M.Tech. Thesis Presentation

Geo-Service Portal

Submitted by
Deepak Punjabi - 15IT60R17

Under Guidance of
Prof. Soumya K. Ghosh

Spatial data

is data containing Information about

the locations and shapes

of geographic features

and the relationships between them,

usually stored as coordinates and topology.

Motivation

- Applications of spatial data
- Problems with currently available solution

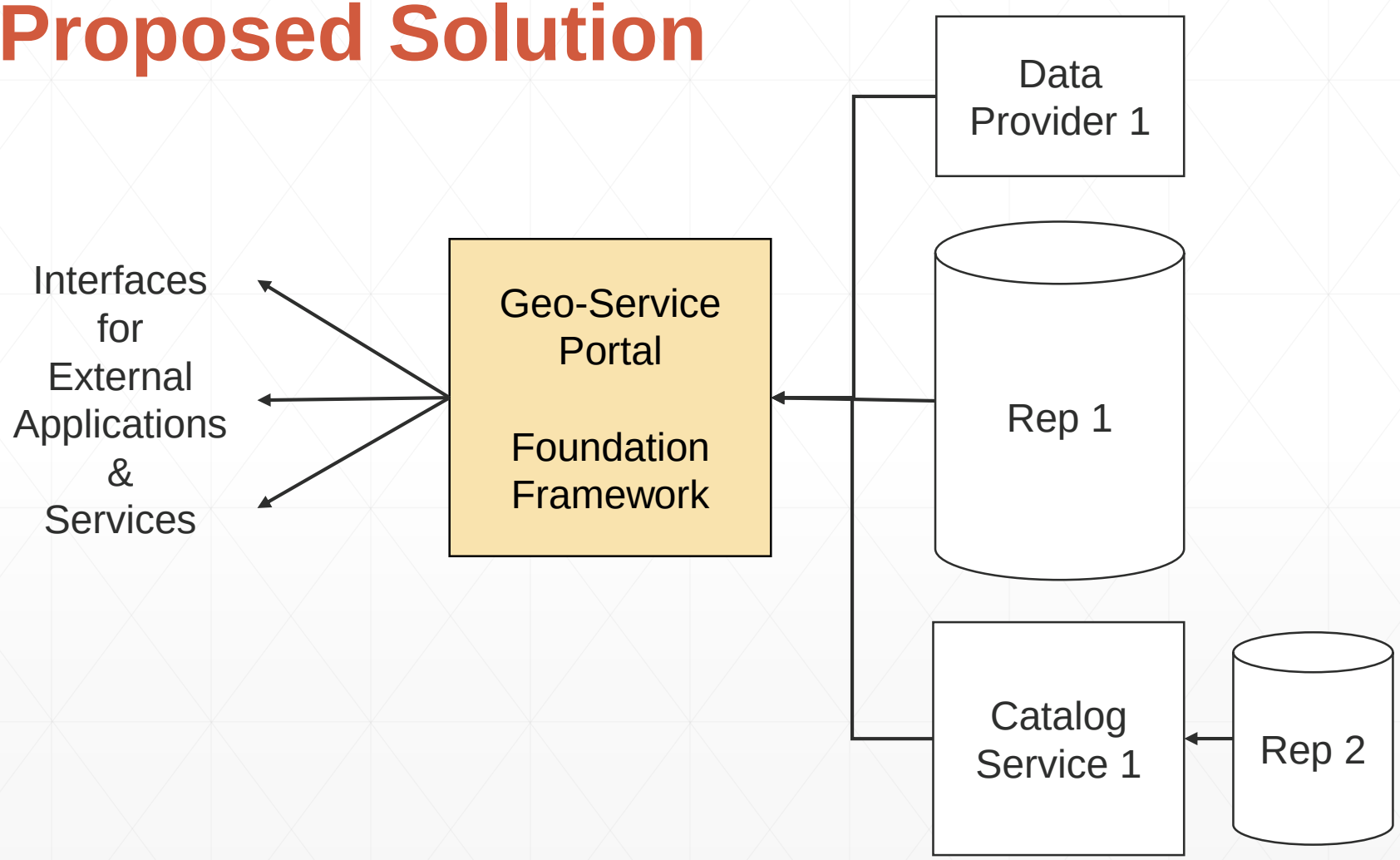
Use Cases

1. Remote Sensing
2. Area affected by flood/disease
3. Spatio-Temporal analysis
4. Spatial data mining
5. Telecom & Network Services
6. Urban Planning and Hot spot analysis
7. Navigation
8. And many more…

Current Challenges

1. General search engines are not good for searching spatial data.
2. Spatial data contains complex data types & operations.
3. Not all spatial data is publicly available.

Proposed Solution



Problem Statement

- Defining the problem
- Defining the solution objectives

Aim

Build a catalog service for web
to crawl, store, maintain, and publish
metadata information about spatial data
and it's providers and to utilize this information
to perform efficient query orchestration.

Objectives

1. Build a topical crawler to crawl the web and store geo-spatial metadata.
2. Build an OGC compliant catalog service to publish and search accumulated metadata.
3. Build Query orchestration service to perform real-time query with heterogeneous data sources and cost matrices associated with them.

Solution Model

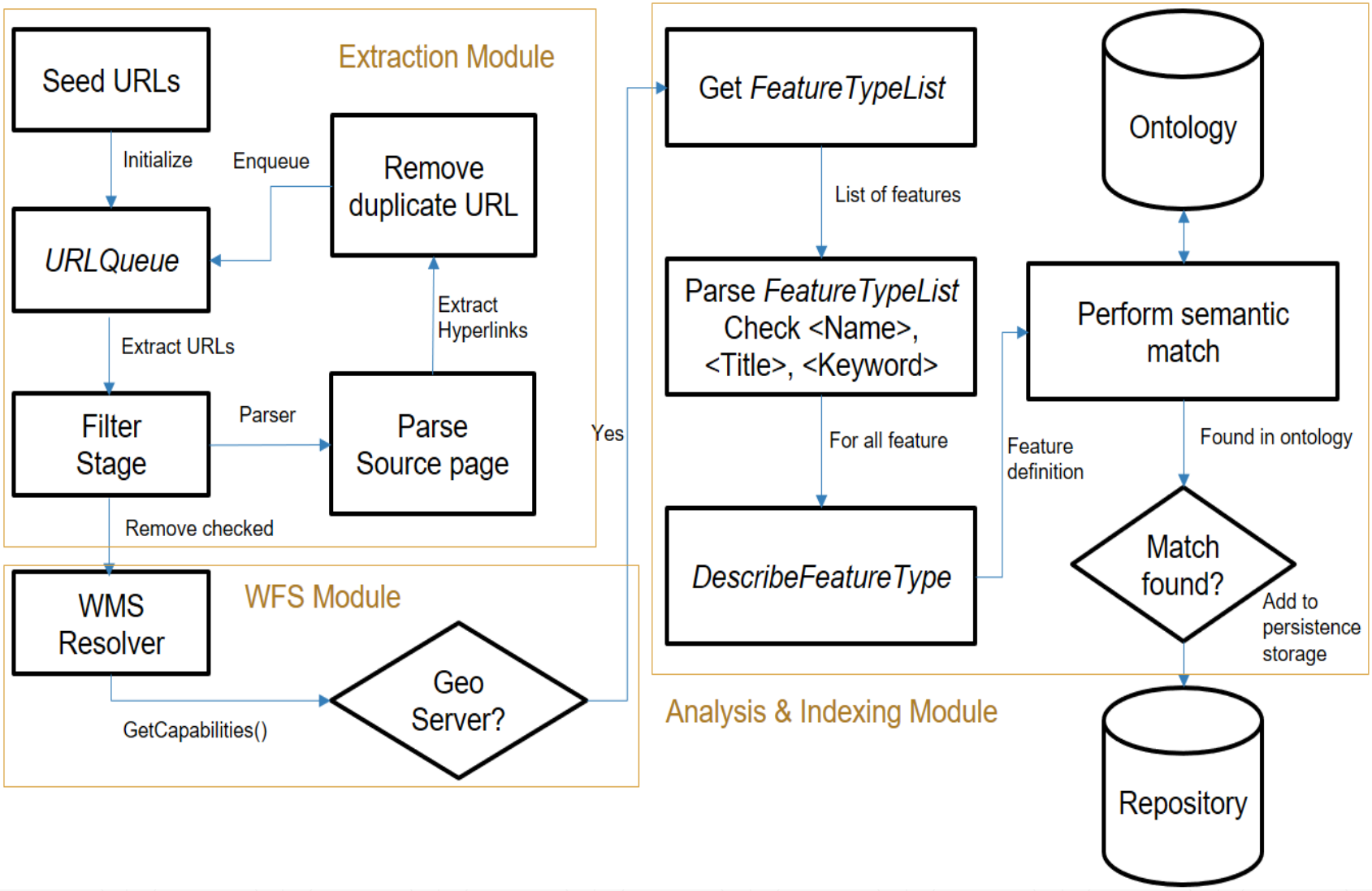
- Spatial Web Crawler
- Catalog Service
- Query Processing

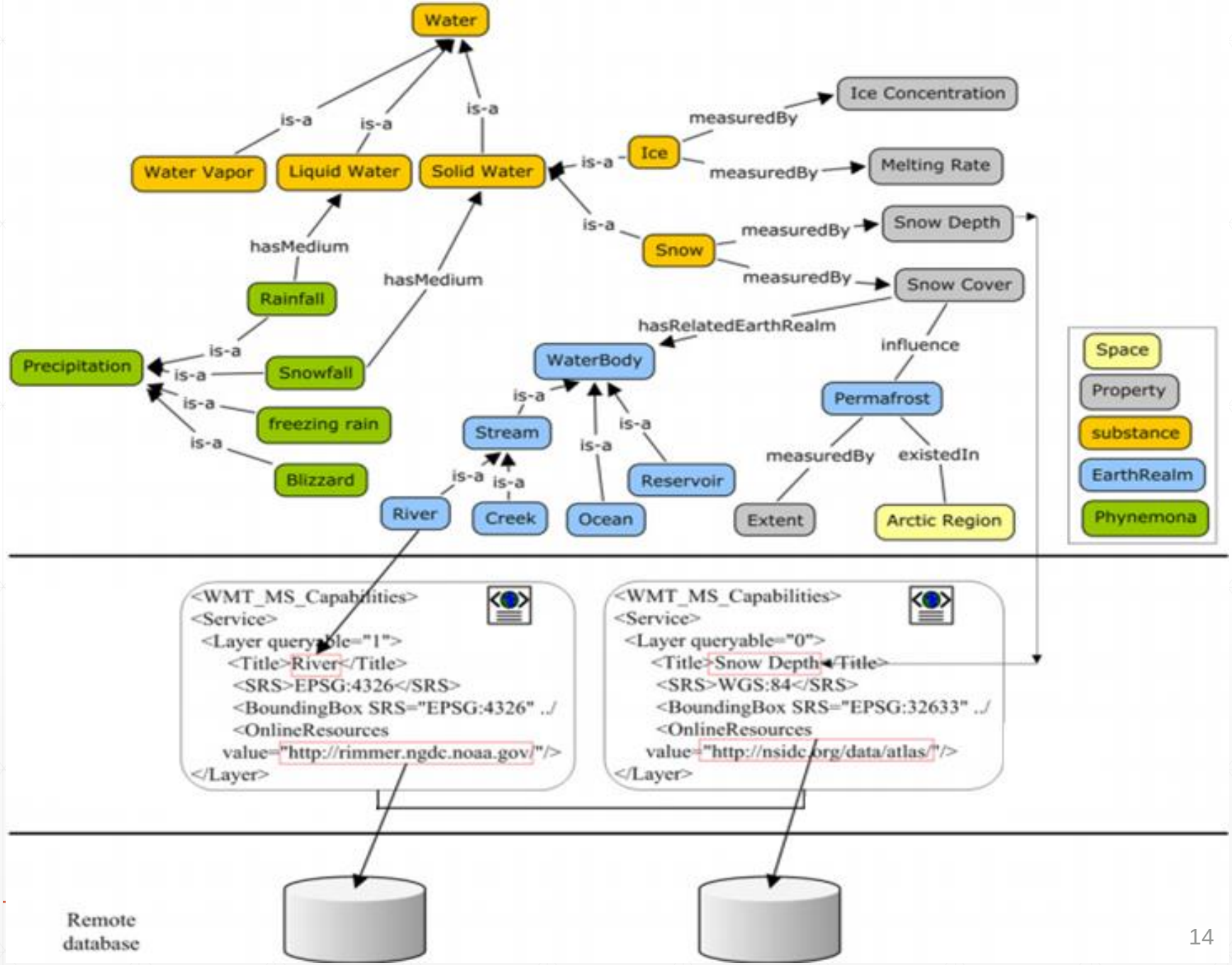
3 stage approach

Spatial web crawler: Objectives

1. Build a spatial web crawler which crawlers through geo-servers which offers WFS based OGC compliant services.
2. Build a domain specific vocabulary(ontology) for this features which can be helpful to compare found features with wanted features.
3. Perform semantic matching of found features from crawled web-pages with given ontology for filtering the correct features and storing them in the permanent repository.
4. Perform an evaluation of the given spatial web crawler using metrics and test URL seed sets.

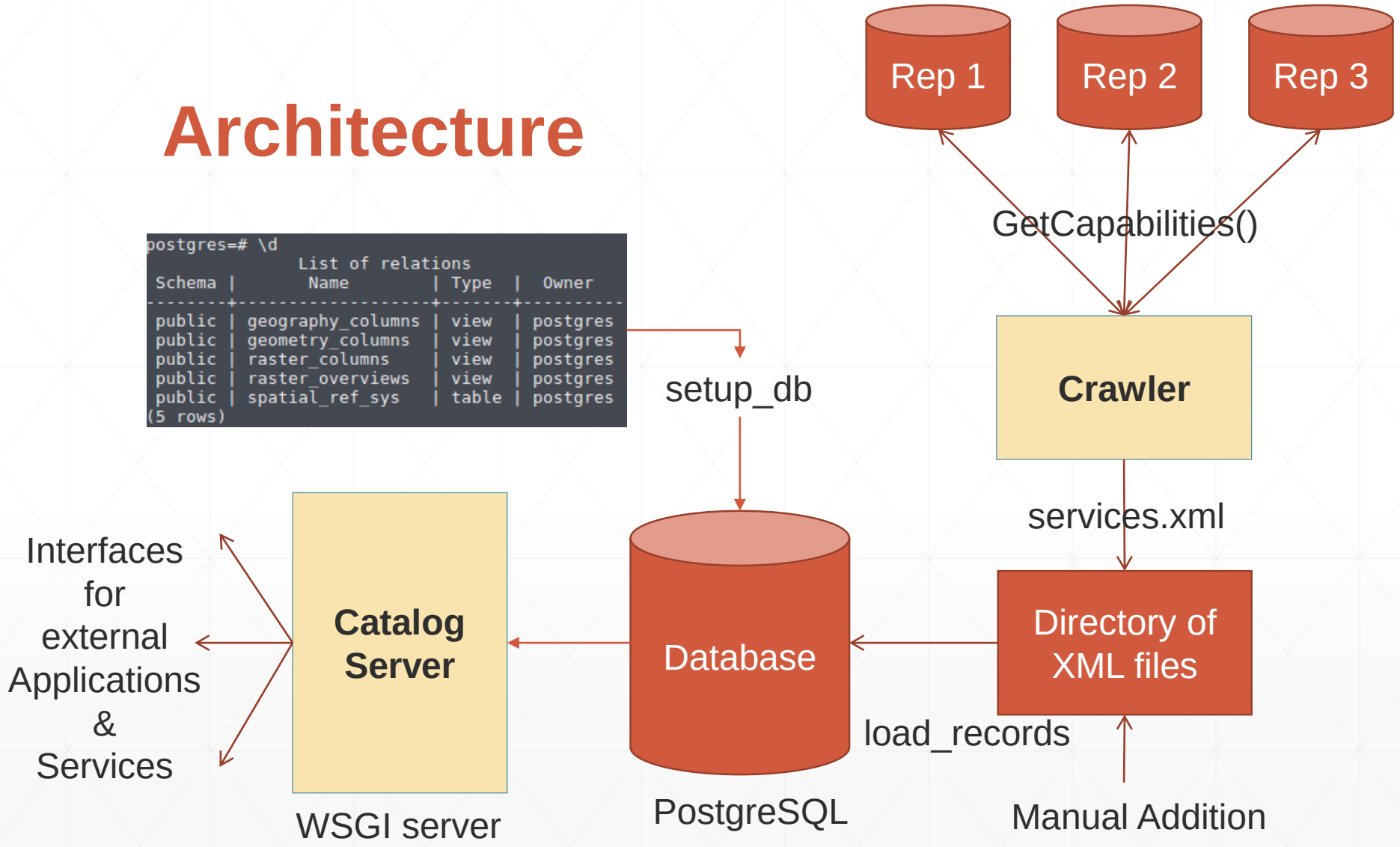
Spatial web crawler: Architecture





source: Li, W., *et al*. Computers & Geosciences,2011.

Architecture



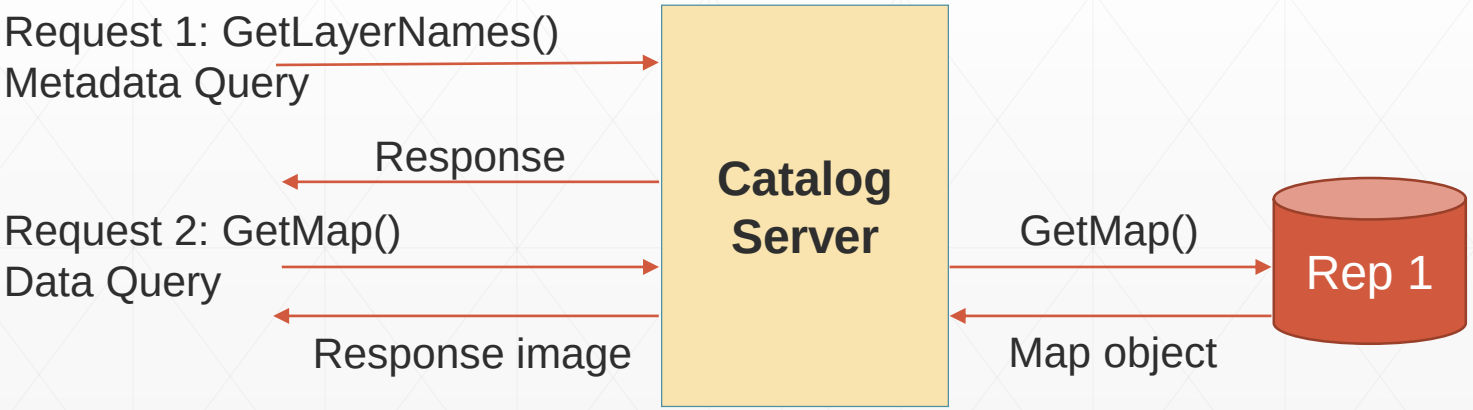
Database Setup

- First extend the database for Geo-spatial operations and data structure support. < *create extension postgis ;* >
- Setup tables and their schema. < *setup_db* >
- Services xml files are used to populate the database. Import data from xml files using < *load_records* > command.

```
postgres=# \d
                List of relations
 Schema |       Name        | Type  |  Owner
--------+-------------------+-------+----------
 public | geography_columns | view  | postgres
 public | geometry_columns  | view  | postgres
 public | raster_columns    | view  | postgres
 public | raster_overviews  | view  | postgres
 public | spatial_ref_sys   | table | postgres
(5 rows)
```

Query Processor

- Database holds all metadata information about the data available from repositories.
- Type of queries
  - Metadata Query
  - Request for the data object



Results

List of available Layers:

- kgp:POPULATION
- kgp:bnk block boundary
- kgp:bnk block hq
- kgp:bnk district boundary
- kgp:bnk drainage
- kgp:bnk grampanchayat boundary
- kgp:bnk mouza boundary
- kgp:bnk road

List of available Operations

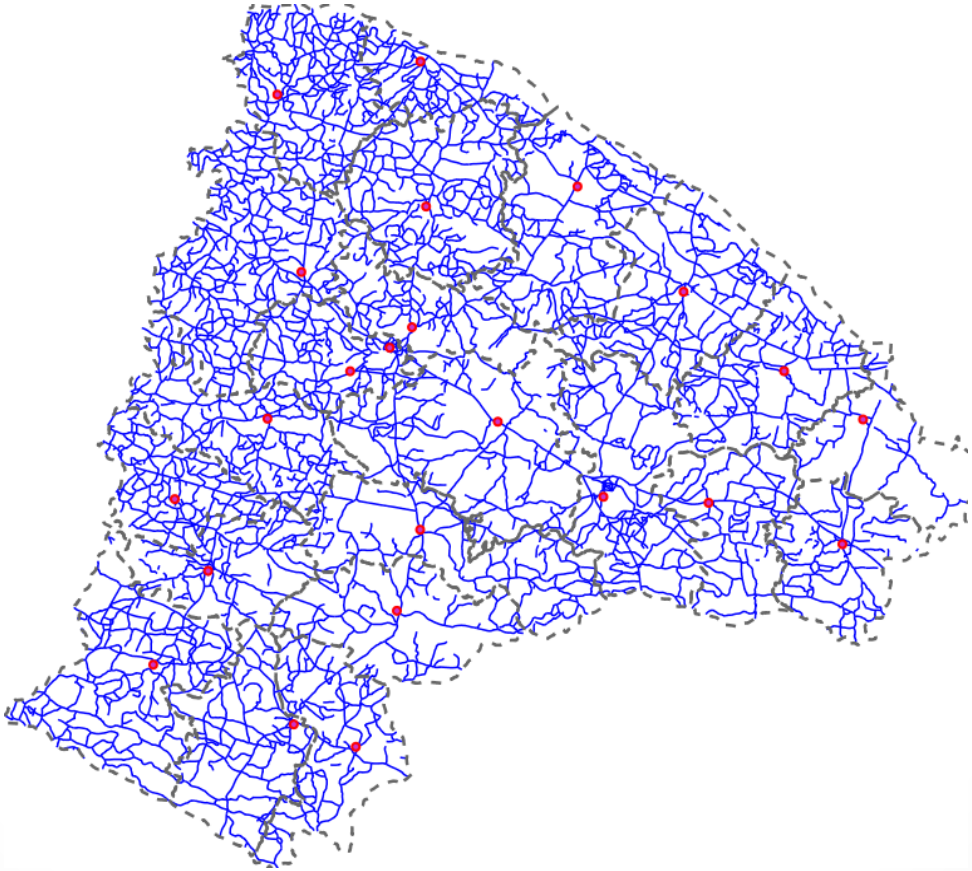
- GetCapabilities
- GetMap
- GetFeatureInfo
- DescribeLayer
- GetLegendGraphic
- GetStyles

GetMap

GetMap returns a map image of the layer(s) in available formats.

Options:

- Layers=kgp:bnk_road
- Width=768
- Height=679
- Format=image/png



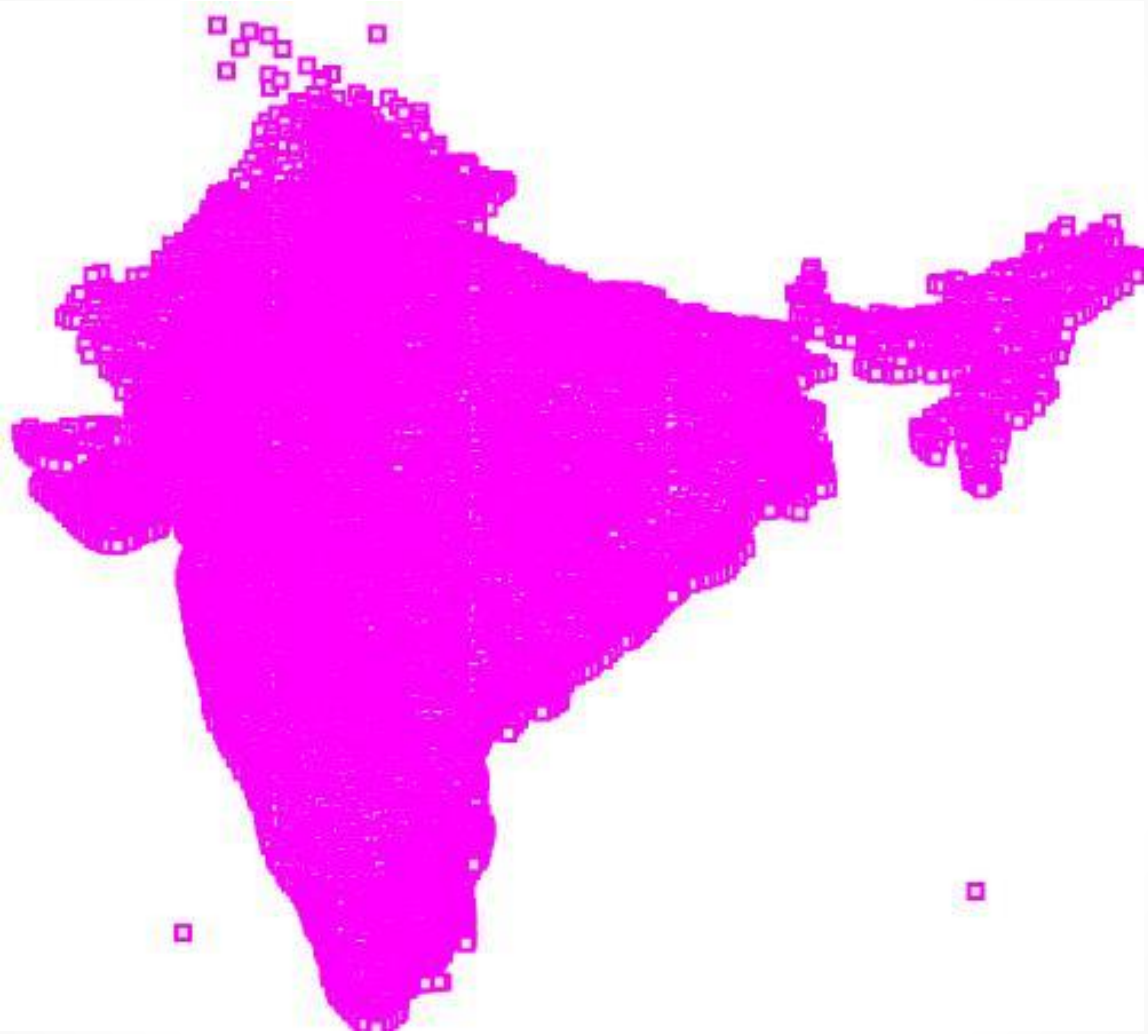
DrawMap

DrawMap Overlays different map images on top of each other.

Useful to find affected area.

Options:

- Layers = { kgp:bnk_road, kgp:bnk_block_hq, kgp:bnk_block_boundary }
- Width = 768
- Height = 679
- Format = image/png



* This image shows population density in India, without any information on boundaries.

Information about specific layer *

- Title | POPULATION
- Name | kgp:POPULATION
- Is Queryable | 1
- Is Opaque | 0
- Bounding Box |
- minx | 68.52669525146484
- miny | 8.086045265197754
- maxx | 97.3387680053711
- maxy | 35.8697509765625

Conclusion

- Geo-service portal acts as a underlying framework or foundation for various kind of higher level use cases.
- Building an OGC compliant web service catalog can also be beneficiary as already available software and services can use the registry for various kinds of services with little to no modification of their original code-base.

Future Work

- Build a cloud based implementation for the spatial web crawler, catalog service and query processing.
- Build interfaces and implementation for more complex queries.
- Provide parallel query processing for same data occurring in multiple repositories.
- Implement model for ranked retrieval.

References

- Patil, Sonal, Shrutilipi Bhattacharjee, and Soumya K. Ghosh. ***A spatial web crawler for discovering geo-servers and semantic referencing with spatial features.*** International Conference on Distributed Computing and Internet Technology. Springer International Publishing, 2014.
- Li, Wenwen, Chaowei Yang, and Chongjun Yang. ***An active crawler for discovering geospatial web services and their distribution pattern. A case study of OGC Web Map Service.*** International Journal of Geographical Information Science 24.8 (2010): 1127-1147.
- Najork, Marc. ***Web crawler architecture.*** Encyclopedia of Database Systems. Springer US, 2009. 3462-3465.
- Ahlers, Dirk, and Susanne Boll. ***Location-based Web search.*** The Geospatial Web. Springer London, 2009. 55-66.
- Li, W., et al. ***Semantic-based web service discovery and chaining for building an Arctic spatial data infrastructure.*** Computers & Geosciences 37.11 (2011): 1752-1762

References (continue)

- Jiang, Jun, Chong-jun Yang, and Ying-chao Ren. ***A Spatial Information Crawler for OpenGIS WFS***. Sixth International Conference on Advanced Optical Materials and Devices. International Society for Optics and Photonics, 2008.
- ***http://geopython.github.io/pycsw-workshop/***
- ***https://geopython.github.io/OWSLib/***

Thank You