

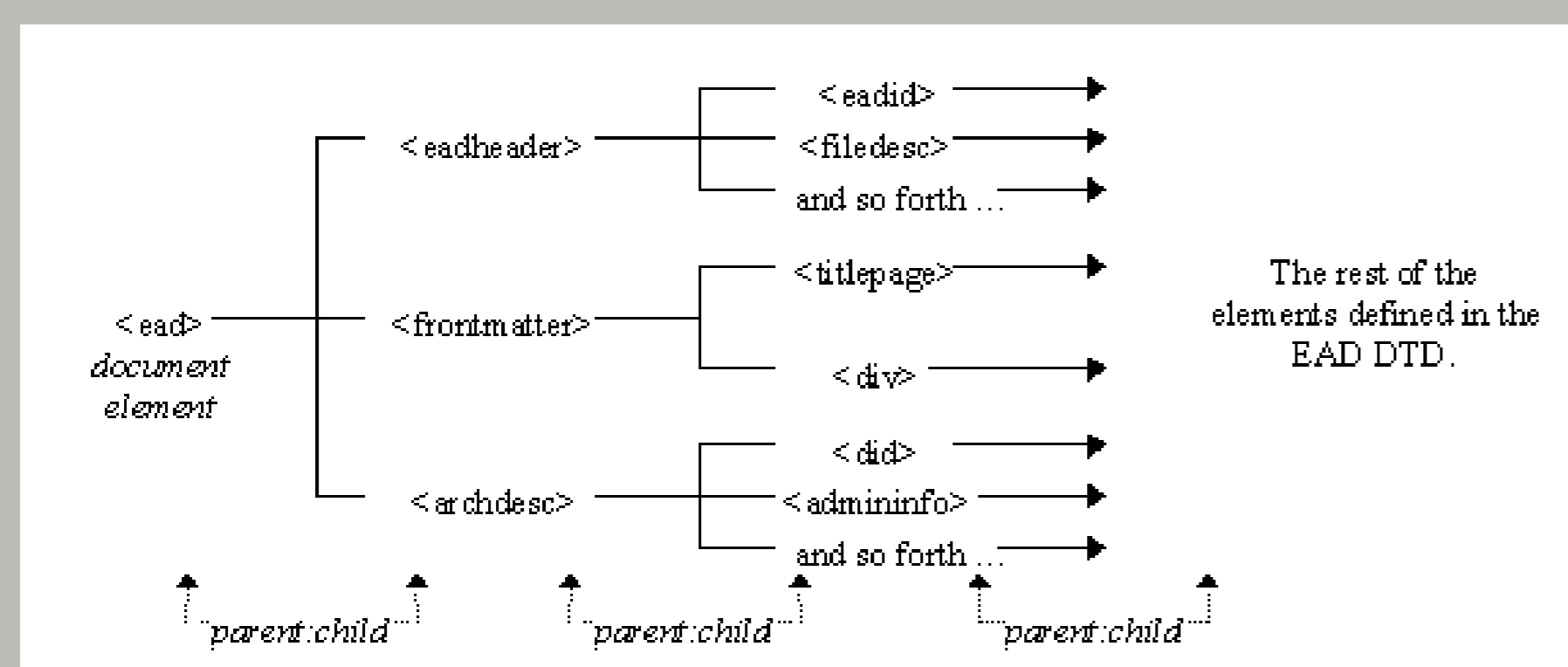
# Access to Archival Finding Aids: Context Matters

Khairun Nisa Fachry, Jaap Kamps, Junte Zhang  
University of Amsterdam



## 1. Introduction

- An **archival finding aid** is a **multilevel description** proceeding in a top-down fashion, first describing the whole archive, then its major sub-components, and so on.
- Modern archival finding aids are encoded in XML using the **Encoded Archival Description** standard.
- Archival finding aids can be extremely **lengthy and complex!**



## 2. Project Research Objectives

- Study **effective retrieval techniques** tailored to focused retrieval on archival finding aids, taking into account the user's profile and context, the structural context, and the contextual content, of the unit to return.
- Enhance user access to archival material** through digital finding aids from multiple sources, and in compliance with archival standards.

## 3. Datasets

- Archival finding aids from different institutions, with difference in length, complexity and language.

Institute	Files	Length (char)			Lang.
		Min	Max	Mean	
National Archives (NL)	2,174	6,157	9,757,729	180,776	Dutch
IISG (NL)	2,866	1,847	2,121,778	26,244	Multilingual
Archives Hub (UK)	3,120	1,609	889,118	8,872	English

## 4. Retrieving Encoded Archival Description More Effectively – README Approach

*How to retrieve and present archival material in context?*

### Preserve Provenance

- Our system ranks and retrieves the XML elements separately, but we **group results** from the same archival description.
- The hitlist presents **query-sensitive results** grouped per finding aid (thus same archival creator), and the text nodes with keyword highlighting.

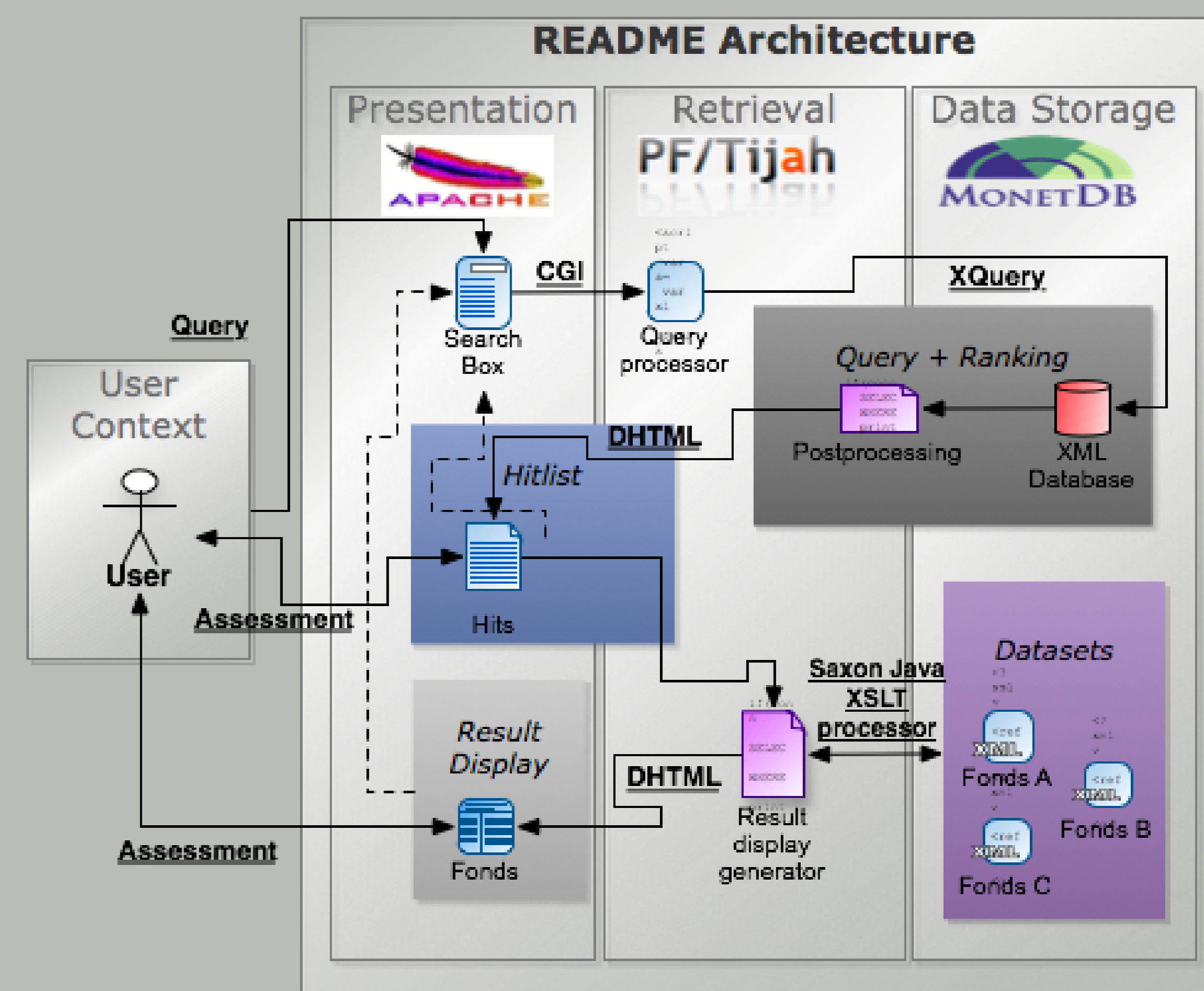
### Preserve Document Structure

- To support the conceptual understanding of the structural dependencies, a global view of the finding aids is presented that **preserves the document structure** as a tree.
- We traverse the XML tree with the **XPath axes in XQuery**, and while we traverse this tree, we select the titles, headings, and other structural dependencies belonging to relevant text nodes.
- We preserve the results in **document order**.

### Deep Linking and Direct Access

- Using XQuery, we group the top ranking elements by article. As we merge the contextual relevancy, **any item that is merged (such as the XML structure) is clickable**, and directly takes the user to the relevant part of the full finding aid.
- There are several choices that a user can make using this result list, a clear advantage is that the **overview of the finding aids is not being lost**, maximally exploiting the found results and their full context.

## 5. System Architecture



## 6. README in Action

Figure: Hitlist with context

Figure: Result display of a fonds

## 7. Conclusions

- Designed and implemented a system that retrieves and presents **archival material in context**.
- In-depth and exhaustive studies conducted with **users**.
- System supports users in **navigation** and **assessing relevancy**.