GeoSpatial Data Model – Backbone for realization of SDI?

Soumya K. Ghosh
Spatial Data Science Lab
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
skg@cse.iitkgp.ac.in

Present scenario

- Mostly all organizations/ departments are digitally enabled
- Data collected and Application developed as per organizational goal
- Use of software packages for data and application management
- States are having space application centres/ data centres (SAC/SDC) for geospatial information management
  - A facilitating centre for serving user/line departments
- User departments wants “solution of their problems”!

Why Data Models?

- Major challenge in information integration and interoperability
- Domain knowledge remain primarily with the user/line departments
- Increasing pressure of SAC/SDC to support user/line departments requirements
- Somewhat "disconnect" between SAC/SDC and user/line departments regarding data/ applications
  - Users wants solution to their problem – Not data
- Need to shift "Data-Driven" to "Service-Driven" philosophy
- Data Model is the foundation for: Interoperability, Sharing, Service realization

SDI – Major Players!

- NSDI, DST
- State SDIs
- Central Government Departments
- State Governments
- State Line Departments
- Industry
- Academic Institutes
- Citizen ?

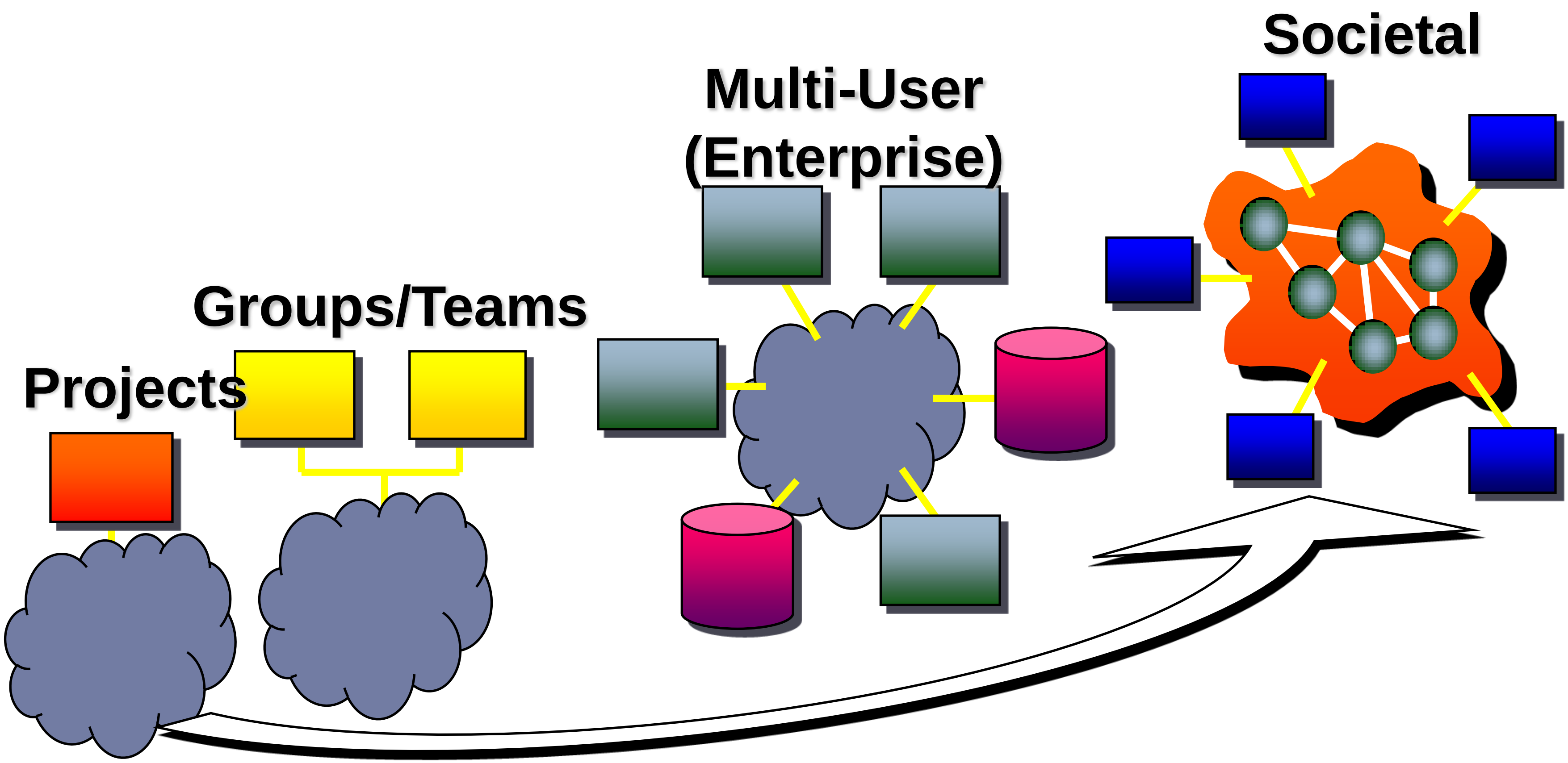
SDI “wishes” and “needs”

- A seamless “infrastructure” – Agile/ Scalable/ Upgradable/ ….
  - All “good things” in a “box” (like, Telcom services ?)
- Modeling on Domain knowledge
- Emphasis on Services, Process, Process Synchronization, Decision Support
  - This will drive Data collection/ organization
- Central Govt/ State Govt/ District Administration/ Line Departments/ Public etc. – finds value in SDI
- Common policy on data organization, processes/ applications

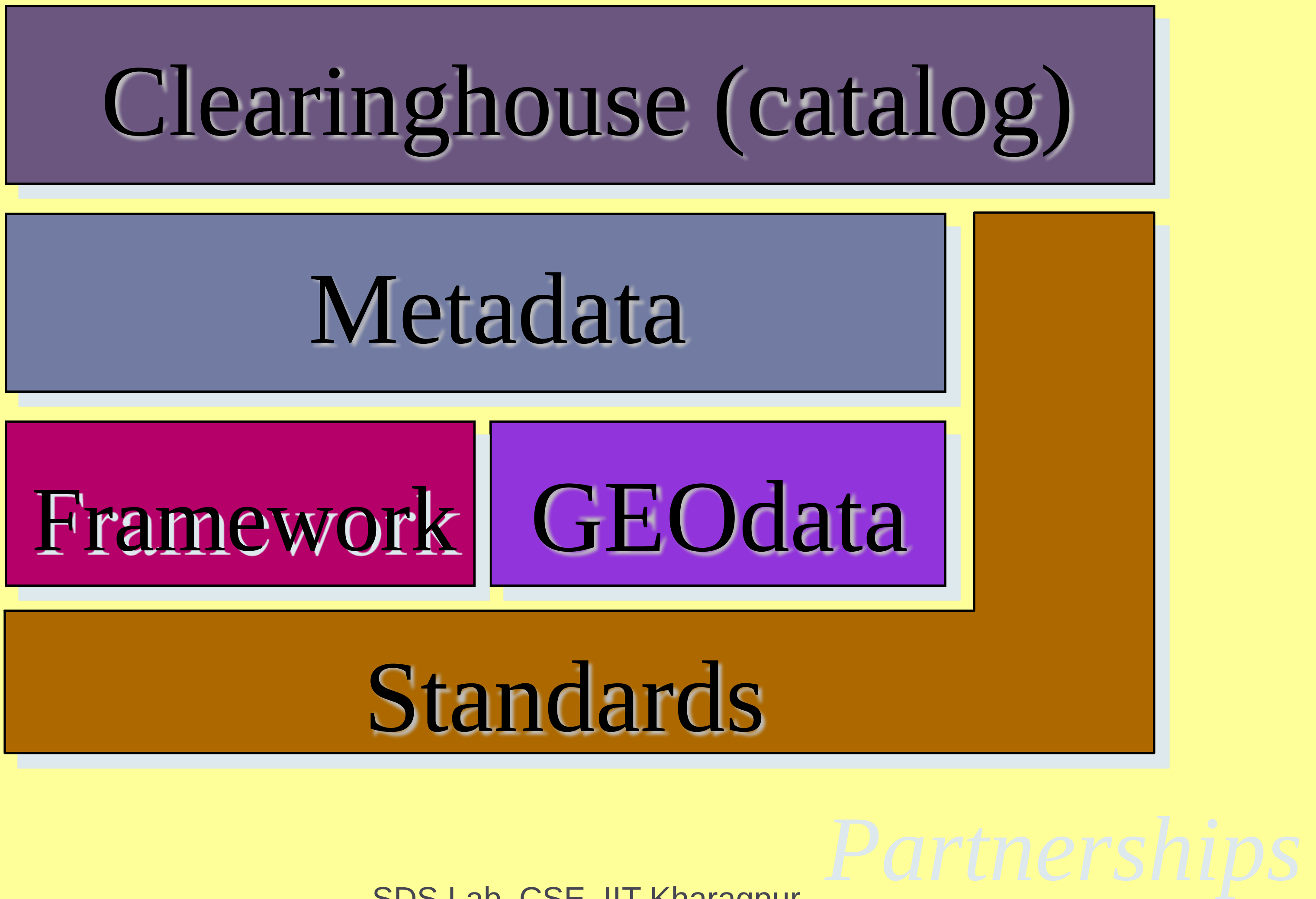
Possible Approach
(some may be already in practice)

- Model "Data", "Process", "Workflow" by the Domain Experts – involve end users.
- Start with Central Government Schemes – NREGA/ ICDS/ …
- Get rid (partially) of Hardware/ Software procurement??
- Try to "Institutionalize" the line-departments' processes (Start with a sample set)
- NSDI, DST may initiate some use-cases
- Development of Human Resources (School level? Sponsored school projects on Geoinformatics?

Geospatial Applications - Evolution



NSDI Components

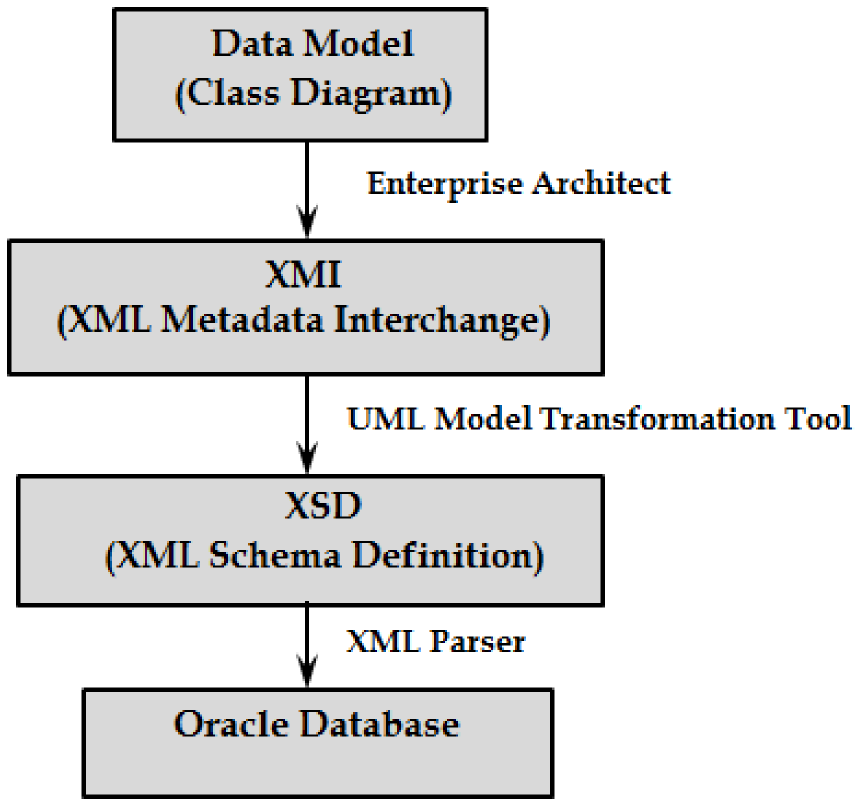


Need for more than SDI ?

- "Huge" volume of Data and Metadata
- Need of Services and Service Orchestration
- Evolving Standards and Policies
- On-demand Spatial Services
- Scaling of Services
- Need for Geospatial Cloud

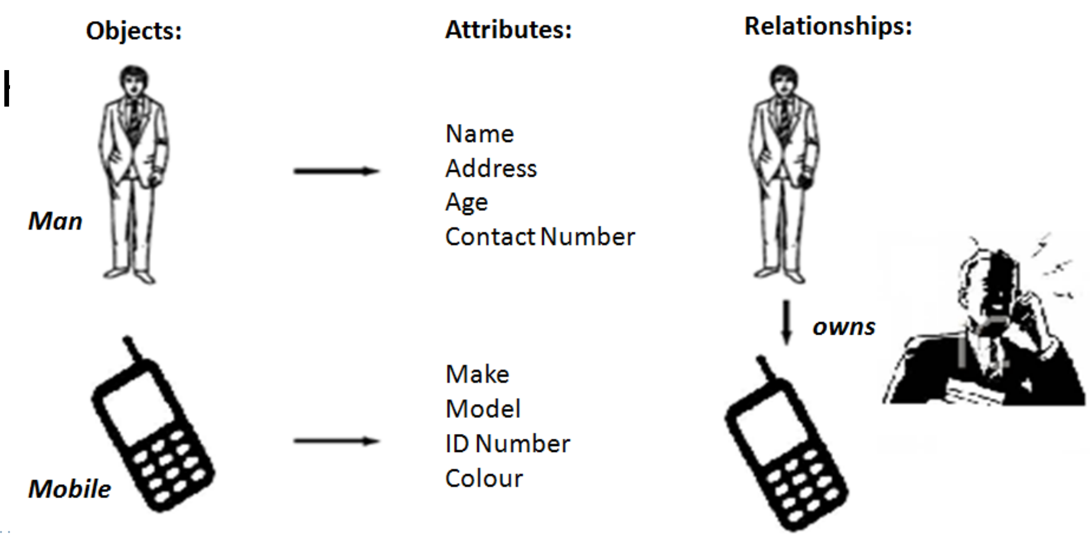
Overall Process Flow

The entire process of mapping a data model into Oracle database can be represented with the following block diagram



What is Data Model?

- *Data model* is a description of the objects represented by a computer system together with their properties and relationships; these are typically "real world" objects such as products, suppliers, customers, and orders
- Data modeling in software engineering is the process of creating a data model for an information system by applying formal modeling techniques
- The data model consists of three interrelated pieces of information: the data ct, and the relationsl



Data Model Contd.

- ***Data objects:*** A data object is a representation of almost any composite information that must be understood by software. The composite information means something that has a number of different properties or attributes

- ***Attributes:*** Attributes define the properties of a data object and take on one of three different characteristics. They can be used to,
  - name an instance of the data object
  - describe the instance
  - make reference to another instance in another table

- ***Relationships:*** Data objects are connected to one another in different ways. Consider two data objects, man and mobile. A connection is established between man and mobile because the two objects are related. Here, the relationship is '*owns*'.

Data Model Contd.

- **Cardinality:** Cardinality is the specification of the number of occurrences of one object that can be related to the number of occurrences of another object. Cardinality is usually expressed as simply 'one' or 'many'. For example, a man may own many mobiles, but mobile usually belongs to one man. Taking into consideration all combinations of 'one' and 'many', two objects can be related as,
  - ***One-to-one (1 : 1)*** *An occurrence of object 'A' can relate to one and only one occurrence of object 'B', and an occurrence of 'B' can relate to only one occurrence of 'A'.*
  - ***One-to-many (1 : *)*** *One occurrence of object 'A' can relate to one or many occurrences of object 'B' ', but an occurrence of 'B' can relate to only one occurrence of 'A'.*
  - ***Many-to-many (* : *)*** *An occurrence of object 'A' can relate to one or more occurrences of 'B', while an occurrence of 'B' can relate to one or more occurrences of 'A'.*
- **Modality:** The modality of a relationship is 0 if there is no explicit need for the relationship to occur or the relationship is optional. The modality is 1 if an occurrence of the relationship is mandatory

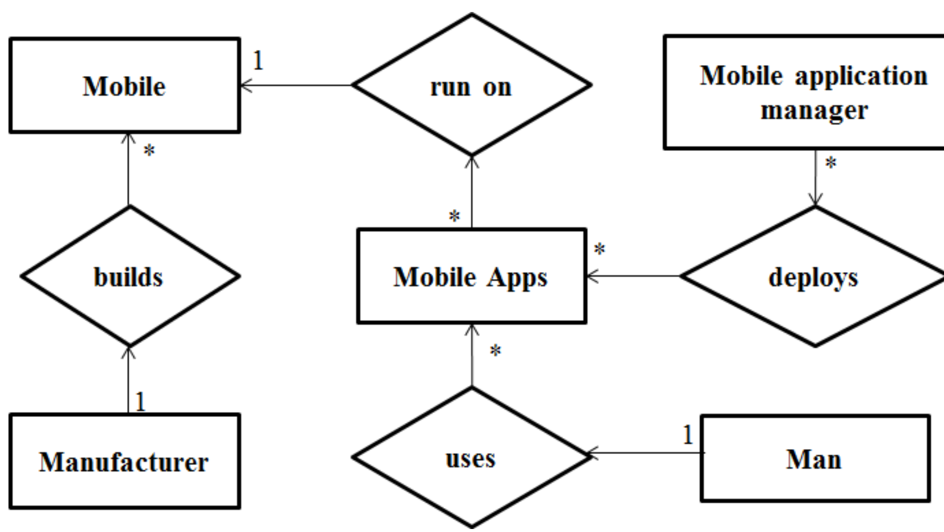
How can it be represented?

The **two** most popular ways to represent data models are,

- Entity/Relationship diagrams (**ERD**s)
- **UML** (Unified Modelling Language) diagrams

Entity/Relationship Diagrams

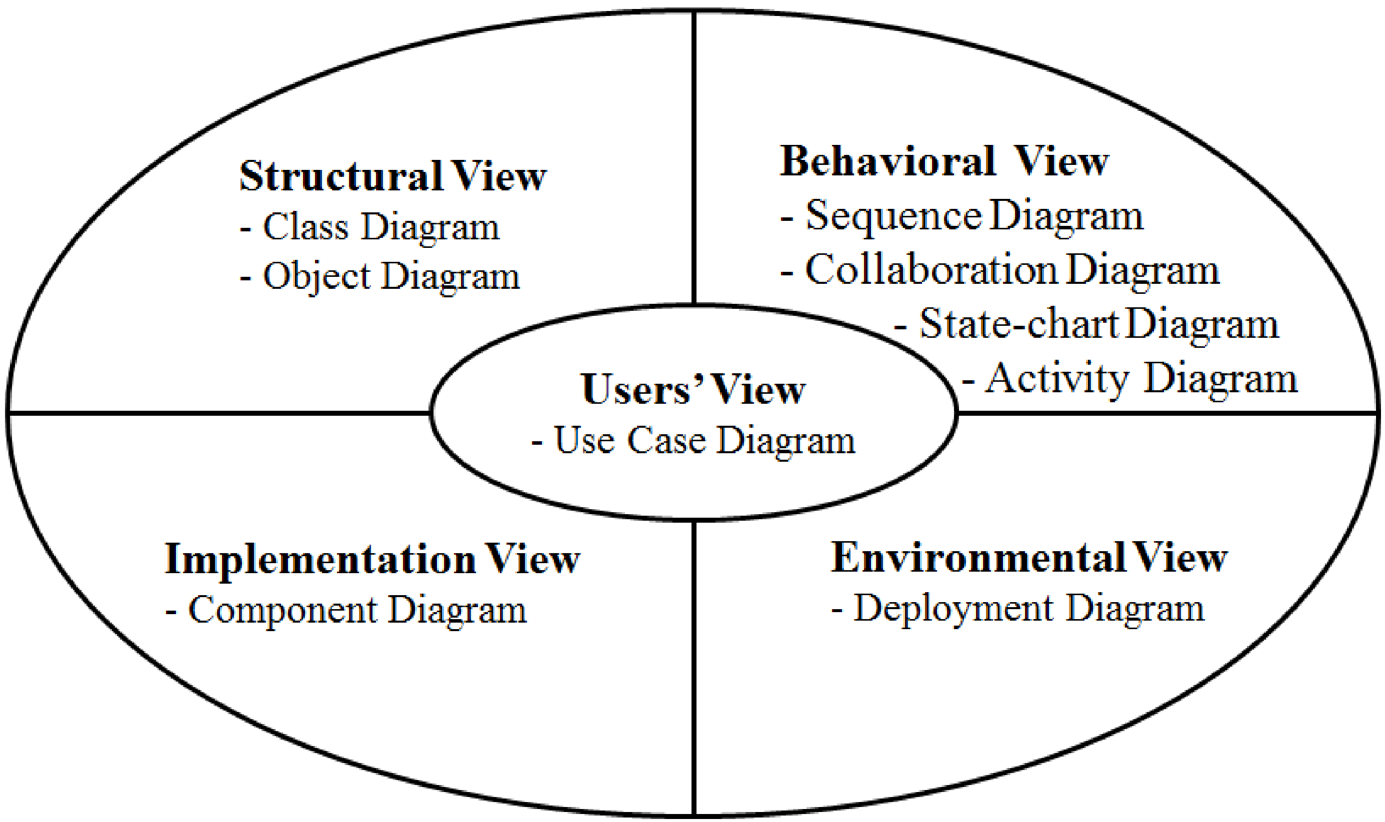
- The object/relationship pair is the cornerstone of the data model
- These pairs can be represented graphically using the entity/relationship diagram
- A set of primary components are identified for the ERD: data objects, attributes, relationships, and various type indicators
- The primary purpose of the ERD is to represent data objects and their relationships



UML Diagrams

- UML can be used to construct nine different types of diagrams to capture five different views of a system
- UML diagrams provide different perspectives of the software system to be developed and facilitate a comprehensive understanding of the system
- Such models can be refined to get the actual implementation of the system
- The UML diagrams can capture the following five views of a system,
  - *User's view*
  - *Structural view*
  - *Behavioral view*
  - *Implementation view*
  - *Environmental view*

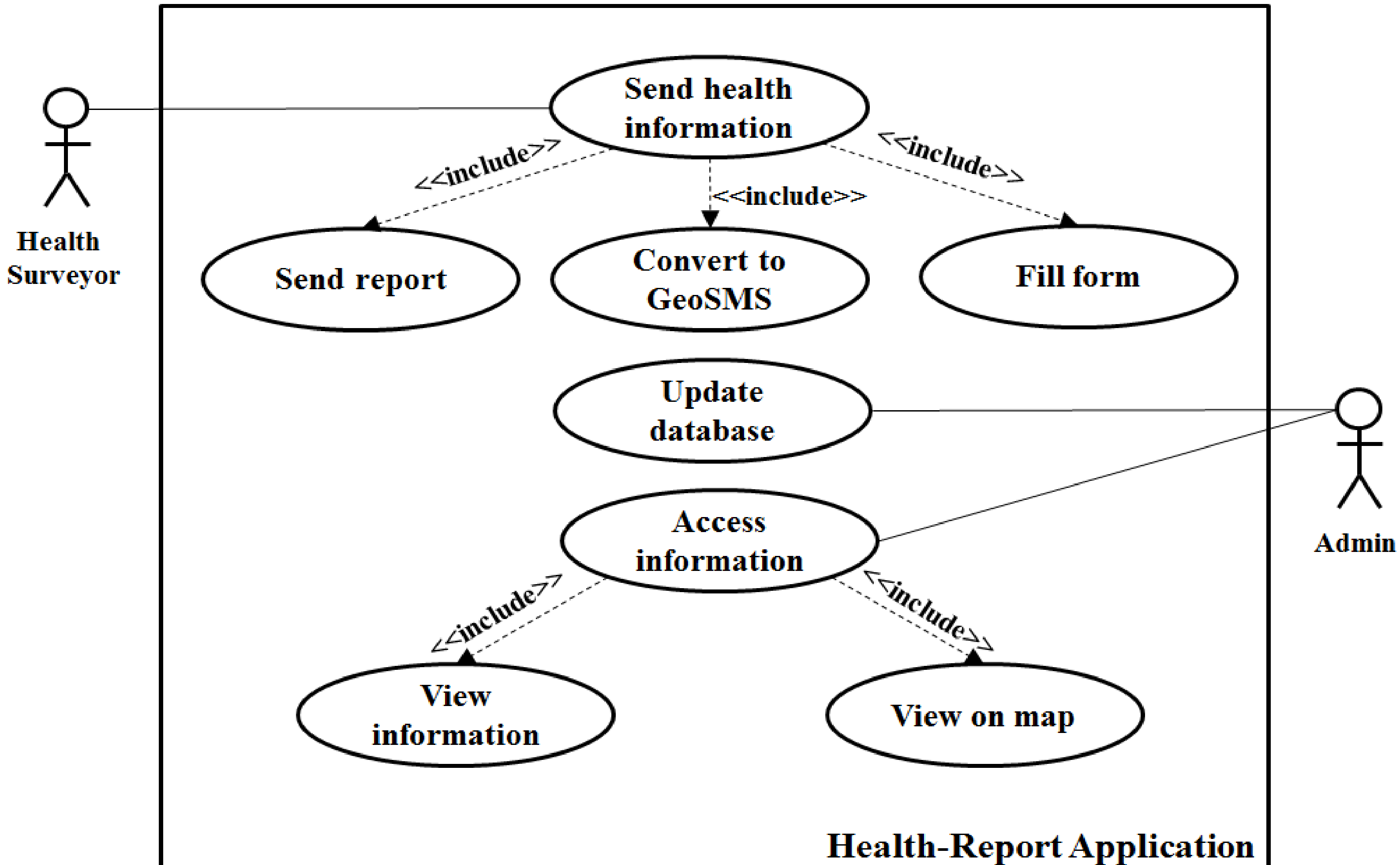
UML Diagrams Overview



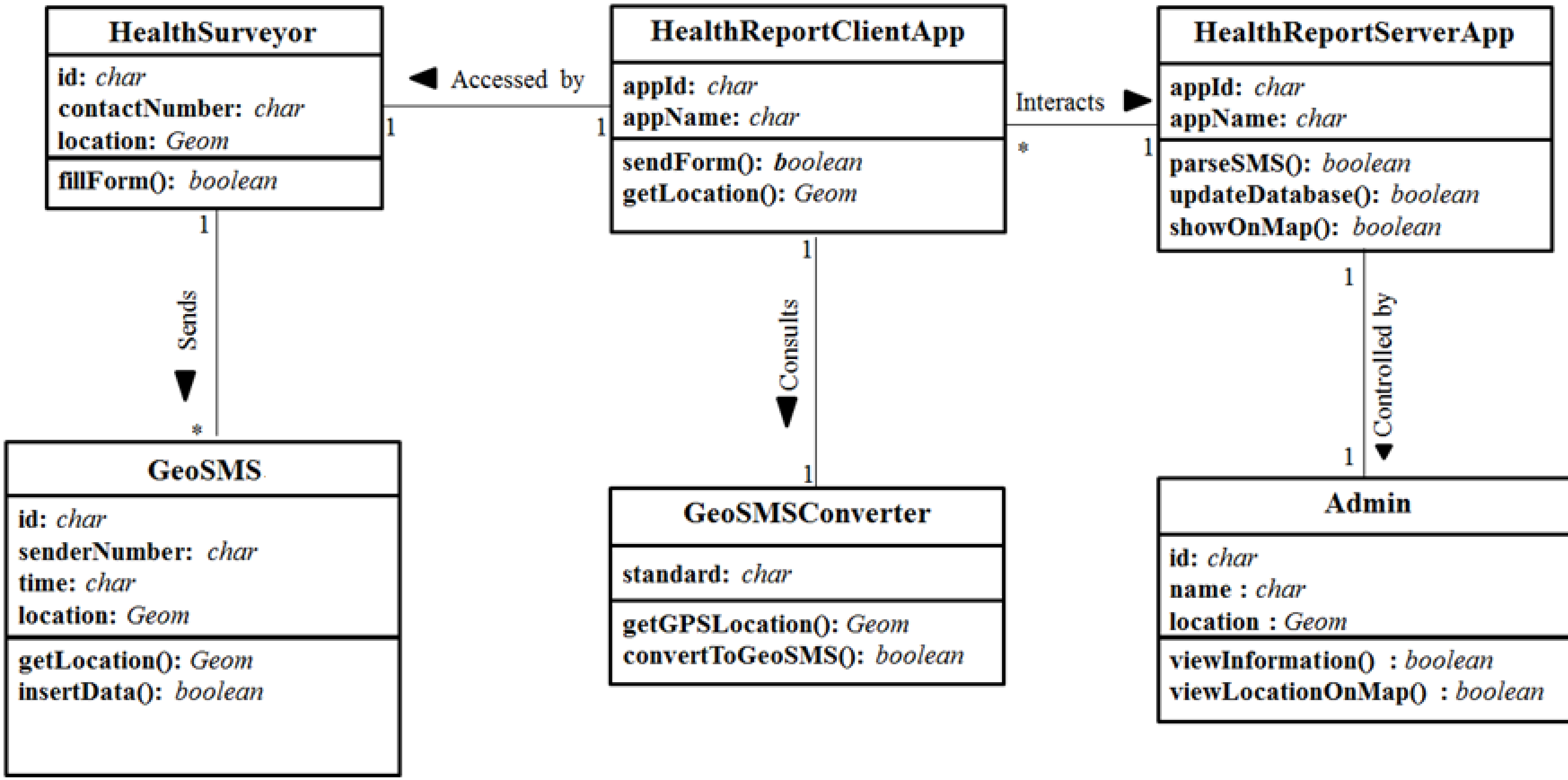
UML Diagrams Overview Contd.

- ***User's view:*** This view defines the functionalities (facilities) made available by the system to its users. It captures the external users' view of the system in terms of the functionalities offered by the system. It can be considered as the central view and all other views are expected to conform to this view
- ***Structural view:*** The structural view defines the kinds of objects (classes) important to the understanding of the working of a system and to its implementation. It also captures the relationships among the classes (objects). The structural model is also called the static model, since the structure of a system does not change with time
- ***Behavioral view:*** The behavioral view captures how objects interact with each other to realize the system behavior. The system behavior captures the time-dependent (dynamic) behavior of the system
- ***Implementation view:*** This view captures the important components of the system and their dependencies
- ***Environmental view:*** This view models how the different components are implemented on different pieces of hardware

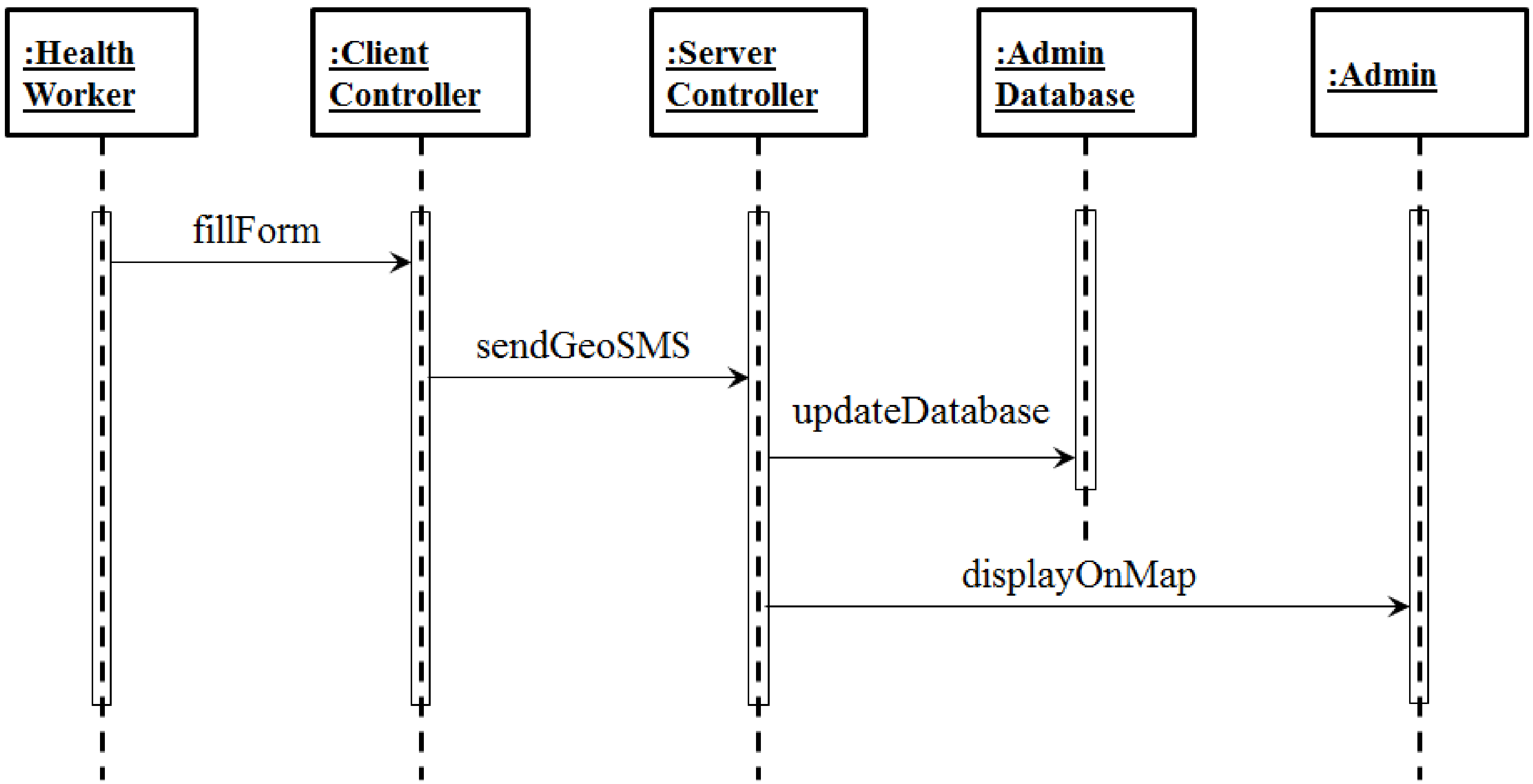
Use Case Diagram
(Health-Report Application)



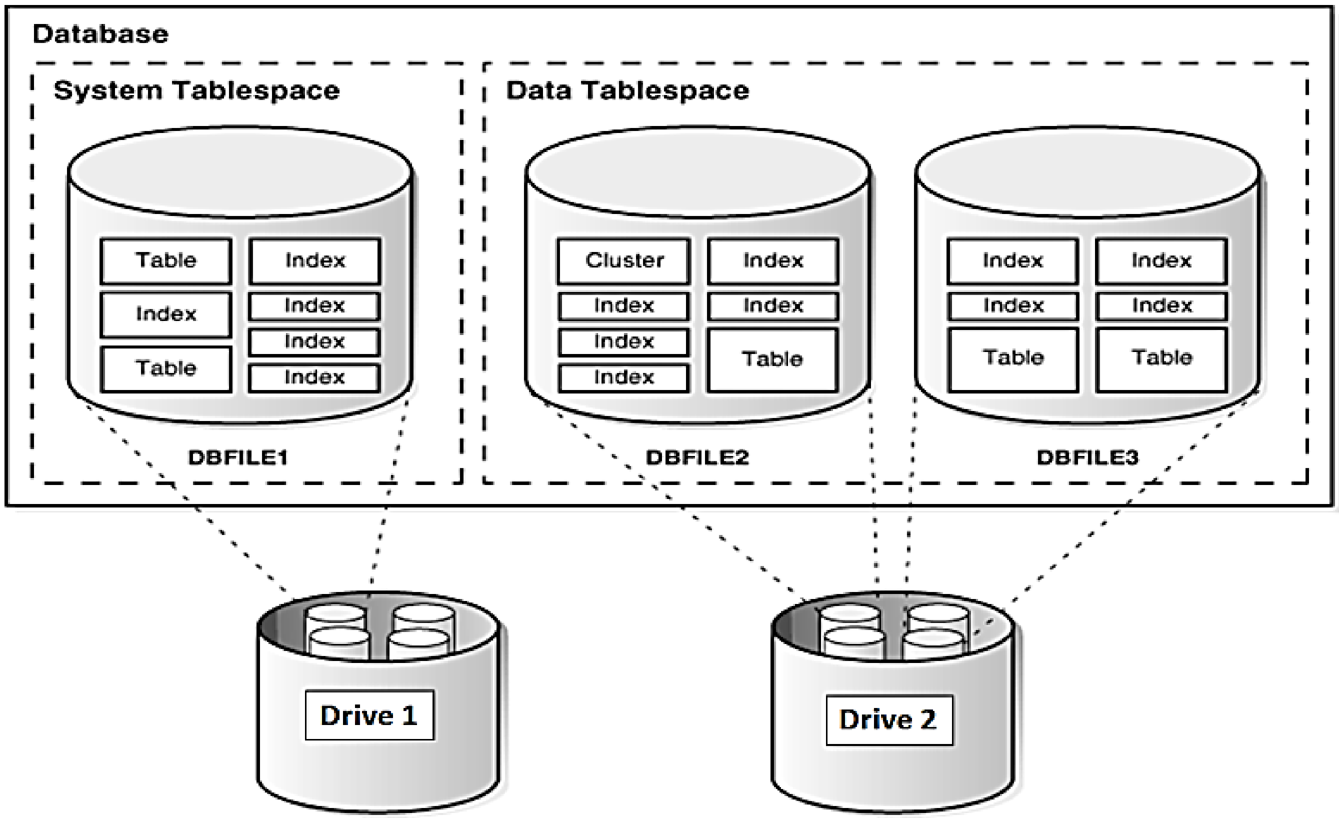
Class Diagram (Health-Report Application)



Sequence Diagram (Health-Report Application)



Database Schema



Schema Representation through GML

- An XML schema is a description of a type of XML document, typically expressed in terms of constraints on the structure and content of documents of that type
- These constraints are generally expressed using some combination of the followings,
  - grammatical rules governing the order of elements
  - Boolean predicates that the content must satisfy
  - data types governing the content of elements and attributes
- XSD is one XML schema 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
  - To determine the document's validity

Schema Representation through GML

```
<?xml version="1.0" encoding="UTF-8"?>
- <schema targetNamespace="http://www.opengis.net/examples" elementFormDefault="qualified" version="2.1.1" xmlns:ex="http://www.opengis.net/examples"
  xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema">
  - <annotation>
      <appinfo/>
      <documentation xml:lang="en">Generated OGC GML 2.1.1 Schema from UML application schema. The main UML module: </documentation>
    </annotation>
      <!-- import constructs from the GML Feature and Geometry schemas-->
    <import namespace="http://www.opengis.net/gml" schemaLocation="feature.xsd"/>
      <!-- =========== global element declarations ==========-->
    <element substitutionGroup="gml:_Feature" type="ex:EARootClassType" name="EARootClass"/>
    <element substitutionGroup="gml:_Feature" type="ex:AdminType" name="Admin"/>
    <element substitutionGroup="gml:_Feature" type="ex:HealthReportServerAppType" name="HealthReportServerApp"/>
    <element substitutionGroup="gml:_Feature" type="ex:HealthReportClientAppType" name="HealthReportClientApp"/>
    <element substitutionGroup="gml:_Feature" type="ex:HealthSurveyerType" name="HealthSurveyer"/>
    <element substitutionGroup="gml:_Feature" type="ex:GeoSMSConverterType" name="GeoSMSConverter"/>
    <element substitutionGroup="gml:_Feature" type="ex:Interface1Type" name="Interface1"/>
      <!-- =========== featureMember elements ==========-->
      <!-- =========== featureMember types ==========-->
      <!-- =========== END featureMember types ==========-->
      <!-- =========== type definitions ==========-->
  - <complexType name="EARootClassType">
    - <complexContent>
      - <extension base="gml:AbstractFeatureType">
          <sequence/>
        </extension>
      </complexContent>
    </complexType>
  - <complexType name="AdminType">
    - <complexContent>
      - <extension base="gml:AbstractFeatureType">
        - <sequence>
            <element type="ex:intType" name="Id"/>
            <element type="ex:charType" name="name"/>
            <element type="ex:charType" name="location"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
</schema>
```

From Logical Data Model to XMI

- XML Metadata Interchange (XMI)
- Enterprise Architect
- Creating logical data model using Enterprise Architecture
- Representing logical data model in XMI format using Enterprise Architect

XML Metadata Interchange (XMI)

- XMI (*XML Metadata Interchange*) is a standard, proposed by Object Management Group (OMG), for exchanging metadata information ("data about data") via *Extensible Markup Language (XML)*

(The ***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.)

- The ***main purposes*** behind the proposal of XMI:
  - ✓ To help system analyst to exchange their data models with each other
  - ✓ To facilitate in exchanging information about data warehouses

Enterprise Architect

What is Enterprise Architect:

- Enterprise Architect is an intuitive, flexible, scalable, feature-rich, and powerful ***UML analysis and design tool*** for building robust and maintainable systems and processes in a wide range of industries

- It combines the power of the latest UML specification (www.omg.org) with a high performance, intuitive interface, to bring an integrated and advanced toolset to the whole development team

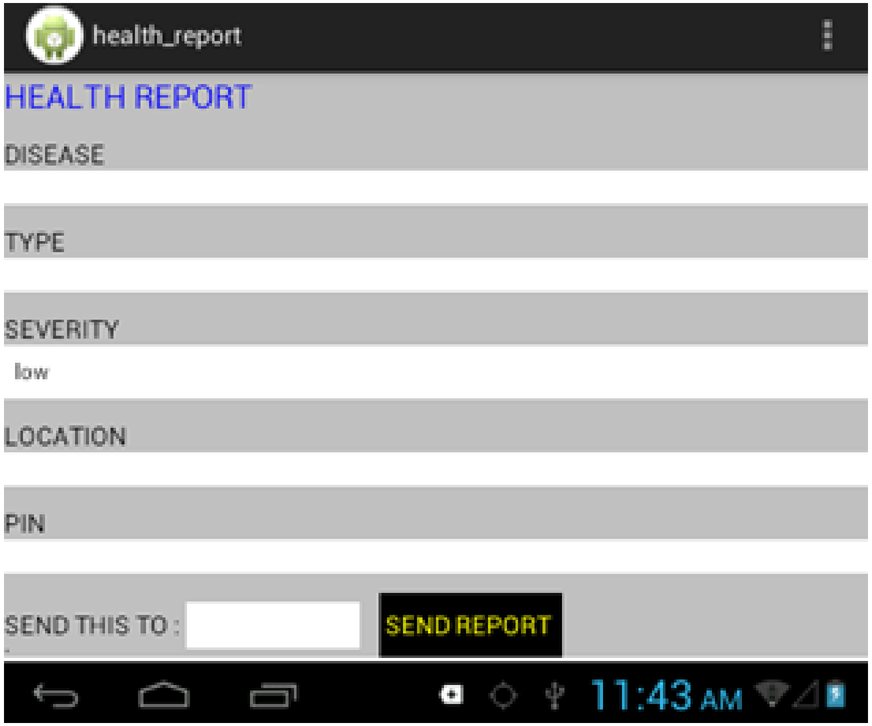
- It is useful to different steps in a software development process

- ***Key Features of Enterprise Architect:***
  - ✓ Modeling with UML
  - ✓ Project Sharing Import/Export
  - ✓ Database modeling
  - ✓ Support to many languages
  - ✓ Visual Execution, Analyzer, Debuggers, Build, Testing
  - ✓ Simulation
  - ✓ Automation and Scripting etc.
- ***Key Benefits of Enterprise Architect:***
  - ✓ Design and Build Diverse Systems Using UML
  - ✓ Model and Manage Complexity
  - ✓ Develop Personal Views and Extracts of the Model
  - ✓ Generate Documentation
  - ✓ Generate and Reverse Engineer Source Code
  - ✓ Visualize, Inspect and Understand Complex Software
  - ✓ SOA (Service Oriented Architecture) support
  - ✓ Systems engineering support etc.

Data Acquisition

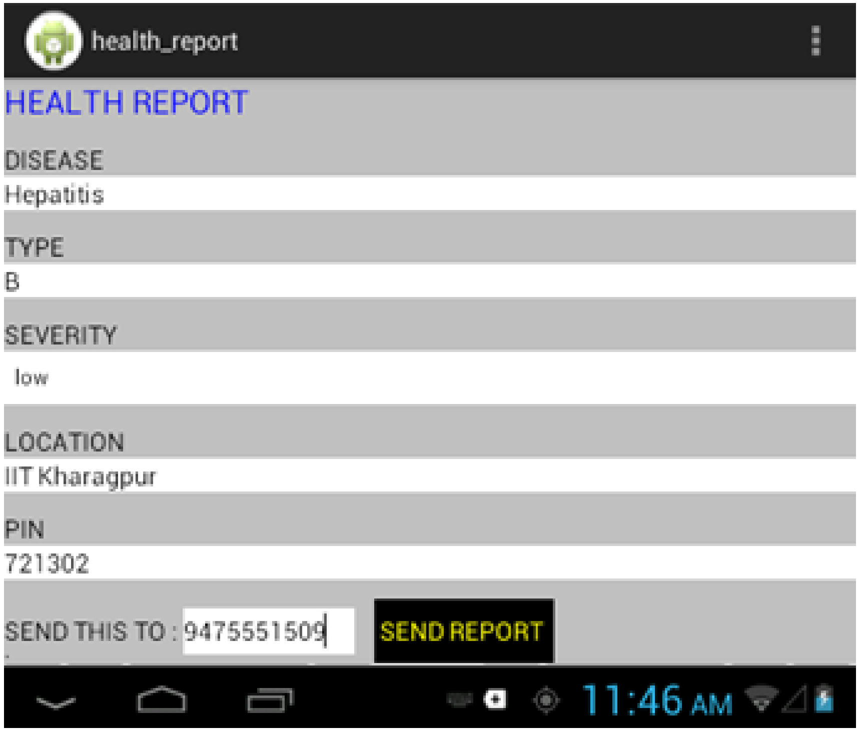
- Once the database is created, any new record of data acquired will be pushed to the EGIS though the application program.
- An example for data acquisition using the *Health Report* application is discussed following.

Health Report Application



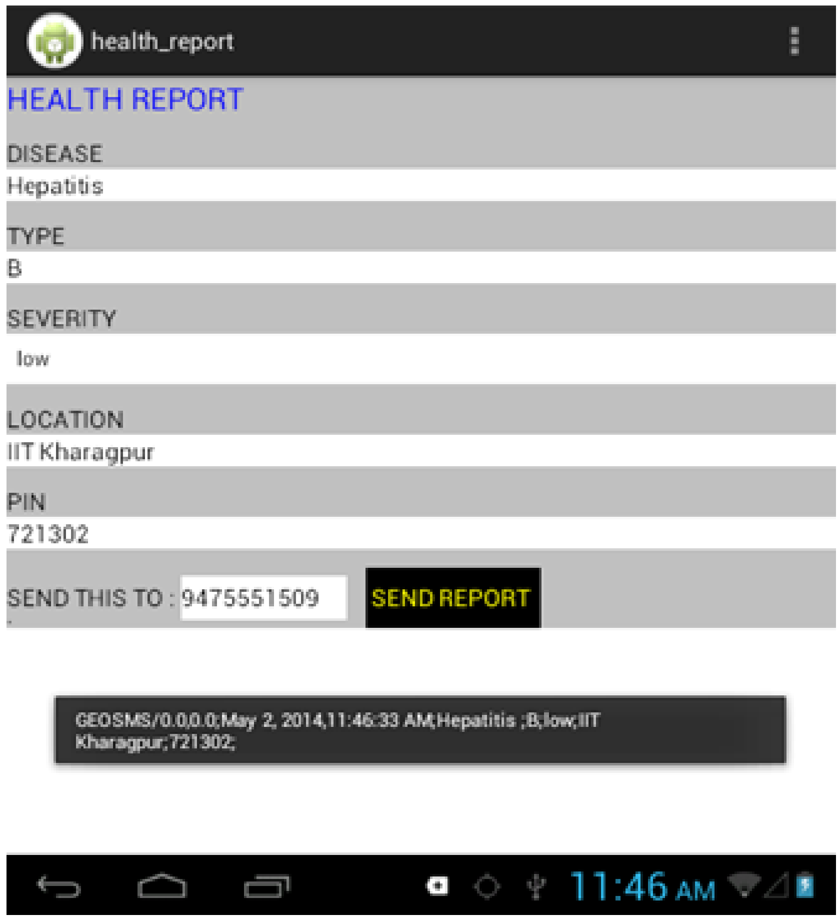
Interface of the Form for Health Data Acquisition

Health Report Application



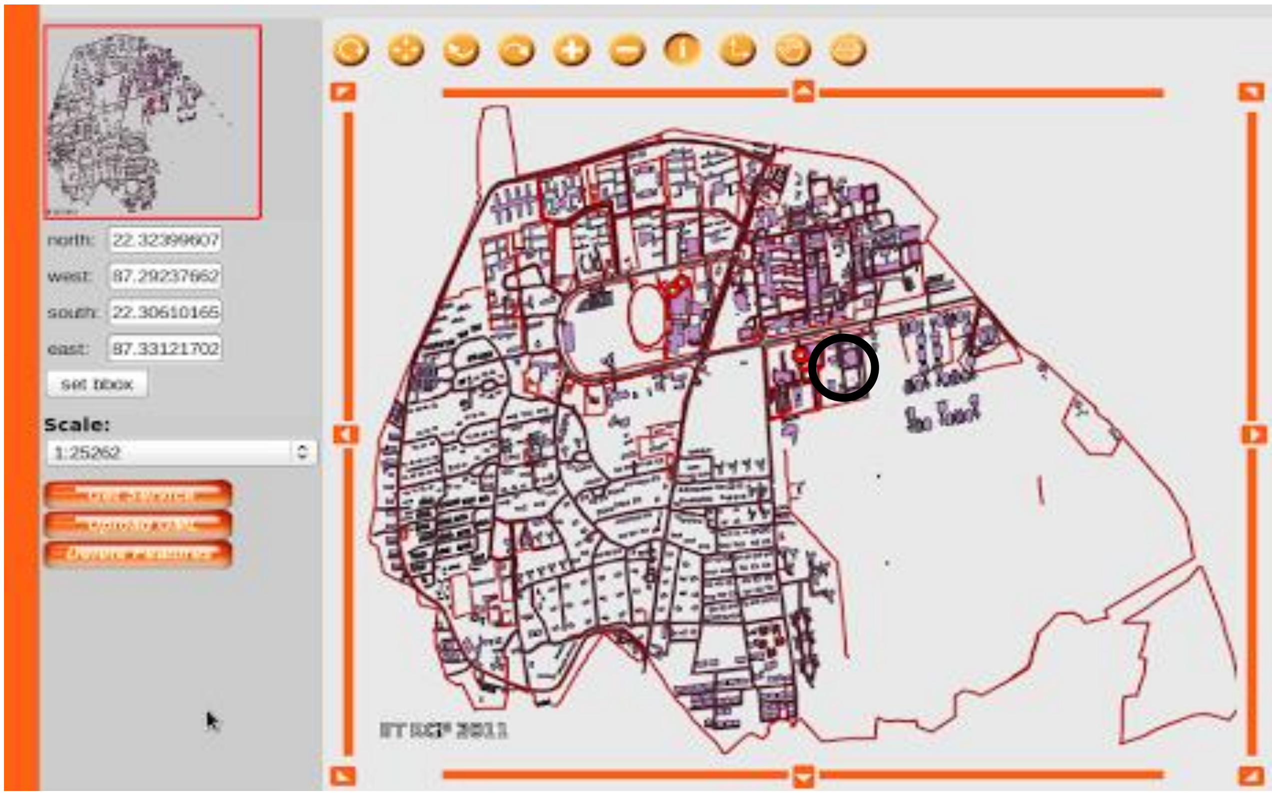
Filled Form ready to be sent to the server

Health Report Application



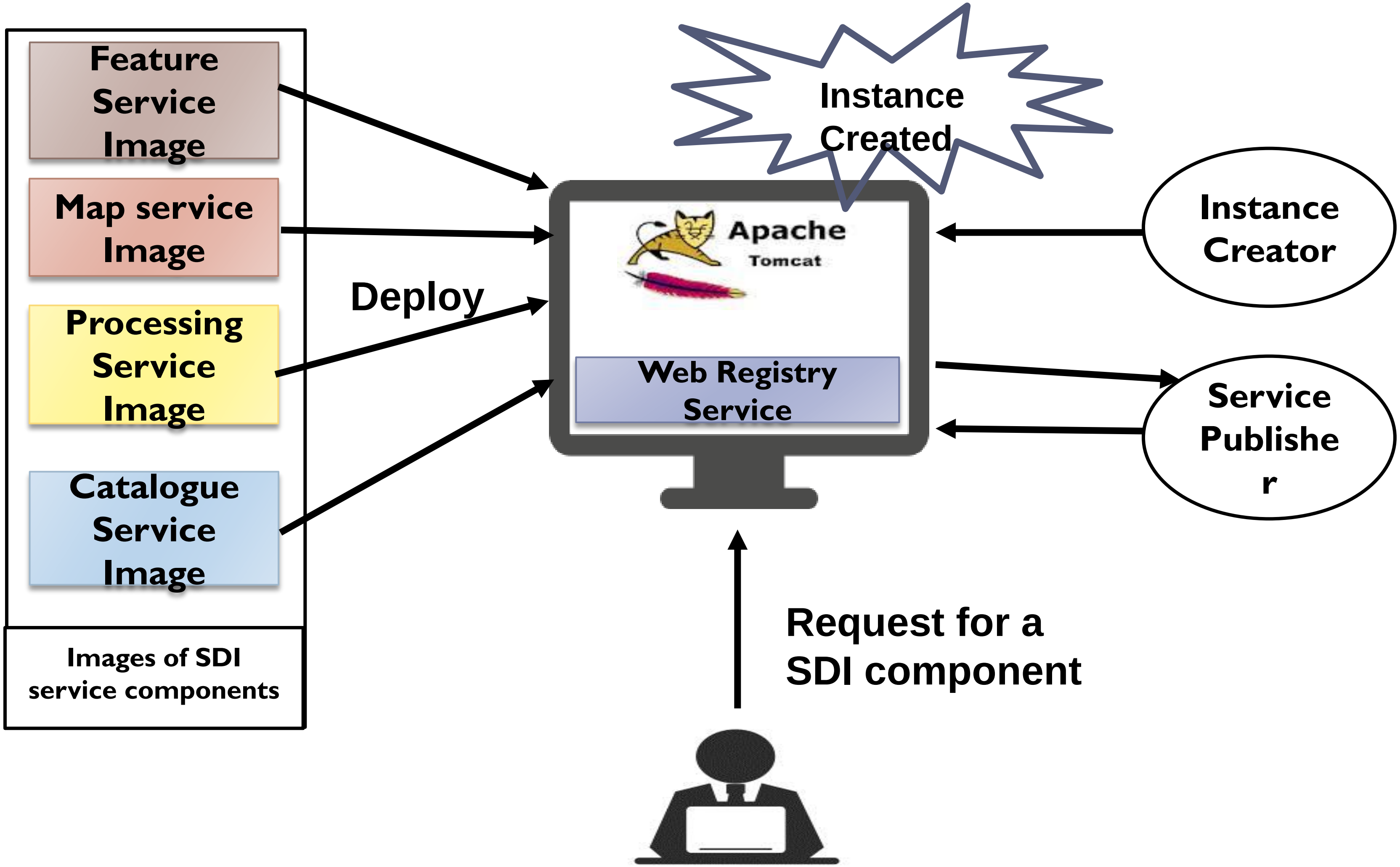
Form Converted into GeoSMS

Health Report Application



Location on the Map

GeoSpatial Cloud - Services

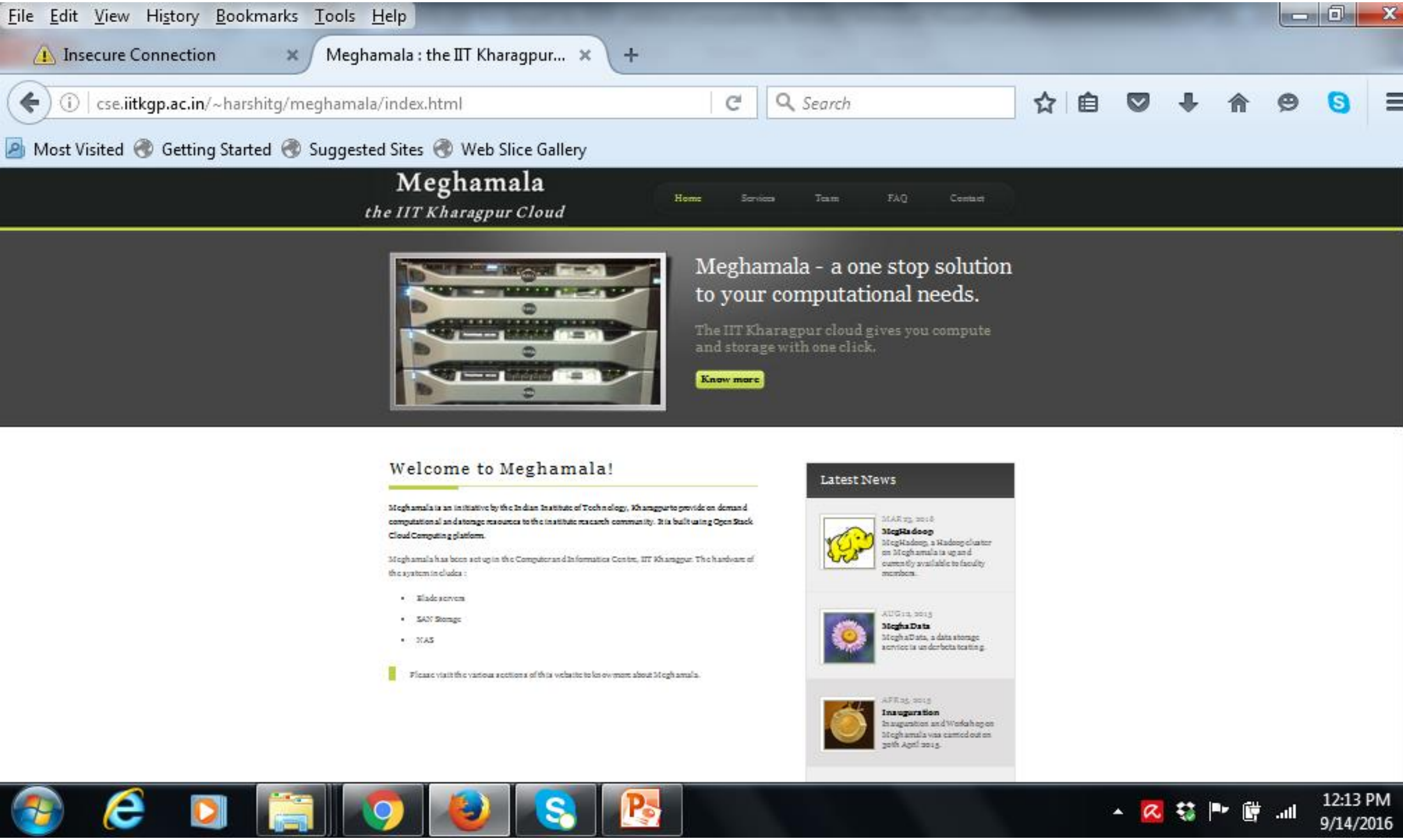


Experimental GeoSpatial Cloud @IITKgp

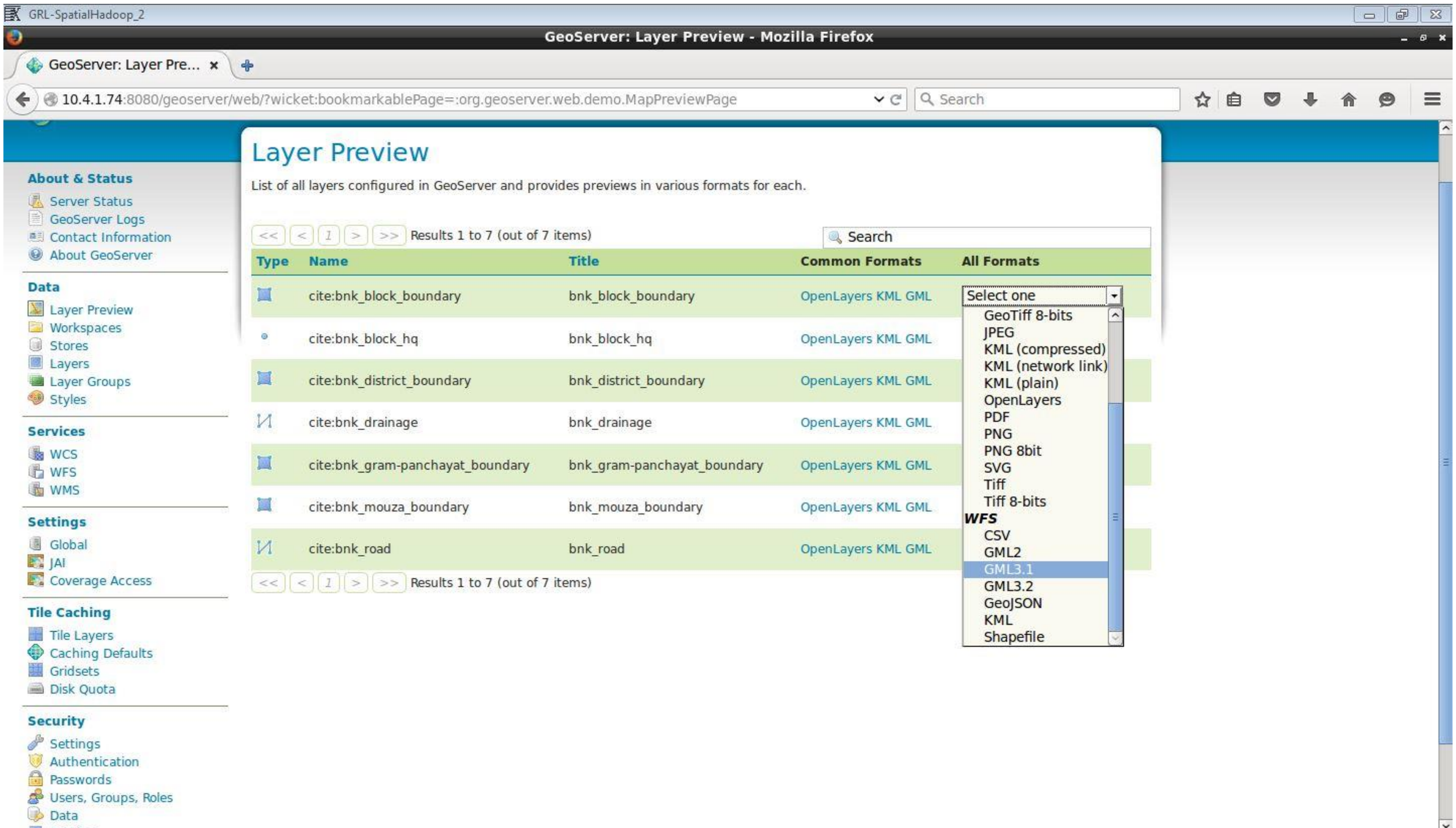
on

Meghamala

(Open Source Private Cloud of IITKgp)

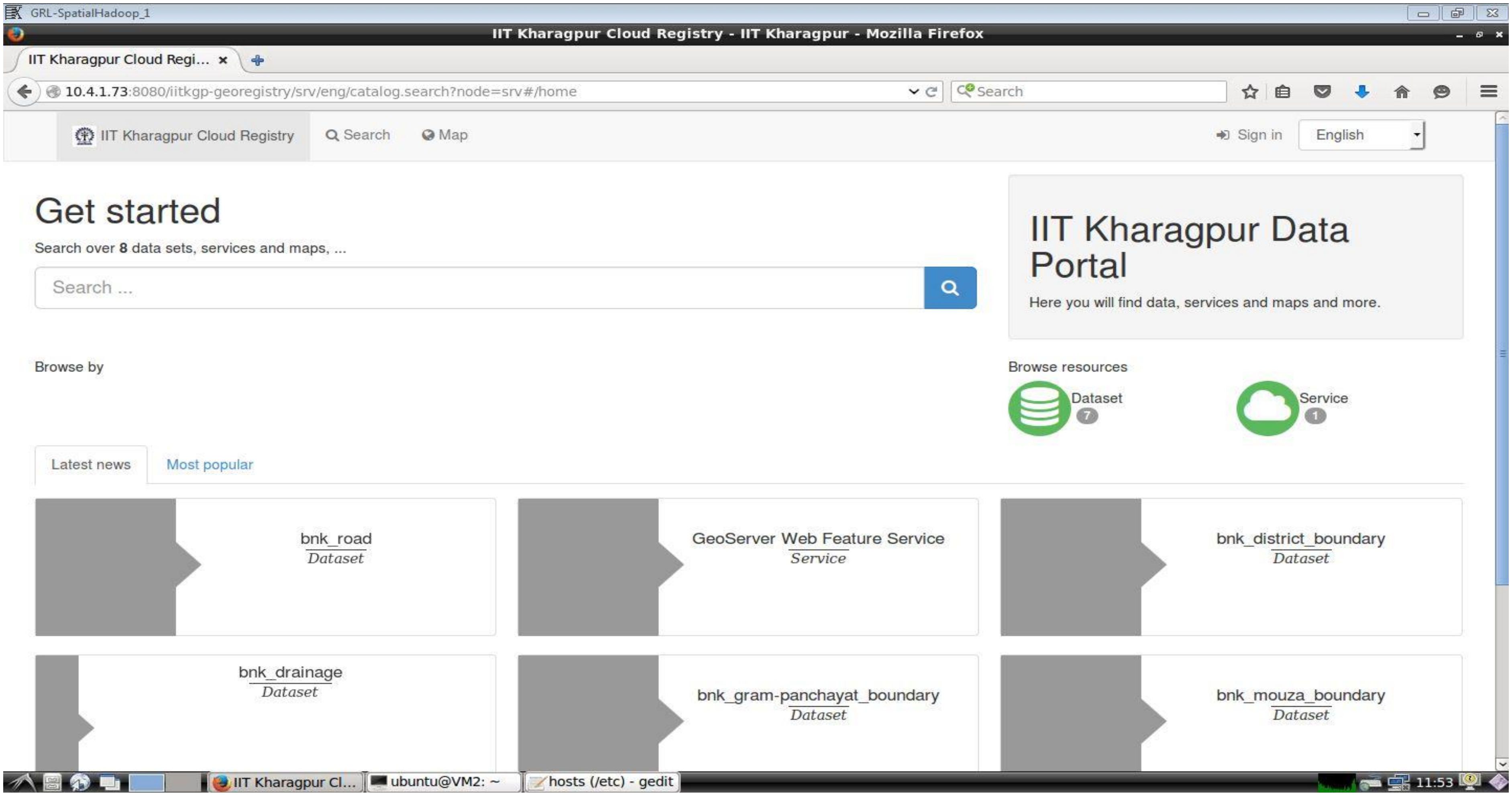

File Edit View History Bookmarks Tools Help
Insecure Connection
Meghamala : the IIT Kharagpur...
cse.iitkgp.ac.in/~harshitg/meghamala/index.html
Search
Most Visited
Getting Started
Suggested Sites
Web Slice Gallery
Meghamala
the IIT Kharagpur Cloud
Home
Services
Team
FAQ
Contact
Meghamala - a one stop solution to your computational needs.
The IIT Kharagpur cloud gives you compute and storage with one click.
Know more
Welcome to Meghamala!
Latest News
MegHadoop
MeghaData
Inauguration
12:13 PM
9/14/2016

WFS on IITKgp GeoCloud

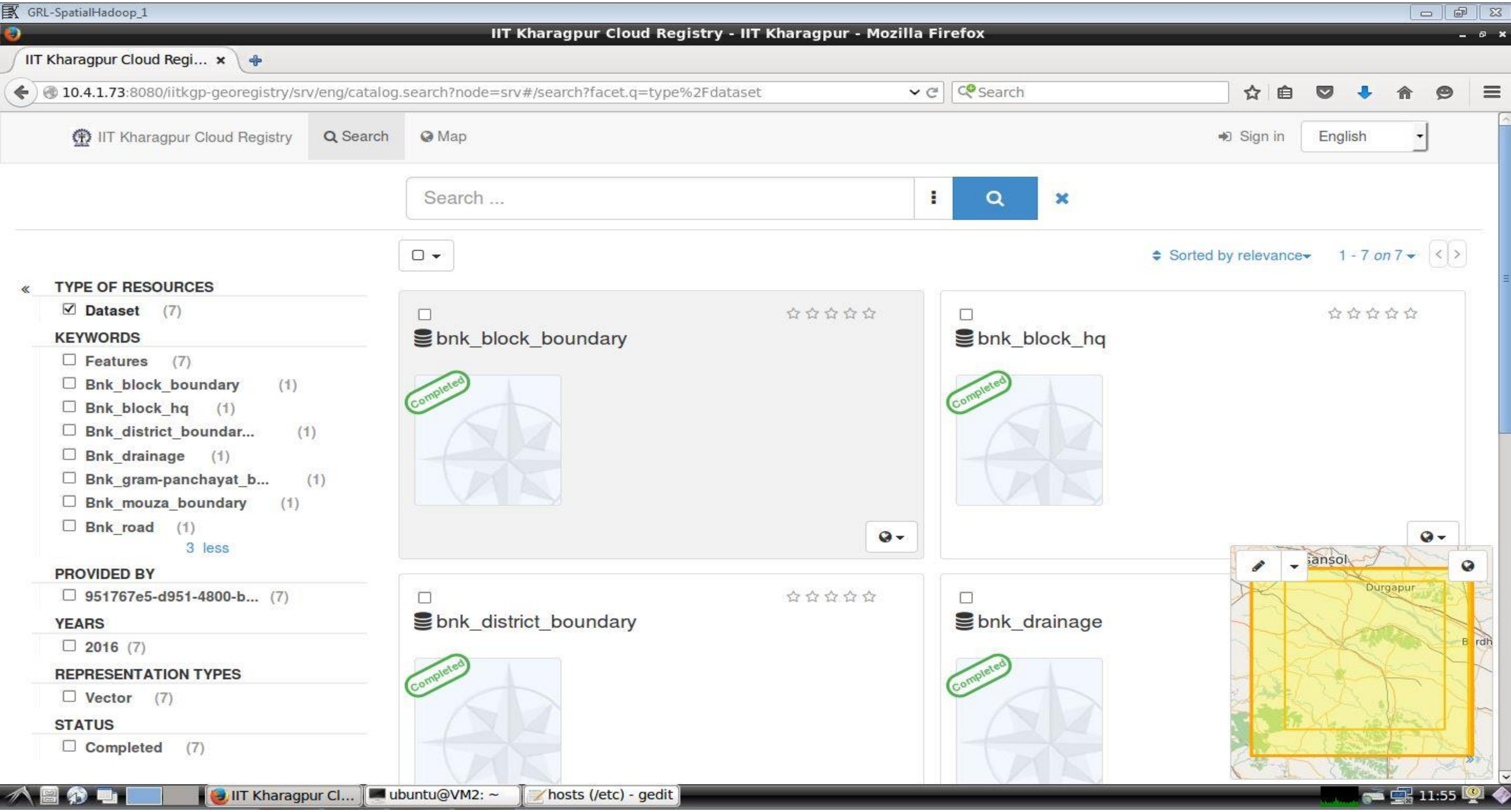


WMS on IITKgp GeoCloud

Registry on IITKgp GeoCloud



Registry (with WFS) on IITKgp GeoCloud



Summary

- SDI for Socio-economic development is a necessity – rather than a option
- NSDI, Organizational-SDIs, SSDIs should be seamlessly integrated – Service level, Loosely coupled, Policy binding
- Stakeholders should find SDI appealing as a market
- Evolving as a GeoSpatial Cloud for catering "on-demand" services for SDI – part of National GIS?
- *S&T trend: Spatial Data Science, Spatial Big Data, Fog Computing, Geo-IoT and Geo-CPS*

Thank you!

Soumya K. Ghosh
Professor
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India
skg@cse.iitkgp.ac.in