

# MTP Progress Report

Deepak Punjabi  
15IT60R17

## Problem Statement:

### **Geo-service Portal**

The objective of this project is to build a catalog service for web to crawl, store and publish metadata information for spatial repositories to perform query orchestration.

## Motivation:

The information about spatial data can be found to be much useful in many fields of industry. However this data is not generally publicly easy to found. The methods to access these data are also heterogeneous and many times require licensing and using proprietary technologies. A central catalog service can be built to keep track of this various repositories, also the data can be made available through simple unified calling mechanisms. With the availability of large amount of heterogeneous data from different multiple sources one can do many type of spatio temporal analysis.

## Objectives:

**Phase I** - Build a topical crawler to crawl the web and store geo-servers found and their corresponding metadata.

**Phase II** - Build a OGC compliant catalog service to publish and search accumulated metadata.

**Phase III** - Build Query orchestration service to perform realtime query with heterogeneous data sources and cost matrices associated with them.

**Phase IV(Extended)** - Build a cloud based implementation of catalog service to easily access and manipulate the registry.

## Work Done:

Phase I: Built a sample topical crawler that can verify for a given url that if it offers OGC web service or not. This can be done in loop for verifying subsequent url's if they offer geoservices or not. A better approach would be to built level based crawler that crawls till some certain layers.

Phase II: Build a sample geoserver parser to extract all the information available in the geoserver. Tested various models to implement web services.

Phase III: Can perform static queries, like describing layers of a geoserver, extracting other metadata and presenting. Presenting a map based on the information available.

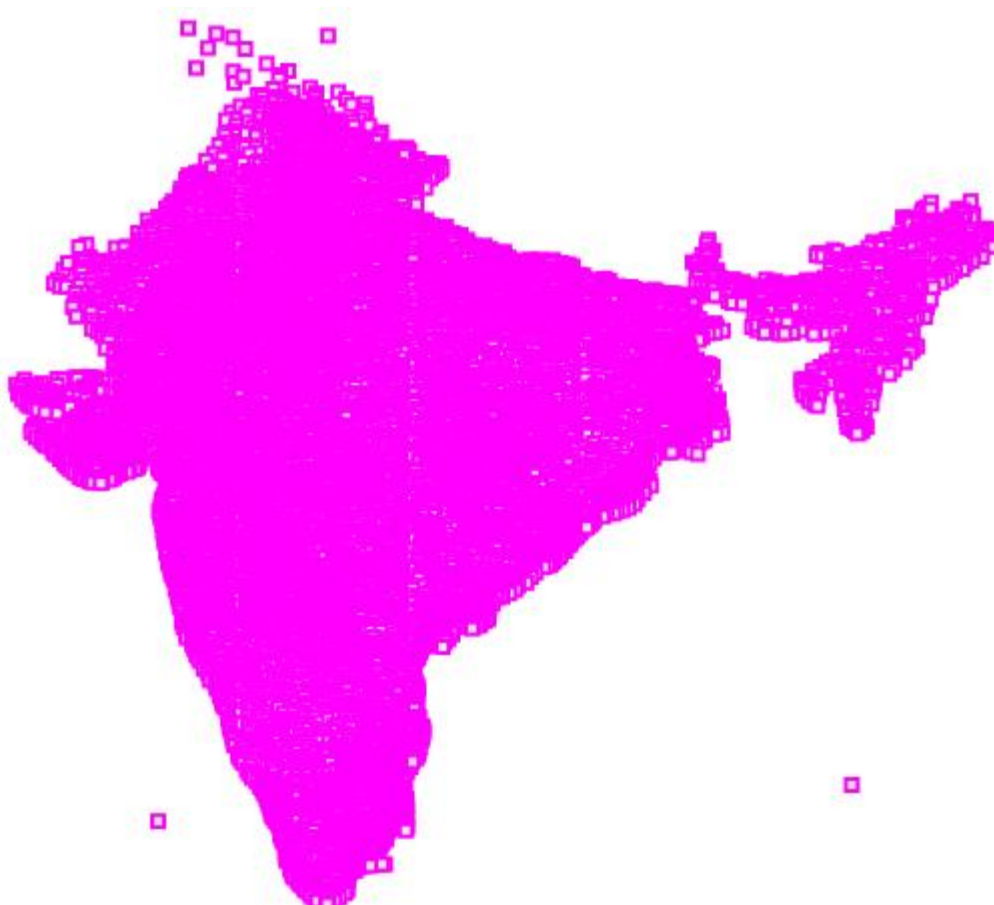
## Results :

I am attaching log file for the results at the end of this document based on the work done so far.

## Discussion:

- Standardize the schema for storing geospatial information in the database so that it can be efficiently retrieved and shared.
- Integrate different solutions to build a system.

## Logs:



/usr/bin/python3.5 /home/niku/PycharmProjects/mtp/sample\_reg.py

**Service Metadata** ---> OGC:WMS

**Title** ---> GeoServer Web Map Service

**List of available layers** --->

kgp:POPULATION

kgp:bnk\_block\_boundary

kgp:bnk\_block\_hq

kgp:bnk\_district\_boundary

kgp:bnk\_drainage

kgp:bnk\_gram-panchayat\_boundary

kgp:bnk\_mouza\_boundary

kgp:bnk\_road

**Available operations** --->

GetCapabilities

GetMap

GetFeatureInfo

DescribeLayer

GetLegendGraphic

GetStyles

**Format Options** --->

image/png

application/atom+xml

application/atom+xml

application/openlayers

application/pdf

application/rss+xml

application/rss+xml

application/vnd.google-earth.kml

application/vnd.google-earth.kml+xml

application/vnd.google-earth.kml+xml

application/vnd.google-earth.kml+xml;mode=networklink

application/vnd.google-earth.kmz

application/vnd.google-earth.kmz+xml

application/vnd.google-earth.kmz+xml

application/vnd.google-earth.kmz;mode=networklink

atom

image/geotiff

image/geotiff8

image/gif

image/gif;subtype=animated

image/jpeg

image/png8

image/png; mode=8bit  
image/svg  
image/svg+xml  
image/svg+xml  
image/tiffimage/tiff8  
kml  
kmz  
openlayers  
rss  
text/html; subtype=openlayers

----- **Details about particular layer: POPULATION** -----

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

**styles --->**  
kgp:Population