Institute of Information Science
Academia Sinica
中央研究院 資訊科學研究所

Efficient GML-native Processors for Web-based GIS: Techniques and Tools

(Presented at ACM-GIS06)

Open GIS & Internet GIS 2006 Conference

December 8, 2006

IIS Sinica

Chia-Hsin Huang @
Open Geo-spatial Information Team.IIS.Sinica
jashing@iis.sinica.edu.tw

Agenda

- Motivations
- Contributions
- GML-based Web GIS & GML-native Processors
- Performance Analysis
- Conclusions & Future Work

Institute of Information Science
Academia Sinica
中央研究院 資訊科學研究所

Motivations

A GML Fragment

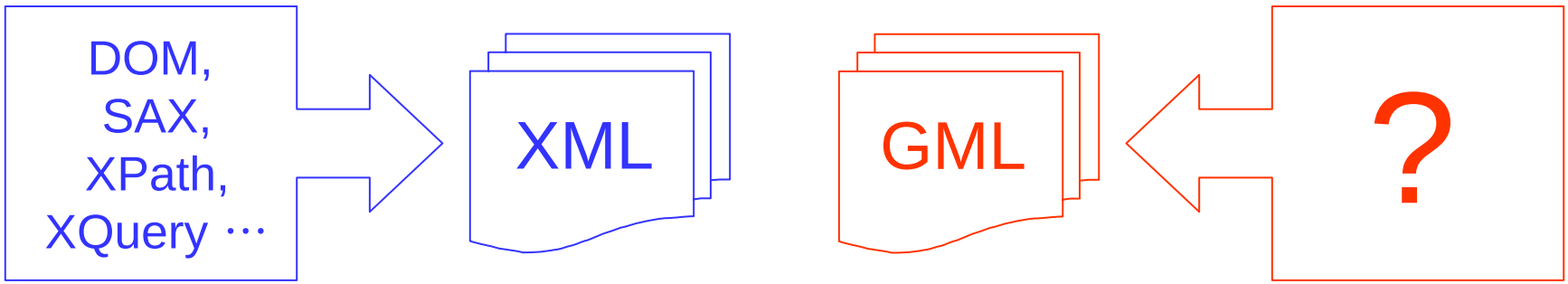
```
1  <Streams>
2    <streamsMembers>
3      <Rivers>
4        <riversMembers>
5          <FootPrint id="21001000000-11">
6            <name> … </name>
7            <length> … </length>
8            <multiCenterLineOf>
9              <MultiCurve>
10               <curveMembers>
11                 <LineString>
12                   <coordinates>302745.435072,2442803.380075 ...
13                 </LineString>
14               </curveMembers></MultiCurve>
15           </multiCenterLineOf>
16         </FootPrint>
17       </riversMembers>
18     </Rivers>
19   </streamsMembers>
20 </Streams>
```

Geospatial Data (Coordinates)

XML/GML Tags

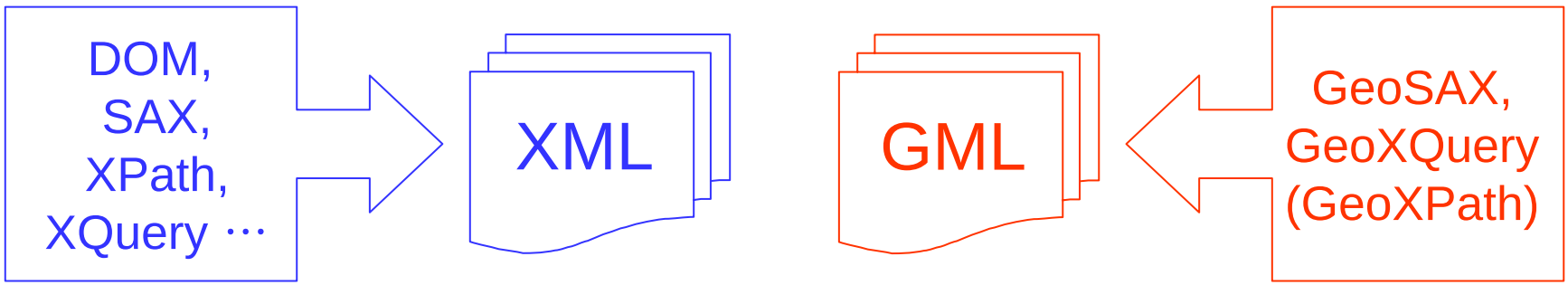
Issues of Manipulating GML Docs.

- GML, providing rich vocabulary and flexible document structure, has been considered as an effective approach to express increasingly complicated geospatial data, as well as non-geospatial data.
- Although GML is a kind of XML, the existing XML processors (DOM, SAX, XPath, and XQuery) are not suitable for processing GML.



Solutions

- GIS databases,
  - Open source software, PostgreSQL/PostGIS.
  - Many people choose this way.
- Extending the existing XML processors
  - We now are talking about this way.

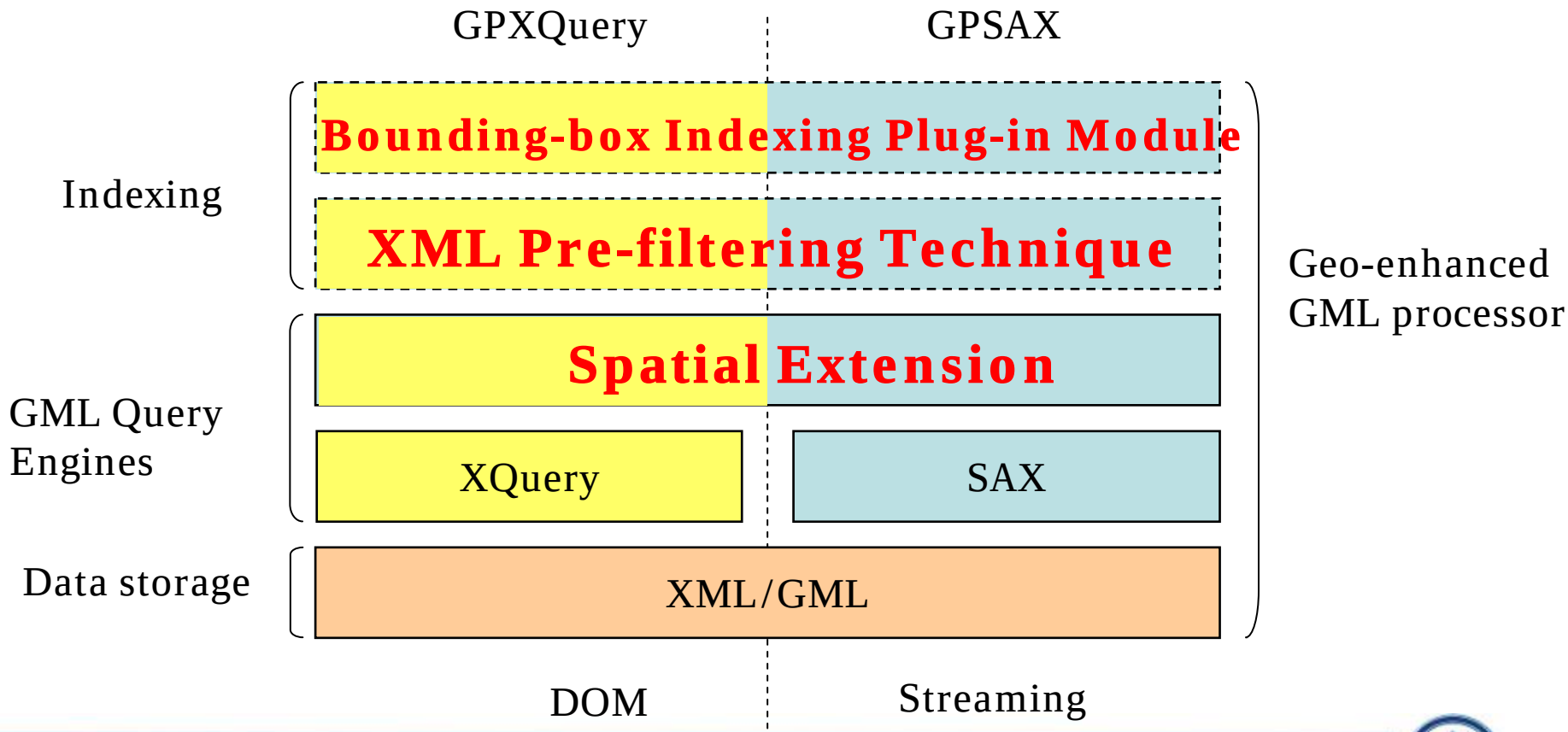


Institute of Information Science
Academia Sinica
中央研究院 資訊科學研究所

Contributions

Contributions

- Proposing two efficient GML-native processors.
- Enabling the GML processors to query large GML docs.
- Building a GML-based Web GIS using the GML processors.



GML Instance

```
<Streams>
  <streamsMembers>
    <Rivers>
      <riversMembers>
        <FootPrint id="21001000000-11">
          <name> … </name>
          <length> … </length>
          <multiCenterLineOf>
            <MultiCurve>
              <curveMembers>
                <LineString>
                  <coordinates>302745.435072,2442803.380075 ... 302882.868766,2442187.96039</coordinates>
                </LineString>
              </curveMembers></MultiCurve>
          </multiCenterLineOf>
        </FootPrint>
      </riversMembers>
    </Rivers>
  </streamsMembers>
 </Streams>
<Roadways>
  <roadwaysMembers>
    <GeneralRoad>
      <generalRoadMembers>
        <FootPrint id="4230904000-31">
          <name> … </name>
          <length> … </length>
          <multiCenterLineOf>
            <MultiCurve><curveMembers>
                <LineString>
                  <coordinates>301219.608538,2442423.503538 ...  308235.358961,2434884.578507</coordinates>
                </LineString>
              </curveMembers>
            </MultiCurve>
          </multiCenterLineOf>
        </FootPrint>
      </generalRoadMembers>
    </GeneralRoad>
  </roadwaysMembers>
</Roadways>
```

Querying GML by Feature ID

using XQuery Expression with Geospatial Extension Libraries

Geospatial extension for XQuery

```
1.  declare namespace my="http://www.sinica.edu.tw/";
2.  declare namespace gml="java:GML.XQGeoExtensions";
3.  declare namespace svg="java:GML.XQSVGExtensions";

4.  declare function my:get_geo1() as element() {
5.    for $var1 in doc("lanyu.xml")//Rivers//FootPrint[@id = "21001000000-11"]
6.  return <result1>{$var1}</result1> };

7.  declare function my:get_geo2() as element() {
8.    for $var1 in doc("lanyu.xml")//Roadways//FootPrint[@id ="4230904000-31"]
9.  return <result1>{$var1}</result1> };
```

Geospatial Operations

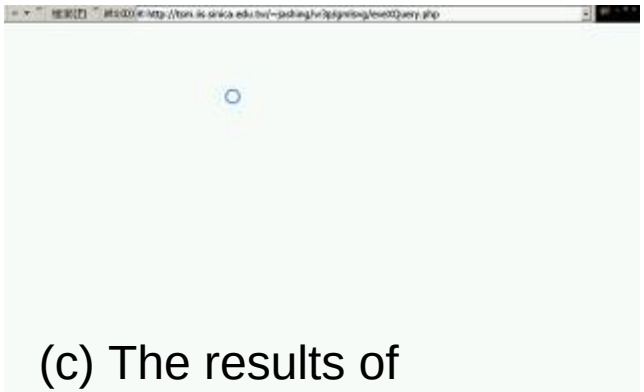
```
10. svg:GML2SVG(gml:Buffer(
    gml:Intersection (my:get_geo1() , my:get_geo2() ), 50))
```

Calculating the buffer of the intersection of a road and a river

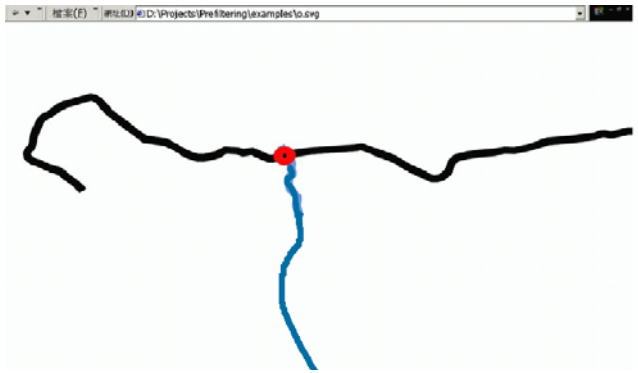
Query Results

(a) A road.

(b) A river.



(c) The results of buffering the intersection of the road and the river



(d) Combine and recolor (a), (b), and (c) in a SVG map.

Querying GML by Layers

using XQuery Expression with Geospatial Extension Libraries

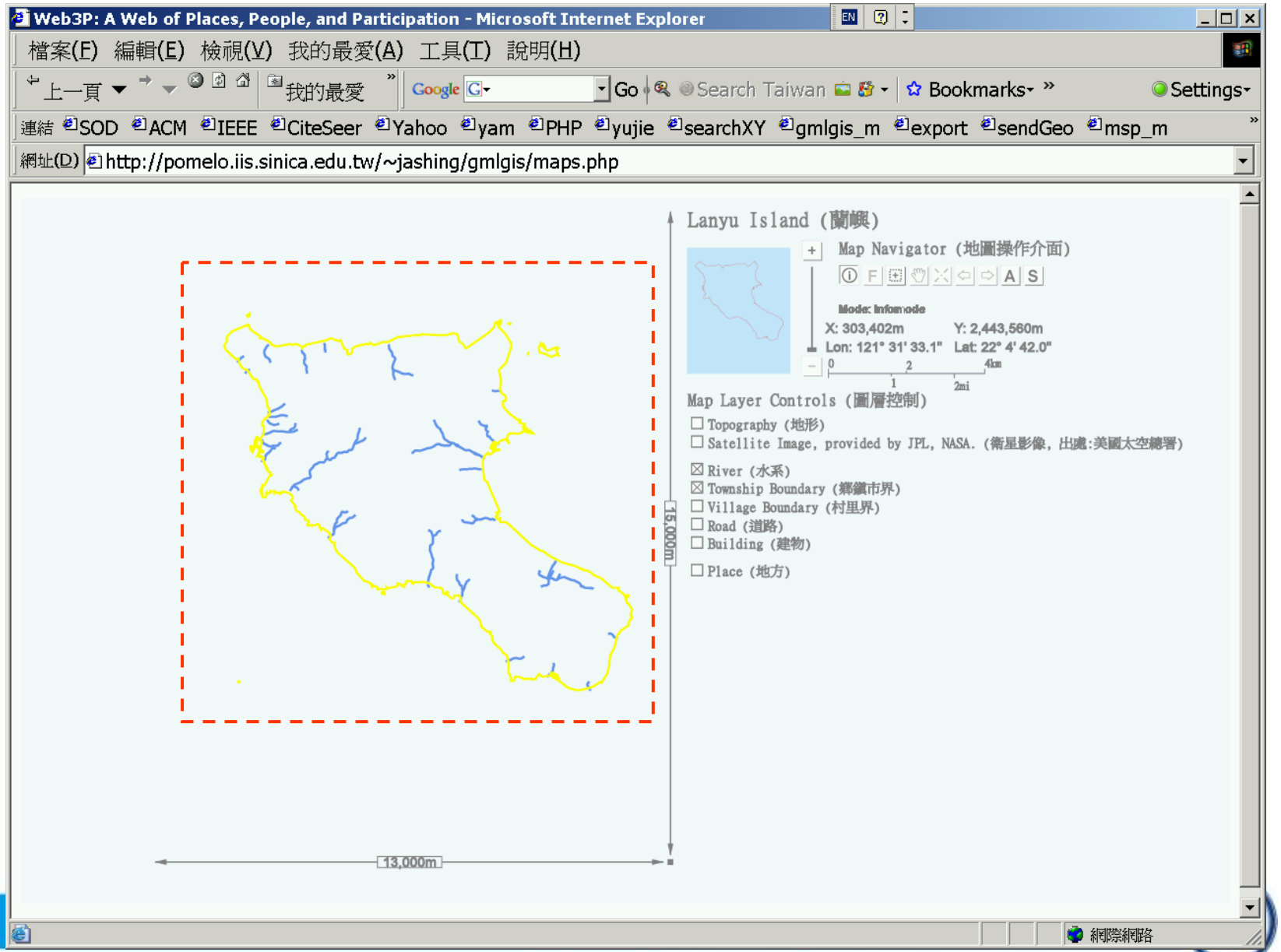
Geospatial extension for XQuery

```
1.  declare namespace my="http://www.sinica.edu.tw/";
2.  declare namespace gml="java:GML.XQGeoExtensions";
3.  declare namespace svg="java:GML.XQSVGExtensions";

4.  declare function my:get_rivers() as element() {
5.    <result>{
6.      for $var1 in doc("lanyu.xml")//Rivers//FootPrint
7.      return $var1
8.  }</result>};

9.  svg:GML2SVG(my:get_rivers())
```

Transformation

Selecting the *Rivers Layer*

Querying Results



Querying GML by a Layer and a BBox

using XQuery Expression with Geospatial Extension Libraries

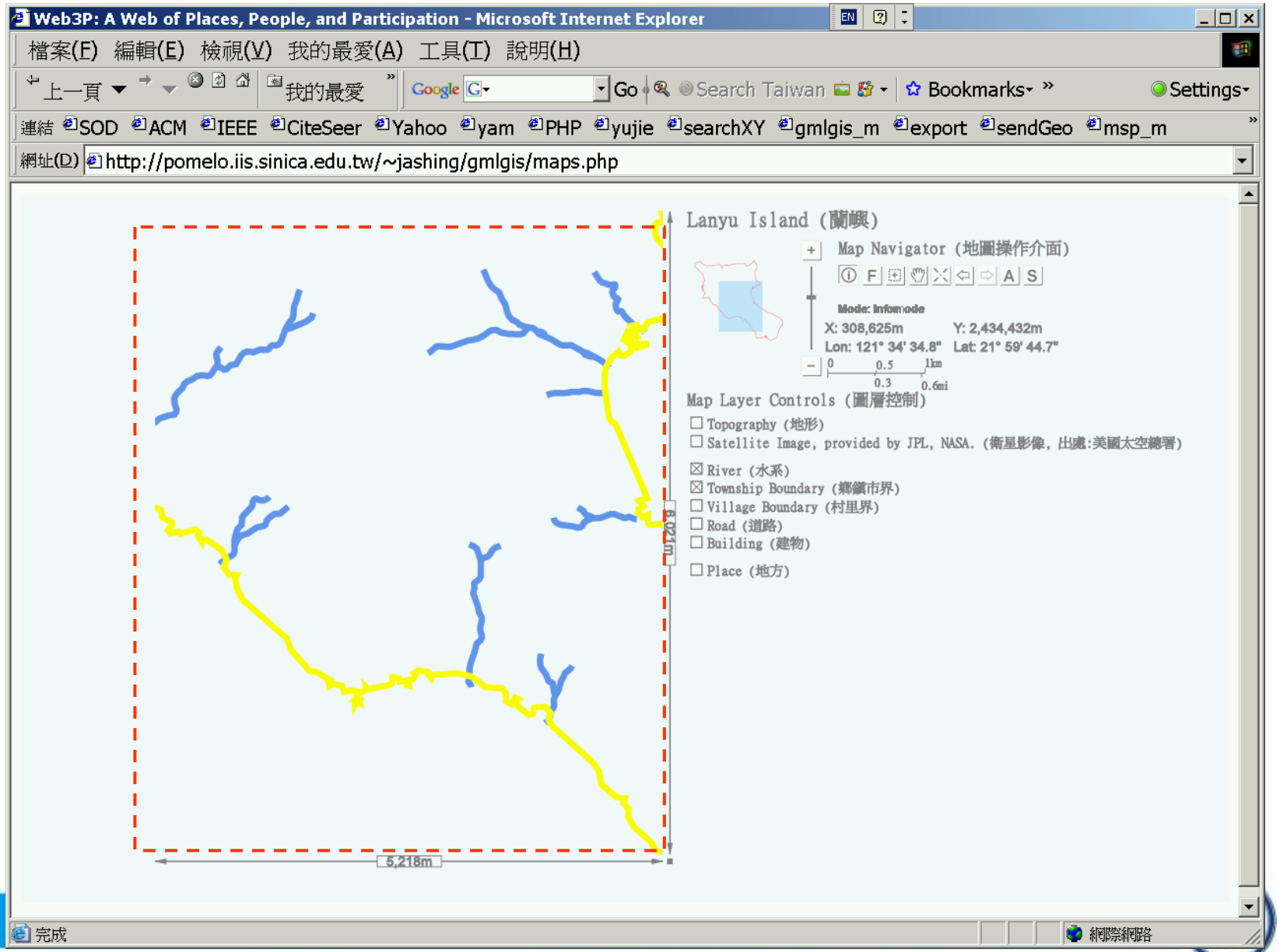
Geospatial extension for XQuery

```
1.  declare namespace my="http://www.sinica.edu.tw/";
2.  declare namespace gml="java:GML.XQGeoExtensions";
3.  declare namespace svg="java:GML.XQSVGExtensions";

4.  declare function my:get_rivers() as element() {
5.    <result>{
6.      for $var1 in doc("lanyu.xml")//Rivers//FootPrint
7.      return $var1
8.  }</result>};
9.  svg:GML2SVG(gml:queryByBox(
    308636,2429765,305529,2440453, my:get_rivers()))
```

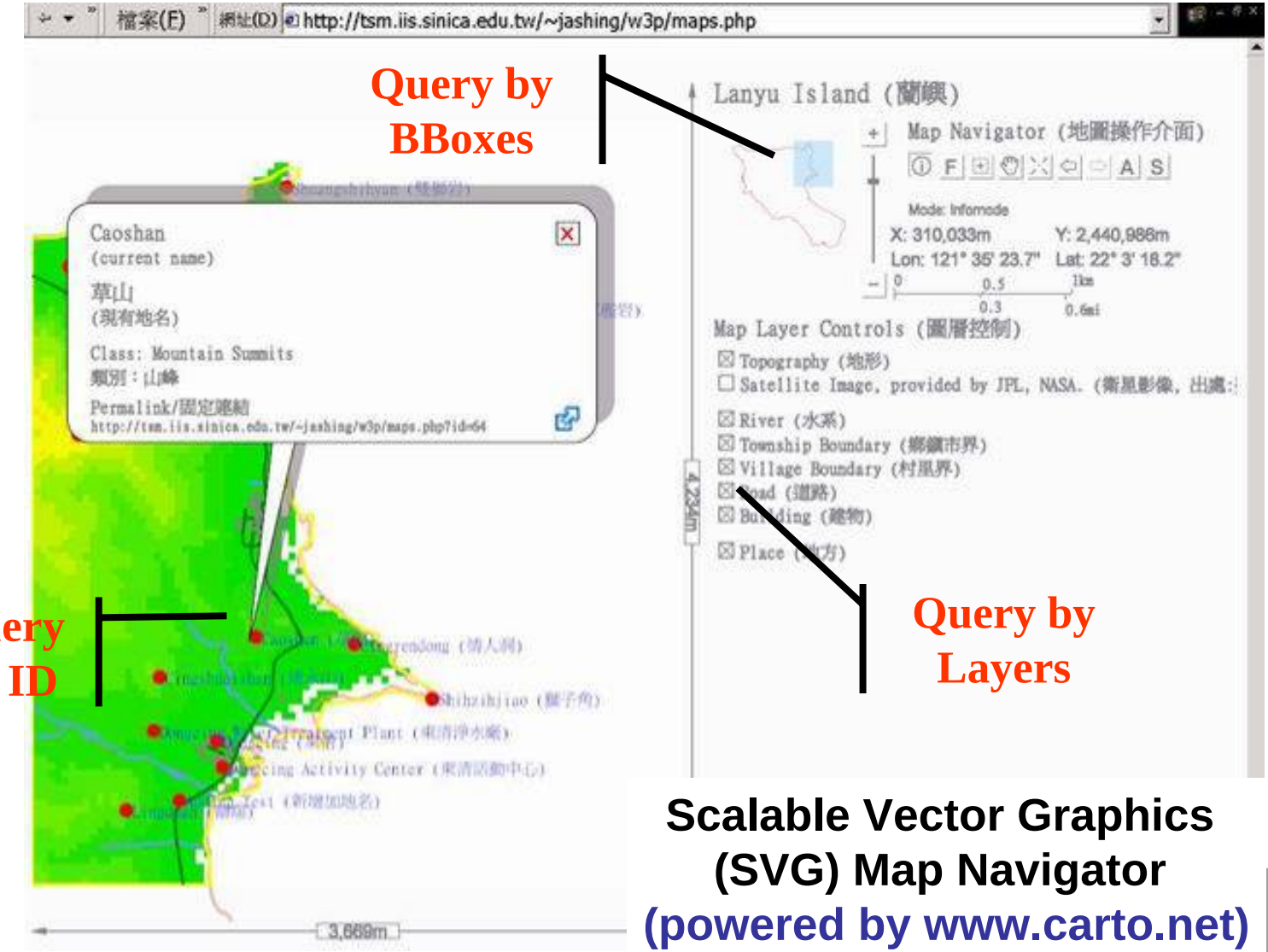
Geospatial Operations

Selecting the *Rivers Layer* in a *BBox*

Querying Results



Snapshots of the GML-based Web GIS

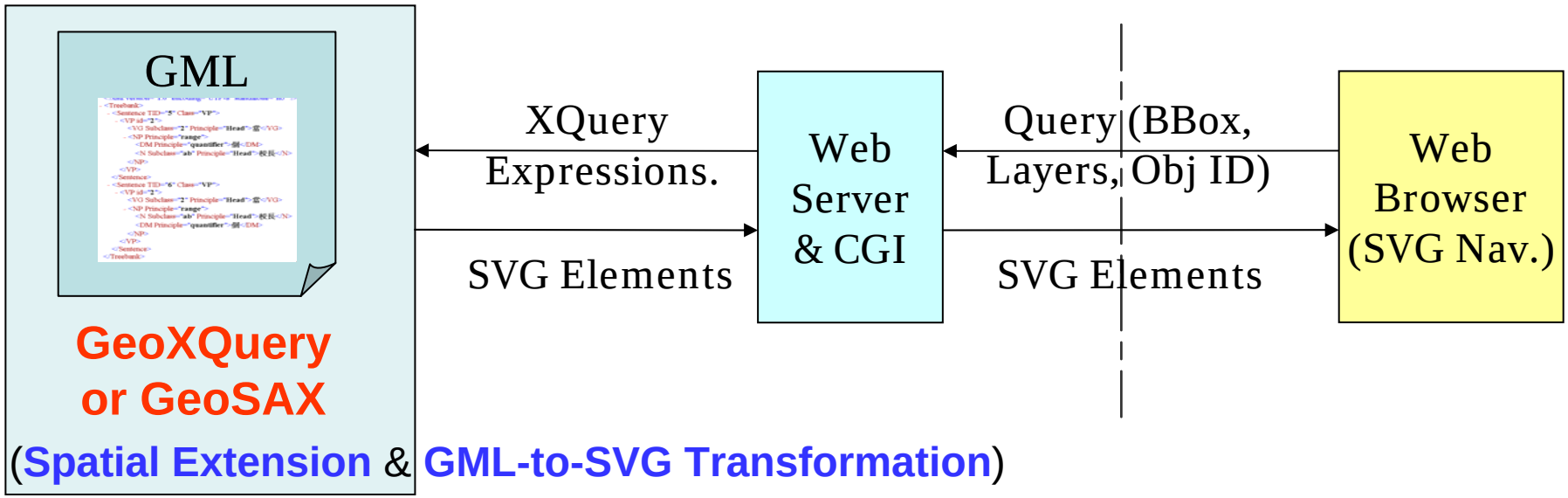


Scalable Vector Graphics (SVG) Map Navigator (powered by www.carto.net)

Institute of Information Science
Academia Sinica
中央研究院 資訊科學研究所

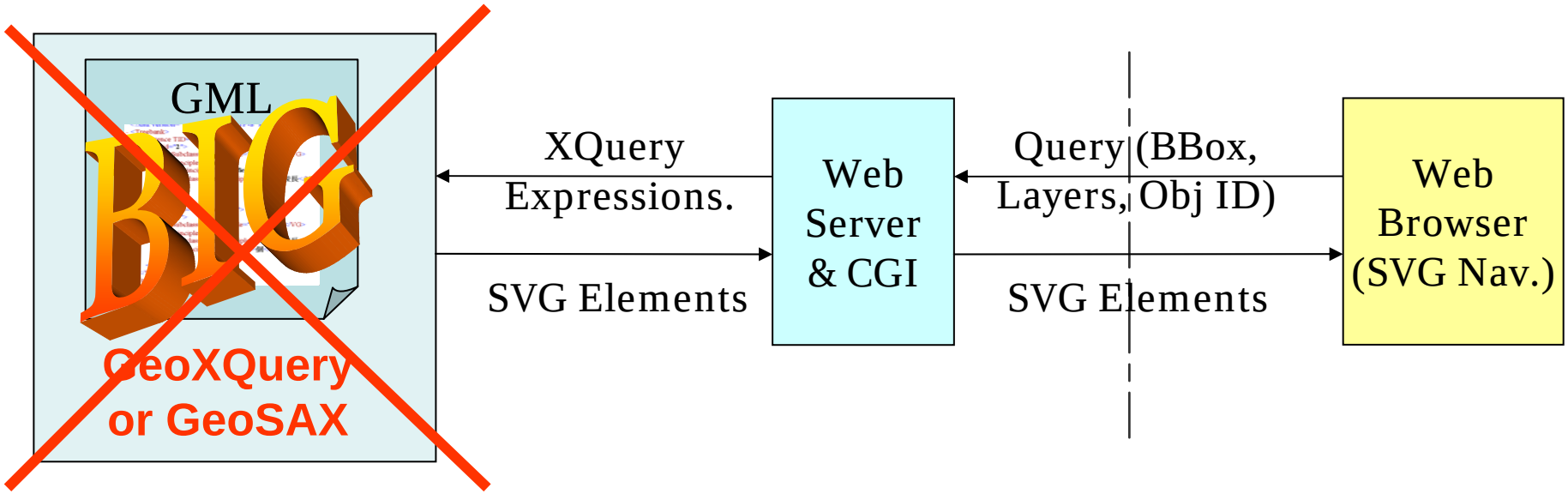
GML-based Web GIS and GML-native Processors

System Architecture of the GML-based Web GIS



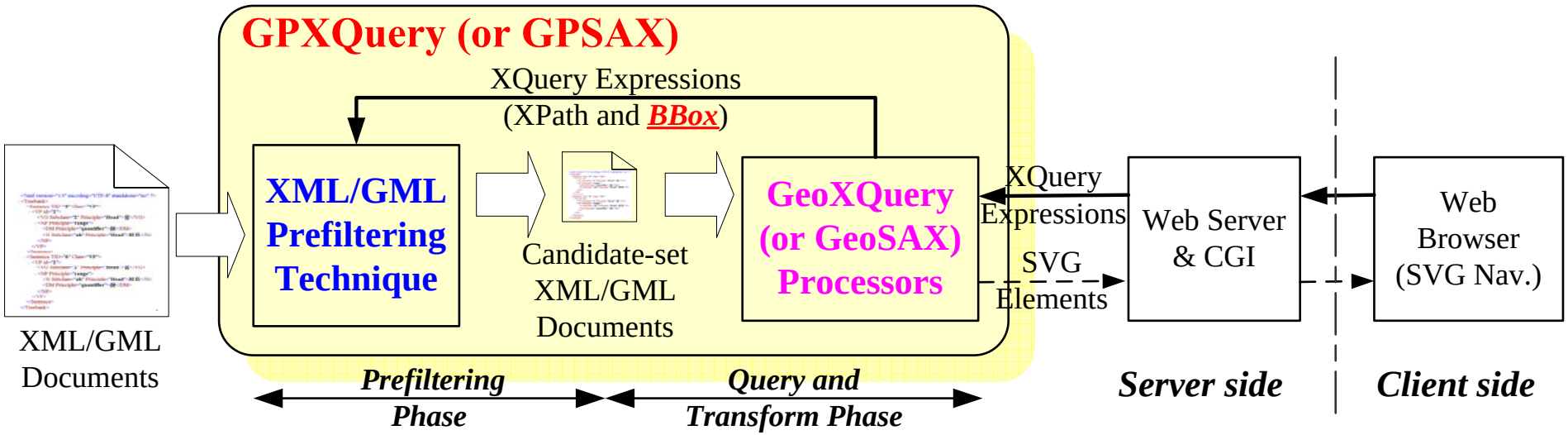
- GeoXQuery – a GML query engine [Boucelma and Colonna, 2004]
  - Extending the Saxon Java XQuery processor by calling spatial functions libraries of JTS (Java Topology Suite).
- GeoSAX -- a GML streaming parser
  - Extending the Sun's SAX parser to support the spatial functions.

Problems in the GML Solution



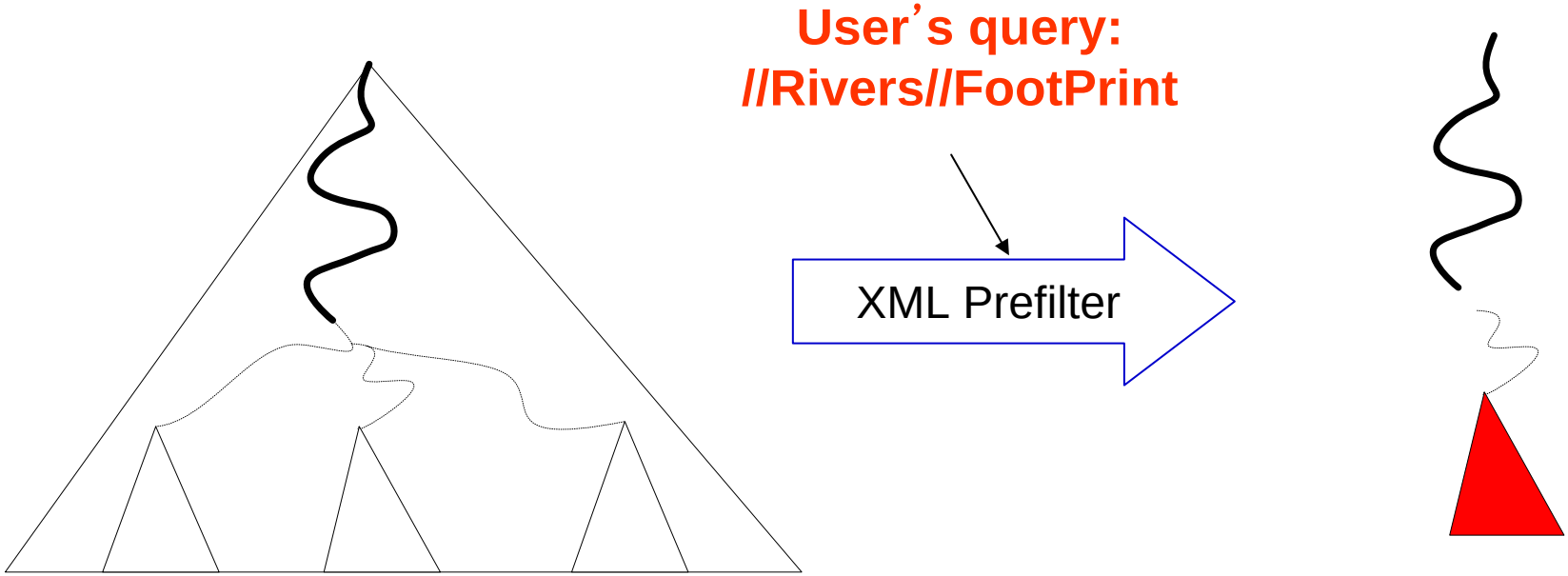
- If the GML documents are **Large**
  - GeoXQuery may not work (DOM data model consumes a huge amount of main memory.)
  - GeoSAX needs a stream-based query algorithm.

Integrating with an XML Pre-filter



- Using an XML Pre-filter Technique [Huang et al. 2006.] to cut off uninteresting XML/GML fragments by approximately executing user query.
  - However, the prefilter does not support the functionality of prefiltering Geospatial data.
  - I.e., cannot handle the BBox query constraint.

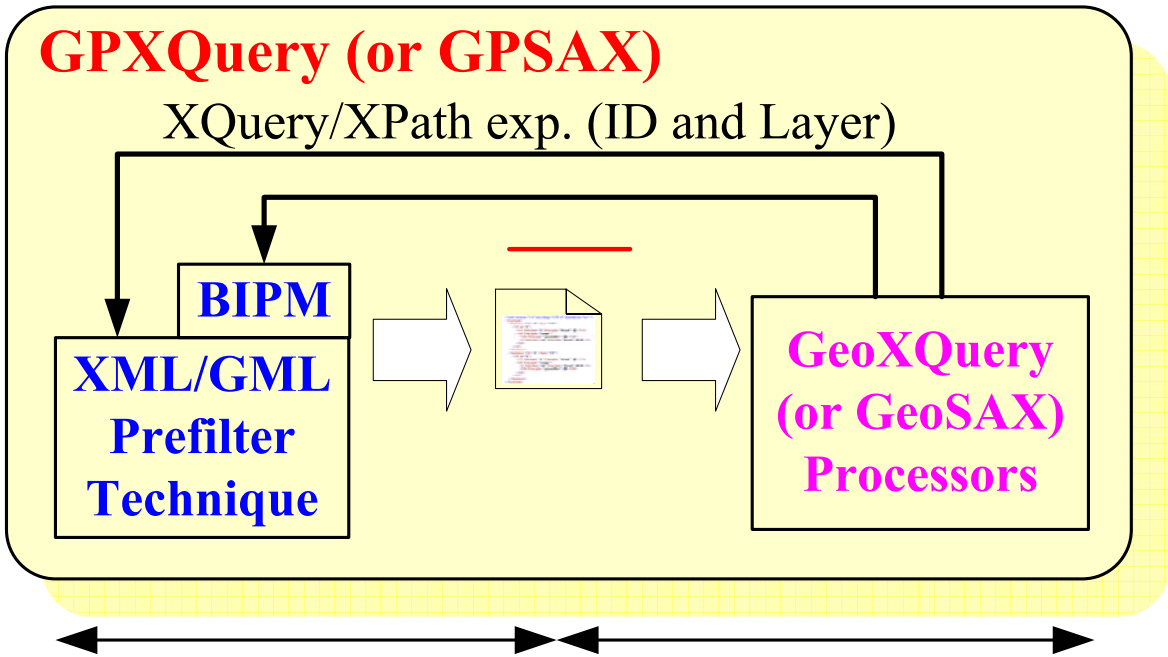
XML Prefiltering Technique - Idea



Returning to GeoXQuery/GeoSAX

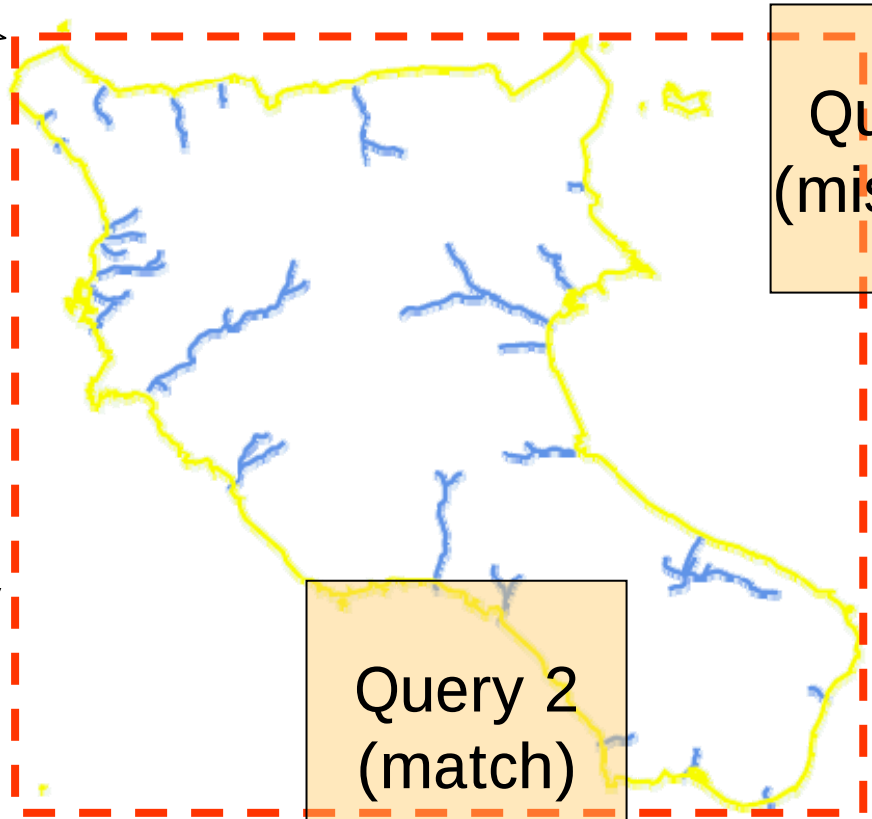
Bounding-Box Indexing Plug-in Module (BIPM) for the XML Pre-filter

- ***B***ounding-box ***I***ndexing ***P***lug-in ***M***odule (BIPM) is developed for the XML pre-filtering technique to perform geospatial filtering functionality.
- *BIPM* can index the boundary of each geographical feature in the documents and provides an intersection operation to query indexed features.



Bounded Box and Query

The Bounded Box (BBox) of the Geo-obj.



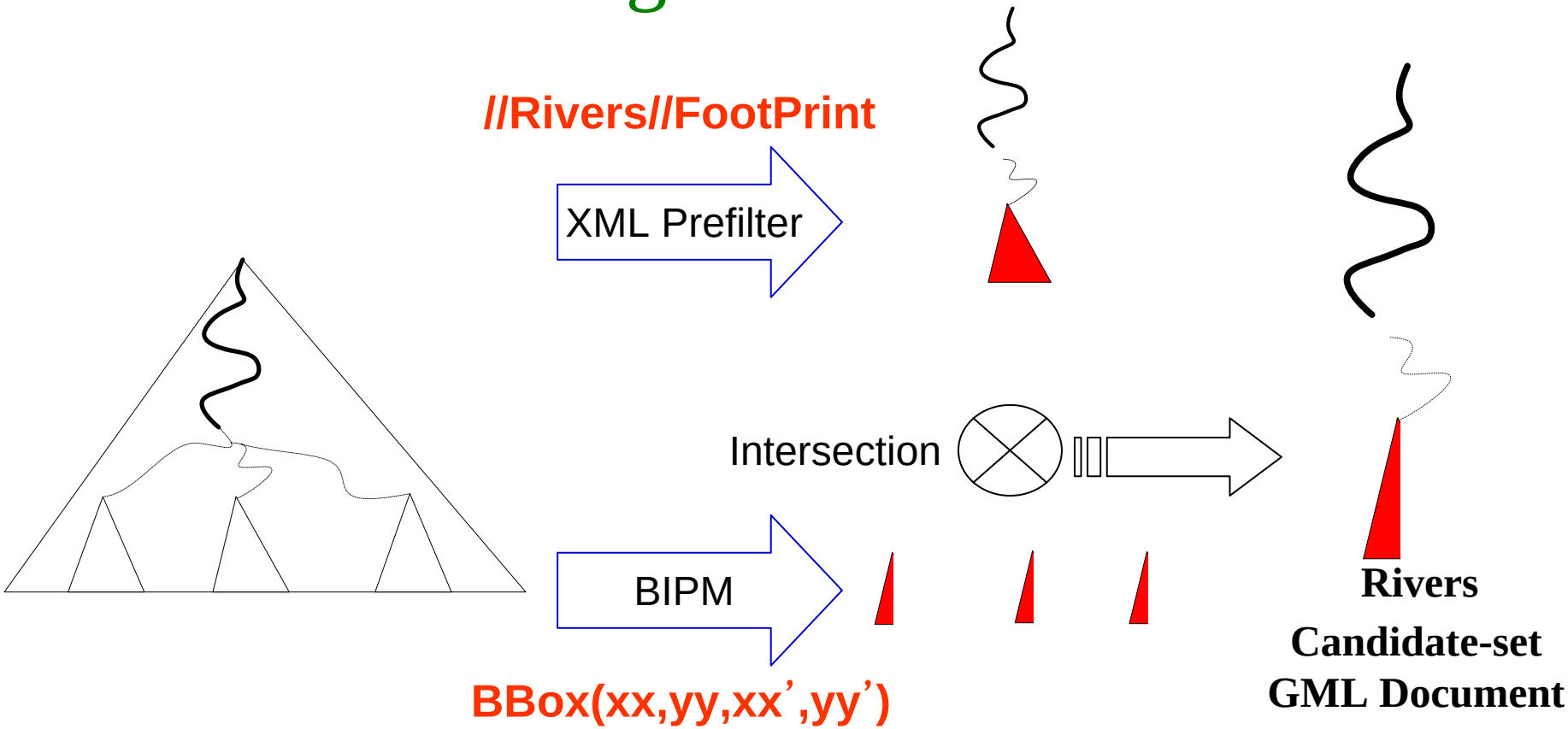
Query 1 (mismatch)

Query 3 (unmatch)

Query 2 (match)

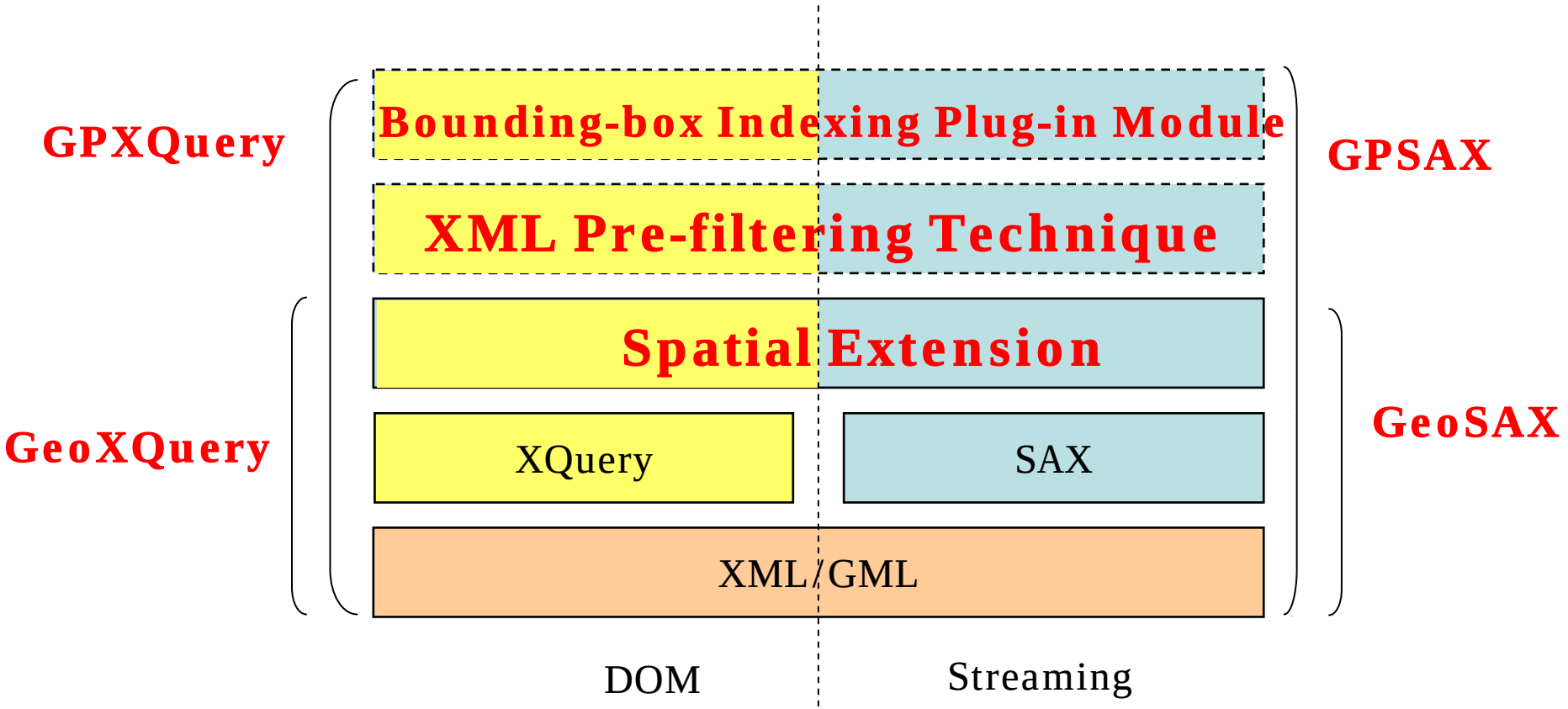
Matching Process:
1. Check BBox
2. Check boundary

Bounding-Box Indexing Plug-in Module - Idea



Final Pre-filtering Results

GML-native Processors



Institute of Information Science
Academia Sinica
中央研究院 資訊科學研究所

Performance Analysis

Environment and Datasets

- Six GML processors
  - *GeoXQuery*
  - *GPXQuery* with BIPM
  - *GPXQuery* without BIPM
  - *GeoSAX*
  - *GPSAX* with BIPM
  - *GPSAX* without BIPM

- Two datasets
  - **1.1 GB** GML document (the Taipei city)
  - **152 MB** GML document (the Xinyi area)

- Setup
  - an Intel Pentium-4 PC running at 2.53 GHz with **1 GB** DDR-RAM,
  - a 120 GB EIDE hard disk,
  - the MS Windows 2000 server.
  - Java 2 (Standard Edition V.1.4.2).

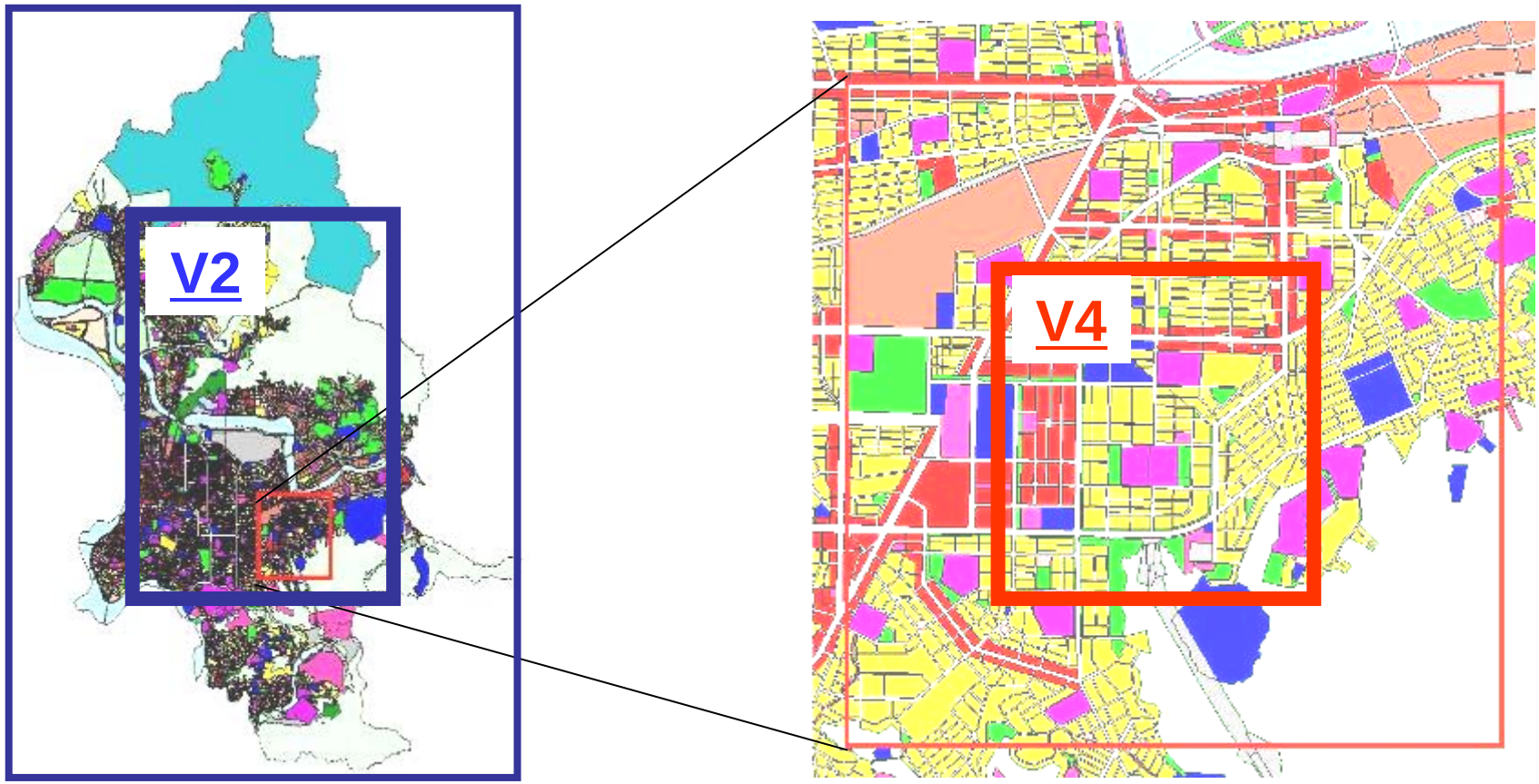
Query Constraints

Feature Class	XPath Expression	Size of the Fragment
FC_1	/TGML/ThemeGroup/ EnergySupplyUtilityGroup/ EnergySupplyUtilityMembers/ EnergySupplyUtility/FootPrint (Meaning: Selecting the Energy Supply Utility layer)	In Xinyi: 8KB In Taipei: 28KB
Predicate	**XPath Expression with a Predicate**	
P	//FootPrint [@id="18749"] (Returning a geo-feature)	
BBOX	**Coordinates**	**Range** (m^2)
V_2	(300000, 2767100)-(309775, 2782575)	9775 * 15475
V_4	(305991, 2768093)-(308233, 2770930)	2242 * 2837

Datasets

Large dataset—Taipei, 1.1 GB

Small dataset—Xinyi, 152 MB



Querying by a Feature ID XQuery-based Processors

The query returns a geo-feature.

Doc. 152MB	Run Time (s)	Memory (MB)
GeoXQuery	42	1065
GPXQuery with BIPM	**6**	**179**
GPXQuery without BIPM	**5**	**305**

Doc. 1.1GB	Run Time (s)	Memory (MB)
GeoXQuery	N/A	N/A
GPXQuery with BIPM	**38**	**179**
GPXQuery without BIPM	**32**	**305**

N/A means that the processor run out of memory and did not finish

The pre-filtering technique lowers resource consumption.

Querying by a Layer and a BBox XQuery-based Processors

The query returns the Energy Supply Utility layer in V_4.

Doc. 152MB	Run Time (s)	Memory (MB)
GeoXQuery	21	655
GPXQuery with BIPM	**2**	**39**
GPXQuery without BIPM	**2**	**39**

The query returns the Energy Supply Utility in V_2.

Doc. 1.1GB	Run Time (s)	Memory (MB)
GeoXQuery	N/A	N/A
GPXQuery with BIPM	**12**	**298**
GPXQuery without BIPM	**10**	**305**

The pre-filtering technique lowers resource consumption.

Querying by a BBox XQuery-based Processors

The query returns geo-features in V_4.

Doc. 152MB	Run Time (s)	Memory (MB)
GeoXQuery	N/A	N/A
GPXQuery with BIPM	**196**	**958**
GPXQuery without BIPM	N/A	N/A

The query returns geo-features in V_2.

Doc. 1.1GB	Run Time (s)	Memory (MB)
GeoXQuery	N/A	N/A
GPXQuery with BIPM	N/A	N/A
GPXQuery without BIPM	N/A	N/A

BIPM can **efficiently filter out** uninteresting geographic features.

Querying by a Feature ID SAX-based Processors

The query returns a geo-feature.

Doc. 152MB	Run Time (s)	Memory (MB)
GeoSAX	50	2
GPSAX with BIPM	**5**	23
GPSAX without BIPM	**4**	39

Doc. 1.1GB	Run Time (s)	Memory (MB)
GeoSAX	233	2
GPSAX with BIPM	**36**	298
GPSAX without BIPM	**29**	305

The pre-filtering technique lowers the run time but increases memory consumption.

Querying by a Layer and a BBox SAX-based Processors

The query returns the Energy Supply Utility layer in V_4.

Doc. 152MB	Run Time (s)	Memory (MB)
GeoSAX	57	2
GPSAX with BIPM	**2**	39
GPSAX without BIPM	**1**	39

The query returns the Energy Supply Utility in V_2.

Doc. 1.1GB	Run Time (s)	Memory (MB)
GeoSAX	242	2
GPSAX with BIPM	**11**	298
GPSAX without BIPM	**9**	305

The pre-filtering technique lowers the run time but increases memory consumption.

Querying by a BBox SAX-based Processors

The query returns geo-features in V_4.

Doc. 152MB	Run Time (s)	Memory (MB)
GeoSAX	158	18
GPSAX with BIPM	512	57
GPSAX without BIPM	1343	57

The Cost of pre-filtering GML docs.

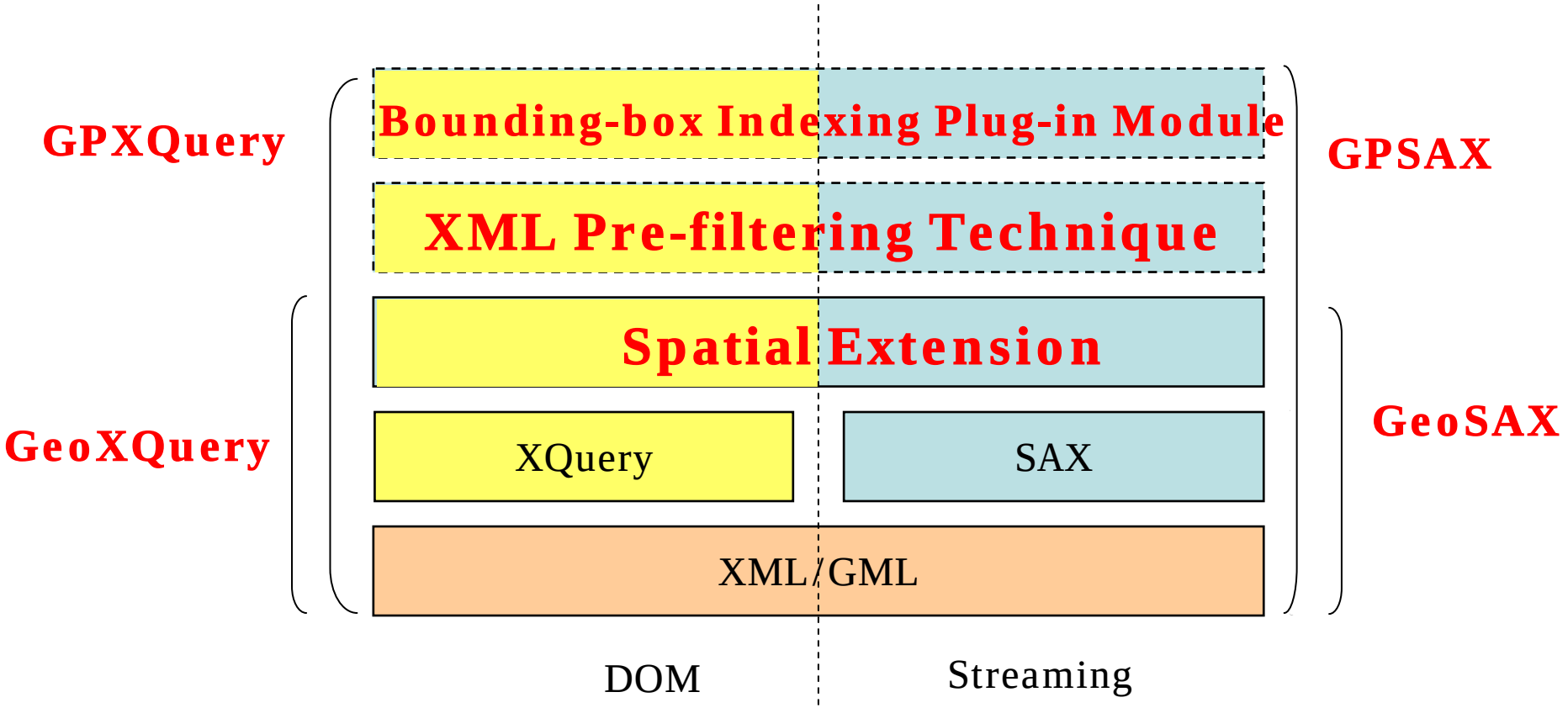
The query returns geo-features in V_2.

Doc. 1.1GB	Run Time (s)	Memory (MB)
GeoSAX	1008	243
GPSAX with BIPM	5968	767
GPSAX without BIPM	13802	489

Performance Characteristics Analysis

- **SAX-based methods** (*GeoSAX* and *GPSAX*) are more suitable for simple queries run against huge documents since they consume relatively **less memory** as compared to XQuery-based methods (*GeoXQuery* and *GPXQuery*).
- The **pre-filtering technique lowers resource consumption**, particularly about memory space usage and disk access time.
- The **geospatial plug-in module** of the pre-filtering technique can **efficiently filter out** uninteresting geographic features with **a little additional effort**.

CONCLUSIONS



- The *GPXQuery* and *GPSAX* processors have been employed to develop a GML-based Online GIS, along with a geospatial query interface and a SVG map navigator.

Future Work

- The GML-based GIS reported here is still impractical for dealing with gigabyte-sized geospatial datasets.
- We suggest the following areas for improvement.
  - Efficient index management system for the prefilter
  - XML/GML documents updating
  - Schema level operation

Institute of Information Science
Academia Sinica
中央研究院 資訊科學研究所

Thank you for your attention!!

Questions and Comments

All the software packages are accessible at
http://www.iis.sinica.edu.tw/~jashing/prefiltering/

Acknowledgements:
Many thanks to 1) the reviewers of the paper,
2) National Science Council of Taiwan,
3) Open Geospatial Information Team at IIS/AS, and
4) Juliana Williams and Andreas Neumann at www.carto.net.