

1.2 Related systems

The project builds upon and integrates several existing systems and software components that are already used within the domain of research information management. These systems form the technical and conceptual foundation of the Research AI Portal.

1.2.1 Pure (CRIS)

The Research Portal is managed by the backend Pure. It is the primary and authoritative source for institutional research metadata, including publications, abstracts, persons, organizations, projects, and persistent identifiers. Pure is managed by Elsevier (and the only non-open-source software that is used), and is used by almost all universities in The Netherlands.

Existing integrations and scripts for retrieving and processing Pure data are already available. Within this project, these integrations will be consolidated, stabilized, and integrated into a coherent application architecture.

1.2.2 Ricgraph

Ricgraph is an internal research knowledge graph that models relationships between research entities such as persons, organizations, publications, concepts, and collaborations.

Ricgraph provides structured, graph-based insights that are not explicitly available in Pure, such as collaboration networks and semantic relationships. In this project, Ricgraph is used in a read-only manner as a structured knowledge layer that complements Pure metadata and supports entity- and collaboration-based queries.

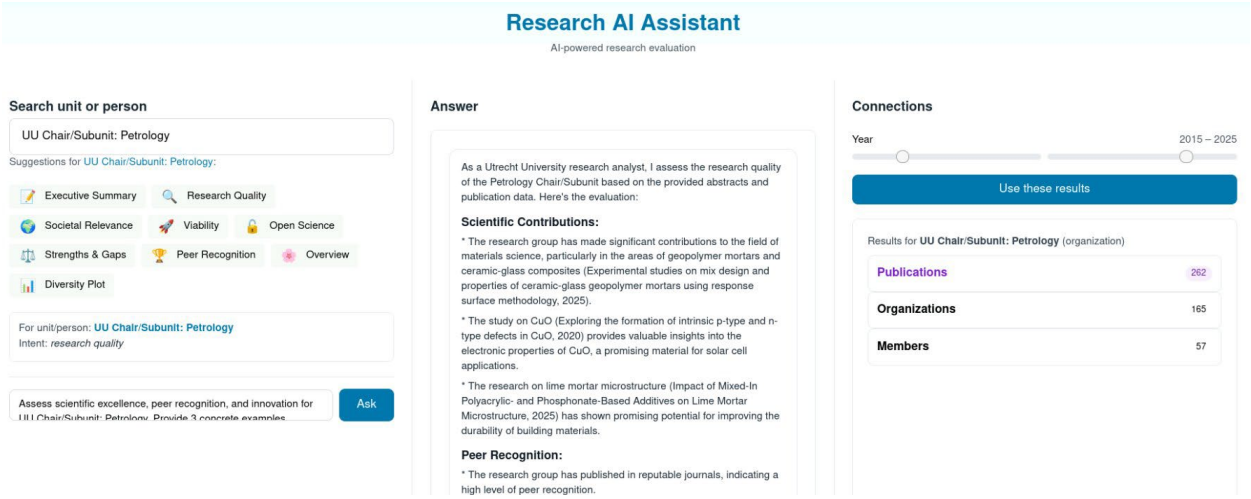
1.2.3 OpenAlex

OpenAlex is used as an external, open data source to enrich research outputs with concepts, fields of science, and contextual metadata. OpenAlex data is linked to Pure records via persistent identifiers (e.g. DOI) and serves as an enrichment layer rather than a source of record.

1.2.4 Existing Research AI prototype

The client has developed working prototype and proof-of-concept components, including:

- data retrieval and enrichment scripts,
- early Retrieval-Augmented Generation (RAG) pipelines,
- prompt templates for research-related use cases.



These components demonstrate feasibility and form the starting point for this Software Project. The focus of the project is to transform these prototypes into a robust, maintainable, and reusable software system.

1.2.5 Related research analytics tools

While various research analytics dashboards (e.g. Business Objects) and reporting tools exist, these typically rely on predefined reports and require specialist expertise. The proposed system differs by enabling interactive, AI-assisted exploration while maintaining strict transparency about data sources, authorization boundaries, and limitations.

All documentation, example code, and system descriptions related to these existing systems will be made available to the student team at the start of the project.

1.3 Domain knowledge

The domain of this project is research information management and research intelligence within a university environment, with a specific focus on the responsible and controlled use of AI models in institutional software systems.

Students are not expected to have prior domain knowledge. During the project, they will learn from the client about:

- how research information is structured within a university (e.g. publications, persons, organizations, projects, collaborations);
- the role of a Current Research Information System (CRIS), such as Pure, in research administration, evaluation, and reporting;
- how graph-based representations, such as Ricgraph, are used to model relationships and collaborations between research entities;
- how external open data sources (e.g. OpenAlex) can complement internal institutional data;
- how AI language models are used as components within a software system, rather than as standalone or experimental tools.

With respect to **AI usage,** students will learn:

- how Retrieval-Augmented Generation (RAG) is used to constrain AI model output to verified institutional data;
- how prompt templates are designed for specific, repeatable use cases (e.g. researcher profiles, funding CVs, organizational narratives);
- how to ensure traceability and explainability, including source attribution and explicit handling of uncertainty (“no answer possible”);
- how authorization and policy constraints influence what an AI system is allowed to generate; the limitations and risks of AI models in policy-sensitive and data-governed environments.

The required domain knowledge is available and will be **transferred to the students** through:

- an introductory presentation by the client in the first project week;
- Access to the repository of the proof-of-concept research AI
- example datasets prompt templates, and API specifications;
- ongoing clarification during sprint reviews and refinement sessions.

The emphasis of the domain knowledge transfer is on applied AI within a professional software engineering context, rather than on machine learning theory or model training.

1.4 Why is the project interesting for the students

This project is interesting for students because it combines modern software engineering, applied AI, and real-world institutional systems in a professional and meaningful context.

The project closely resembles professional software development in a research IT and data-intensive environment and provides experience that is directly relevant for careers in software engineering, data engineering, and applied AI within public and institutional organizations.

Students work on a system that is not a toy example, but is based on real research data, real users, and real constraints. Rather than experimenting with isolated AI models, students will learn how to embed AI models responsibly into a larger software architecture, where correctness, transparency, and authorization are essential. With this project, you will gain much sought after skills for your professional career.

Key aspects that make this project attractive include:

- working with large-scale, structured data from systems such as Pure and Ricgraph;
- integrating AI language models through Retrieval-Augmented Generation instead of unconstrained generation;
- designing and implementing policy-aware authorization that directly affects what the AI system is allowed to generate;
- building a bilingual application (English and Dutch) with configurable language support;
- contributing to a fully open-source system intended for reuse by multiple universities.

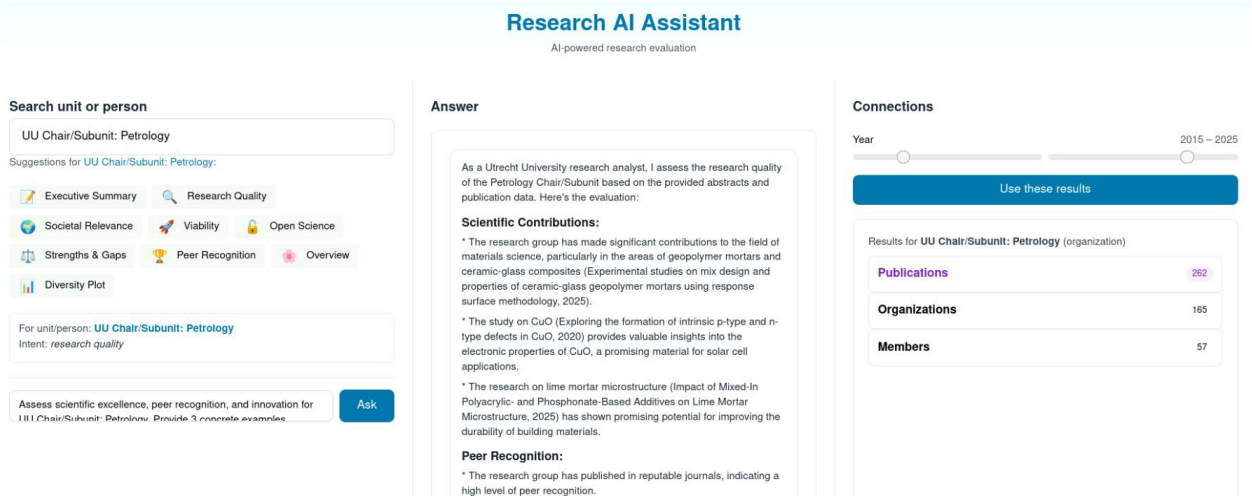
2 Main functionalities

2.1 Main functionalities

The system provides AI-assisted access to institutional research information through a secure, explainable, and configurable web application.

A significant part of the technical foundation already exists in the form of prototypes, scripts, and working integrations developed by the client. The focus of this Software Project is therefore not to start from scratch, but to integrate, consolidate, extend, and productize these existing components into a coherent, maintainable system.

A short overview of the existing web app is this:



In the left column, one can search for an organizational unit or a person. Once this is chosen, two things happen:

1. Via Ricgraph all connections are retrieved and transferred to the right column
2. Context aware pre-programmed prompts will appear. These are not all functional at this moment, hence different prompt buttons appear for persons and organizations.
3. The prompt button is executed. The answer appears in the middle column.

The main functionalities are grouped as core baseline functionality and extensions and refinements.

2.1.1 Core baseline functionality

These functionalities already exist partially and will be reused, integrated, and hardened during the project.

Research data integration

- Retrieval of organizational data, research metadata and abstracts from Pure (CRIS).

- Retrieval of structured relations between persons, organizations, publications from Ricgraph
- Enrichment of research outputs using OpenAlex, linked via persistent identifiers.

Entity-based data model

- Support for three entity types:
    - Persons (researchers)
    - Organizations (faculties, departments, etc.)
    - Collaborations (between organizations, and persons. For now, this only appears for persons and organizations. The project closely resembles professional software development in a research IT and data-intensive environment and provides experience that is directly relevant for careers in software engineering, data engineering, and applied AI within public and institutional organizations)

- Consistent identifiers and entity context handling across the system.

AI-assisted querying (RAG)

- Existing Retrieval-Augmented Generation pipelines used as a foundation.
- Answers are generated strictly based on retrieved institutional data.
- Source attribution is included with every generated answer.
- Explicit “no answer possible” behavior when data is insufficient.

Bilingual operation

- Support for English and Dutch in prompts and AI-generated output.
- Language selection based on user preference.

2.1.2 Extensions and refinements

These functionalities turn the existing components into a complete, usable product.

Authentication and authorization

- Authentication via SURFconext.
- Role- and policy-based authorization (e.g. allowing organization-level access while restricting person-level data).
- Enforcement of authorization rules at both API and user interface level.

Structured prompt packs

- Well-defined prompt collections per entity type, for example:
    - Persons: NWO-style CV generation, structured researcher profiles, executive summary, societal impact

    - Organizations: SEP-style narratives, trends over time, research strengths and positioning, layman's summary, executive summary, societal impact.
    - Collaborations: joint strengths, thematic overlap, partnership summaries.

- Each prompt produces structured, predictable output suitable for reuse.

User interface

- A web-based interface supporting:
    - entity selection,
    - prompt selection,
    - conversational interaction,
    - transparent presentation of sources and metadata.

- Clear visual indication of authorization boundaries.

Search and discovery

- Search functionality for persons, organizations, and collaborations.
- Respecting authorization constraints in both search results and detail views.

Configurability and portability

- Configuration-driven setup to allow deployment at other universities.
- Clear separation between institution-specific configuration and generic application logic.

Extra data import

- Media items
- Activities
- Prizes

2.1.3 Explicit non-goals

To keep the project feasible within the given timeframe, the following are explicitly out of scope:

- Training or fine-tuning of language models.
- Automated ingestion of arbitrary external data sources.
- Complex visual analytics dashboards.

2.2 Visualization and artwork

The system is experienced as a professional web application intended for researchers, research managers, and policy advisors.

- The interface focuses on:
    - clarity and transparency,

    - structured interaction rather than free-form experimentation,
    - explicit visibility of data sources and limitations.

- The visual style is clean, minimalistic, and functional, aligned with existing research-support tools.

- No specialized artwork is required.

3 Technology Requisites

3.1 Choices and options

Several implementation choices, including backend framework selection, frontend stack, internal modularization, and supporting tooling, may be made by the student team in consultation with the client, within the constraints outlined below.

Backend

The project involves the development of a data-intensive, AI-assisted web application that builds on existing institutional components. The backend is implemented in Python, as integrations with Pure, Ricgraph, and AI components already exist in this language. A modern Python web framework (such as FastAPI or Flask) will be used; the final choice may be made by the student team provided it remains compatible with the existing codebase.

Frontend

The application has a web-based frontend. The specific frontend approach may be chosen by the student team, as long as it results in a maintainable and accessible user interface.

AI functionality

AI functionality is implemented using large language models within a Retrieval-Augmented Generation (RAG) architecture. LLMs are treated as replaceable components; model training and fine-tuning are out of scope. Prompt templates are considered part of the application logic and must be versioned and documented.

Database

A vector database is used for embedding-based retrieval. External systems such as Pure, Ricgraph, and OpenAlex are accessed in a read-only manner via APIs or exports, with harvested data stored locally to support reproducibility

Authentication.

Authentication is handled via SURFconext using standard federation protocols. Authorization is implemented at the application level and must be enforced consistently across APIs and user interfaces.

Portability

The system must be open source, portable to other institutions, configurable without code changes, transparent in its AI output, robust against incomplete data, and security-aware. AI models may run locally or via a configurable backend, and specialized hardware such as GPUs may be used during development.

3.2 Existing software components

The project builds on an existing Research AI software stack developed by the client and on widely used open-source components. Client-provided software includes integrations with Pure and Ricgraph, as well as prototype Research AI components such as data pipelines, Retrieval-Augmented Generation logic, and prompt templates. These components are provided under an open-source license. In addition, the system uses established open-source frameworks and libraries for web development, vector storage, authentication, and AI processing. Large language models are treated as replaceable components, with a preference for open-source solutions and an architecture that avoids vendor lock-in.

3.3 Hosting of the system

For development a server with GPU is needed when developing and testing with local LLM's.

3.4 Data and examples

The project uses institutional research data provided by the client in a controlled manner. This includes research metadata from Pure (such as publications, abstracts, persons, organizations, and identifiers) and access to Ricgraph to retrieve relationships between research entities.

Relevant, open enrichment data from OpenAlex may be used in accordance with its license. The client is the data owner or authorized steward. Example datasets and predefined scenarios will be provided for development and demonstration purposes.

3.5 Test environment

The system will be developed and tested in a controlled, non-production environment approved by the client. Testing will cover core functionality, integration with external systems, authentication and authorization behavior, and user-facing clarity and transparency.

The client will provide representative test data and example scenarios, in collaboration with stakeholders like policy officers and research managers. Testing will be performed with involvement of the client and selected domain experts, and feedback will be incorporated during sprint reviews and project milestones.

4 Contact information

First point of contact:

David Grote Beverborg (d.h.j.grotebeverborg@uu.nl), IT Engineer at ITS

Project participants:

For use cases, user stories and user research:

Theije Visser (t.m.visser@uu.nl), Information analyst at UBU and LEG Faculty

For all things about Ricgraph

Rik Janssen (r.d.t.janssen@uu.nl), Business analyst at ITS