

XIRAF

Ultimate Forensic Querying

DFRWS - August 15, 2006

Wouter Alink, Raoul Bhoedjang
Netherlands Forensic Institute

Peter Boncz, Arjen de Vries
Centrum voor Wiskunde en Informatica



Introduction

XIRAF

*“An XML Information Retrieval
Approach to Digital Forensics”*

Collect, manage, and query information
extracted from digital evidence

Outline

- Problem statement
- XIRAF approach
- XIRAF architecture
- Forensic application areas
- Initial experiments
- Conclusion

Typical investigation steps

1. Media capture
2. Feature extraction
3. Analysis
4. Reporting

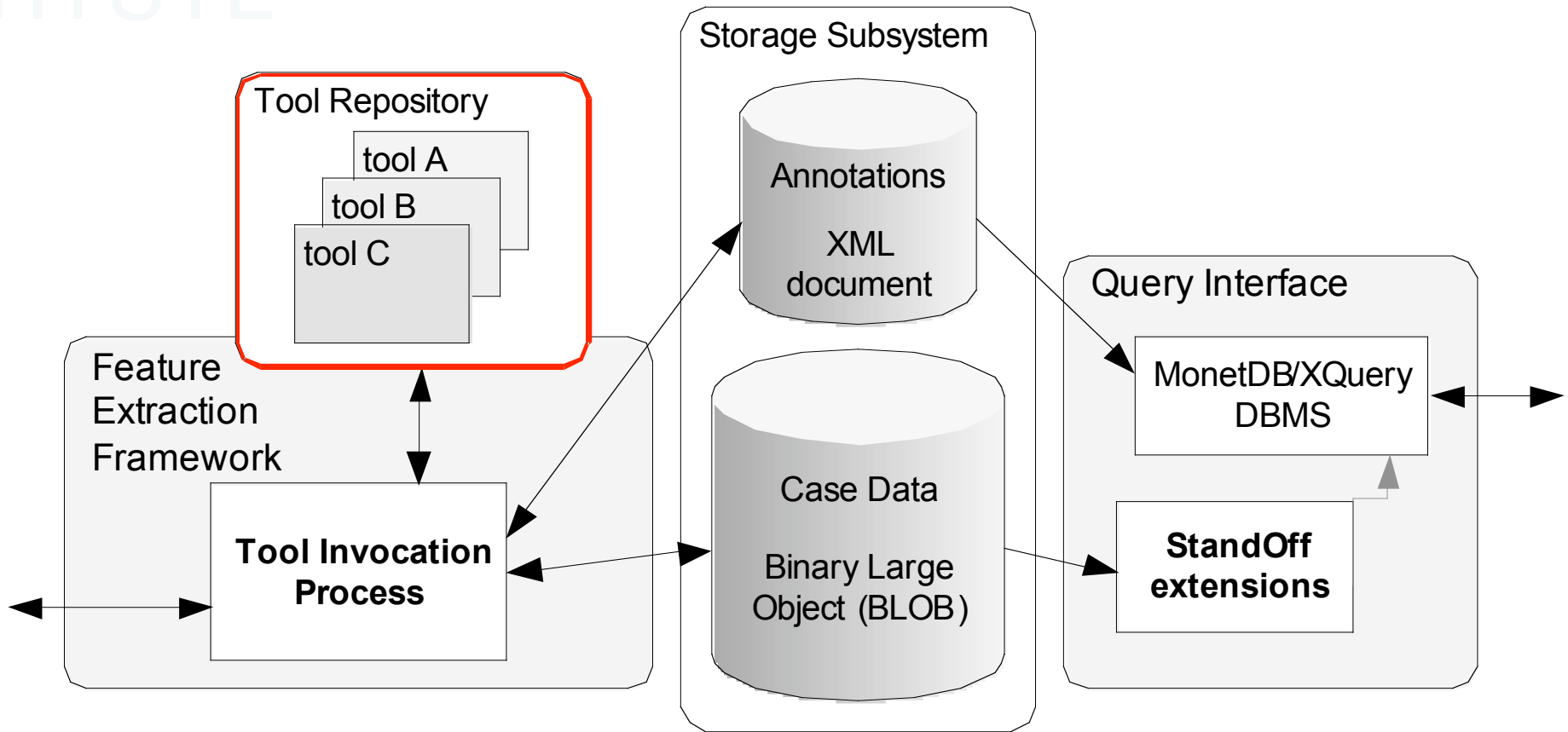
Problem identification

- Large amounts of data
 - Investigation restricted by deadlines
 - Too much information to track manually
- Diversity of data and tools
 - Many different formats
 - Many stand-alone forensic tools

Approach

- Clean separation between feature extraction and analysis
- A single, XML-based output format for tools
- XML database technology to analyze extracted features
- Use of existing forensic analysis tools

XIRAF architecture

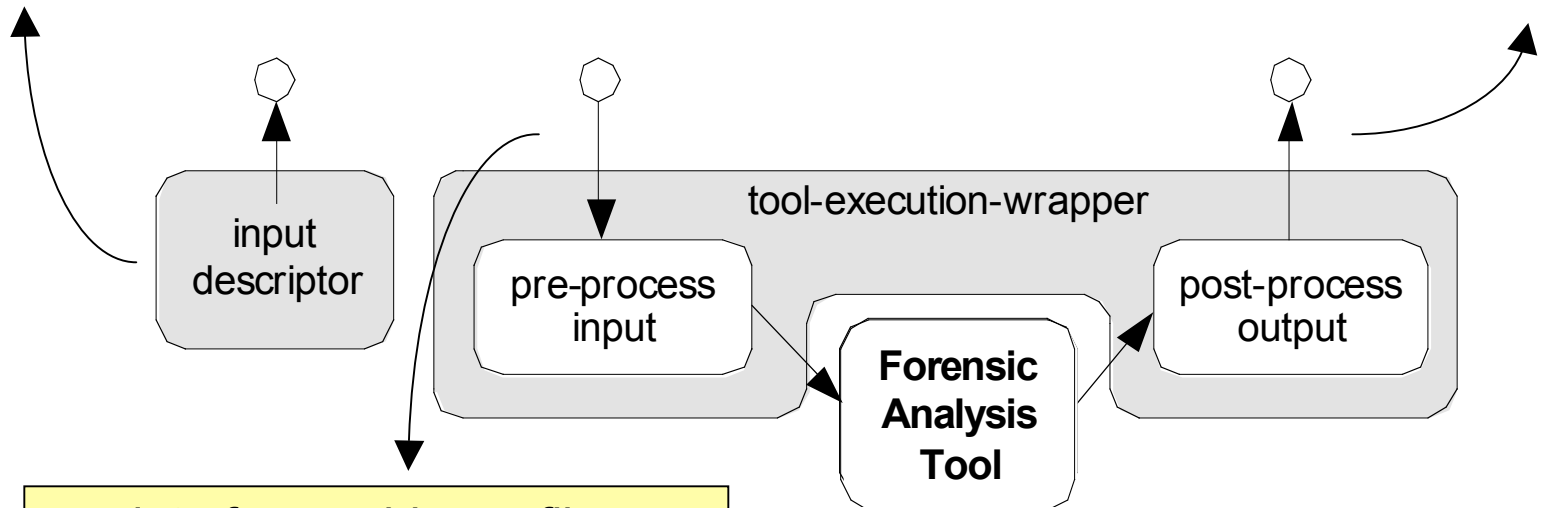


Tool wrapper

```
<photo>  
  <camera>Canon</camera>  
  <taken-on>  
    <date>15-12-2005</date>  
  </taken-on>  
</photo>
```

```
//file[mime="image/jpeg"]
```

- metadata (features/traces)
- new view of the original data



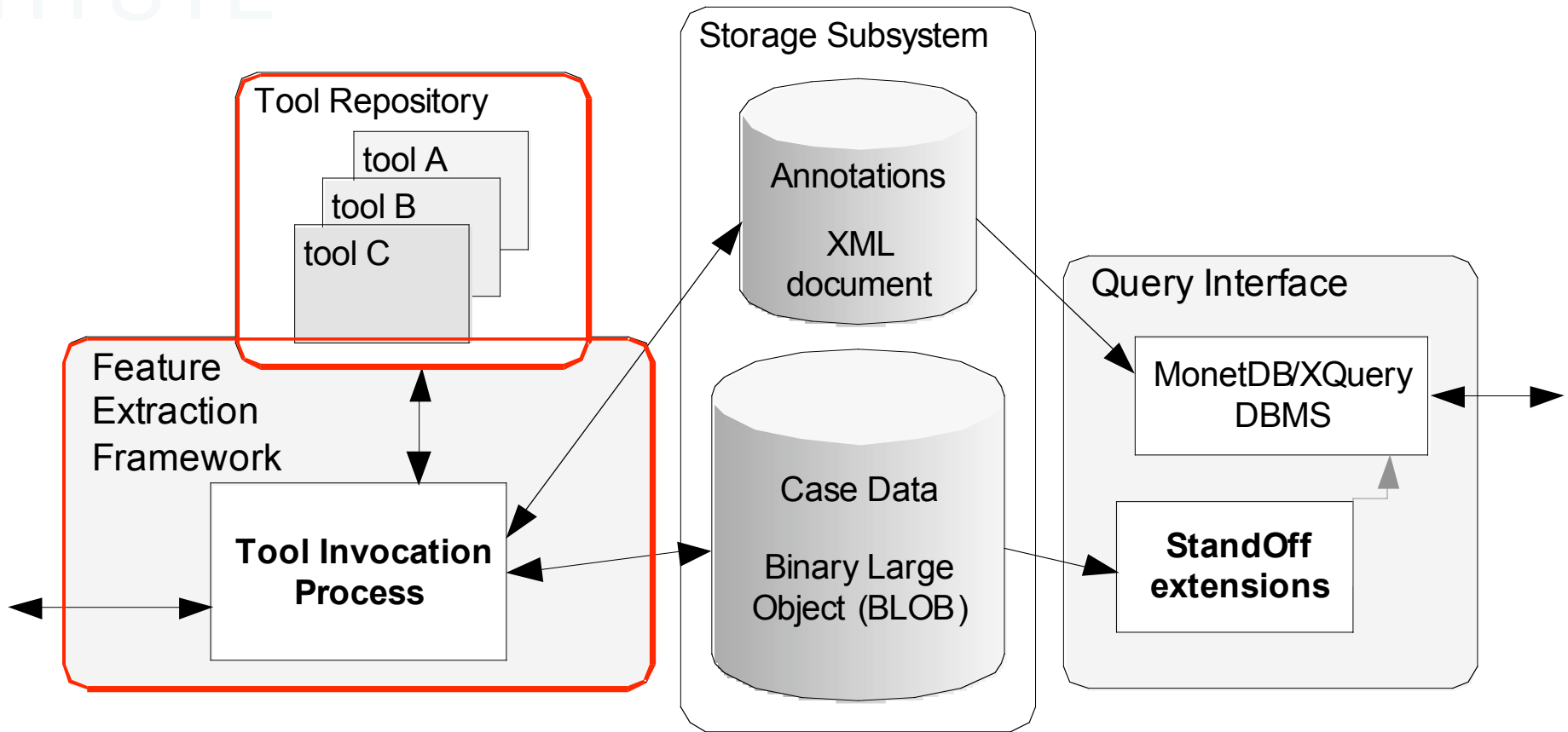
- data from evidence files
`Photo03.jpg`
- Optional:
additional metadata



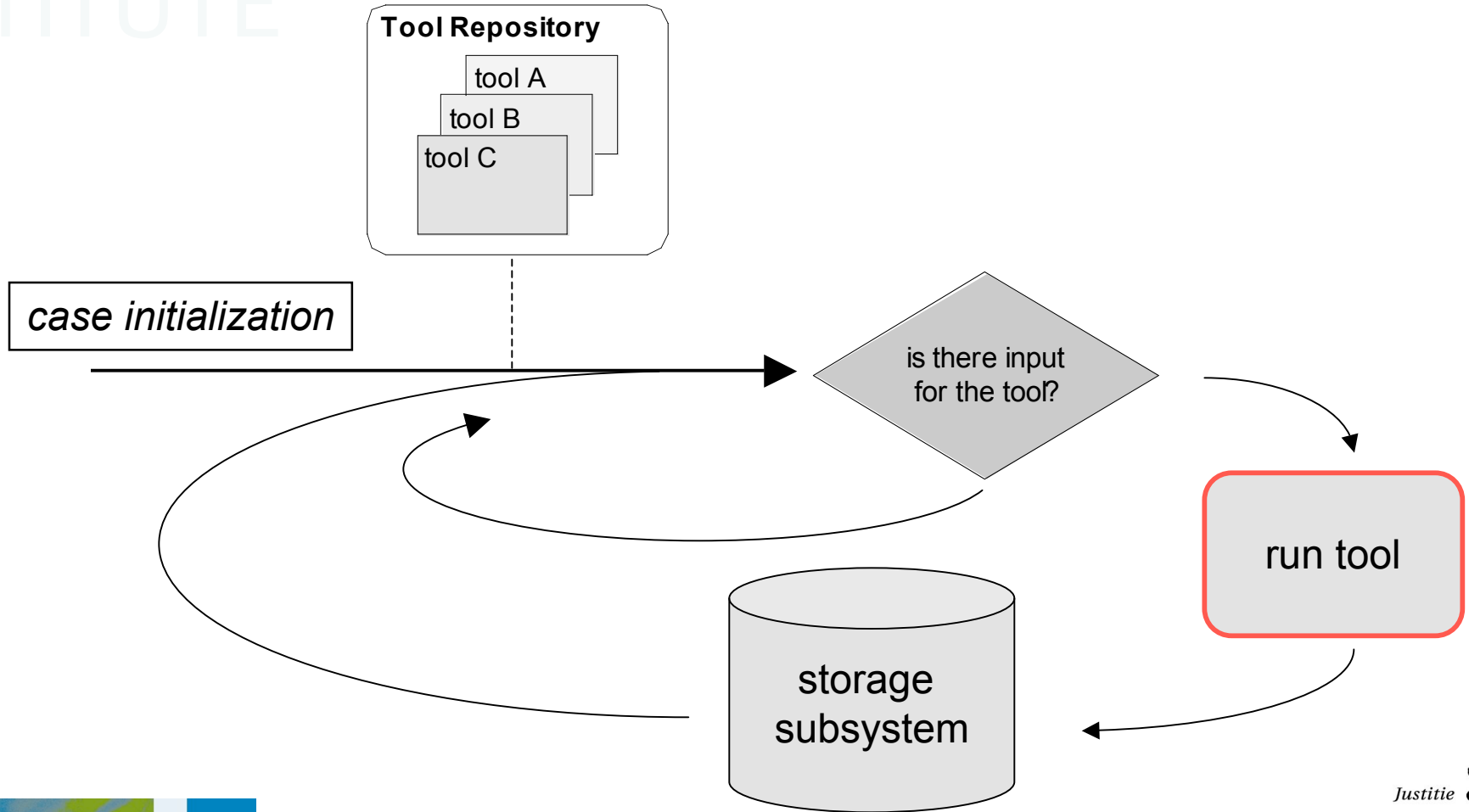
Tool repository

- Feature extraction tools
- Gain knowledge about an 'object':
 - volume
 - file-system
 - image
 - email
- Some of the wrapped tools:
 - file-system dissector
 - windows registry analyzer
 - EXIF-data parser
 - carving tool
 - IE-history parser
 - Hashing tool

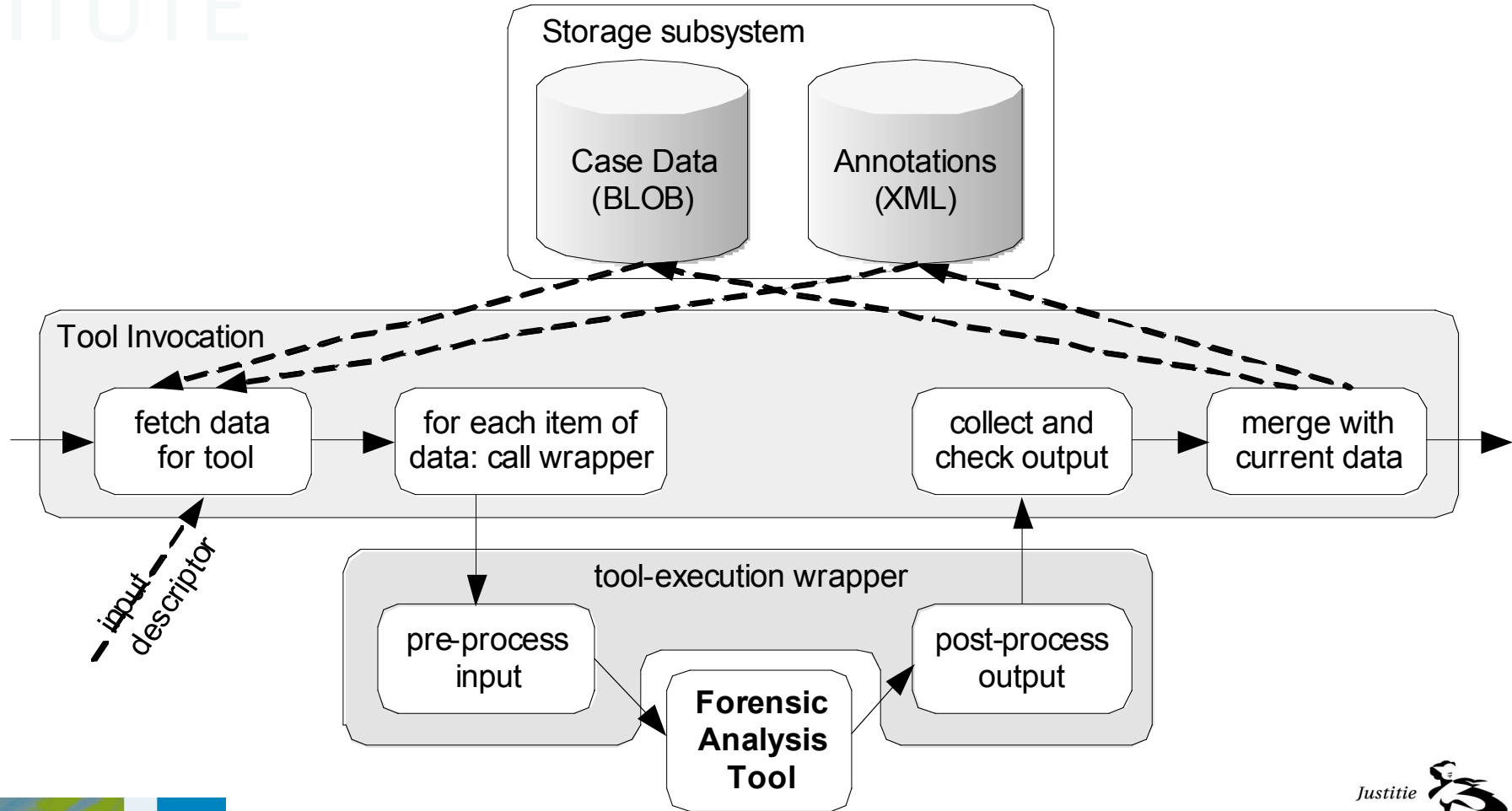
XIRAF architecture



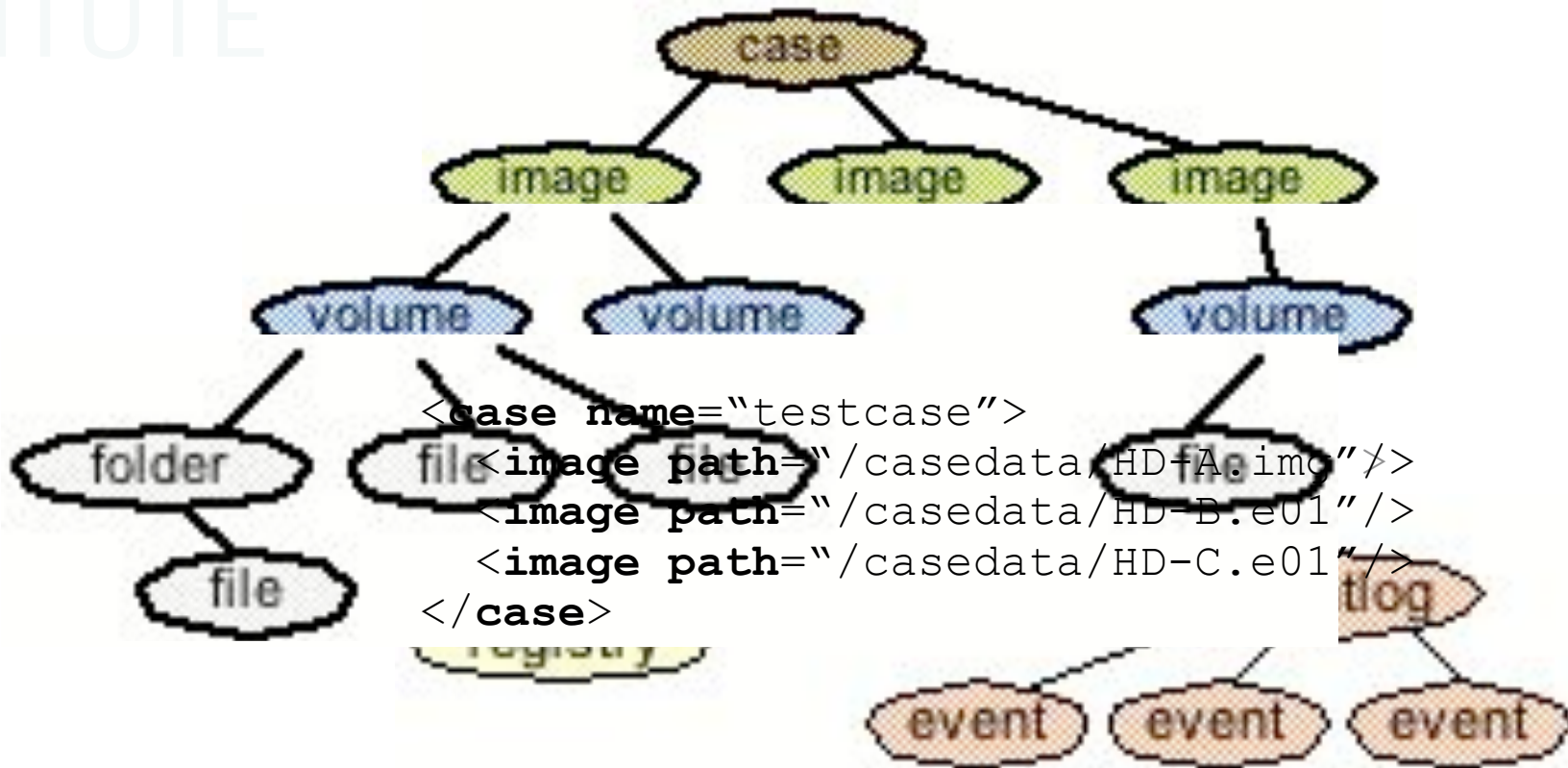
Feature extraction framework



Feature extraction framework



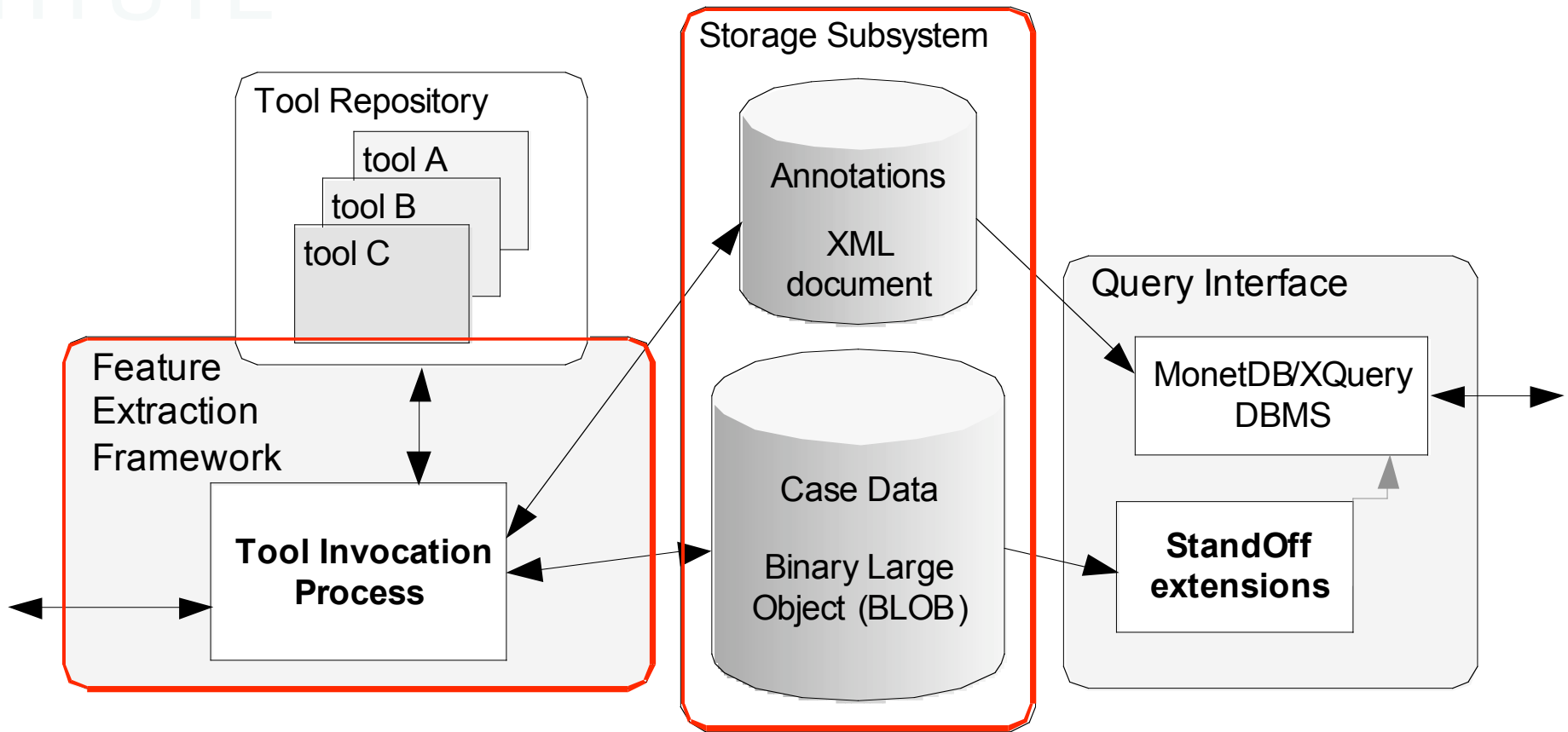
Feature extraction



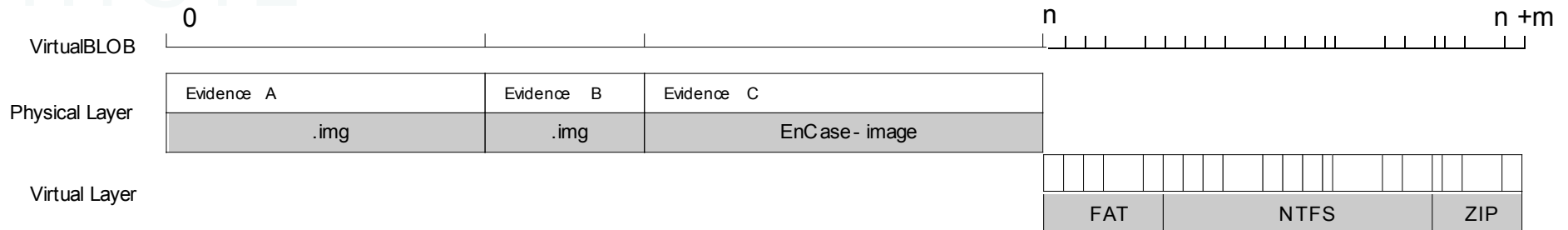
```
<image path="/casedata/HD-C.e01">  
  <volume label="MP3"/>  
</image>  
</case>
```



XIRAF architecture



Virtual BLOB and XML



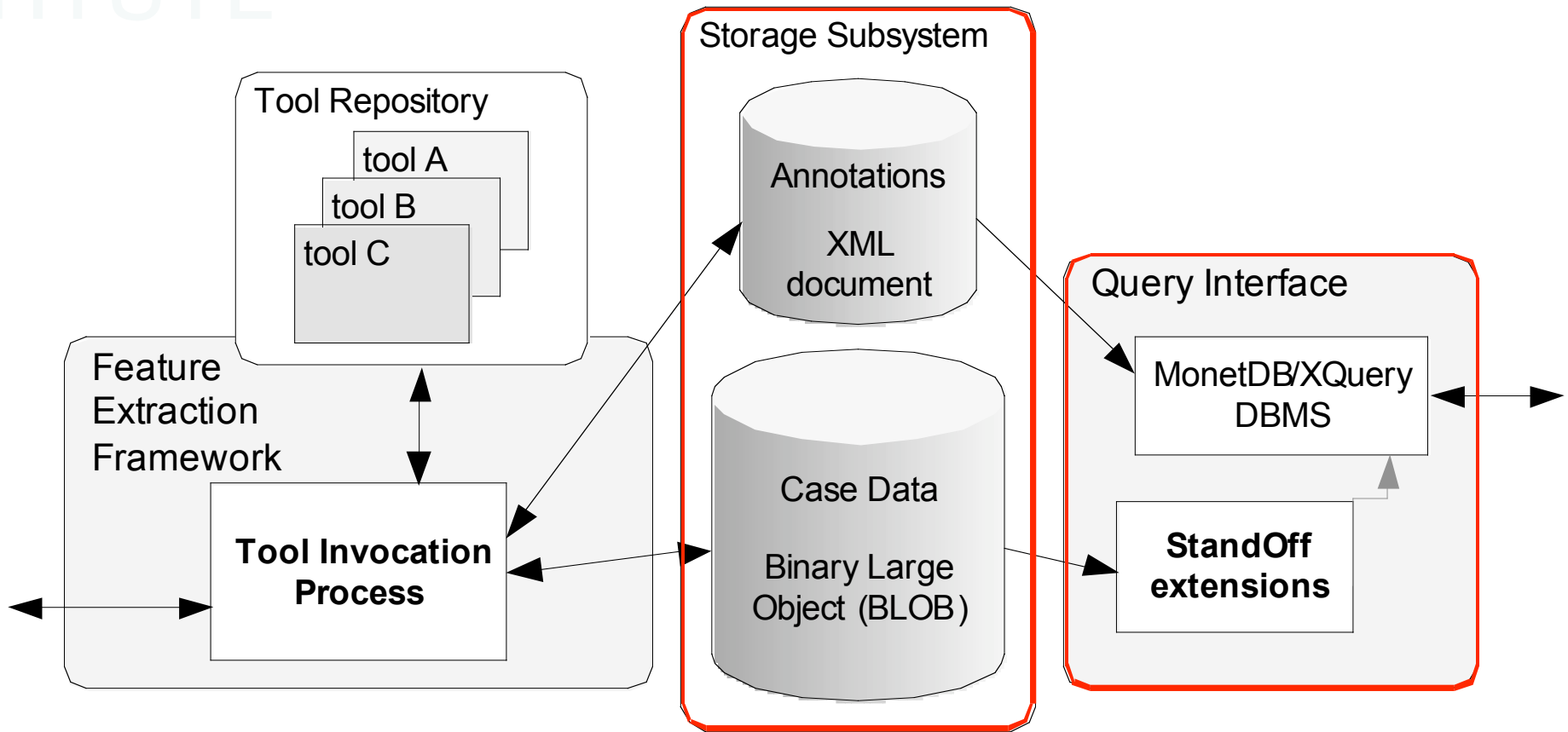
```
<case name="testcase">
  <image path="Photo\data\p1\HD-A.e01" start="70000" end="74999" />
  <image path="Photo\data\p1\HD-B.img" start="20000" end="29999" />
  <image path="Photo\data\p1\HD-C.e01" start="30000" end="59999" />
</case>
<modified><date>2006-08-15T09:10:00</date></modified>
</file>
...
<volume type="FAT" start="1000" end="19999" />
<volume type="NTFS" start="35000" end="39999" />
```



Storage subsystem

- Virtual BLOB mapping
 - evidence files
 - alternative representations
- Single XML document
 - extracted features
 - references to layout

XIRAF architecture



XQuery language

- Database language:
 - large XML documents
 - sorting/grouping/selecting/(updating)
- Example: timeline
 - different tools produce date-elements

```
for $i in doc("case.xml")//date
order by $i
where $i > $lowerbound
    and $i < $upperbound
return $i
```

Forensic application areas

- search for keywords, MD5s, URLs

```
for $i in doc("case.xml")//file
for $j in doc("CP-hashes.xml")//md5
where $i/md5 = $j
return <file> { $i/@name } </file>
```

```
let $word_list :=
    doc("terrorism-words.xml")//word
for $i in doc("case.xml")//*
where some $j in $word_list
    satisfies blob-contains($i,$j)
return element { name($i) } { $i/@* }
```



Benefits

- Exploit exhaustive runs of tools
- Use knowledge from previous investigations
- Integrated data schema
- Added functionality:
 - XQuery extensions to relate XML to Virtual BLOB content



XIRAF Query Page

Project: **javaPatrick**
Number of files: 89017
Number of folders: 4471

Please select an object of interest:

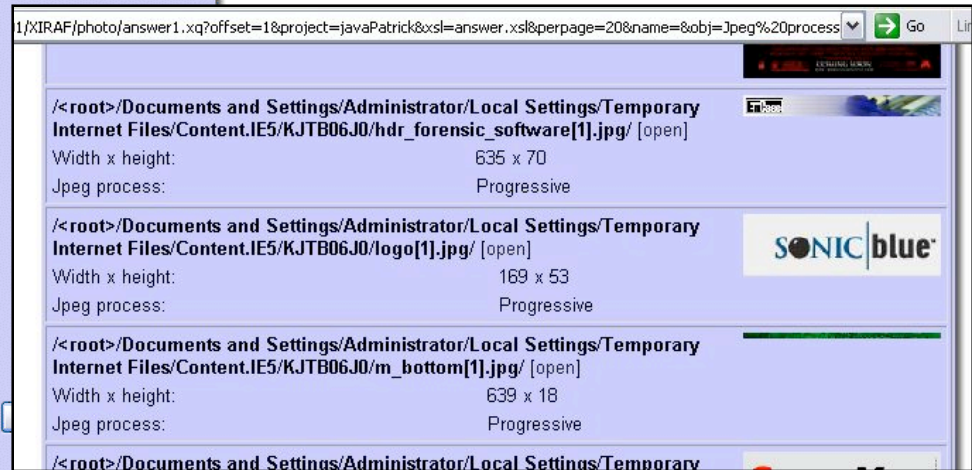
Limiting Results

The item should:

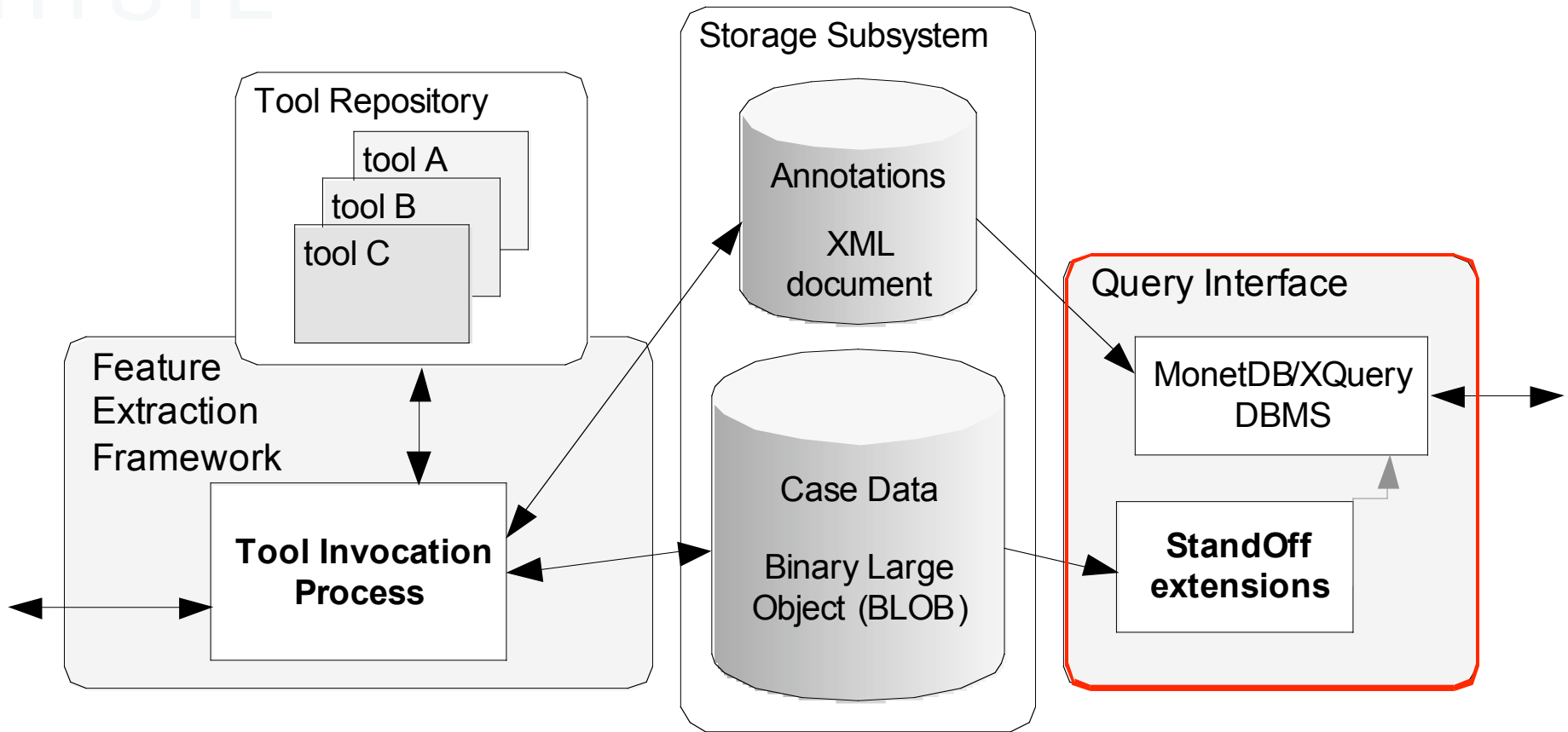
- contain the keyword
- contain date between
and
- contain folder (4471)
with key :
- contain a
with keyword (optional):

[back to project page](#) | [explain page](#)

```
let $d := doc("case.xml")
for $i in $d//%object_of_interest%
where $i/descendant::%contains%[so-contains(%keyword_1%)]
and $i/ancestor::%contained%[so-contains(%keyword_2%)]
and (some $j in $i//%date%//date
satisfies $j >= %lowerbound% and $j < %upperbound%)
return element { name($i) } { $i/@* }
```



XIRAF architecture



Initial Experiments

- Evidence: 2 hard disks
 - (2 x 120GB)
- ~200MB XML
 - ~2.5M elements
- Recognized ~90000 files
 - file-systems / unallocated space
- ~500000 timestamps
 - file-system, registry, EXIF, .LNK, log-entry, cookie, etc

Conclusion

- Separation of feature extraction and analysis seems a viable approach
- Integrated querying of multiple tools becomes possible



Status & Future Work

- Prototype implementation (Java/Python)
- Make system production-ready
- More tools, query patterns
- Connect XIRAF to existing knowledge-bases

More information

- xiraf-info@holmes.nl
- <http://www.forensischinstituut.nl/>
- <http://monetdb.cwi.nl/>