

An aerial view of a city with a river, greenery, and buildings. The text is overlaid on this background.

Towards Geo-spatial Application Provisioning: *Modeling Spatial data and Computation in Geo-Cloud*

Shreya Ghosh, Arindam Dasgupta

Under the supervision of

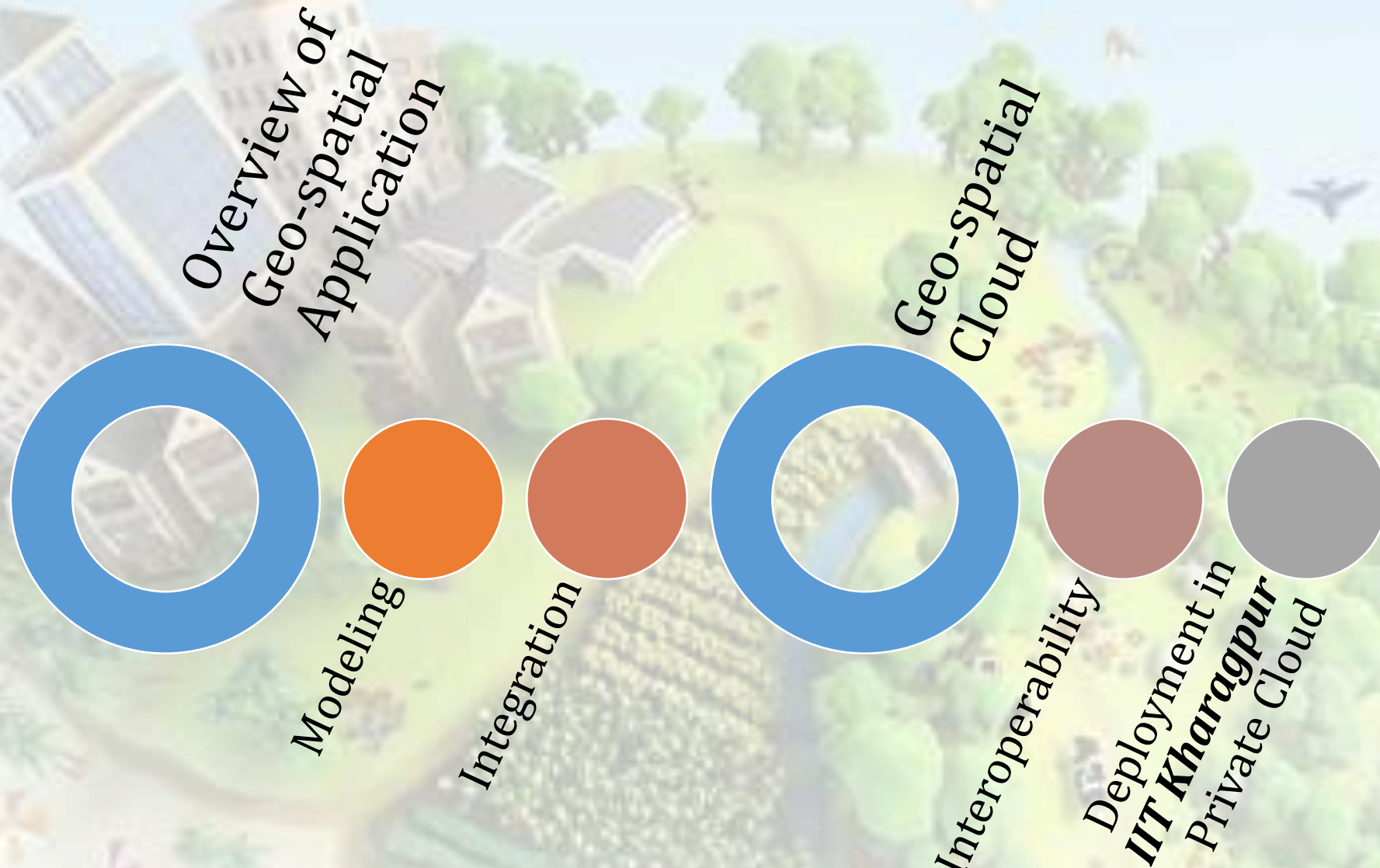
Prof. Soumya K Ghosh

Spatial Data Science Lab

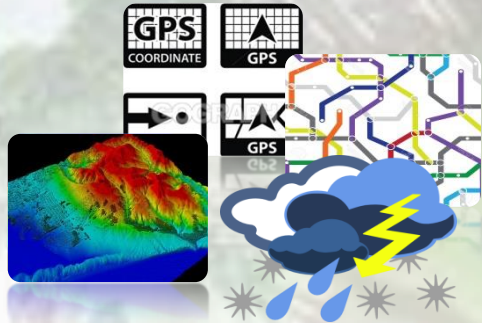
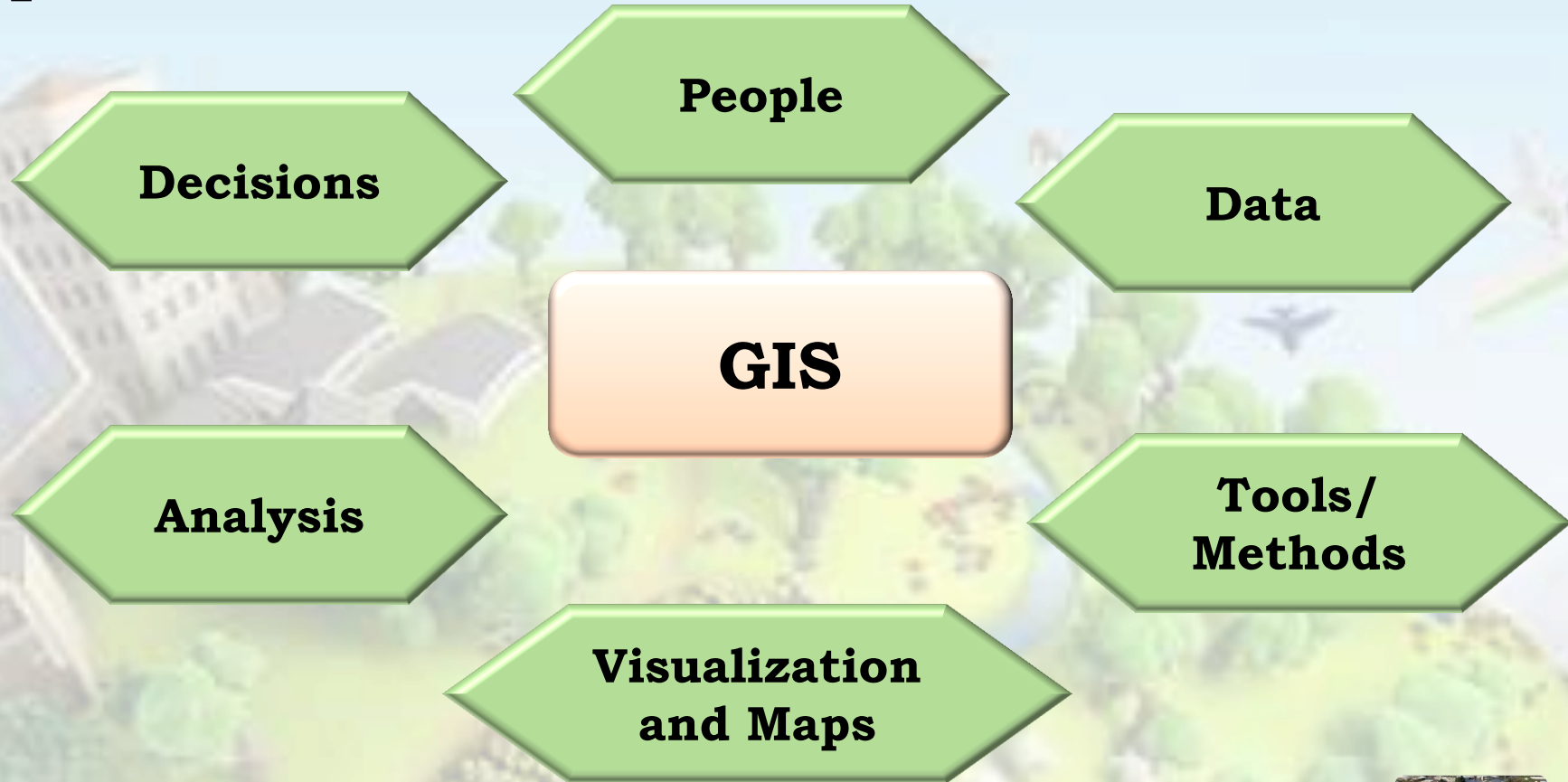
Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Contents



Geospatial Application



Identify



Analyze



Inference

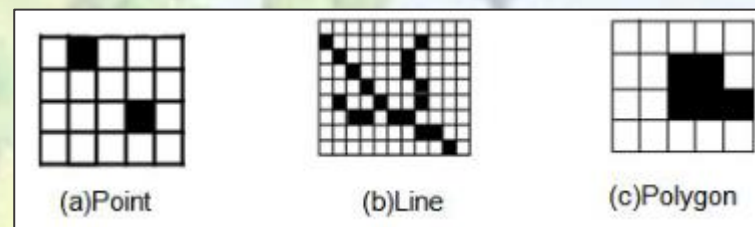
Spatial Data Modeling

- Spatial data is comprised of objects in multi-dimensional space

Vector Data Model



Raster Data Model



- Need a structural representation of spatial data sets – *easy to share, access and analyze!*



Spatial Data Modeling?



Logical Data Modeling



Logical Data Model to XMI and XSD



Logical Data Model to Database Schema



Logical Data Modeling

A **logical data model** or logical schema is a data model of a specific problem domain expressed independently of a particular database management product or storage technology (physical data model) but in terms of data structures such as relational tables and columns, object-oriented classes, or XML tags.

A logical data model (class diagram) includes

- **entities** (tables)
- **attributes** (columns/fields) and
- **relationships** (keys)

Logical data model of a ROI

<i>lulc_kolkata</i>
<i>id [primary-key]: varchar</i> <i>lulc_code: varchar</i> <i>shape_leng: number</i> <i>shape_area: number</i> <i>shape: geom</i>
<i>find_lulc(lulc_code)</i>

<i>population_kolkata</i>
<i>id [primary-key]: varchar</i> <i>density: varchar</i> <i>edu_density: varchar</i> <i>child_density: varchar</i>
<i>find_maxDen ()</i>



Logical Data Model (Class Diagram) of Spatial Data Repositories

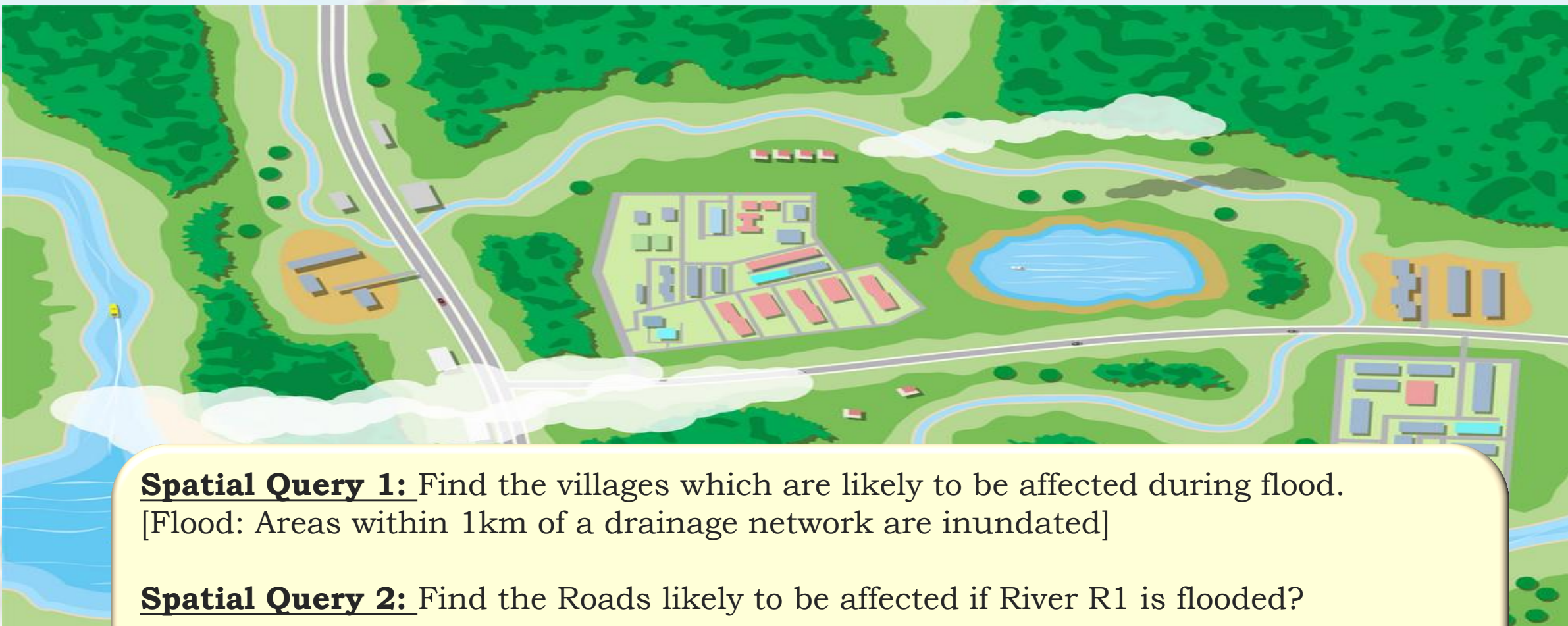
DEMO



Example Scenario

Consider 4 independent repositories of a region **P**, namely, **ROAD (R)**, **DRAINAGE (D)**, **VILLAGE (V)** and **ADMIN BLOCKS (A)**.

[Road: Polyline; Drainage: Polyline; Village: Point; Admin: Polygon]



Spatial Query 1: Find the villages which are likely to be affected during flood.
[Flood: Areas within 1km of a drainage network are inundated]

Spatial Query 2: Find the Roads likely to be affected if River R1 is flooded?

Spatial Query 3: To setup a new industry the requirement is: It should be in Admin Blocks A2 or A7, 2km from NH, no Drainage within 1km, within 5km of villages with working population (20-50yrs) greater than 100

Consider 4 independent repositories of a region **P**, namely, **ROAD (R)**, **DRAINAGE (D)**, **VILLAGE (V)** and **ADMIN BLOCKS (A)**. [Road: Polyline; Drainage: Polyline; Village: Point; Admin: Polygon]

Spatial Query 1: Find the villages which are likely to be affected during flood.

[Flood: Areas within 1km of a drainage network are inundated]

```
SELECT V.vill_id, V.vill_name FROM VILLAGE V, DRAINAGE D
WHERE OVERLAP(V.shape, BUFFER(D.shape,1000))=1;
```

Spatial Query 2: Find the Roads likely to be affected if River R1 is flooded?

```
SELECT R.road_id, V.vill_name FROM VILLAGE V, DRAINAGE D
WHERE OVERLAP(V.shape, BUFFER(D.shape,1000))=1
AND D.dr_name="R1";
```

Consider 4 independent repositories of a region **P**, namely, **ROAD (R)**, **DRAINAGE (D)**, **VILLAGE (V)** and **ADMIN BLOCKS (A)**. [Road: Polyline; Drainage: Polyline; Village: Point; Admin: Polygon]

Spatial Query 3: To setup a new industry the requirement is: It should be in Admin Blocks A2 or A7, 2km from NH, no Drainage within 1km, within 5km of villages with working population (20-50yrs) greater than 100

```
        Create VIEW REG AS (  
SELECT INTERSECT(V.shape,A.shape) AS REG_SHAPE FROM ROAD R, DRAINAGE D, VILLAGE V,  
        CITIZEN C WHERE  
        OVERLAP(V.shape, BUFFER(D.shape,1000))=0 AND  
OVERLAP(V.shape, BUFFER(R.shape,2000))=1 AND COUNT(C.citizen_id)>=100 WHERE  
        C.age>20 AND C.age<50 AND C.residential==V.vill_id)  
  
SELECT INTERSECT (REG_SHAPE,A.shape) FROM REG, ADMIN_BLOCKS A  
        WHERE  
        A.block_name IN ("A2","A7") AND  
OVERLAP(A.shape, BUFFER(REG_SHAPE,5000))=1
```


Logical Data Model to XMI and XSD



❑ **XML** is a *markup language* that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable

❑ Main purposes behind the proposal of XMI were as follows:

- ❖ To help programmers, using the Unified Modeling Language (UML), with different languages and development tools to exchange their data models with each other
- ❖ To facilitate in exchanging information about data warehouses

❑ **XSD** is an XML schema definition language which can be used to express a set of rules to which an XML document must conform in order to be considered 'valid' according to that schema

❑ It is designed with the intent that determination of a document's validity would produce a collection of information, adhering to specific data types

DEMO



DEMO

Logical Data Model to Database Schema



- ❑ A database schema of a database system is *its structure described in a formal language supported by the database management system (DBMS)*
- ❑ It refers to the organization of data as a blueprint of how a database is constructed (*divided into database tables in case of Relational Databases*)
- ❑ In the context of Oracle databases, a *schema object is a logical data storage structure*
- ❑ In general, the *language by which the database schema is described is called Data Definition language (DDL)*

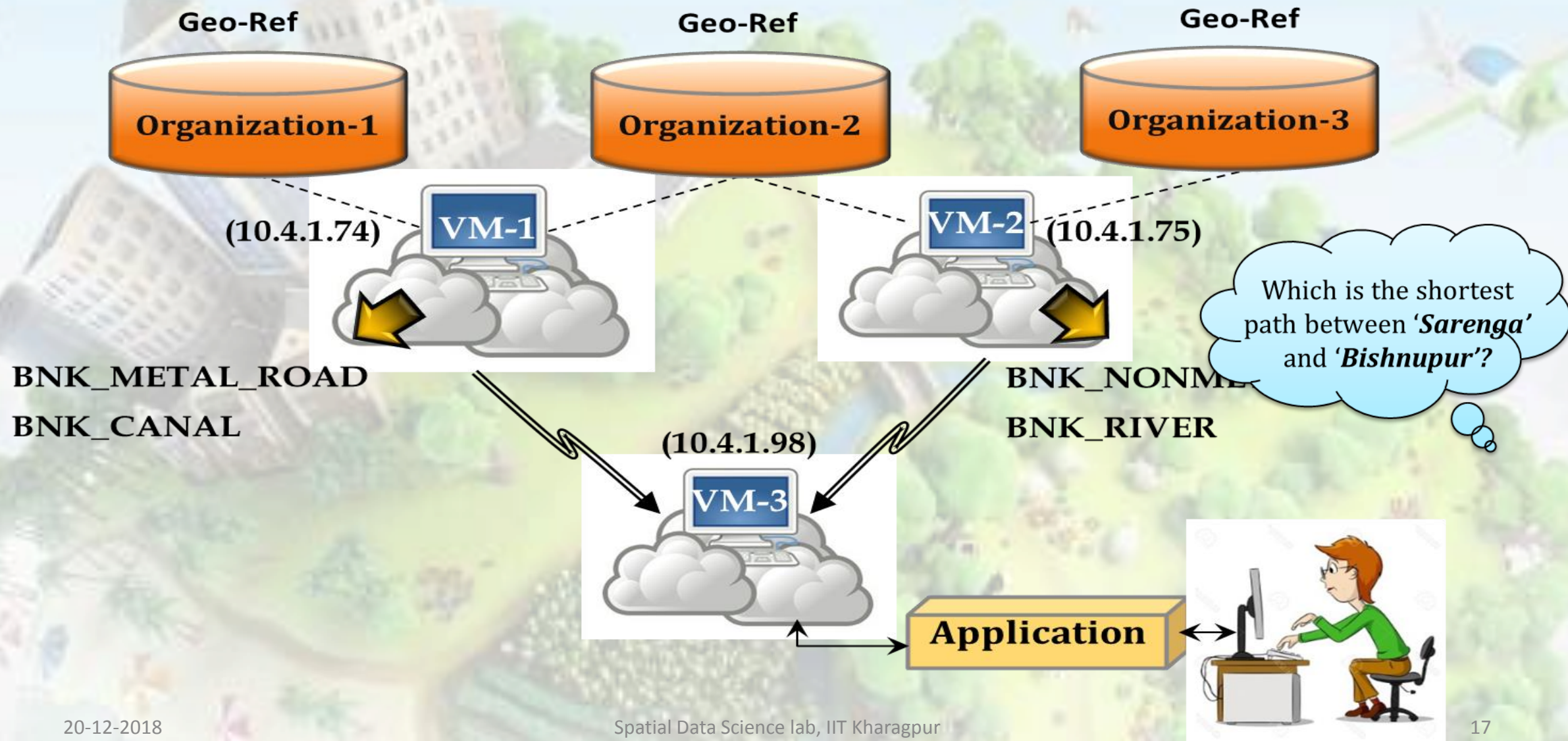
DEMO

Integration of Data Models?

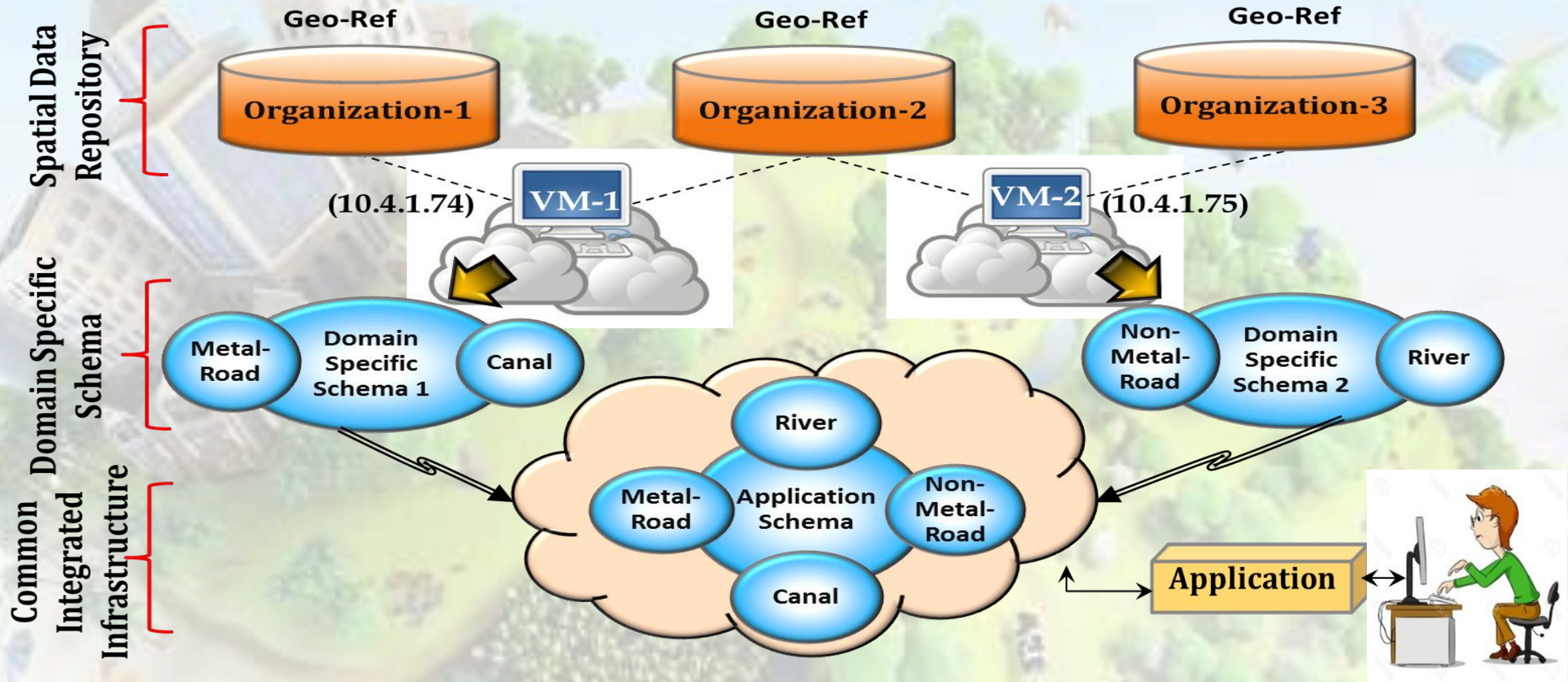


- Every organization has own data repository of geo-databases
- *Data repositories are isolated* because of no sharing mechanism
- *The formats of data in repositories are varying among organizations*
- Meaning of another organization's data *difficult to interpret*
- Sharing of data will be very *effective* if data repositories are integrated

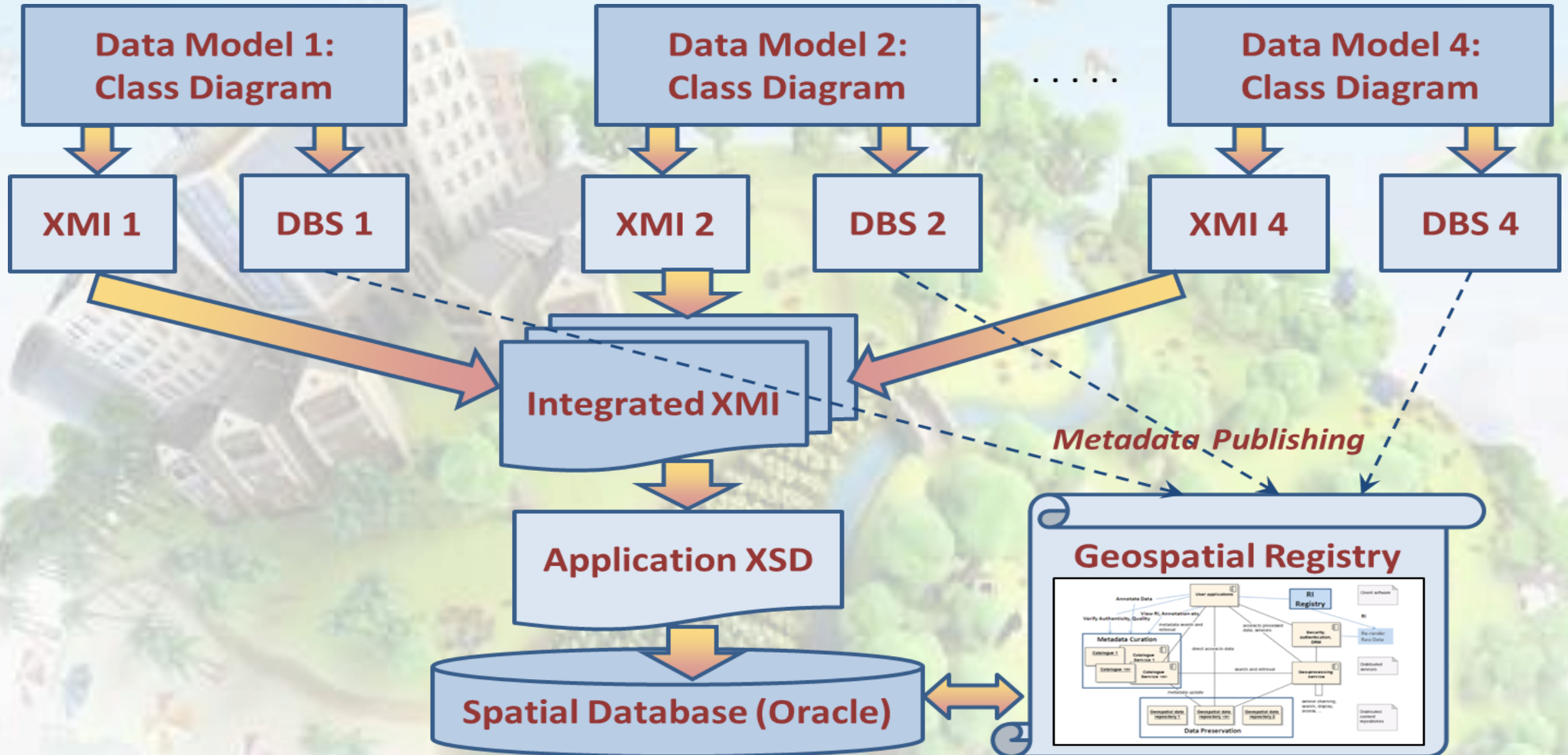
Integration of Data Models?



Integration of Data Models



Integration of Data Models



Integrated XMI Generation: GUI Design



Effective if organizations model their own database in UML

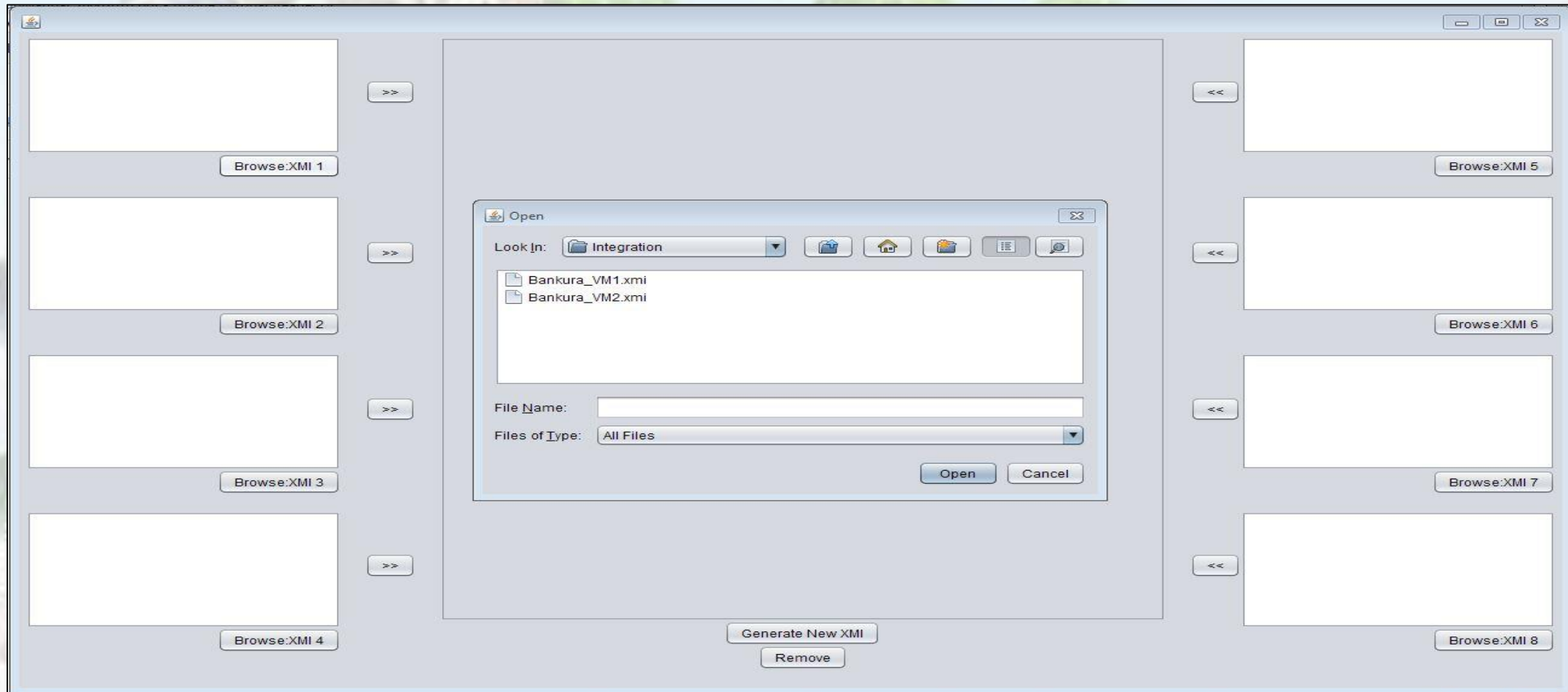
Exchange of XMI support for all type of UML tools

Useful data can be ***extracted*** from others XMI

Integrate individual data into a single XMI

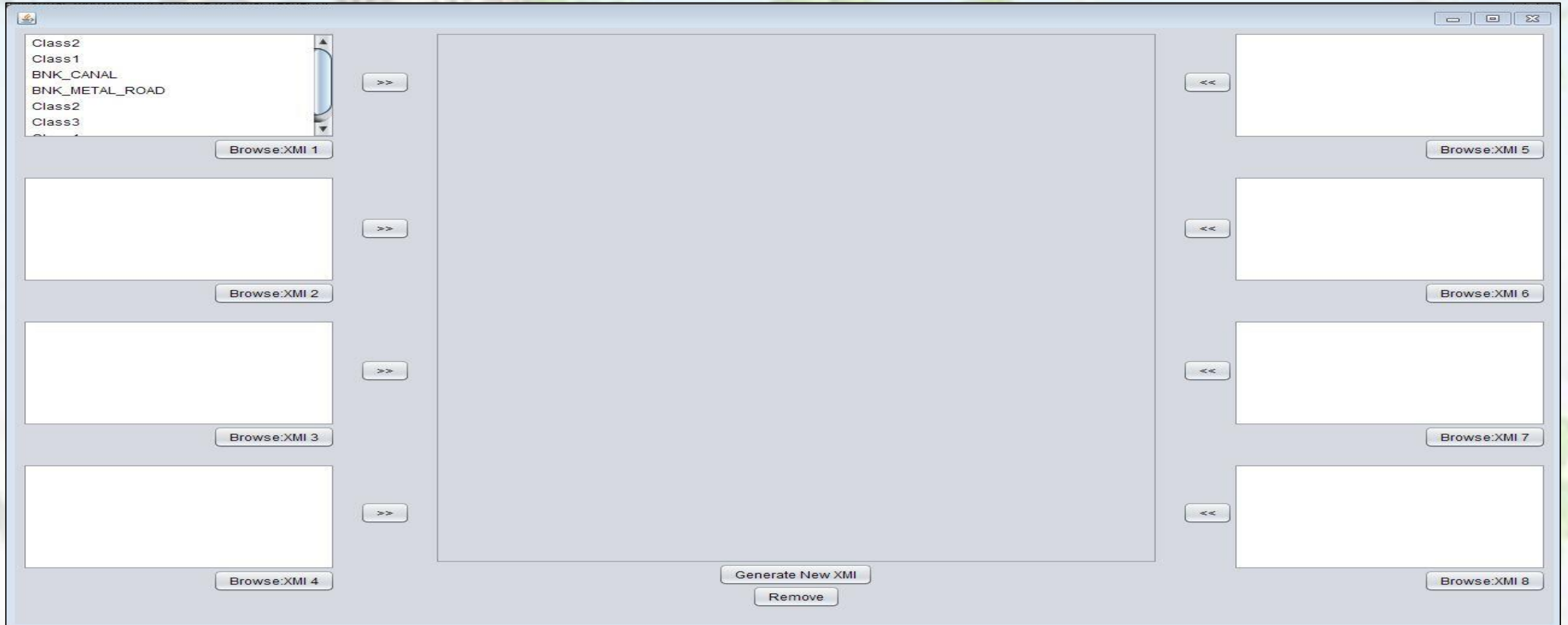
Integrated XMI Generation: GUI Design

Browse XMI Models



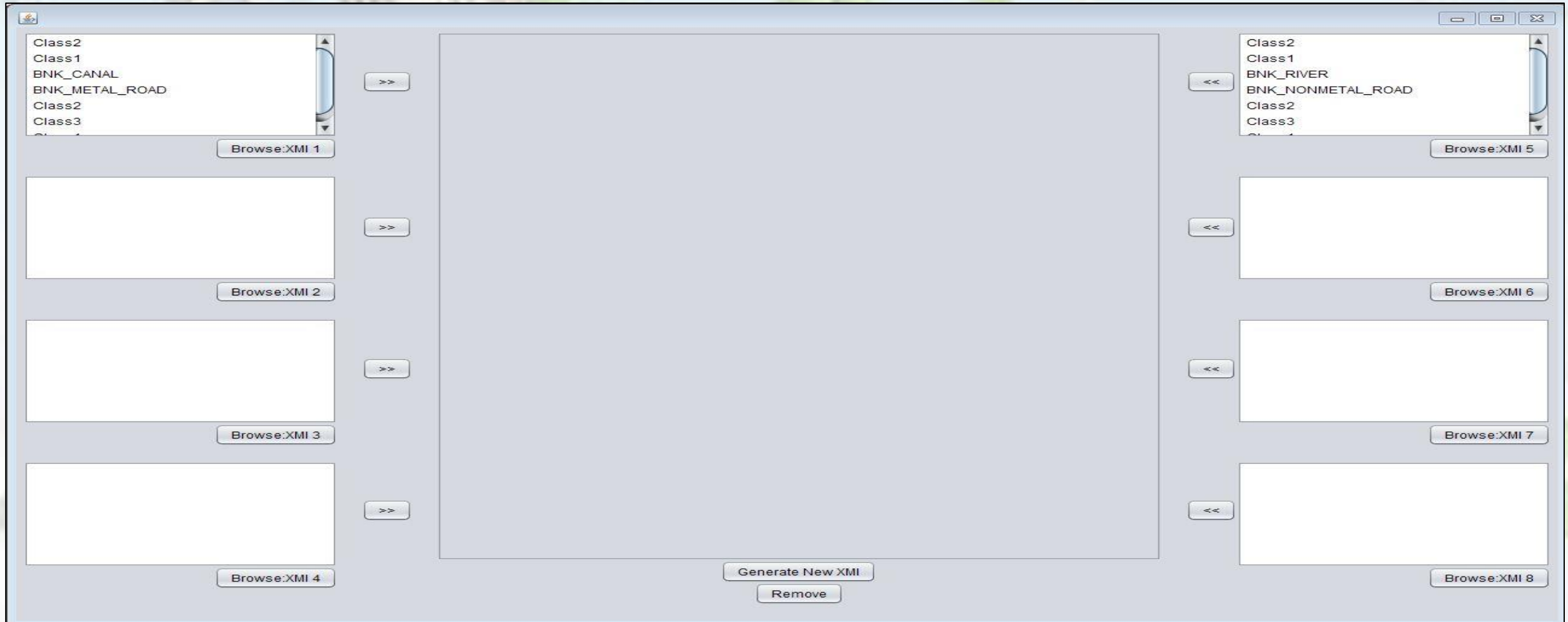
Integrated XMI Generation: GUI Design

Retrieve Classes



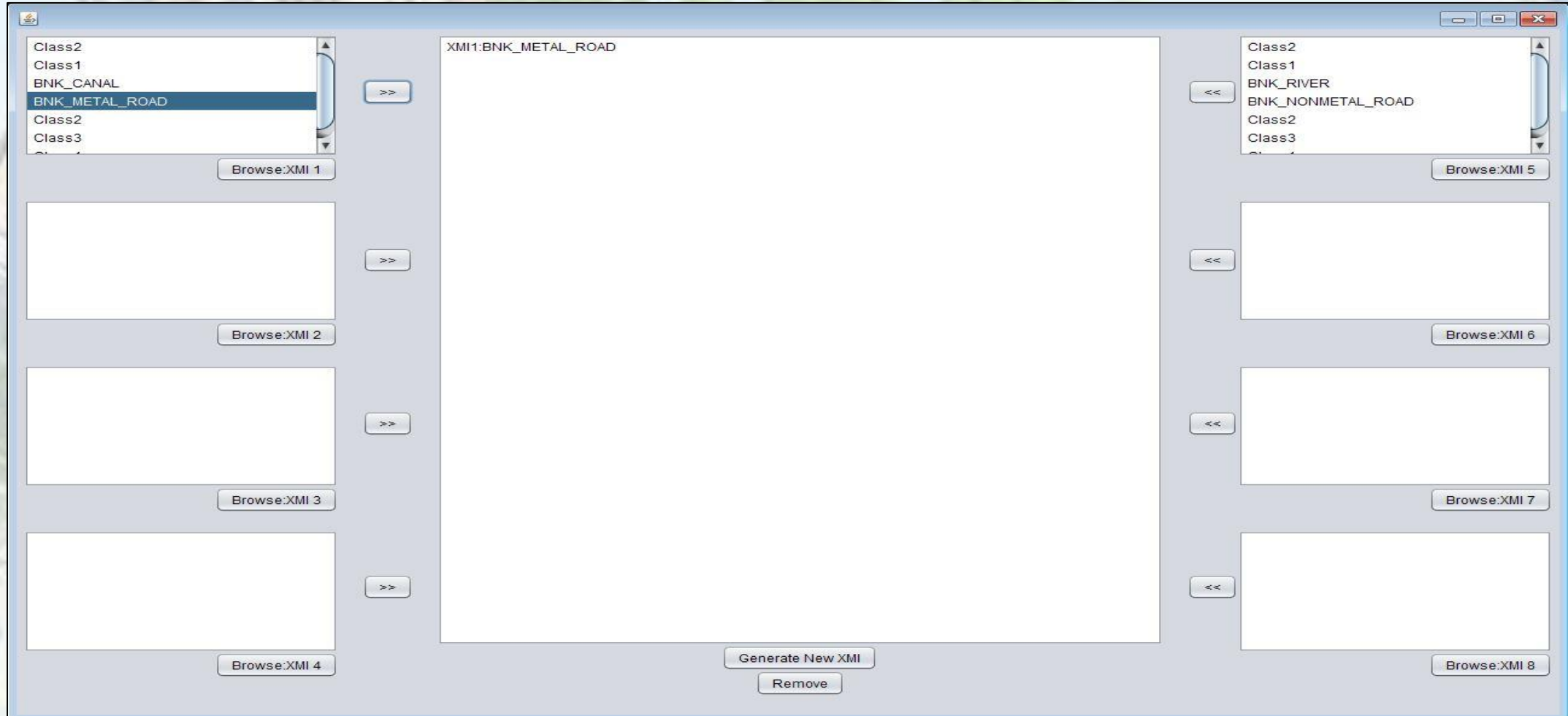
Integrated XMI Generation: GUI Design

Retrieve Classes



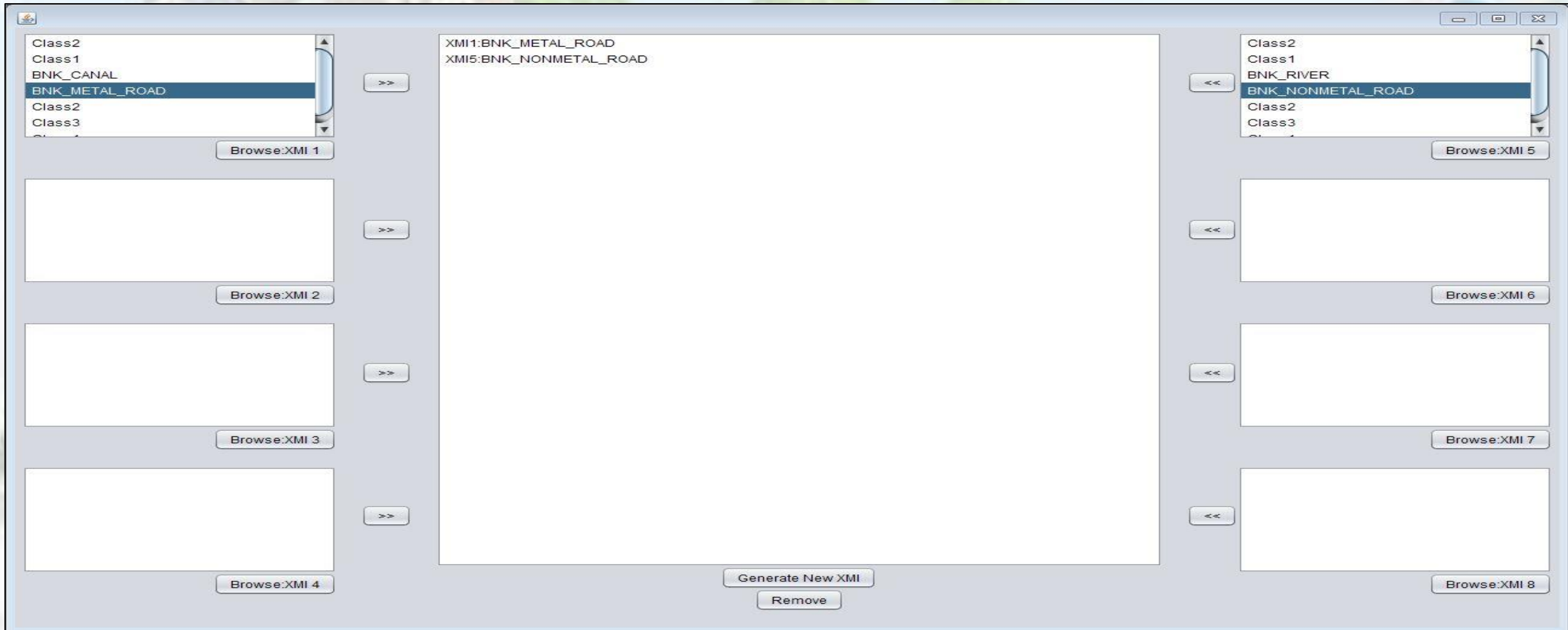
Integrated XMI Generation: GUI Design

Specific Class Selection



Integrated XMI Generation: GUI Design

Specific Class Selection (contd.)



Generating Integrated UML



The screenshot displays the Enterprise Architect (EA) software interface. The title bar reads "Integrated_DataModel2 - EA - 30 Day Trial". The ribbon menu is set to "Publish", with sub-ribbons for "Documentation", "Model Exchange", "Technology", and "Technologies". The "Publish" sub-ribbon includes tools like "Reusable Assets", "Publish As...", "CSV", "Package Control", "Export XMI", and "Import XMI". The "Technology" sub-ribbon has a "Publish" tool. The "Technologies" sub-ribbon lists "ArcGIS", "ArchiMate", "ODM", "GML", "NIEM", and "NIEM 3.0".

On the left, the "Project Browser" shows a tree structure with "Model" and "Class Model". Below it, the "Properties" window is open for the "Class Model", showing "General Settings" (Name, Scope, Type, Stereotype, Alias, Complexity, Version, Phase, Language, Filename) and "Project" and "Advanced" sections.

The main workspace is currently empty, showing a "Start Page" tab. The bottom status bar includes a zoom slider and labels for "CAP", "NUM", "SCRL", and "CLOUD".

Importing Integrated XMI



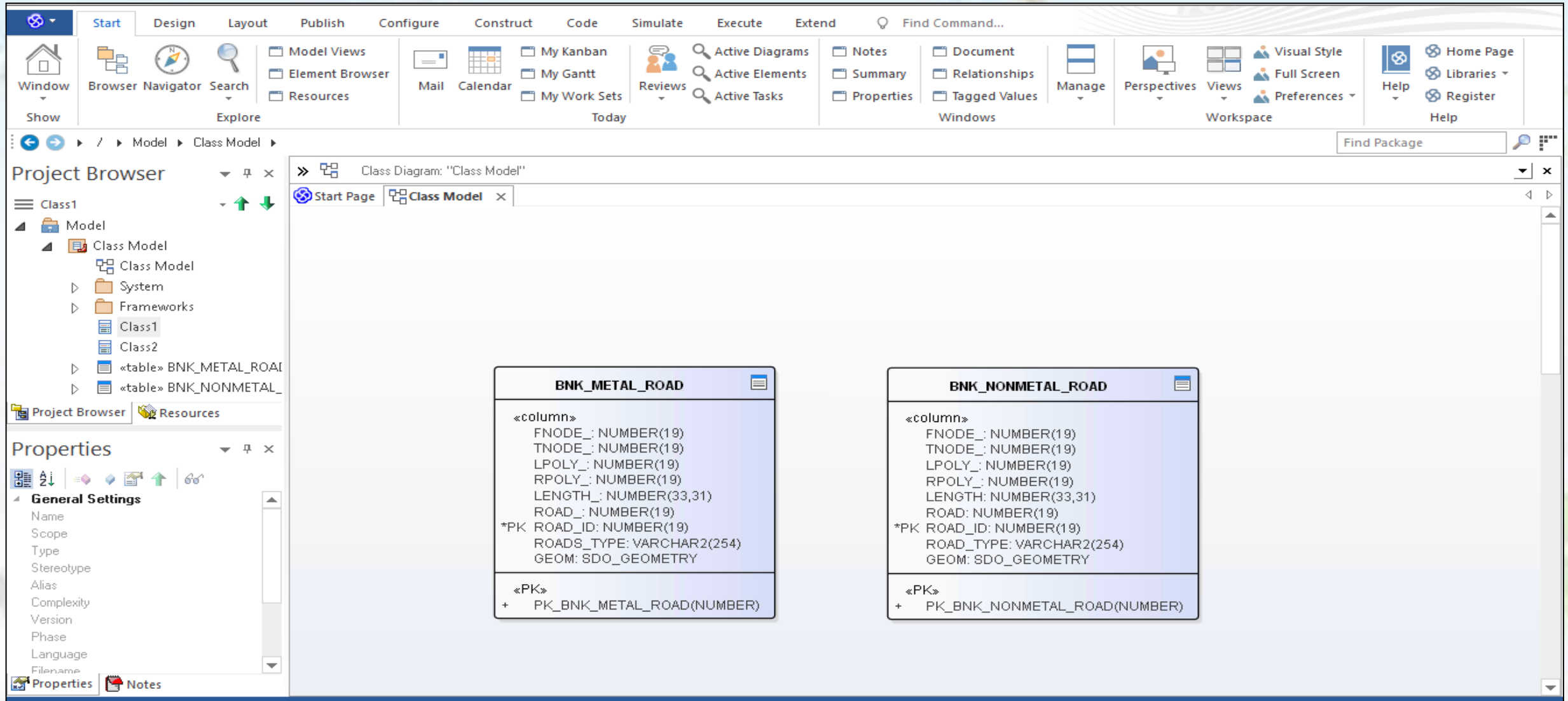
The screenshot displays the Bentley Systems software interface, titled "Integrated_DataModel2 - EA - 30 Day Trial". The ribbon menu is set to the "Publish" tab, with the "Extend" sub-tab active. The "Import XMI" button is highlighted, and its dropdown menu is open, showing two options: "Import Model XMI..." and "Batch Import Controlled Packages...".

The interface includes several panels:

- Project Browser:** Shows a tree view with "Model" expanded to "Class Model".
- Properties:** Shows a list of properties under "General Settings", including Name, Scope, Type, Stereotype, Alias, Complexity, Version, Phase, Language, and Filename. It also has sections for "Project" and "Advanced".
- Resources:** A panel for managing resources.

The status bar at the bottom left contains the text: "Import a model from file based XMI content (Ctrl+Alt+I)". The bottom right of the status bar shows system tray icons for CAP, NUM, SCRL, and CLOUD.

Integrated UML Class Diagram



The screenshot displays the Oracle SQL Developer interface with the 'Class Model' UML diagram open. The diagram shows two classes, `BNK_METAL_ROAD` and `BNK_NONMETAL_ROAD`, both represented as column objects. The `BNK_METAL_ROAD` class has a primary key `ROAD_ID` and a foreign key `PK_BNK_METAL_ROAD` that references the `ROAD_ID` of the `BNK_NONMETAL_ROAD` class. The `BNK_NONMETAL_ROAD` class also has a primary key `ROAD_ID` and a foreign key `PK_BNK_NONMETAL_ROAD`.

BNK_METAL_ROAD

```

«column»
  FNODE_: NUMBER(19)
  TNODE_: NUMBER(19)
  LPOLY_: NUMBER(19)
  RPOLY_: NUMBER(19)
  LENGTH_: NUMBER(33,31)
  ROAD_: NUMBER(19)
  *PK ROAD_ID: NUMBER(19)
  ROADS_TYPE: VARCHAR2(254)
  GEOM: SDO_GEOMETRY

«PK»
+ PK_BNK_METAL_ROAD(NUMBER)
  
```

BNK_NONMETAL_ROAD

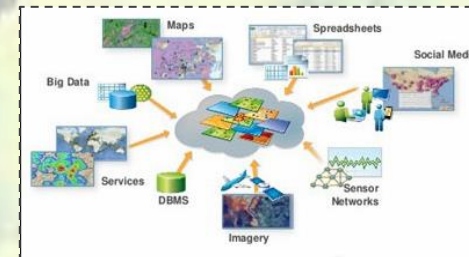
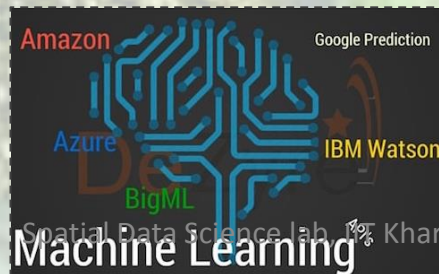
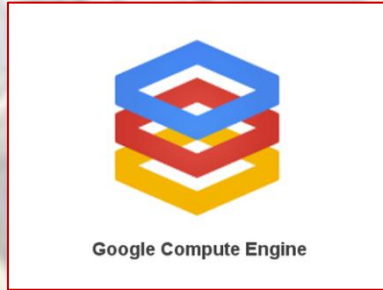
```

«column»
  FNODE_: NUMBER(19)
  TNODE_: NUMBER(19)
  LPOLY_: NUMBER(19)
  RPOLY_: NUMBER(19)
  LENGTH_: NUMBER(33,31)
  ROAD_: NUMBER(19)
  *PK ROAD_ID: NUMBER(19)
  ROADS_TYPE: VARCHAR2(254)
  GEOM: SDO_GEOMETRY

«PK»
+ PK_BNK_NONMETAL_ROAD(NUMBER)
  
```

Geo-Cloud Application







Geo-Cloud Computing paradigm?

- On-demand self service
- Ubiquitous Network Access
- Resource Pooling
- Location Independence
- Rapid Elasticity
- Measured Services
- Virtualization

- Private and public organization wants to share their spatial data
- Provision of Geo-spatial services
- Geo spatial computations

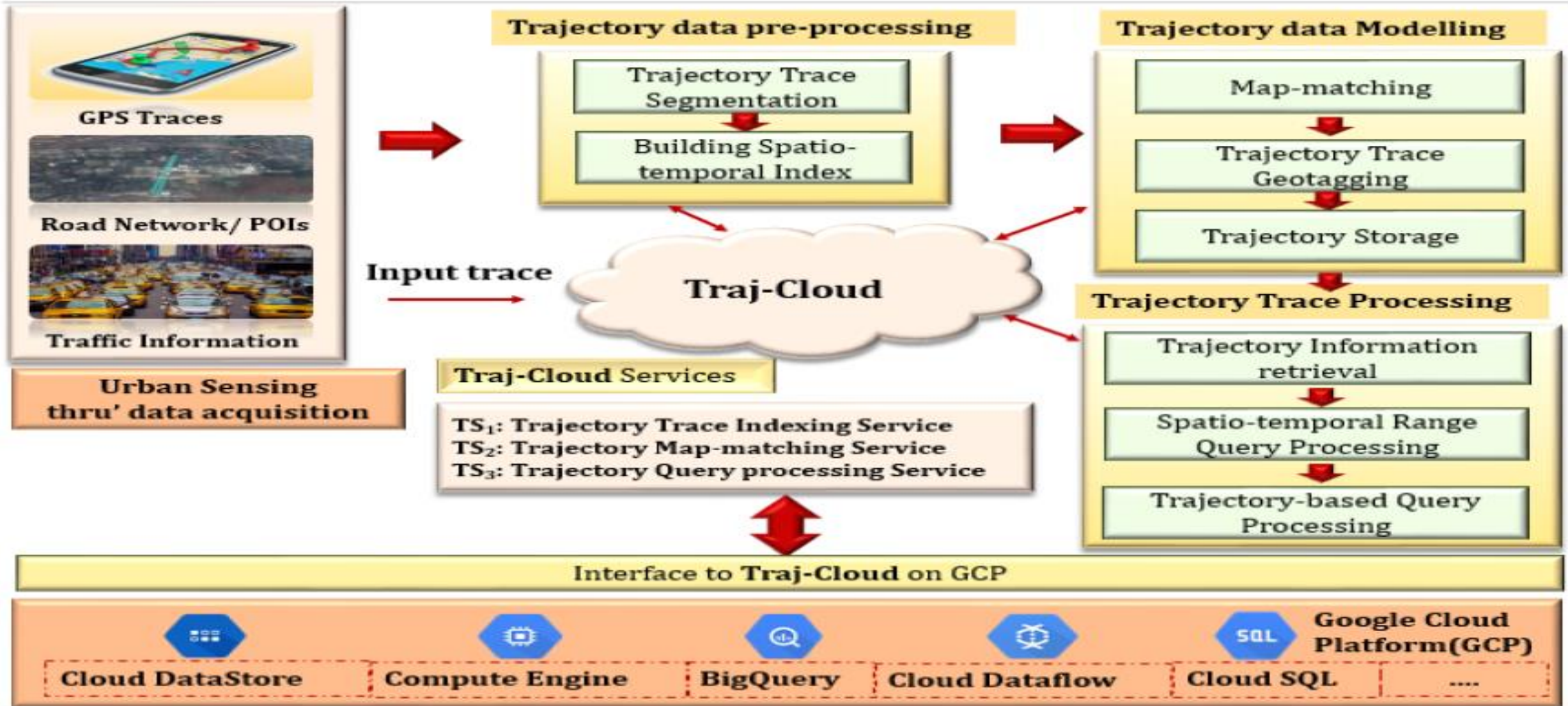


Use Case 1: Traj-Cloud

Traj-Cloud is a SaaS cloud which provides three major mobility services, namely,

- ❑ ***trajectory-indexing*** to efficiently handle huge real-time trajectory updates;
- ❑ ***geo-tagging, map-matching*** services in a distributed computing platform (i.e., Google Compute Engine) and
- ❑ ***trajectory-processing*** to resolve ***mobility based spatio-temporal queries*** to improve efficiency and reduce response time

Traj-Cloud



Traj-Cloud Service: Trajectory Query Processing



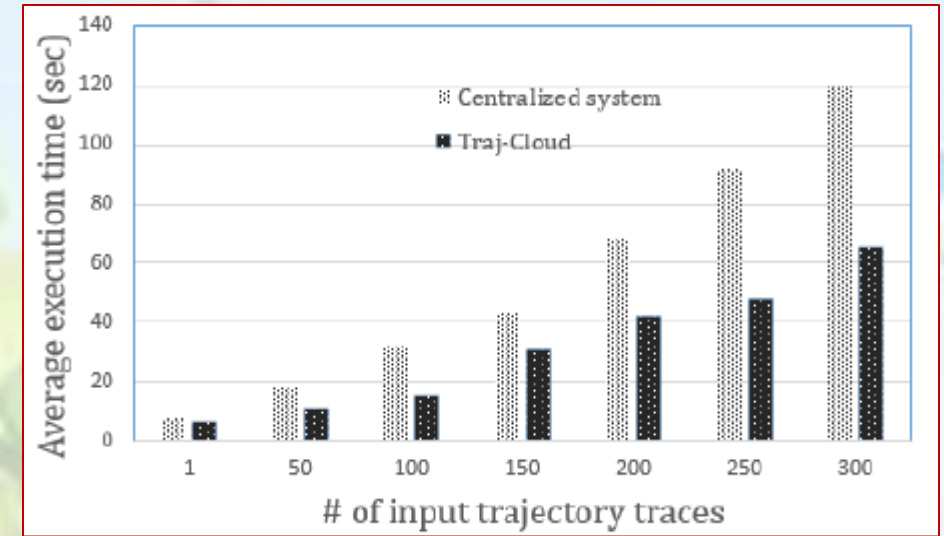
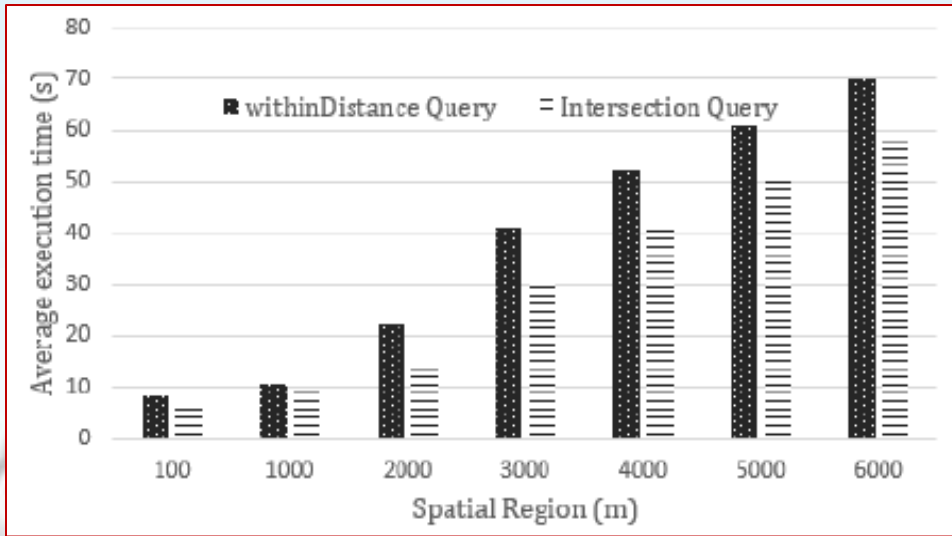
- Find all the petrol stations within 500m distance of a vehicle's trajectory

```
Select poi_Id from POI P, Traj T
where P.id="petrol_st" and
overlap(P.shape, Buffer(T.shape, 500))=1 ;
```

- Find find trajectory segments passing through the residential area of a city

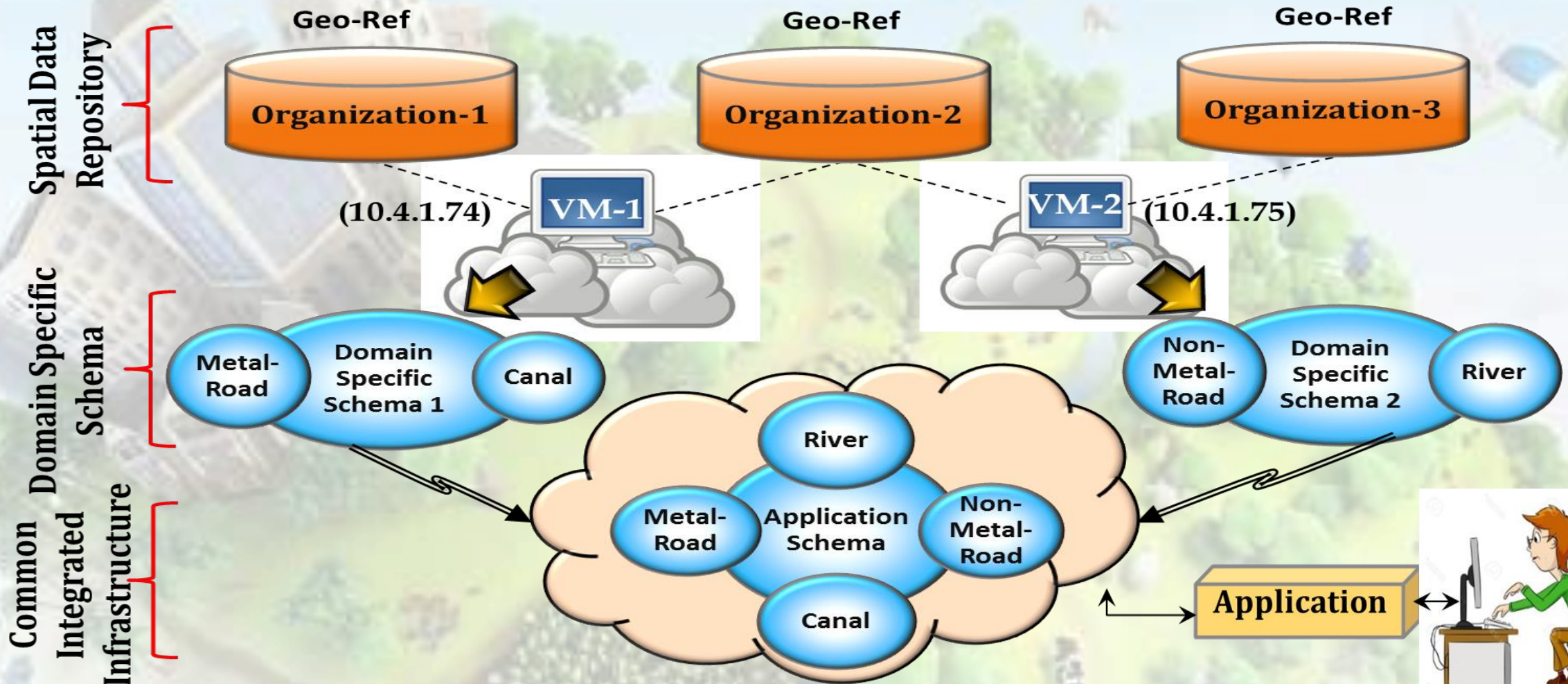
```
Select traj_Id from Traj T, Region R
where R.id="residential_zone" and
cross(T.shape, R.shape)=1 ;
```


Efficiency of Traj-Cloud



Type of Query	Linear Scan (Standalone)	Linear Scan (BigQuery)	Indexing (BigQuery)	Indexing + MapReduce (BigQuery + CloudSQL + Dataproc)
Point (10)	18.8 s	14.7 s	10.6s	6.2s
Range (10)	40.3 s	32.8s s	24.1s	13s
Point (100)	200.6 s	180s s	104s	85.8s
Range (100)	800.2s	450.1s	180s	142s
Point (500)	1780s	1000s	630.7s	432s
Range (500)	2540s	1860s	1020s	780s

Use case 2: Integration of Data Models



Integrated UML Class Diagram

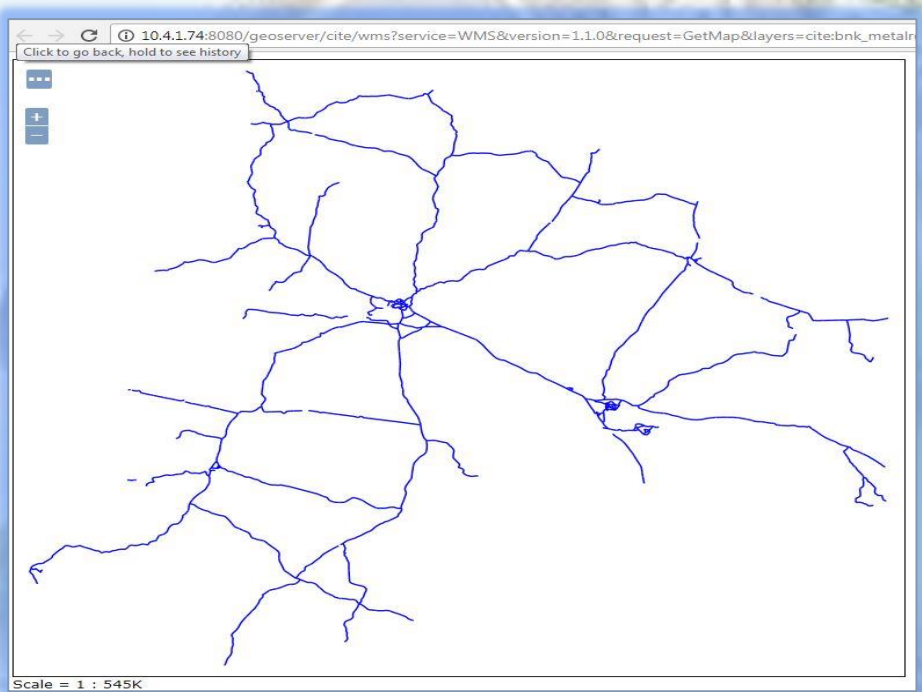


The screenshot displays a software interface with a top menu bar (Start, Design, Layout, Publish, Configure, Construct, Code, Simulate, Execute, Extend, Find Command...), a toolbar with icons for Window, Browser Navigator, Search, Model Views, Element Browser, Resources, Mail, Calendar, My Kanban, My Gantt, My Work Sets, Reviews, Active Diagrams, Active Elements, Active Tasks, Notes, Summary, Properties, Document, Relationships, Tagged Values, Manage, Perspectives, Views, Visual Style, Full Screen, Preferences, Help, Home Page, Libraries, and Register. The main workspace shows a UML Class Diagram titled "Class Model" with two classes:

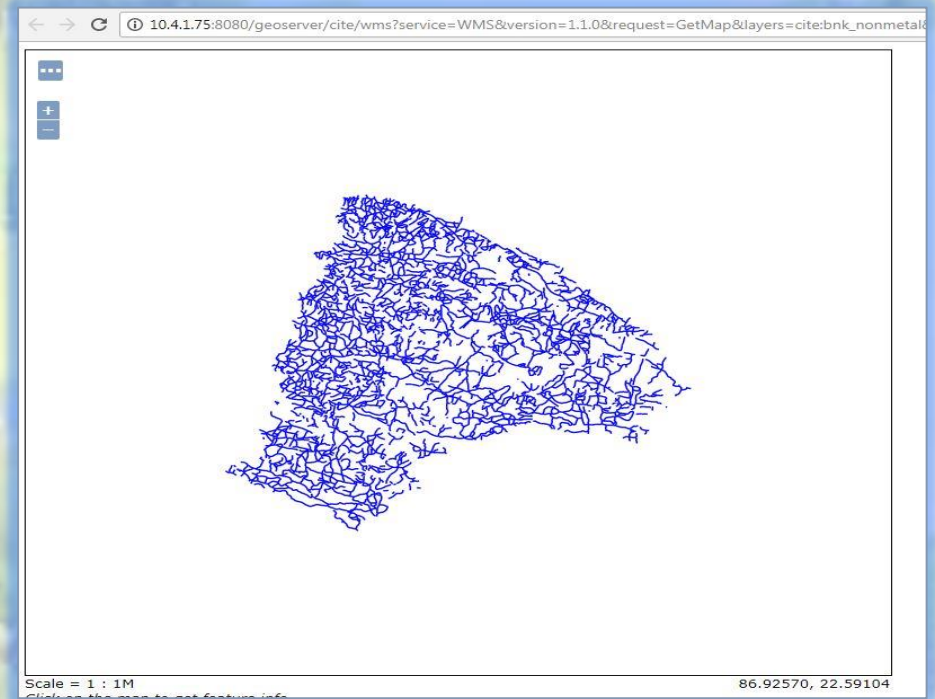
```
classDiagram
    class BNK_METAL_ROAD {
        <<column>>
        FNODE_: NUMBER(19)
        TNODE_: NUMBER(19)
        LPOLY_: NUMBER(19)
        RPOLY_: NUMBER(19)
        LENGTH_: NUMBER(33,31)
        ROAD_: NUMBER(19)
        *PK ROAD_ID: NUMBER(19)
        ROADS_TYPE: VARCHAR2(254)
        GEOM: SDO_GEOMETRY
        <<PK>>
        + PK_BNK_METAL_ROAD(NUMBER)
    }
    class BNK_NONMETAL_ROAD {
        <<column>>
        FNODE_: NUMBER(19)
        TNODE_: NUMBER(19)
        LPOLY_: NUMBER(19)
        RPOLY_: NUMBER(19)
        LENGTH_: NUMBER(33,31)
        ROAD_: NUMBER(19)
        *PK ROAD_ID: NUMBER(19)
        ROADS_TYPE: VARCHAR2(254)
        GEOM: SDO_GEOMETRY
        <<PK>>
        + PK_BNK_NONMETAL_ROAD(NUMBER)
    }
```

The left sidebar shows the Project Browser with a tree view: Class1 > Model > Class Model > Class Model > System > Frameworks > Class1 > Class2 > «table» BNK_METAL_ROAD > «table» BNK_NONMETAL_ROAD. The bottom left shows the Properties panel with General Settings.

Service Integration for Query in Cloud

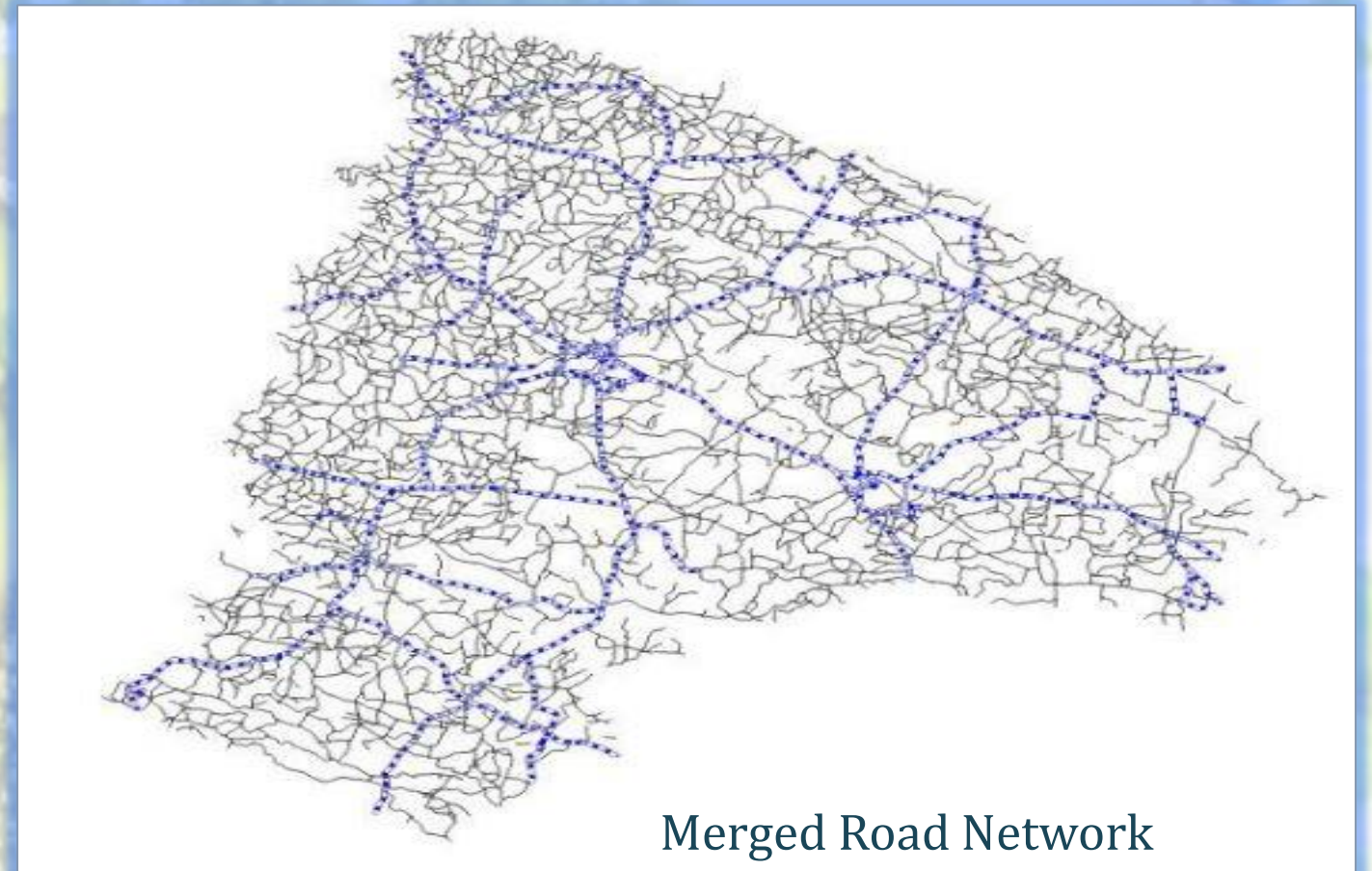
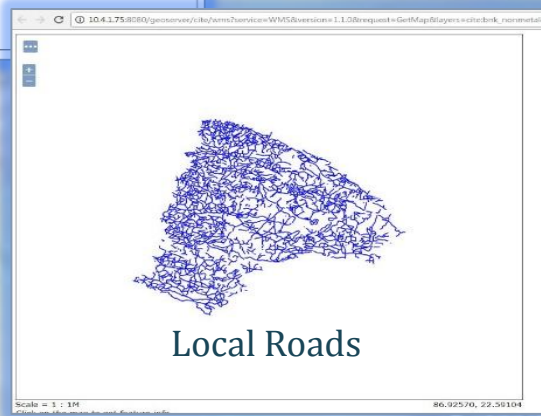
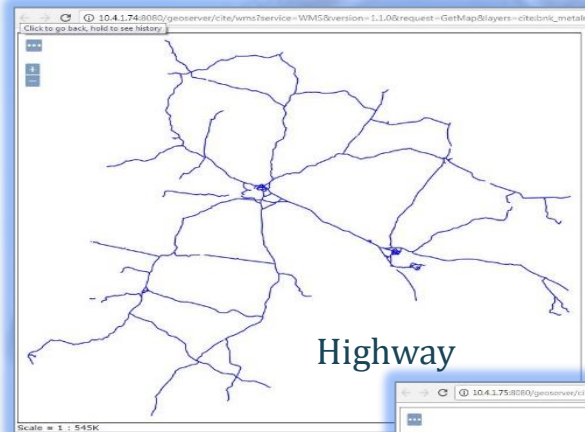


Highway

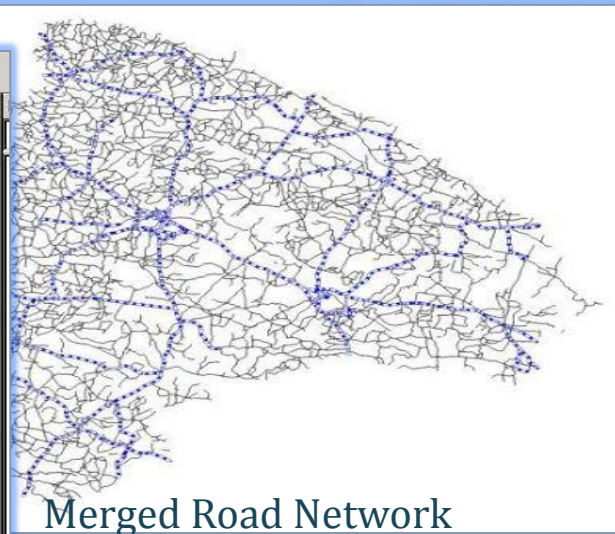
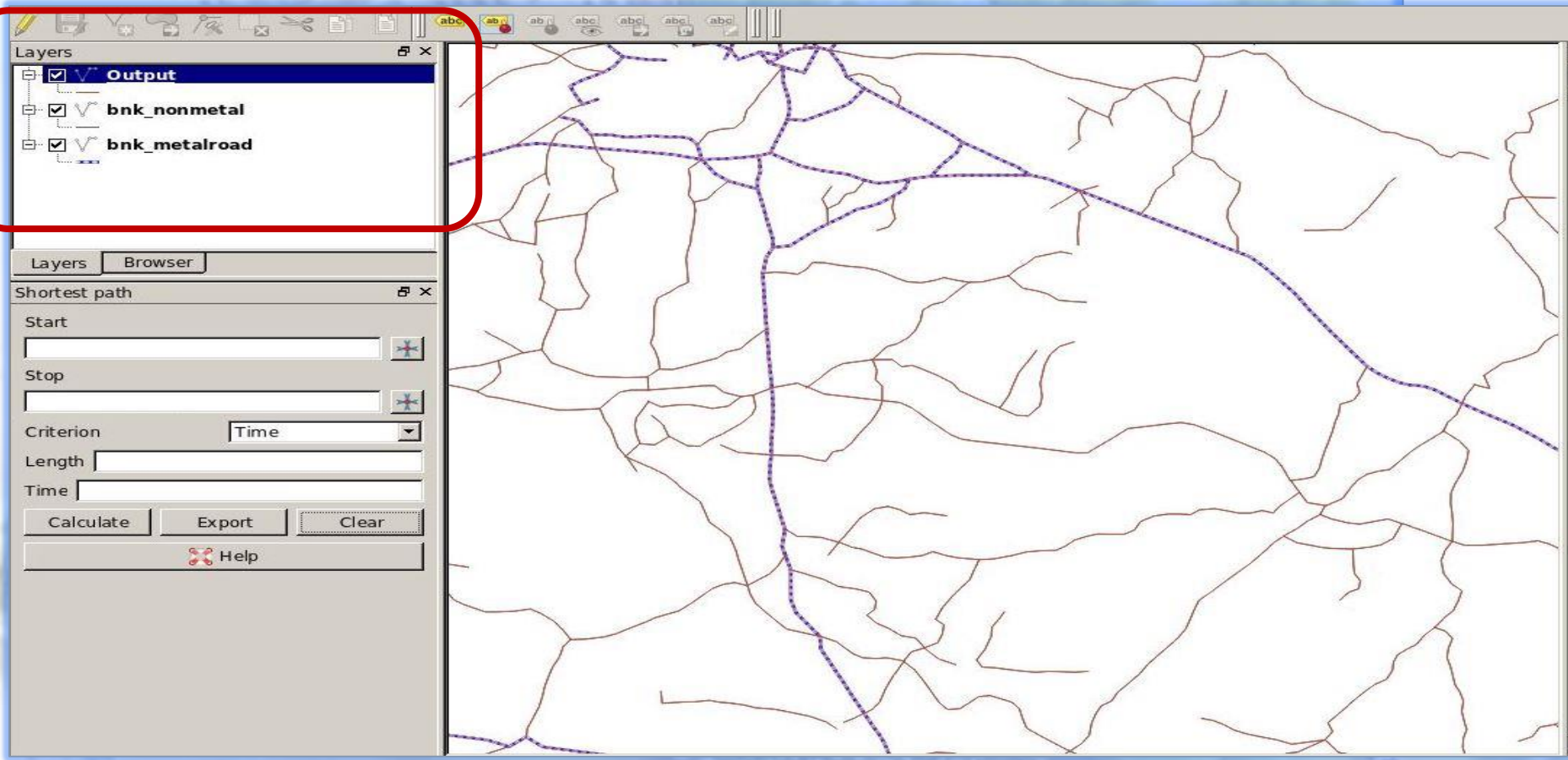


Local Roads

Service Integration for Query in Cloud

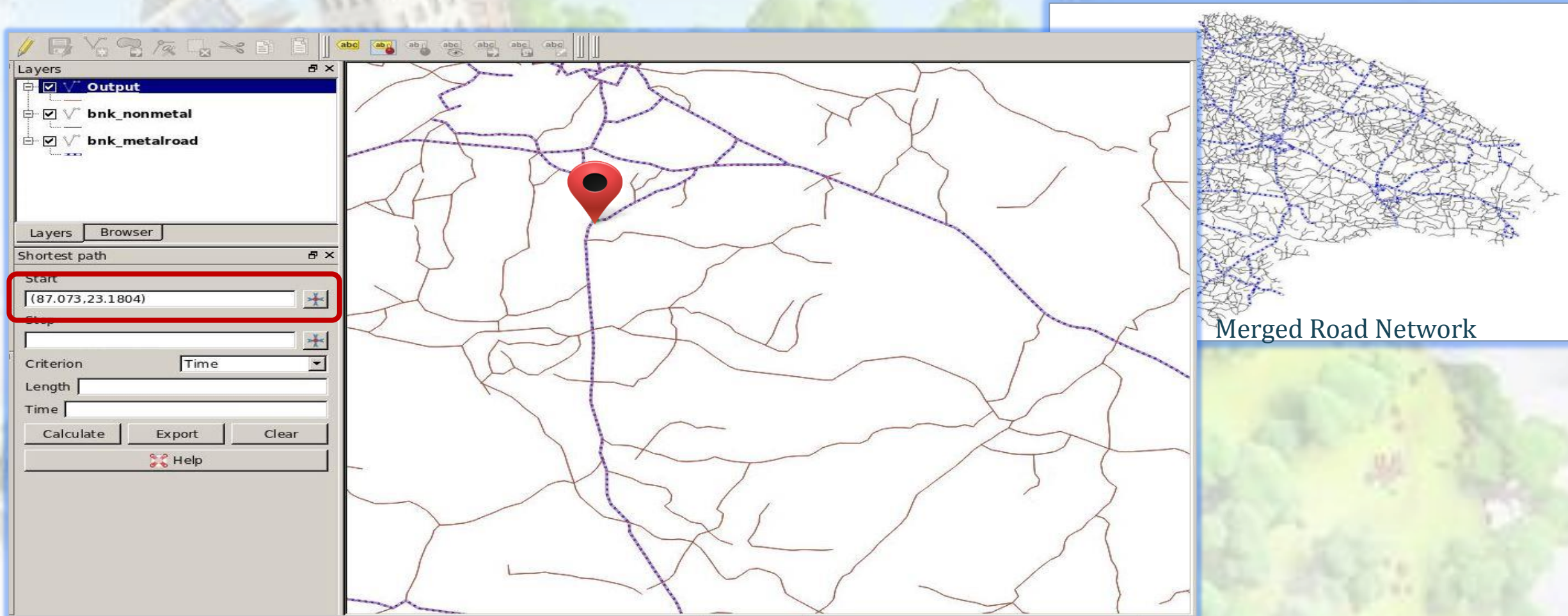


Service Integration for Query in Cloud



Shortest Path Calculation

Service Integration for Query in Cloud

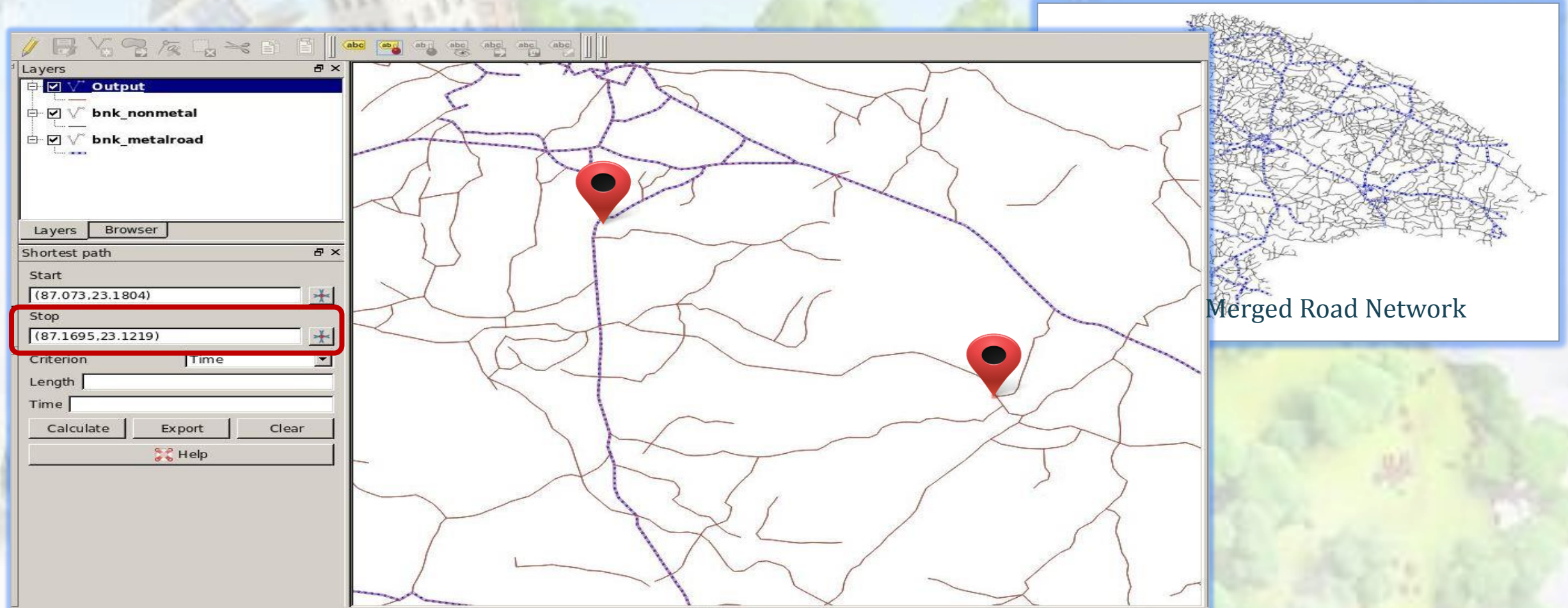


The screenshot displays a GIS application window with a map of a road network. A red location pin is placed on the map. The 'Shortest path' panel is highlighted with a red box, showing the start coordinates (87.073, 23.1804). The 'Layers' panel on the left shows three layers: 'Output', 'bnk_nonmetal', and 'bnk_metalroad'. The 'Shortest path' panel includes a 'Start' field with the coordinates (87.073, 23.1804), a 'Stop' field, a 'Criterion' dropdown set to 'Time', and 'Length' and 'Time' input fields. There are 'Calculate', 'Export', and 'Clear' buttons, along with a 'Help' icon.

Merged Road Network

Shortest Path Calculation

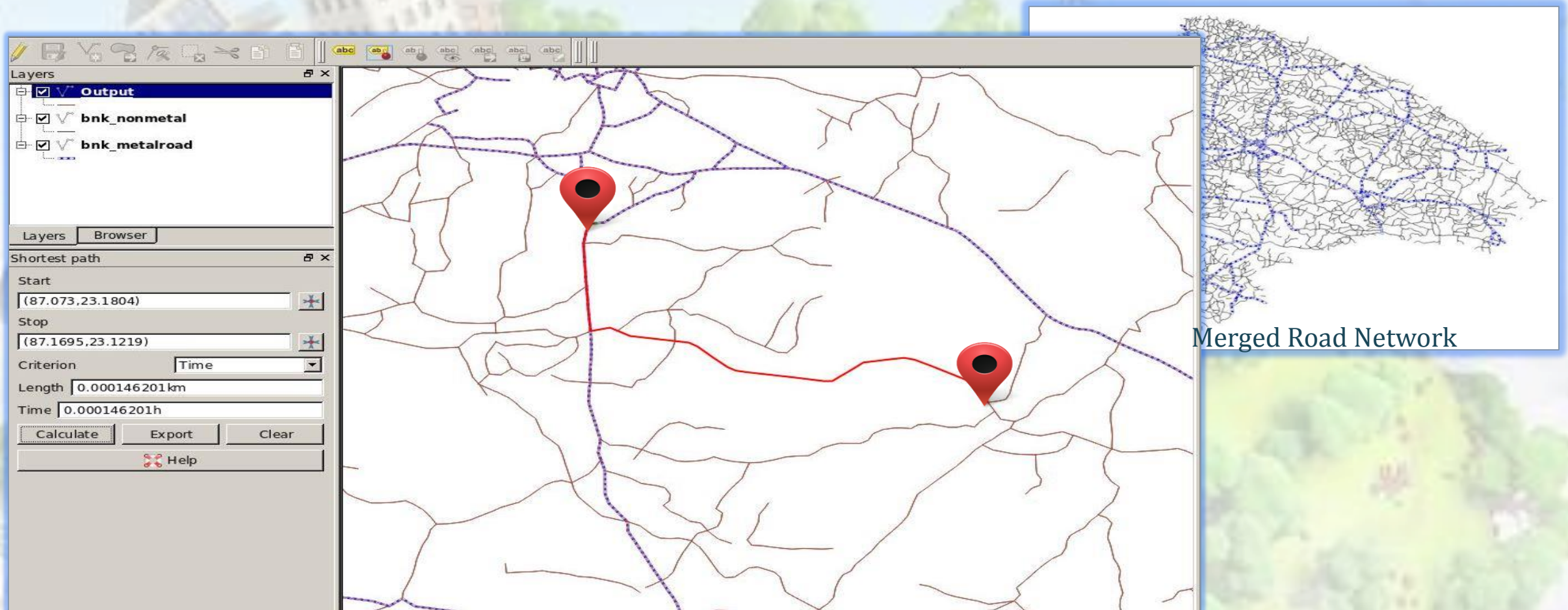
Service Integration for Query in Cloud



The screenshot displays a GIS application window with a 'Layers' panel on the left and a 'Shortest path' panel below it. The 'Layers' panel shows three layers: 'Output', 'bnk_nonmetal', and 'bnk_metalroad'. The 'Shortest path' panel has a 'Start' field with coordinates (87.073, 23.1804) and a 'Stop' field with coordinates (87.1695, 23.1219), both highlighted with a red box. The 'Criterion' is set to 'Time'. The main map area shows a road network with a purple dashed line indicating the shortest path between two red location pins. An inset window on the right shows a 'Merged Road Network' as a dense blue network of lines.

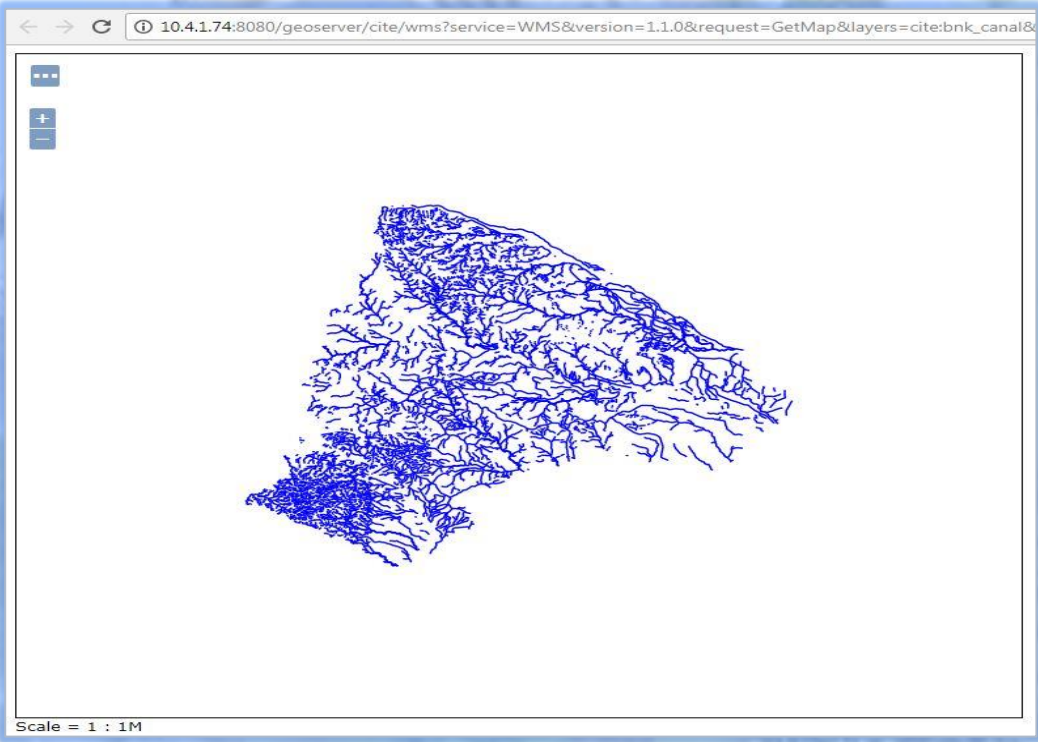
Shortest Path Calculation

Service Integration for Query in Cloud

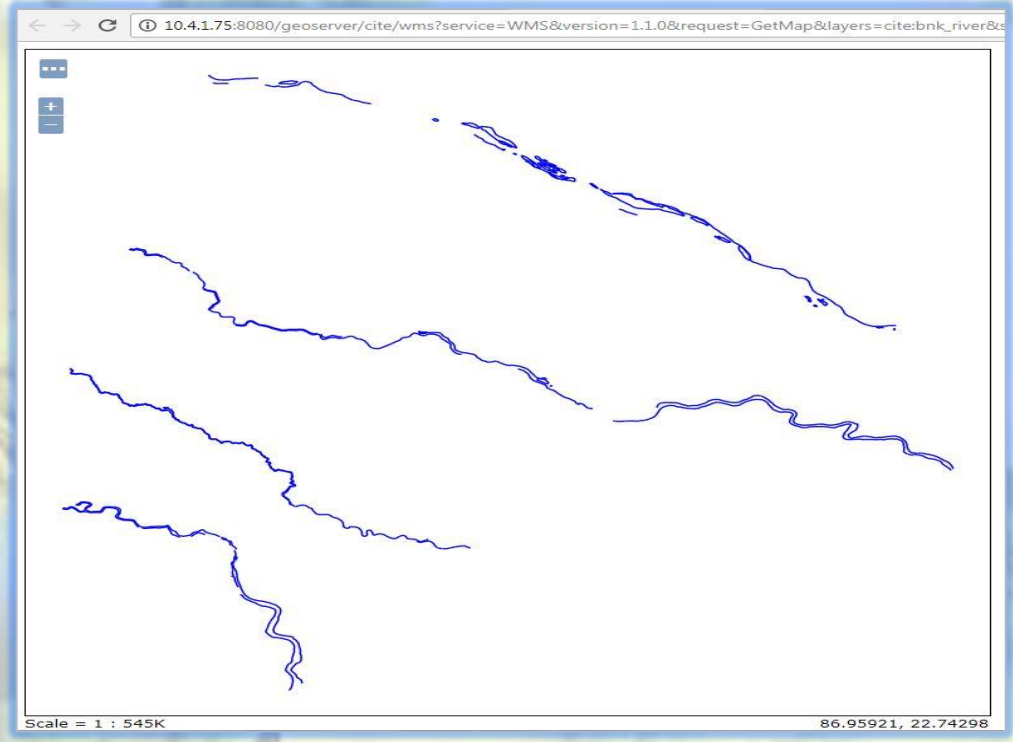


Shortest Path Calculation

Service Integration for Query in Cloud

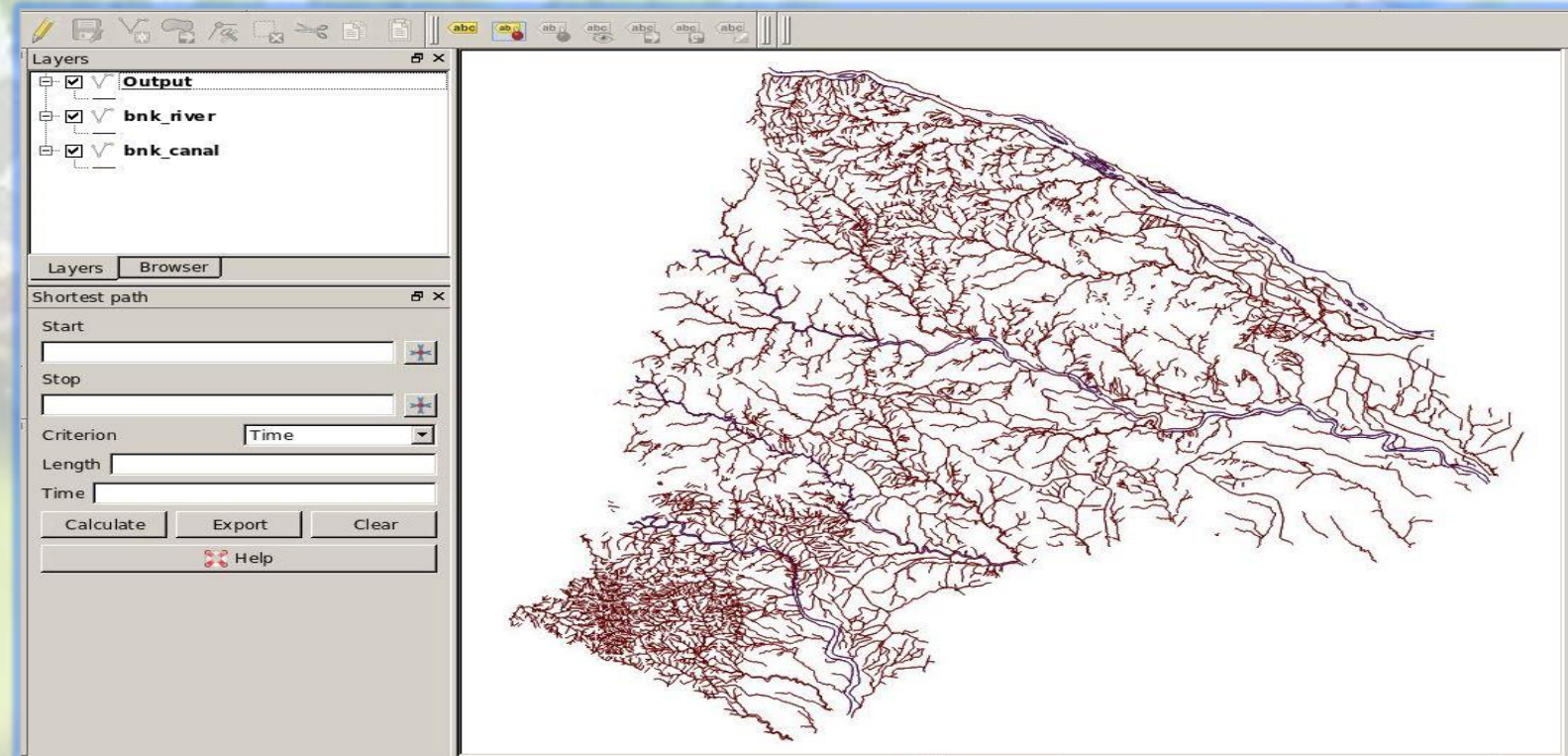
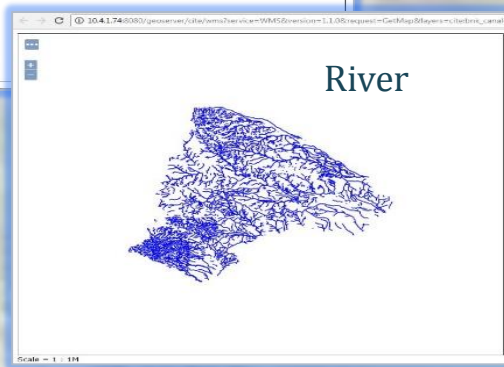
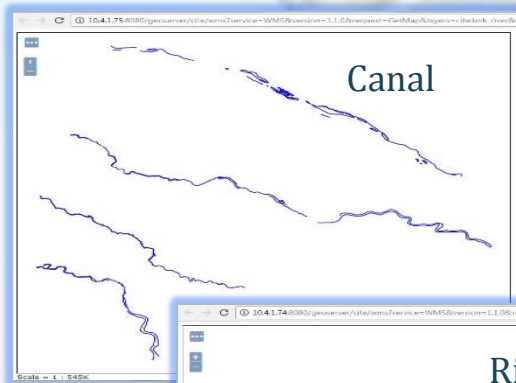


Canal



River

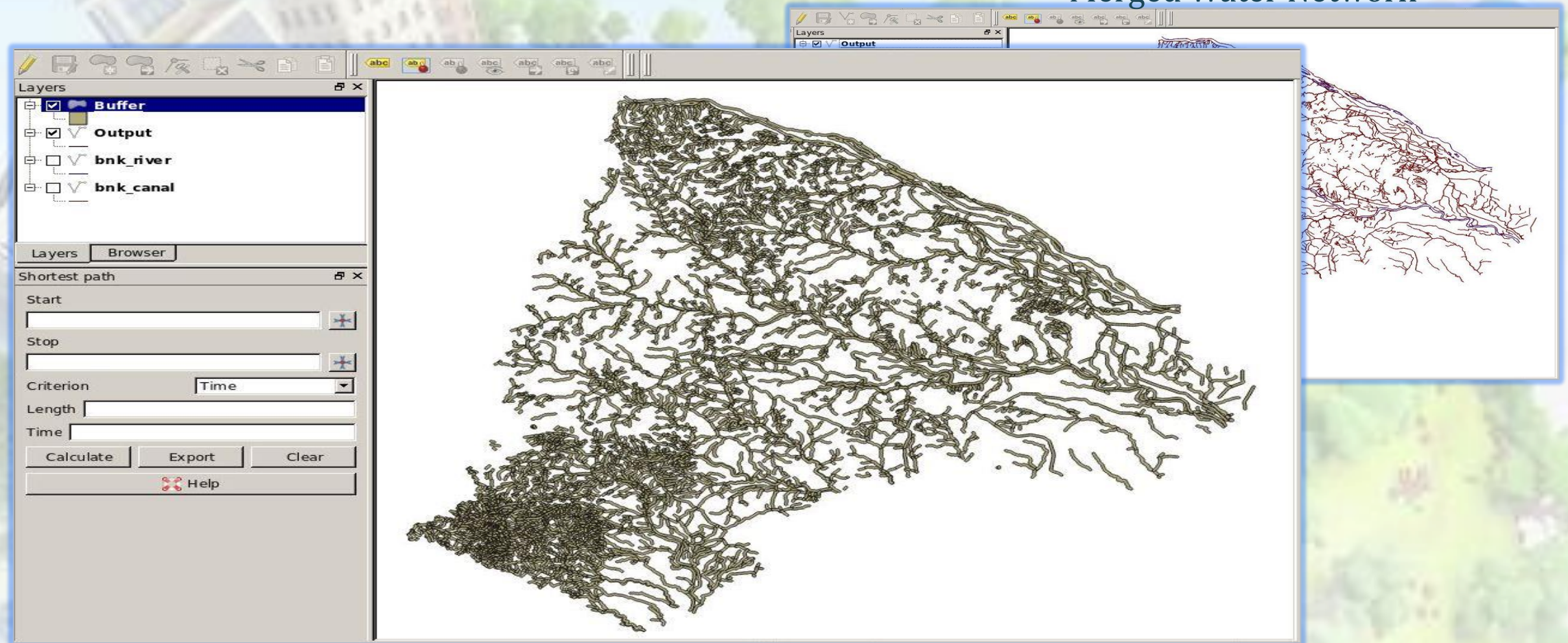
Service Integration for Query in Cloud



Merged Water Network

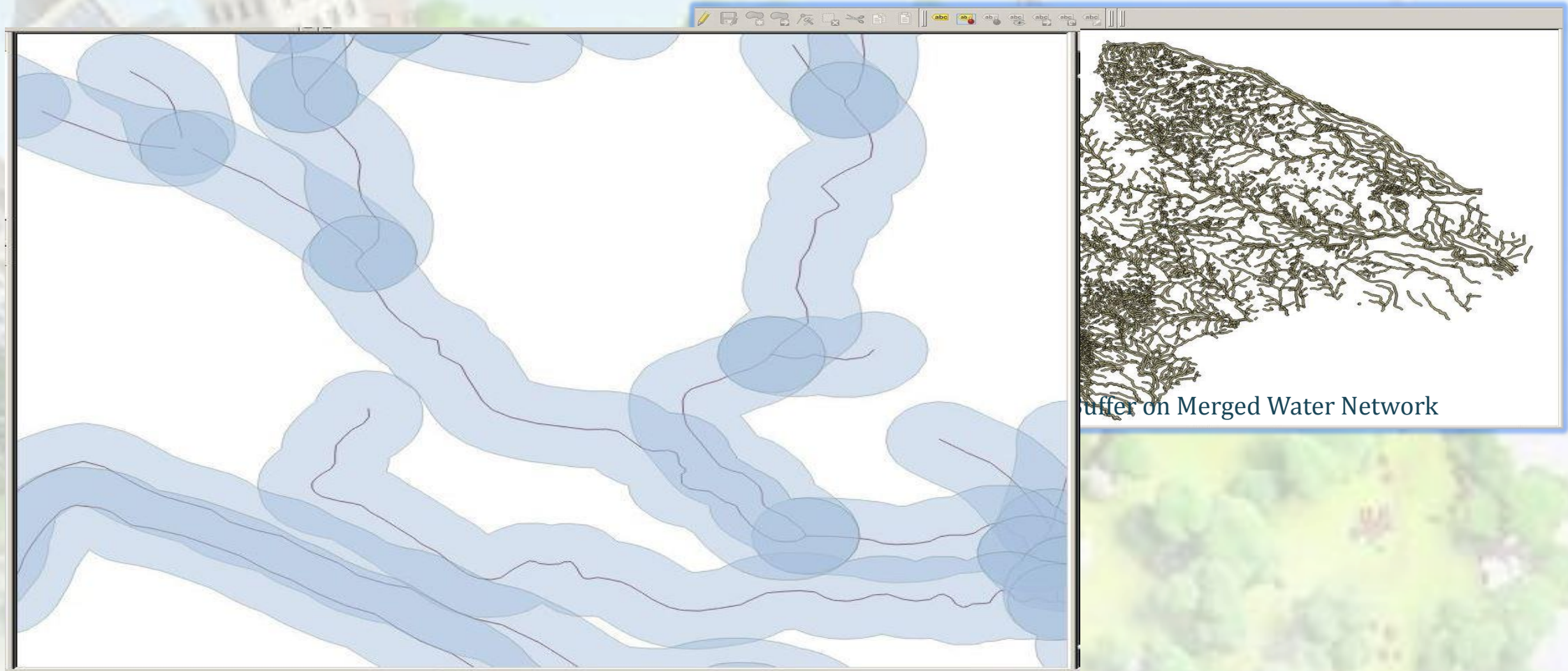
Service Integration for Query in Cloud

Merged Water Network



Buffer on Merged Water Network

Service Integration for Query in Cloud



Buffer on Merged Water Network (Zoomed)

Contact us:

Spatial Data Science Lab

sdsl.iitkgp@gmail.com

**Department of Computer Science and
Engineering**

IIT Kharagpur

Thank You!!

