Design of Web based Geospatial Data Services Framework – An Implementation of Web-based Climate Services

Thesis submitted in partial fulfilment

of the requirements for the degree of

Master of Science in

Computer Science and Engineering by Research

by

AARATHI RAMESH MUPPALLA
20173018
aarathi.muppalla@research.iiit.ac.in

International Institute of Information Technology, Hyderabad

(Deemed to be University)

Hyderabad - 500032, INDIA

June 2023

International Institute of Information Technology

Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled **“Design of Web based Geospatial Data Services Framework - An Implementation of Web-based Climate Services”** by Aarathi Ramesh Muppalla, has been carried out under my supervision and is not submitted elsewhere for a degree.

-------------------------	----------------------
Date	Adviser: Dr K. S.Rajan

Acknowledgments

My sincerest thanks to my supervisor, Prof.K.S.Rajan, Head LSI and Registrar IIIT-H for his kind guidance, motivation, understanding, continued support and wealth of knowledge to complete the thesis work.

I owe a debt of gratitude to the management of IIIT-H and all the staff members of IIIT-H for timely help in administrative and financial tasks.

I sincerely thank the OGC network for the Geospatial Standards and for their efforts to build the opensource solutions for the OGC Standards.

I sincerely thank Dr. Prakash Chauhan Director, NRSC/ISRO for his kind support and encouragement to carry out my thesis work. I greatly thank Shri. Vinod M Bothale, Former Associate Director, NRSC/ISRO for his immense support and motivation to complete the thesis work.

I sincerely thank my colleagues of NRSC/ISRO for their support and encouragement.

I sincerely thank my husband, Mr Mahesh Pathakoti, Scientist-SE at NRSC/ISRO and PhD student at LSI, IIIT-H for his continuous support and encouragement that helped me complete this thesis work.

Abstract

Spatial Data has increased tremendously in the last two decades from various data sources including satellite sensor data, Internet of Things data, data collected from remote sensing instruments and crowd sourced data. With the advent of mobile phones, devices and internet applications, a vast amount of spatial data is also available from various mobile and web applications. These datasets have lots of applications in different fields like Urban planning, waste management, Traffic monitoring and planning, Climate science studies, Climate resilience, Agriculture, Forest survey, disaster management, smart city planning etc. These datasets have huge volume, variety, veracity, and velocity, making the problem of spatial data analysis a big data problem. To convert these datasets to usable applications in timely manner will ensure the effective utilization of the application. A unified framework that collects, processes, and helps users analyze the output over internet will be beneficial to the geospatial domain researchers and policy makers for enabling faster decision making. Apart from these, web based geospatial applications also has applications for general citizens like smart homes, transportation etc.

Climate change has been significantly affecting our environment and society since ages. Another important application of web based geospatial framework is climate change study. This is one of the urgent needs for which informed action has to be taken immediately.

Sustainable Development Goal 13 (SDG-13) calls for urgent climate action to combat climate change [1]. This calls for focused study to create mitigation plans at global, country and regional levels. For this, climate research should be carried out in faster way. Also, along with climate science expertise, traditional climate research involves significant data analysis and knowledge in computing is essential. Considering this, there is a need for common platform that provide web-based climate services for climate researchers, decision makers and policy makers that enable faster research and availability of research output results. It is also required for these services to be discoverable, available, interoperable, and reusable. Hence, using published and widely used standards for this platform is also essential for enabling wider use.

Open geospatial Consortium (OGC) has been developing and maintaining the standards needed for Geospatial data management, publishing, sharing, and processing such datasets. OGC Web Services standards WMS, WFS, and WPS are suitable for providing data access to

users in the form of maps and data. OGC Web Services standard WPS, allows users to utilize the geospatial data processes that are published. Hence OGC Web Services standards are suitable for implementing this framework. Also, these services are widely used and available as open-source solutions. However, OGC Web Services are implemented in Remote-Procedure-Call (RPC) architectural style using Extensible Markup Language (XML) over HTTP protocol. Currently, resource-oriented architecture like RESTful API is being used by various modern web applications and are competing against traditional service-oriented architectures like RPC. Hence, standards that implement RESTful APIs like OGC API standards are more suitable for this framework. OGC API components maps, tiles, features, coverages, and processes offer similar services to OGC Web services. Additionally, OGC API records implements a mechanism for the available datasets and tools to be discoverable. Considering this, OGC APIs are more suitable for this framework. Extensive work is being carried out by the OGC community to publish all the standards of OGC APIs and many opensource software like pygeoap, GeoServer and MapServer are also implementing these standards. Considering this, we chose both OGC API and OGC Web Services standards for the design of this framework. Also, most of the geospatial software worldwide offers tools and services compatible with OGC Web Services currently. By implementing both standards, our design will follow the W3C best practices for sharing data as well as being in compliant with most available and widely used geospatial software. Phasing out the OGC web services can be carried out when the OGC APIs are published and implemented by major geospatial software and service providers.

SensorThings API is another standard that is used in this design for ingesting sensor data into the portal. After ingestion, this data is designed to be served using OGC API Features, Maps and Environmental Data Retrieval components making sensor data useful to different sectors of users.

Considering the current requirements and the above-mentioned standards, in our research work, we identify the challenges involved in developing a unified web based spatial data framework and design a web based spatial data framework that will address these challenges. Further, we chose climate science as a use case to demonstrate this framework.

We designed this framework to accept data from multiple sources like Satellite derived products, Internet of Things datasets, Instrument datasets and citizen datasets in various geospatial formats by using open-source solutions that implement the OGC Web Services, OGC API and SensorThings API standards. We designed the modules and the workflows to address the functional requirements of this framework. Apart from the functional requirements, we also identified the need for the framework to be scalable and highly available for the framework to be effectively used by larger audience. We added big data solutions Elasticsearch and open data cube that make the framework scalable and handle big data.

We implemented three use cases in climate science based on the framework principles and successfully demonstrated handling of Internet of things data, scalability of IoT data, development of on-the-fly dashboards and the usability of publishing geospatial data processes for faster research.

Although, we demonstrated the usage of framework for IoT and satellite derived data sets, further work needs to be carried out to implement other geospatial datasets and formats. Usage of Big Data solutions for large datasets will also be beneficial to handle the processing of large spatial datasets and provides future scope for development.

Table of Contents

List of Figure

List of Abbreviations

Intergovernmental Panel on Climate Change (IPCC),
United Nations (UN)
Global Climate Observing System (GCOS)
United Nations Framework Convention on Climate Change (UNFCCC)
ECV (Essential Climate Variables)
National Information system for Climate and Environment Studies (NICES)
Indian Space Research Organization (ISRO)
India Meteorological Department (IMD)
National Remote Sensing Centre (NRSC)
National Spatial Data Infrastructure (NSDI)
Geographic Information System (GIS)
Spatial Data Infrastructure (SDI)
Open Geospatial Consortium (OGC)
Internet of Things (IoT)
Earth Observation (EO)
Unmanned Aerial Vehicle (UAV)
National Aeronautics and Space Administration (NASA)
Greenhouse Gas Analyzer (GGA)
Fourier Transform InfraRed (FTIR)
Network Common Data Form (NetCDF)
Hierarchical Data Format (HDF)
General Regularly distributed Information in Binary form (GRIB)
Tag Image File Format (TIFF)
JavaScript Object Notation (JSON)
Digital line graph (DLG)
Geography Markup Language (GML)
Keyhole Markup Language (KML)
Low Range Wide Area Network (LoRaWAN)
REpresentational State Transfer (REST)
Hyper Text Transfer Protocol (HTTP)
Secure Hyper Text Transfer Protocol (HTTPS)
Application Programming Interface (API)
SensorThings API (STA)
Chloro fluoro carbons (CFCs)
hydro chlorofluorocarbons (HCFCs)
Total column Ozone (TCO)
National Aeronautics and Space Administration (NASA)
Total Ozone Mapping Spectrometer (TOMS)
Ozone Monitoring Instrument (OMI)
Copernicus Climate Change Service (C3S)
Global Framework for Climate Services (GFCS)

Climate Data Store (CDS)
Digital Object Identifier (DOI).
Quality of Service (QoS)
Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)
OGC Catalog Services for web (OGC-CSW)
OGC web services (OWS)
Web Map Service (WMS)
Web Feature Service (WFS)
Web Coverage Service (WCS).
OGC API - Environmental Data Retrieval (OGC API - EDR)
Sensor Web Enablement (SWE)
Observations & Measurements Schema (O&M)
Sensor Model Language (SensorML)
Transducer Markup Language (TML)
Sensor Observations Service (SOS)
Sensor Planning Service (SPS)
Sensor Alert Service (SAS)
Web Notification Services (WNS)
Message Queuing Telemetry Transport (MQTT)
Universal Resource Locator (URL)
Key-Value Pair (KVP)
Spatio Temporal Asset Catalog (STAC)

Chapter -1: Introduction

1.1 Motivation

Spatial Data has increased tremendously in the last two decades from various data sources including satellite sensor data, Internet of Things data, data collected from remote sensing instruments and crowd sourced data. With the advent of mobile phones, devices and internet applications, a vast amount of spatial data is also available from various mobile and web applications.

These datasets when effectively used has lots of applications in different fields. Some of the examples include Urban planning, waste management, Traffic monitoring and planning, Climate science studies, Climate resilience, Agriculture, Forest survey, disaster management, smart city planning etc. As these datasets have different sources, they are highly variable in terms of spatial and temporal resolutions and coverages. For example, the datasets collected from satellites has large spatial coverage but less spatial resolution whereas a smart home sensor has limited spatial coverage and high temporal resolution. The veracity of these datasets due to the accuracies that are dependent on the data source, mechanism to acquire and process data, also poses a challenge to integrate these datasets and effectively use them for a common purpose. Some of these datasets have very high temporal resolution contributing to the large velocity, especially the data acquired from IoT sensors and citizen handheld devices. These features make spatial data as spatial big data, making it challenging to collect, process and analyze. [9]

Most of the spatial datasets are available for public and researchers for usage over Internet. Extracting information specific to the user application from these publicly available data along with the user data on timely manner will ensure the effective utilization of the application. Collection, processing and analyzing spatial big data needs expertise in multiple domains including spatial statistics, computer science and the application domain in which the end use of the product is planned. Also, executing all the necessary steps involved in spatial big data processing is time consuming and ultimately leads to the delay in the development of application. To address these problems, it is necessary to build a framework that does these processes and provides output over internet that will reduce the time and knowledge required for the users to develop their applications. It is also necessary to build this framework using widely used standards as this will help to make the data and tools discoverable for wider usage and accessible across multiple platforms. Such portal reduces the turn-around time for developing spatial applications, makes it easier to use across multiple platforms and makes it easier for the users to generate the products or applications.

Considering this, in our research work, we identify the challenges involved in developing a unified web based spatial data framework and design a web based geospatial data services framework that will address these challenges. Further, we chose climate science use cases to demonstrate this framework.

1.2 Climate Change Context

Climate change is significantly affecting our environment and society since ages. It plays a vital role in extreme climate weather and climate events, more diseases and economic loss. The Intergovernmental Panel on Climate Change (IPCC), which is the United Nations body for assessing the science related to climate change, mentions in its sixth assessment report that, climate change is a threat to human well-being and planetary health. In its technical summary report, it is mentioned that there is a critical need for more targeted research to develop appropriate country-level, locally specific, policy and land management response options [2][3].

Apart from the expertise in climate science, good quality climate data is one of the important requirements for climate science studies. The main source for this data is satellite data, on-site instrument data, products derived from satellite data and data collected by drones and other aerial vehicles. These days' low cost IoT sensors are also available or collecting climate related data. Along with the data, it is also essential to improve use of climate information available for better understanding and predicting climate change and weather conditions and to enable policy makers to take timely actions to mitigate the effects of climate on society.

United Nations (UN) established the Global Climate Observing System (GCOS) in 1992 with climate experts from Atmosphere, Ocean, and Terrestrial domains. One of the main activities of GCOS is to coordinate and ensure the availability of climate observations for all potential users. GCOS also identifies gaps in observation systems and proposes actions to respond to data and information needs of international frameworks, such as the United Nations Framework Convention on Climate Change (UNFCCC) and the IPCC. Identification of Essential Climate Variables (EVCs) is one of the key tasks of GCOS. There are 54 ECVs are in three domains, namely, atmosphere, ocean and terrestrial [4][5].

Climate services are services that provide climate information such as ECVs to the common public, researchers and policy makers for climate studies and assisting decision making.

Making these services available over web, helps the services to reach wider audience. Web based climate services are available from different web-portals worldwide. Some countries have well developed climate services, and many countries are still developing such services. Maria Teresa Miranda Espinosa et al [8], discusses the various Data catalogues/platforms that provide access to ECV datasets and accesses their discoverability, accessibility, and usability. They identified that there is a lack of standard metadata information, the scarcity of data dissemination via web services, lack of proper user support to discover and access ECVs. It was suggested that improving the discoverability, interoperability, and accessibility to quality-controlled ECV data records using web services is required along with browsing, filtering, retrieval and ingestion mechanisms, automated data processing pipelines to better enable community driven tools and have value added applications.

In India, Climate science studies are currently carried out by various agencies in government, academic and private sectors. Climate services and climate related datasets are available public use through web portals of Indian Space Research organization (ISRO) (bhuvan.nrsc.gov.in, mosdac.gov.in), India Meteorological Department (IMD, https://mausam.imd.gov.in/ , https://cdsp.imdpune.gov.in/ (IMD Climate data service portal)), government open data portal (data.gov.in), Central pollution control board (cpcb.nic.in) and Indian Institute of Tropical Meteorology (tropmet.res.in) etc.

National Information system for Climate and Environment Studies (NICES) is a multi-institutional program established at NRSC on 28 September 2012 with the major objective to generate ECVs over Indian region and disseminate or effective utilization across scientific community and public. Since its inception, NICES has disseminated various products under Bhuvan web-portal of National Remote Sensing Centre (NRSC) /ISRO (bhuvan.nrsc.gov.in) under Ocean, Terrestrial, Atmosphere and Cryosphere Categories. These products are available in different formats with different spatial resolutions ranging from 250m to 25Km and temporal resolution ranging from one day to one year. Visualization and metadata of the generated product is also possible in this website by selecting the required product. Metadata follows NSDI metadata standard version 2.0.

These portals have open data sets for climate studies and meteorological data for data downloads and data visualization. These datasets are of different spatial and temporal resolutions, different formats, and the way to access and analyze these datasets differ widely across portals and products.

Considering this, for the climate services to be effectively utilized by the intended users, these services should be discoverable, accessible, reusable and have scientific credibility to the data. As all the datasets have geospatial and temporal components associated with them, these datasets have to be treated as geospatial datasets with temporal characteristics. With this motivation, in this thesis work, we studied the current available standards that are suitable for web-based climate services and web based geospatial frameworks that are suitable for climate services. We also attempt to provide a framework using Open geospatial Consortium (OGC) standards and open-source tools.

Web based climate services is a vast topic and involves multidisciplinary working groups and technical know-how in various domains. Although in our study, we attempt to provide an overall framework, we focus more on the Internet of Things and in-situ instrument data and its role in the proposed framework.

1.3 Objectives

The main objective of this thesis is mentioned below

I. Study the current available software, standards and frameworks for WebGIS and web-based climate services

II. Identify the requirements for the framework for Geospatial data services including standards and its implementations

III. Design and development of the Geospatial data services framework

IV. Implement case studies demonstrating the effectiveness of the framework design.

Considering the large volume of datasets with different spatial and temporal resolutions, the major challenges are like that of challenges involved in big data applications. Scalability, Performance, and processing of these datasets are to be considered along with the functional requirements of the framework. Additionally, as these datasets have spatial component, additional requirements on handling spatial queries and spatial data storage formats and databases are to be considered.

1.4 Outline of the thesis

In Chapter-1: Introduction, we discuss the motivation for this thesis work, Climate science context, objectives, and challenges. In Chapter-2, we discuss the different requirements, OGC standards, open-source software, and climate sources. In Chapter-3, considering the framework requirements and the suitable OGC standards, we design web-based geospatial services framework. We also discuss the climate research studies carried out to understand the general steps involved, problems observed and further discuss the limitation of the research and how this can be improved using a unified web-based spatial services platform.

Finally, we present our conclusions in Chapter-8 along with the future work that is required to improve the architecture of the framework designed.

Chapter - 2: Literature Review

Several studies are carried out towards web based geospatial data services framework. Kim, M et al. [19], design such framework using then announced OGC web services standards. Gonj, J et al. [20], discusses the technology and standards on spatial data sharing with focus on OGC web services standards. Bhuvan, (bhuvan.nrsc.gov.in), which is an Indian Geo-Platform of ISRO, is an implementation of web based geospatial data services. This geo-portal has various spatial datasets for visualization, downloads and analysis and hosts web geospatial applications in multiple domains.

Global Climate Observing system (GCOS) is initiated by World Meteorological Organization to ensure that the observations and information required to address climate-related issues are collected and distributed, which is discussed in detail in 2.4.1. Copernicus Climate Change Services (C3S) is another large scale implementation of web based geospatial data services which is discussed in detail in 2.4.2. This web portal is a valuable resource for Global Framework for Climate Services (GFCS). Further, Giuliani et al. [6] reviewed geospatial solutions to share climate data.

Considering this, in this chapter, we discuss the currently available open-source GIS solutions, geospatial standards and study the available frameworks and web portals that provide such services with drawbacks.

2.1 Software components of Web GIS Solutions

Data publishing, Data visualization and Data Processing are three major requirements of Web GIS solutions. In this section we discuss the various available open-source solutions that are suitable for web GIS applications.

2.1.1 FROST Server

FROST server is the first opensource implementation of OGC SensorThings API (STA). It provides a unified way to connect IoT devices, applications, and data over internet. Following are the features of FROST Server.

- Supports SensorThings API sensing and tasking
- Data Ingestion: Over HTTP and MQTT
- Data encoding: STA output, GeoJSON and CSV
- REST interface

- Based on OASIS OData URL patterns and query options
- OpenAPI Plugin

The architecture of FROST server is shown in the below figure.

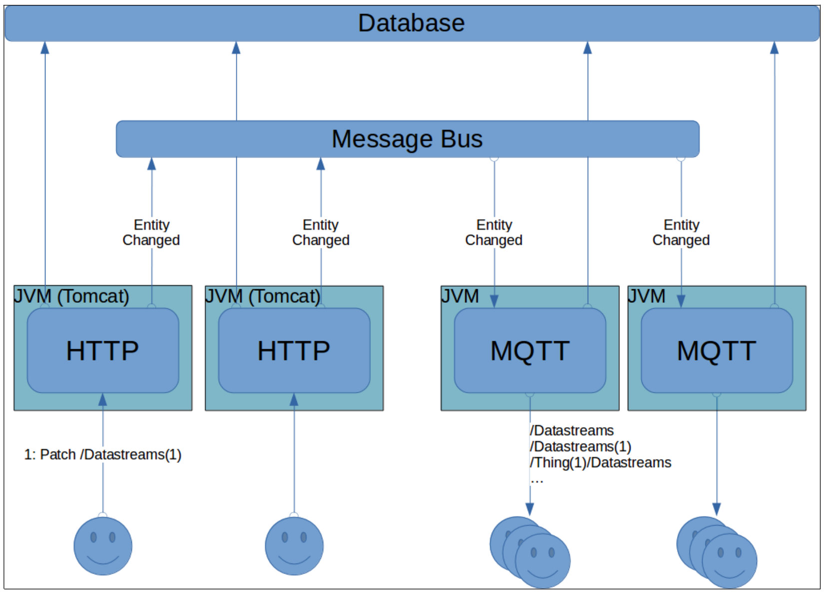

Figure 1: FROST Server Architecture

FROST Server is developed in Java, uses Tomcat server and PostgreSQL database. FROST server is planning to implement SensorThings API Plus (STA+) in its next version. 52°North STA is another opensource solution that can be considered for this purpose.

Frost Server currently supports only PostgreSQL database. Considering the volume of sensor data, it may not be the correct choice as relational databases doesn't perform well for large datasets.

2.1.2 Pygeoapi

Pygeoapi is an opensource implementation of OGC APIs, which is currently in the development stage. OGC APIs are discussed in detail in section 2.5. Pygeoapi currently supports OGC API Tiles, OGC API Features, OGC API Records, OGC API Coverages, OGC API Processes, OGC API Environmental Data Retrieval (EDR), and Spatio-temporal Asset Catelog (STAC). This also supports SensorThings API as data input for OGC API Features. OGC API Maps and OGC API GeoVolumes are in the development stage now and are not implemented in pygeoapi.

Pygeoapi is currently in development stage and lots of features are to be implemented and aligned with the latest OGC API releases.

2.1.3 GeoServer

GeoServer is an opensource implementation of OGC Web Services standards WMS, WFS, WMTS, WCS, and WPS. It also implements CSW. This is one of the widely used opensource software for sharing geospatial datasets over the internet, and the standards supported are widely used. GeoServer currently implements OGC API services as plugin. This includes OGC API – Features, OGC API – Maps, OGC API – Coverages, OGC API Records and OGC API Processes. Most of these features are in development stage.

2.1.4 MapServer

MapServer is an opensource implementation of OGC Web Services standards WMS, WFS, WMTS and WCS. This is one of the widely used opensource software for sharing geospatial datasets over the internet, and the standards supported are widely used. MapServer currently implements OGC API – Features, but this is currently in infancy.

2.1.5 Desktop GIS Software

Desktop GIS solutions are widely used by GIS professionals. Hence, the services provided by the framework should be capable of being used by the widely used desktop GIS software. QGIS, ArcGIS, Grass etc are some of the widely used desktop solutions. These software support OGC web services and some of the OGC API components. Full support for OGC API is in the roadmap of these software.

2.1.6 Spatial data Processing tools and libraries

Python and R are the used by the geospatial community for spatial data processing. These programming languages have various libraries that support spatial data for processing, analysing and visualization. Geospatial Data Abstraction Library (GDAL) is also widely used library for spatial data.

Openlayers, leaflet and mapbox are libraries for creating and displaying interactive maps on web. Hence, the design of the framework will benefit by supporting these tools and libraries.

2.1.7 Open Data Cube

Considering the large growing datasets and the worldwide users, it is suggested to use Big Data technologies for this framework. All the geospatial datasets, instead of using directly, can be used using Open Data Cube. It is an opensource project for managing large volumes of earth observation datasets. It is one of the data providers of pygeoapi as a plugin, and hence the data published using open data cube can be disseminated using pygeoapi.

The below table, table-1, summarizes the software components discussed.

Software module	**Data Input**	**Output**	**Suitability**	**OGC Standard**
FROST Server	JSON over HTTP and MQTT	STA output format, CSV, GeoJSON, OPEN API	Sensing and tasking of IoT Sensor data. Can also be used for instrument data, citizen centric data	SensorThings API
GeoServer and MapServer	Many spatial dataformats including Shapefile, GeoTIFF, CSV, KML, PostGIS, GeoPackage etc	XML output based on OWS standards	Vector and raster datasets visualization, analysis and process. Uses legacy XML based outputs	OGC Web Services OGC API in development
PygeoAPI	Major spatial data formats incuding shapefile, GeoTIFF, CSV, KML, PostGIS, NetCDF, GeoJSON, Zarr, elasticsearch, GeoPackage, Open data cube STA etc	JSON based on OGC API standards	Vector, raster, IoT datasets visualization, analysis and process. Storing metadata information. Follows OpenAPI standards and follow recent web technology trends.	OGC API
Desktop GIS	Major spatial data formats and OGC Web Services	Maps and major spatial data formats	Desktop level geospatial visualization and analysis.	Supports OGC web services. Support for OGC APIs in development
Open data cube	Major spatial data formats including shapefile, GeoTIFF, CSV, KML, PostGIS, NetCDF, HDF, GeoJSON, Zarr, GeoPackage, COG etc	Major spatial data formats	Manage and analyse large geospatial datasets	Can be integrated with OGC APIs

Table 1: Comparison of software components

2.2 Database considerations for Web-GIS solutions

2.2.1 PostGIS

PostGIS is an extension of PostgreSQL with capabilities that support spatial datatypes and functions. It is the used spatial database that supports both vector and raster data. It is simpler to use and to perform queries as it supports SQL. It is a relational database and hence supports relations between entities if used correctly. It is easier to use with web applications and frameworks. However, it is difficult to scale for supporting large datasets and the performance also degrades with complex queries made on large datasets. Considering these features PostgreSQL with PostGIS extension is a good choice for the framework to provide reliable datastore and manage relations between entities. This can be used along with a scalable, high performing NoSQL database with fast data retrieval to compensate for the cons of PostgreSQL.

2.2.2 Elasticsearch

Elasticsearch is a NoSQL document store cum search Engine. The following are the major features of Elasticsearch.

- RESTful API: RESTful API using JSON over HTTP. Client libraries are available for many programming languages (python, java, perl, ruby, JavaScript, PHP etc). It can be integrated with Hadoop and Kafka.
- Real Time Analytics: It combines the speed of search instances with the power of analytics for better decision making.
- Ease of Data Indexing: Schema-free and document-oriented as structured JSON and automatically detects the data structure and types.
- Many search options
- Resilient Clusters: Failure detection. Reorganizes and re-balance data automatically
- Speed: Executes complex queries extremely fast
- Distributed approach: This solution is highly scalable, i.e., the solution that is implemented currently in one server and possible to run as a cluster in multiple servers with load balancing, fail-over and high availability features. Routing and re-balancing operations are done automatically when new documents are added

Considering these features, Elasticsearch is a good choice for the framework as it has the capacity to handle vast amount of data streams and can scale in proportion to the available and streaming datasets.

Kibana is open-source software that is used for visualizing and performing real time queries on Elasticsearch data. The data from Elasticsearch can be visualized using histograms, line graphs, pie charts, heat maps, tile maps, data tables, metrics etc. It also has the capability to visualize geo-spatial data and create heat maps. It can perform advanced time series analysis with user interfaces. Queries, transformations, and visualizations can be performed using powerful and easy to learn expressions. Additional features like combining data with graph exploration anomaly detection, unsupervised machine learning and role-based access for dashboards are also available.

2.3 OGC standards for Web GIS

The Open Geospatial Consortium (OGC) is an international consortium of more than 500 businesses, government agencies, research organizations, and universities driven to make geospatial (location) information and services FAIR - Findable, Accessible, Interoperable, and Reusable. Considering this, we studied various OGC standards and their applicability for this framework.

In the current work, we focus more on the data ingestion and utilization mechanisms of IoT data, in-situ instrument data and citizen data. So, the standards suitable for this purpose will be discussed in detail. We also discuss the OGC standards for other component briefly.

First, we start by discussing the major OGC standards that are suitable for this framework and then we discuss their role in the overall framework.

2.3.1 OGC Web Services (OWS)

OGC web services (OWS) are a group of web service standards that are developed to provide standard ways to access geospatial information over internet. It includes Web Service common, Web Map Service (WMS), Web Feature Service (WFS), and Web Coverage Service (WCS). [14]

2.3.1.1 Web Service Common

OGC web Service Common is an OGC standard that specifies aspects of multiple OWS implementation standards which include Web Map Service (WMS), Web Feature Service (WFS), and Web Coverage Service (WCS). It covers request and response operations and their contents, request and response parameters and encodings.

2.3.1.2 Web Map Service (WMS)

WMS interface standard defines a standard for visualizing geospatial map images over internet using HTTP interface. This interface provides map images in the image formats like JPEG, PNG etc as a http response based on the HTTP request that defines the geographic layer name, area of interest, style etc from various geospatial databased (like PostGRESQL) or from geospatial file of formats TIFF, IMG etc. These images can be displayed in a browser application using various opensource libraries like leaflet, openlayers, mapbox etc.

It also supports discovery operation and enables the service discoverable over internet.

Major Open-source implementations: MapServer, GeoServer.

2.3.1.3 Web Feature Service (WFS)

WFS is a standard that defines a way to create, modify and exchange geographic information at the feature and feature property level. Unlike the WMS, which provides map as static file, WFS provides access to geographic information and allows query operations on it. It also allows access to features for locking operations to modify or delete features and transaction operations to create, modify, replace, and delete the underlying data store.

It also supports discovery operation and enables the service discoverable over internet

Major Open-source implementations: MapServer, GeoServer.

2.3.1.4 Web Coverage Service (WCS)

WCS is a OGC standard that defines a way to exchange “coverages” over internet in the forms that are useful for client-side rendering. As defined by OGC a coverage is a “space-time varying phenomenon” such as 1D sensor time series, 2D remote sensing imagery, 3D x/y/t satellite image time series and x/y/z geophysical data, as well as 4D x/y/z/t atmospheric and ocean data. Like WFS and WMS, WCS also allows query operations based on spatial constraints to fetch the required information and discovery operation that allows it to be discoverable to other users. However, instead of responding with static maps like WMS, WCS provide data with detail description and original semantics enabling users to utilize this information for geospatial applications and modeling applications directly. In contrast to WF, which provides discrete geospatial features based on user queries, WCS provide coverages as defined above.

Major Open-source implementations: MapServer, GeoServer.

2.3.1.5 Web Map tile service (WMTS)

WMTS is a standard that defines distribution of geospatial maps over internet. It provides a complementary approach to WMS for tiling maps. As opposed to WMS, which provides maps as dynamic data based on the user queries, WMTS provides fixed map tiles which are fixed at the server side by fixed spatial constrains (bounding box and resolution). This allows or faster response, as the server just servers the already available files over internet. Like WMS, WMTS is also discoverable over internet.

Major Open-source implementations: MapServer, GeoServer.

2.3.1.6 Web Processing Service (WPS)

WPS Interface Standard provides rules for standardizing how inputs and outputs (requests and responses) for geospatial processing services, such as polygon overlay. The standard also defines how a client can request the execution of a process, and how the output from the process is handled. It defines an interface that facilitates the publishing of geospatial processes and clients' discovery of and binding to those processes. The data required by the WPS can be delivered across a network or they can be available at the server.

Major Open-source implementations: extension for GeoServer, pyWPS

2.3.1.7 OWS Common Security standard

This standard, published in 2019 January, is defined to allow the implementation of Information Assurance Controls and to advertise their existence in an interoperable way with minimal changes to existing implementations of OWS deployed on HTTPS. It uses XML elements that are already part of the Capabilities document structure ensuring that the other OWS implementations need not be changed.

2.3.2 OGC Catalogue Service

OGC Catalogue Services for Web (CSW) defines interface standards to publish and access digital catalogues of metadata for geospatial data and services. These catalogues can be queried based on the metadata attributes and XML standard is used for interface.

These services are required for data discovery.

2.3.3 OGC API

OGC API is a family of OGC standards that are developed to accommodate the current trends of transferring data over internet and transform them to the web-based access of geospatial data. These standards are built upon the OGC Web Services standards WMS, WFS, WCS, WMTS and WPS after considering the years of experience obtained from them It also focuses resource-oriented architecture considering the modern web development practices. Modularization and general alignment with the web practices are the key goals for developing these standards.

These standards are termed as “building blocks” that can be bundled together for creating a geospatial application based in the requirement.

2.3.3.1 OGC Web Services and OGC API

OWS standards based on the then available technologies have historically implemented a Remote-Procedure-Call (RPC) architectural style using Extensible Markup Language (XML) over HTTP protocol. Currently, resource-oriented architecture like RESTful API is being used by various modern web applications and are competing against traditional service-oriented architectures like RPC. [15] With this in view, OGC API family of standards are formed to define Web-APIs that follows this resource-oriented architecture and in compliance to W3C best practices for sharing Data on the Web.

Because of the RESTful approach, OGC API implementations are not backwards compatible with OWS implementations. However, OGC APIs and their functionalities can be mapped to OGC web services implementations.

The building blocks that form the OGC API family of standards are discussed in detail below.

2.3.3.2 OGC API - Common

OGC API Common is currently a draft document that specifies the foundation upon which the OGC APIs are to be built. It specifies API approach to services that follow resource-oriented architecture practices. It has two parts OGC API – Common – Part 1: Core and OGC API – Common – Part 2: Geospatial data, whose draft versions are released in August 2021 and June 2022 respectively. This standard help OGC APIs serve as reusable building blocks.

The purpose of Part 1: Core standard (http://docs.opengeospatial.org/DRAFTS/19-072.pdf) is to define the fundamental rules for the building blocks. Although this standard doesn’t specify fixed format for the API, Open API 3.0 is recommended to be considered for the API designs.

The purpose of the Part 2: Geospatial Data Standard (http://docs.opengeospatial.org/DRAFTS/20-024.pdf) is to provide a means of organizing geospatial datasets and to define operations for the discovery and selection of individual datasets.

2.3.3.3 OGC API - Processes

The OGC API - Processes standard is built upon the OWS standard WPS. This has one part, OGC API – Process – Part 1: core. This standard is approved in August 2021. Like WPS, this standard supports execution of processes and metadata retrieval. Based on the usage, these processes can combine multiple types of geospatial datasets with well-defined algorithms to produces new geospatial datasets. It wraps computational processes into executable processes, which can be offered as a Web API based on client request.

Open-source Implementation: pygeoapi, GeoServer, PyWPS

The open-source implementation of OGC API – Processes pygeoapi implements this functionality by providing a plugin architecture. This is a python module that allows clients/developers to implement their custom processing workflows based on their requirements using Python.

Pygeoapi implements OGC API Processes in two modes: synchronous mode (Online processing) and asynchronous mode (Offline processing). The synchronous mode is for the tasks that are to be executed in real time and asynchronous mode is for the tasks that need more time to execute and hence can be scheduled.

When asynchronous mode is selected, the server responds immediately with a reference to the job. The user can then check the status of the job by polling the server.

Although pygeoapi implements OGC API – Processes as a plugin, the ability to handle large datasets and suitability for more advanced task managing capabilities needs to be added. Currently pygeoapi uses TinyDB for job management.

Open-source Implementations: pygeoapi

2.3.3.4 OGC API - Maps

The OGC API - Maps has a single part, OGC API - Maps Part 1: Core specification (http://docs.ogc.org/DRAFTS/20-058.pdf). This defines a Web API for requesting map images over the Internet and is based on the OWS standard WMS.

Like WMS, OGC API - Maps allows clients to request images based on the client parameters such as size and coordinate reference systems. It also allows discovery operations to make this service discoverable over internet.

Open-source Implementations: pygeoapi, GeoServer

2.3.3.5 OGC API – Environmental Data Retrieval

The OGC API - Environmental Data Retrieval (EDR) standard (https://docs.ogc.org/is/19-086r4/19-086r4.html) provides a family of lightweight query interfaces to access spatio-temporal data resources based on the query parameters that define Position, Area, Trajectory etc. This standard is approved in March 2021.

This API standard is formed with the goal of making data access easy over simple web interface and to achieve data reduction by providing the data that is only needed for analysis by providing a simple set of easy-to-use queries. The major use case kept in mind for forming this API standard, is to utilize environmental data for weather forecasts and applications that require a small subset of the total data available. This API help to simplify the design of systems making it easier to build robust and scalable infrastructure.

Open-source Implementations: pygeoapi, GeoServer

2.3.3.6 OGC API - Tiles

OGC API - Tiles is a draft standard (http://docs.ogc.org/DRAFTS/20-057.pdf) that defines a standard API to retrieve geospatial information as tiles. This is based on OWS standard WMTS. It has four parts: OGC API-Tilespart-1: core, OGC API-Tiles part-2: Tile Matrix sets, OGC API-Tiles part-3: info and OGC API-Tiles part-4: multi-tile. Currently only part-1 is in draft stage. All the other parts are still under discussion.

Open-source Implementations: pygeoapi, GeoServer

2.3.3.7 OGC API - Features

OGC API - Features is a standard that aims to create API building blocks to create, modify and query features on the Web. This standard is based on OWS standard WFS. It has four parts: OGC API – Features - Part-1: Core, OGC API – Features - Part-2: Coordinate Reference Systems by Reference, OGC API – Features - Part-3: Filtering and Common Query Language, and OGC API – Features - Part-4: Create, Replace, Update and Delete.

OGC API – Features - Part-1: Core (https://docs.ogc.org/is/17-069r4/17-069r4.html) is approved in May 2022. This specifies the core capabilities of Features PI in fetching features whose geometries are represented in Coordinate Reference System (CRS) WGS 84. This standard supports discovery, modifying and creating features, complex models, perform queries etc.

OGC API – Features - Part-2: Coordinate Reference Systems by Reference Standard (https://docs.ogc.org/is/18-058r1/18-058r1.html) is published in May 2022. This extends Part 1: Core with the ability to use CRS identifiers other than WGS 84.

OGC API – Features - Part-3: filtering and Common Query Language (CQL2) is a draft standard (https://docs.ogc.org/DRAFTS/19-079r1.html) that defines query parameters for the filtering and the common query language.

OGC API - Features - Part 4: Create, Replace, Update and Delete Standard (https://docs.ogc.org/DRAFTS/20-002.html) is a draft standard that defines the behavior of API allows resource instances to be added, replaced, modified and/or removed for a collection. Although, currently this API is developed as part of OGC API Features, it is being written as a generic standard which can be applied to other resource types. Hence, it is possible, that a major part of this standard may be moved to OGC API – common, retaining the relevant information in OGC API – Features.

Pygeoapi is the complete implementation of OGC API – features. It supports csv, Elasticsearch, MongoDB, GeoJSON, OGR, PostgreSQL, SQLiteGPKG and SensorThings API data providers.

Open-source Implementations: pygeoapi, GeoServer

2.3.3.8 OGC API - Records

OGC API Records is a draft standard (https://docs.ogc.org/DRAFTS/20-004.pdf)That aims to make a resource discoverable by providing required information about the resource. In this context, resources are things that would be useful to a user or developer, such as features, coverages, tiles / maps, assets, machine models, services, widgets, etc. This helps to make the resources discoverable over internet and provide a catalogue of datasets and APIs to access.

Open-source Implementations: pygeoapi, GeoServer

2.3.3.9 OGC API - styles

OGC API Styles is a draft standard (https://docs.ogc.org/DRAFTS/20-009.html) that defines architectural styles for rendering maps, features, or coverages over internet.

The current status of the OGC APIs is shown in the below figure.

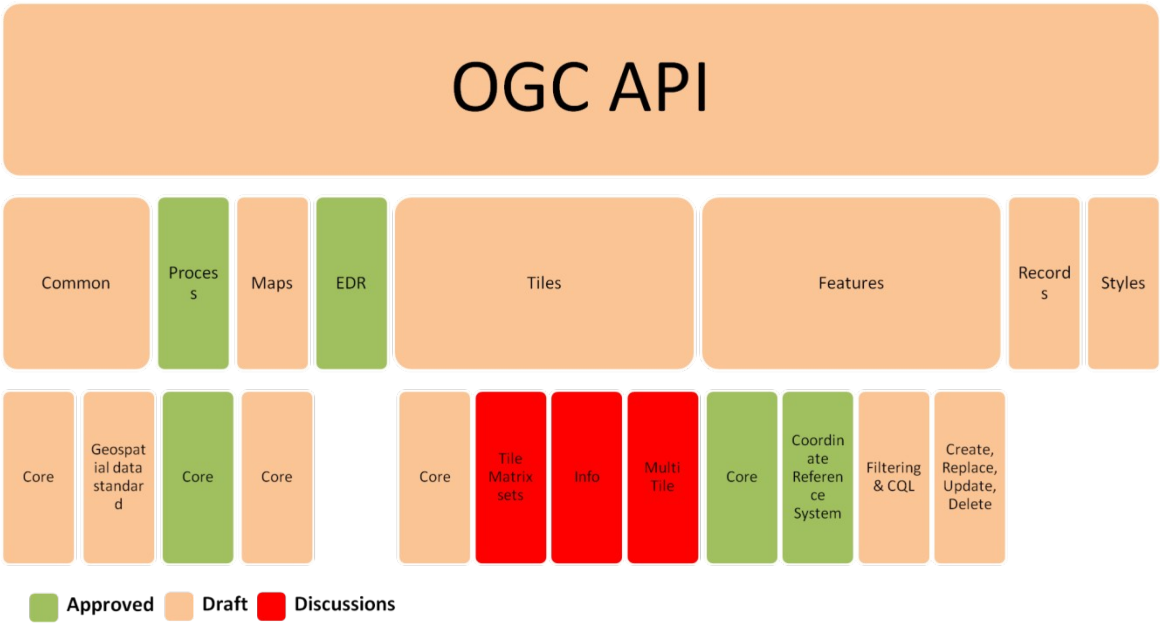

Figure 2: OGC API's current status

2.3.4 Sensor Web Enablement (SWE) working group

This working group works on various best practices for sharing sensor data over internet. The major standards under this working group are given below.

- **Observations & Measurements Schema (O&M)** – Standard models and XML Schema for encoding observations and measurements from a sensor. O&M standard is widely used as data model for many geo-spatial standards. [13]
- **Sensor Model Language (SensorML)** – Standard models and XML Schema for describing sensors systems and processes; discovery of sensors, location of sensor observations, processing of sensor observations, and listing of properties.
- **Transducer Markup Language (TML)** – The conceptual model and XML Schema for describing transducers and supporting real-time streaming of data to and from sensor systems.

- **Sensor Observations Service (SOS) -** Standard web service interface for requesting, filtering, and retrieving observations and sensor system information.
- **Sensor Planning Service (SPS) –** Standard web service interface for requesting user-driven acquisitions and observations.
- **Sensor Alert Service (SAS) –** Standard web service interface for publishing and subscribing to alerts from sensors.
- **Web Notification Services (WNS) –** Standard web service interface for asynchronous delivery of messages or alerts from SAS and SPS web services.

2.3.5 SensorThings API

The OGC SensorThings API provides an open, geospatial-enabled and unified way to interconnect the Internet of Things (IoT) devices, data, and applications over the Web. In contrast to SWE SWG, SensorThings API focuses more on low power computing devices.

It provides two major functionalities.

1. Sensing part: It provides a standard way to manage and retrieve observations and metadata from heterogeneous IoT sensor systems. Version 1.1 of this part is published on August 2021. [16]
2. Tasking Part: It provides a standard way for parameterizing of task-able IoT devices, such as sensors, actuators, drones or even satellites. Version 1.0 of this part is published on January 2019. [17]

This standard is built on the OGC SWE standards, including the ISO/OGC observation and measurement data model and designed specifically for the resource-constrained IoT devices and web community. It utilizes a REST-based API approach modeled on the OData standard

Open-source Implementations: FROST Server, 52°North STA

2.3.6 SensorThings API Plus

SensorThings API plus is a new standard that is being discussed which add user related information to SensorThings API data model. Although, STA captures the sensor information and preservers its relations with the entities, there is no mention of the information related to the user (citizen) who uploaded the data and the project information.

In the context of citizen science, this is highly useful. With this requirement STA Plus standard is first proposed, and also used in citizen science applications

Open-source Implementations: FROST new version is planned

2.3.7 Spatio Temporal Asset Catalog (STAC)

The Spatio temporal Asset Catalog (STAC), although not an OGC standard is worth considering for the design of this framework. This family of specifications aims to standardize the structure and queries of spatio-temporal asset metadata.

This API implements and extends OGC API Features standard, and it is in the roadmap to make this standard an OGC API standard.

Open-source Implementations: pygeoapi

2.3.8 Data Quality DWG

This Domain Working Group is established by OGC to describe an interoperable framework/ model for sharing high quality geospatial information. It is currently in discussion stage and modification of ISO 19157 standard to suit the requirements of geospatial need is being discussed. The goal of this DWG is to provide a machine-actionable register of data quality measures.

2.3.9 Spatial Data on web working group

This working group works in defining the best practices for providing spatial data on web. The best practices document is published in May 2022 (OGC 15-107). The following are the best practices summary mentioned in the document.

- Use globally unique persistent HTTP URIs for Spatial Things
- Make your spatial data indexable by search engines
- Link resources together to create the Web of data
- Use spatial data encodings that match your target audience
- Provide geometries on the Web in a usable way
- Provide geometries at the right level of accuracy, precision, and size
- Choose coordinate reference systems to suit your user's applications
- State how coordinate values are encoded
- Describe relative positioning

- Use appropriate relation types to link Spatial Things
- Provide information on the changing nature of spatial things
- Expose spatial data through 'convenience APIs'
- Support requesting and returning geometries in a specific CRS
- Include spatial metadata in dataset metadata
- Describe the positional accuracy of spatial data
- These best practices need to be followed in proposing the web-based climate services framework.

2.3.10 Climate Resilience DWG

The OGC standards are the widely used standards in the geospatial community. Recently, Climate resilience initiative has been taken up by OGC to "accelerate Climate Change Data Interoperability and Integration with Numerical Model Data, Earth Observation Data, Social Science Data, and non-climatic Data". The overview of Climate Resilience Pilot as explained by OGC is shown in the below figure.

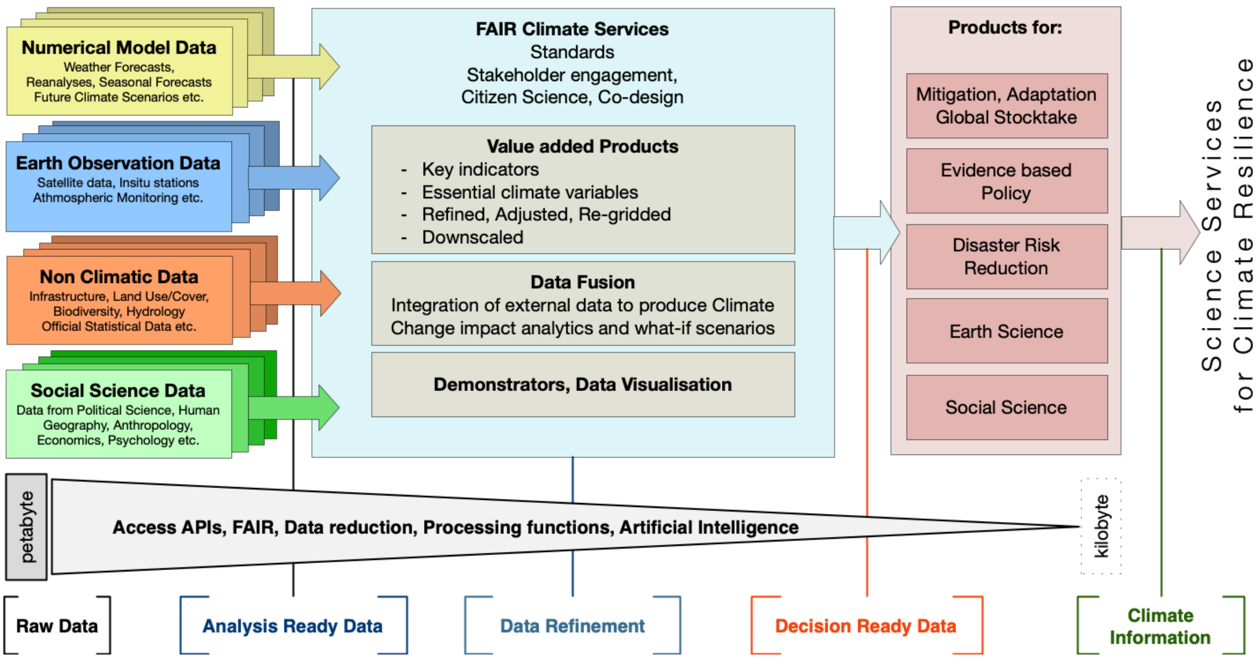

Figure 3: Overview of the Climate Resilience Pilot Processes (source: https://portal.ogc.org/files/?artifact_id=99200)

2.3.11 Metadata and Catalog DWG

This domain working group is formed in 2018 with the objective to provide metadata standards and catalog standards [18]. However, this working group is dormant, and no document is currently published or in discussion under this working group. Considering this ISO 19115: geographic information – metadata can be used for metadata.

2.4 Web based geospatial data services frameworks and solutions

Global Climate Observing system (GCOS) is initiated by World Meteorological Organization to ensure that the observations and information required to address climate-related issues are collected and distributed, which is discussed in detail in 2.4.1. A Global Framework for Climate Services (GFCS) is proposed by World Meteorological Organization [5] to improve the availability and use of climate information for decision making. Copernicus Climate Change Services (C3S) is a valuable resource for GFCS that implements a web based climate services framework covering the major requirements of web based geospatial data services. This is discussed in detail in 2.4.2.

2.4.1 GCOS:

GCOS published The Global Observing system for Climate: Implementation needs in 2016. In this implementation plan, climate products should address

a. Raw Observations:
    a. In-situ observations, remote sensing observations from optical, radar satellites and ground-based weather radars.
    b. Raw observations required so they can be re-processed if required
b. Documented traceability to SI standards:
    a. Metadata are required to record the instrument used, changes and maintenance, calibration intervals, uncertainty and procedures and any other information needed to fully describe the measurement process.
c. Data Recovery: historical data to usable format
d. Processed measurements: Raw observations to usable ECVs
e. Data Analysis: To provide spatially and temporally complete datasets
f. Archiving: To be archived for future climate analysis

GCOS also suggests that the climate data records (CDRs) must be openly accessible to users so that they can be easily and widely exploited.

a. Data Management

GCOS states that data management is of vital importance for the global climate observing system. This is required for proper data collection, data accessible to users for analysis and application. It is also to be ensured that the available data is of required quality and quantity necessary or climate studies, which will ultimately lead to decision making at global and national levels.

Following are the data management principles stated by GCOS.

a. Data access: Available to users on free and unrestricted basis. Open data policies

b. Metadata: The details and history of local conditions, instruments, operating procedures, data processing algorithms and other factors pertinent to interpreting data. This will ensure Improved access and discoverability of datasets

c. Quality Control: A key component of data management is close monitoring of the data streams. This includes ongoing monitoring of the flow of data and quality control of the observations. Data centres need to be adequately supported to perform data monitoring and quality control. This will ensure Long-term, sustainable, provision of timely data and improved data quality

d. Preservation: preservation of data or future generations. Climate observations need to be preserved and remain accessible for future use. Observations and metadata need to be archived indefinitely. Guidelines for loss prevention, including back-up of data and media transfer procedures must be put in place. Data and associated metadata held in data-management systems must be periodically verified to ensure integrity, authenticity and readability.

e. Discoverability: Data and all associated metadata have to be discoverable through portals, catalogues and search engines; data access and use conditions, including licenses, have to be clearly indicated. This will ultimately lead to increased access and use. To assist discoverability and to ensure correct citation of all data, each dataset needs a **unique permanent reference** that will identify each version.

b. Data integration and assimilation:

One of the requirements GCOS mentions is the need for processed datasets for climate applications. These products should be supported by the provision of ancillary information, uncertainty estimates, and evaluations against independent data and comparison with other alternate products. It is also required to continuously produce and develop improved data products. This involves in integrating the data available from various sources and provides a robust platform for data integration and assimilation.

2.4.2 Copernicus Climate Change Service:

Copernicus Climate Change Service (C3S) is hosted on URL https://climate.copernicus.eu/ by the Copernicus Earth Observation Programme of the European Union with a core objective to provide reliable high quality climate data. It is an important resource to the Global Framework for Climate Services (GFCS) and follows user requirements defined by the GCOS. It is a Climate Data Store (CDS) which provides information in the form of ECVs and derived climate indicators.

Major features of C3S are given below:

1. It hosts Climate Data Store (CDS) which contains climate data which can be downloaded using web interface and used in client applications using APIs as python modules.
2. It hosts Climate Data Store (CDS) toolbox, using which users can create applications in python and run them on the CDS infrastructure using the tools available through programming interface. The results can also be downloaded and shared.
3. It provides application gallery that has example applications built using CDS tools and APIs.
4. It provides user support by hosting knowledge base and user forum
5. Quality assurance is carried out for C3S data, tools, applications, and infrastructure
6. It collects user requirements and implements new tools based on it.
7. It implements quality of service (qos) that queues request and schedules and also protects the platform from denial of request attacks.
8. It hosts articles in climate science domain
9. All products and data are assigned a digital object identifier (DOI).
10. The data and products are described using the ISO19115 metadata record standard and are made available through the OAI-PMH and OGC-CSW protocols.

The following are the issues observed on C3S as reported by many users in user forums.

1. Problems with data access and downloads.

2. Problems reported for user created tasks taking too much time

3. Problems related to slow access

4. Problems related to non-working of tool(s).

The CDS architecture of C3S is shown in the below figure. The major components that are involved are explained below.

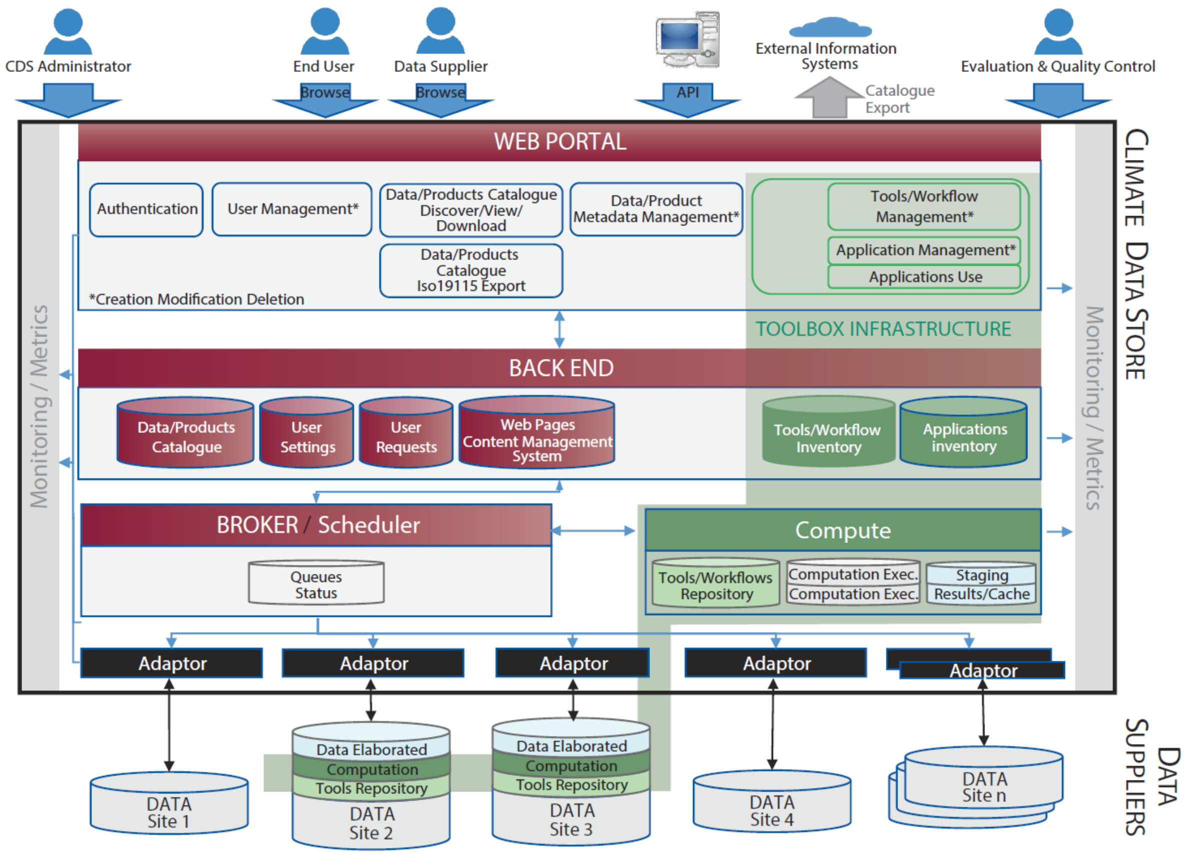

Figure 4: C3S Architecture (source: https://climate.copernicus.eu/climate-data-store)

1. **Web portal:**

This is where the users enter the platform for discovery of data products/tools, retrieve data, access toolbox, visualize and download data/products/results. It also contains user forum, knowledge base and articles related to CDS.

2. **Broker**

This component schedules and forwards data/compute requests to the appropriate destination using adapters. It also implements quality of service. For queuing requests, quality of service component considers the following.

- Pre-defined rules like giving low priority to very large requests

- Considers user profile and type of request
- Execution cost considering the resource utilization involved
- Limits multiple requests from users

The main aim of QoS is to make the services responsive. However, it is to be noted that many users are raising queries related to the slow performance of the system. This feedback is being considered for further improvement.

3. **Adaptors**

These components translate user requests to requests understandable to the CDS infrastructure

4. **Backend**

The backend contains various databases, such as a

- Web content management system for news articles
- A centralized catalogue that describes the data and products in the CDS
- Toolbox catalogue
- Quality control information
- User-related information

5. **Compute**

This component hosts computations that support data retrieval and execution of user requests for using tools given in toolbox.

6. **Toolbox**

It is a catalogue of tools that are classified to perform the following tasks.

- Basic data operations like computation of statistics, averaging, etc
- Workflows that combine the output of tools and feed this as input into other tools to produce derived results
- Applications, which are interactive web pages that allow users to utilize CDS for data analysis, computations, visualization and downloads.

7. **Application programming interface**

This component is a web-based API that allows users to interact with CDS for data operations and usage of tools.

C3S provides a good example for web based geospatial services by covering the major requirements like data visualization, downloads, data access and process through APIs. However, it doesn't use standards for all of these features making the data/tools less discoverable and difficult to use across platforms and other widely used GIS solutions, libraries and tools.

2.5 Climate data sources

Climate data is traditionally available from Earth observation data from satellites, aerial vehicles, and in-situ observation networks. Apart from that there is a rise in Internet of Things (IoT) contributing to climate and weather data using low-cost sensors. Citizen science data, which is the data collected from non-professional volunteers is also helpful in collecting climate data. Socio economic data also plays a key role for climate change studies. The following figure shows the major data sources for climate studies

.

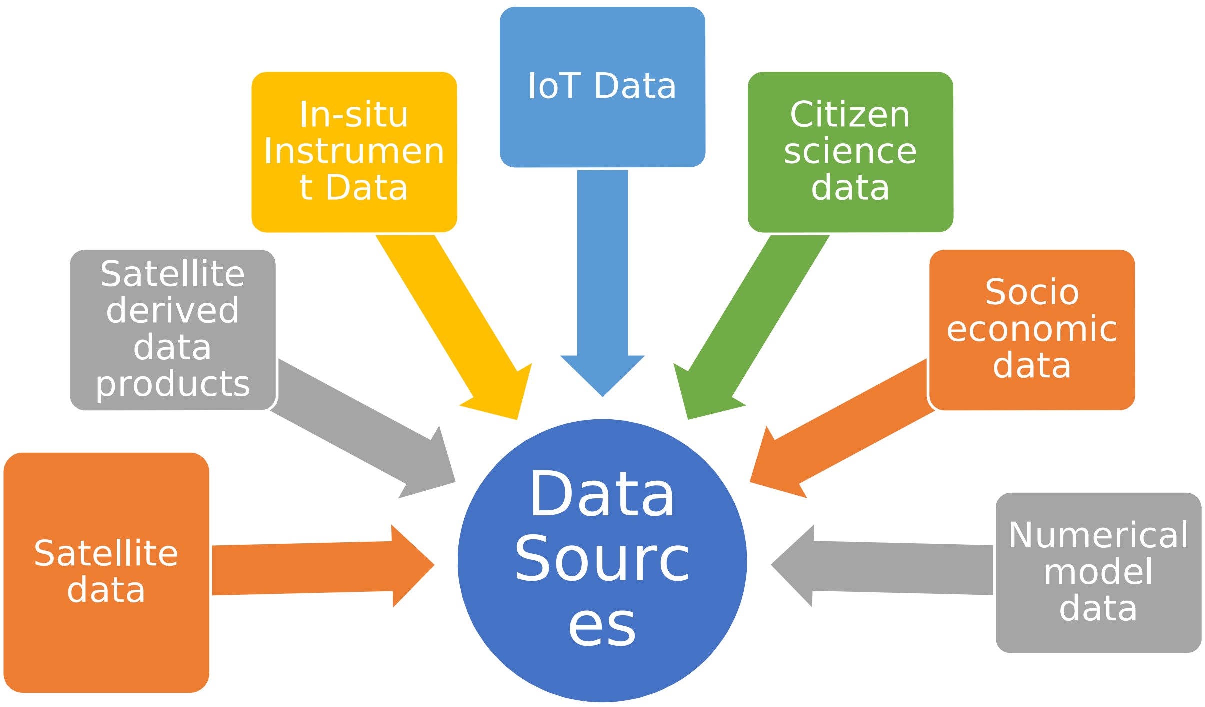

Figure 5: Climate Data Sources

2.5.1 Earth Observation Data

Earth Observation (EO) Data is the data collected using remote sensing technologies usually using satellite sensors and sensors embedded in Unmanned Aerial Vehicles (UAVs). EO derived products are one of the major sources of climate data and used in various applications for studying air pollution, air quality, climate and weather-related calamities like cyclones, dust storms, wild-fires etc. Landsat and Sentinel satellite series are examples of traditional EO programs. Most of these datasets are available for open access.

Large volume of data: The datasets generated by such programs are in tens of Terabytes per day.

Standardization: As these EO datasets are collected from years using similar sensors, their data collection, processing, and dissemination is more standardized. To facilitate interoperability of EO datasets, international organizations Committee on Earth Observation satellites, GCOS and group on earth observations are established. However, the datasets collected from different sensors are not always comparable. Along with this, the validation of these datasets is done by comparing with the instrument data that is available for certain places and hence the quality of the data depends on the availability of ground truth for validation.

Spatiotemporal Resolution of Data: The spatial, temporal and spectral resolution of EO datasets has improved dramatically over the past 50 years due to advancement in sensor technology. However, the spatial resolution is still insufficient to study city level parameters in detail.

Gaps in datasets: Although the increase in the number of deployed satellites and UAVs, and increase in their spatial resolutions, the EO derived products still lack in greenhouse gas emission data as it depends on the experimental algorithms and dependency on ground-based data. However, these gaps may be reduced in the future as new missions like NASA’s GeoCarb are planned to reduce data gaps in time and space. Apart from the government mission, there is a rise in utilizing small scale satellites and UAVs by private companies for environmental studies and greenhouse gas monitoring.

2.5.2 Instrument data:

In-situ instruments provide high quality climate datasets and are used or validating the satellite data. Some of the widely used atmosphere GHGs monitoring instruments are the Greenhouse Gas Analyzer (GGA) and Fourier Transform InfraRed (FTIR) Spectrometer, Isotopic analyzers and Mass Spectroscopy. These instruments measures surface and column level highly accurate CO_2 , CH_4 and H_2O concentration. These measurements are highly decisive for estimating the regional terrestrial regional source and sinks.

Radiosonde and Ozonesonde are balloon-based data, which collects the information on vertical meteorological parameters that are useful for the weather prediction analysis and Ozonesonde also provides vertical information on ozone profiles that is used to study the ozone concentration in the troposphere to stratosphere.

The following are some of the advantages and disadvantages of in-situ field data.

Standardization: There are many different types of sensors for these instruments. Hence, standardization is very less. The quality and accuracy of the data collected from these instruments are good and are very useful for validating satellite datasets. These datasets are collected usually as comma separated text (CSV) or text files depending on the instrument and manufacturer. So, there is a need to modify the data to usable and analysis ready datasets.

Spatial and temporal resolution: The datasets collected from stationary instruments usually have high temporal resolution and the data is collected for a few stations. The datasets collected from mobile instruments like UAVs usually have high spatial resolution but are carried at fixed times depending on the requirement. Some instruments like radiosonde and ozone sonde are one-time experiments carried out at fixed time and fixed locations and are more focused on getting altitude related information. As these instruments are expensive and large in size, it is not cost effective to use these instruments to collect data at different locations.

Data gaps: Most of these instruments are dependent on the environmental conditions and human operators. This cause huge data gaps in the collected datasets.

2.5.3 Internet of Things (IoT) Datasets:

IoT is a system of devices with computing capabilities that are interconnected over network. These devices can communicate with each other and transmit data to servers connected over network. Generally, IoT Solutions contains three major components: Sensor network and network gateway; Network Server for data collection; and Application layer. The role of the network server(s) is to collect, process and store the data from IoT sensors. The major challenge for the IoT data collection module is the ability to handle vast amount of data in real time. Ideally, the data collection module should be distributed in nature with high performance and consistency. Also, the solution proposed should be scalable in nature to cater to the changing requirements. Major requirements of IoT data storage are data integrity, consistency, high availability, high performance, and fault tolerance. Additional requirements like simple use interface for management and data visualization, availability of APIs to multiple programming languages, real time search and analytics, capability to store massive structured and unstructured data sets and capability to perform batch processing can provide additional benefit.

IoT solutions are very helpful for collecting climate data, and to understand climate change mitigation and impact. The following are some of the advantages and disadvantages o using IoT data for climate data collection.

Standardization: There are numerous sensors, gateways, and protocols in IoT applications. Hence, the standardization is currently very less in IoT applications as these sensors and gateways are provided by multiple organizations.

Spatial resolution: As the IoT devices are low cost and some are movable devices, high spatial and temporal resolution data can be collected. However, the spatial extent to which data collected is limited as most of the current IoT projects focuses on smaller spatial extent.

Data gaps: IoT solutions help in reducing the data components as the solution is easily transferable to the required region and times.

Large volume: There are millions of IoT devices available over internet. These datasets are of large volume.

Variety and velocity:

Combining EO and IoT data has its own set of problems.

The following table provides the comparison of the different climate data sources.

S.no	Data Source	Spatial Resolution	Temporal Resolution	Metadata	Key points
1	Satellite data	Low	High	Yes. Very good	Large spatial coverage Huge amount of data Gaps in datasets
2	Satellite-derived products	Low	High	Very good	Large spatial coverage Huge amount of data Changing algorithms Multiple versions and methods of derivation Quality parameters
3	In-situ instrument data	High	Very High	Mostly available	Increasing amount of data Standard methods High quality data No data standardization between instruments Low spatial coverage Can be used to validate satellite data and fill gaps
4	IoT Data	High/ Medium	Very High	Very poor	Increasing devices and hence increasing datasets Low quality data No data standardization Low spatial coverage depending on a number of devices. But can be scalable easily.
5	Citizen Data	High/ Medium	Very High	Very poor	Increasing devices and hence increasing datasets Low quality data No data standardization Low spatial coverage depending on a number of devices. But can be scalable easily. No reliability

Table 2: Comparison of different Climate Data Sources

2.5.4 Data formats

The widely used spatial data formats in climate research are GRIB, NetCDF [10] and HDF. These formats are portable, self describing and have metadata information included in them. Geopackage, an open format for geospatial information is an OGC standard [11].

The other widely used GIS data formats for raster data are

- GeoTIFF – TIFF variant enriched with GIS relevant metadata
- IMG – ERDAS IMAGINE image file format
- Cartesian coordinate system (XYZ) – point cloud
- Digital line graph (DLG) – a USGS format for vector data
- Geography Markup Language (GML) – XML based
- GeoJSON – a lightweight format based on JSON, used by many open source GIS packages usually over internet
- Keyhole Markup Language (KML) – XML based open standard (by OpenGIS) for GIS data exchange
- Spatialite – a spatial extension to SQLite, providing vector geo-database functionality. It is similar to PostGIS, Oracle Spatial, and SQL Server with spatial extensions
- Shapefile – a popular vector data GIS format, developed by Esri

Chapter - 3: Design of framework for spatial data services

In Chapter-2, we discussed the various opensource tools/software for handling geospatial datasets and the widely used PostgreSQL database for geospatial datasets. We also discussed Elasticsearch, its capability to handle spatial datasets and its performance and scalability features. We discussed OGC standards and concluded that both OGC Web Services and OGC API standards are suitable for handling spatial datasets but OGC API standards follow latest web technology trends. Suitability of SensorThings API and SensorThings API Plus for handling IoT and citizen datasets is also discussed. The different spatial dataset standards that the framework should support are also discussed. With this in view, we designed the framework and this chapter covers the framework requirements and design.

3.1 Framework Requirements

Considering the currently available frameworks and solutions discussed in detail, we have arrived at the functional and system requirements that are required for web based geospatial services framework. These are discussed in detail in this chapter.

Following are the requirements of web portal through which web based geospatial services are provided.

1. **User roles:** It should have a mechanism to classify users and user roles.
    a. The main users are **consumers** to utilize the framework and **suppliers** that provide data.
    b. The customers are again to be classified based on their expertise and requirements. Citizen users (Non-professional users), Decision makers and researchers are the proposed categories.
    c. The suppliers are again to be classified based on the type of data they are supplying. Satellite data providers, Satellite derived data product providers, in-situ instrument data providers, IoT data providers and citizen providers are major providers
    d. Proper User Authentication and Authorization should be available.
2. **Data Ingestion**: This should provide APIs and web-based tools in portal for the data suppliers to ingest data into the data centre. Types of data are discussed in data catalogue requirements below.
    a. The framework should provide data ingestion mechanism for various datasets mentioned in Chapter-2 of this document. This includes

Analysis ready datasets, data products that are generated using satellite data and validated thoroughly, datasets from IoT devices, datasets generated by instruments. It is to be noted the current framework is not designed to store raw satellite data and the satellite data is collected and usually stored in the respective data centers of the government space agencies. These requirements convert to following definitive points.

i. API for data ingestion of IoT data
ii. Provision to store and update satellite derived products and other analysis ready products as flat files or in database.
iii. API for data integration of in-situ instrument data.
iv. All the datasets should have proper metadata which includes variable measured with units, instrument used, spatial and temporal resolution, -calibration intervals if applicable, quality flags, uncertainty and procedures/algorithms and any other information needed.

3. **Data Catalogue**: A data store where all data available is visible
    a. It should contain links to all the dataset available.
    b. Data catalogue should categorize data into 5 classes:
        i. Satellite derived products
        ii. In-situ instrument data
        iii. IoT data
        iv. Citizen science data
        v. Other category. This can be social science data or any other data that doesn't belong to any of the classes
    c. Further categorization based on the type of product and its application need to be done
    d. Every dataset should have a unique Identifier and naming convention should be followed.
    e. It should have 'search' feature which allows search based on the product name, the ECV it corresponds to, ECV category (Ocean, Terrestrial, and Atmosphere), spatial and temporal resolution, spatial bounding box, time period, type of data and data quality standard etc.
    f. It should have tools to visualize the data, data quality parameters, metadata and download option.
    g. It should also have RESTful API information on how to access the

selected without downloading based on the required user parameters like resolution, bounding box, time period etc. Examples on how to use the API should also be given.

4. **Tool Catalogue**: List of online and offline tools with examples
    a. Tool catalogue should cater the needs of user who wish to perform online analysis based on the available tools and also to schedule tasks that required offline processing.
    b. The tools should provide data analysis functions that can be performed on the datasets available with clear examples.
    c. The tools that provide results within seconds are online tools and the result should be available for user as visualization and download.
    d. The tools that provide need significant time to be performed are offline tools and the result should be available for user as visualization and download in a separate offline tasks page after the completion of the task scheduled. Examples on scheduling offline tasks should be given.
    e. There should be a fixed timeout placed on all the tools (offline and online) and the same should be informed to the user.
    f. These tools should also be available as APIs for the user to use without coming to the portal. In case of sing APIs for these purposes, the results should be provided back to the user as defined in the tool box API in fixed data format (geo-JSON, NetCDF, grib etc)
    g. These tools should be discoverable and searchable
5. **User Forum:** It should provide a means for the users to get help required for effective usage of website and APIs in the form of chat tools, contact details, frequently asked questions, document and video tutorials etc.
6. **User Analytics:** To monitor the website and API usage of the user
    a. The information related to how any user uses the website and APIs should be recorded and analyzed including the workflow the user used for accessing datasets and tool sets.
    b. Any personal information like IP addresses etc that are recorded for this purpose should be mentioned to the user and his consent for the same should be obtained and recorded.
    c. Any misuse should be detected for taking corrective action.
    d. Based on the usage the recommended products/tools should be shown to the user.

7. **Maintenance:** The data centre and the applications are to be maintained from time to time to ensure that the web site works in all conditions
    a. The website should work under heavy user/work load.
    b. The performance of the website should not degrade
    c. The proper working of the all the workflows should be monitored in real time and any corrections required to be taken immediately.
    d. If a flaw is discovered in any of the workflows/datasets/tools, the same can be informed in the webpage and API as a message stating that it is being currently under maintenance.
    e. The hardware for this should also be monitored and maintained continuously.
8. **Security:** The website should follow strict security guidelines and perform time to time security checks, vulnerability assessments to keep up with the growing security threats.

3.2 Framework Design

In this chapter, we start by discussing the various components that formulate web-based Geospatial services framework, categorize them into sub-components, suggest what standards and open-source tools are suitable for these components.

The web-based Geospatial services are designed considering the user requirements and the OGC standards. Apart from the functional requirements, we also suggest open-source solutions that can aid the framework.

The total architecture is divided into four layers, namely: Data storage layer, low-level application layer, medium-level application layer, and high-level web application layer. In the high-level web application layer, we have the major functional requirements of the framework, as discussed in chapter 3. In medium-level application layer, we have the modules that implement web applications and application security modules to provide users with secure Web and REST API access. In the low-level application layer, we have core geospatial software that supports and implement the basic components that are required to discover, visualize, access, and process features using either opensource solutions or in-house-built solutions. The data storage layer should store all the spatial datasets from different sources in flat files and GIS databases. Implementation of Open Data Cube (ODC) is also beneficial as we will be dealing with big geospatial datasets.

3.2.1 High-level web application layer:

This layer has five major components: Data Ingestion, Data Catalogue, Tool Catalogue, Geospatial Applications, and User Interaction. The OGC Standards that can be used for this platform are shown in Figure 3.

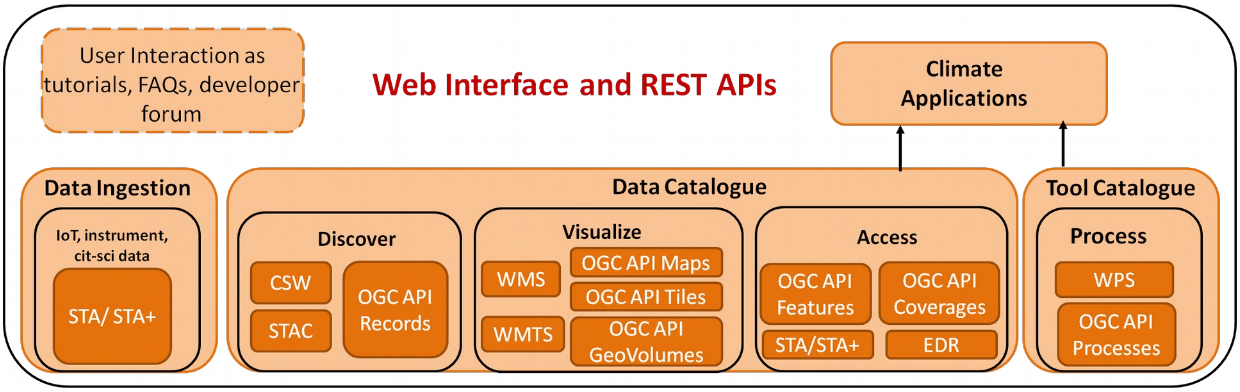

Figure 6: Web components and OGC standards

3.2.1.1 Data ingestion:

This provides a mechanism for data suppliers to securely share their datasets using the web user interface and REST APIs. The mechanism to enable data suppliers to ingest data is shown in Figure 4.

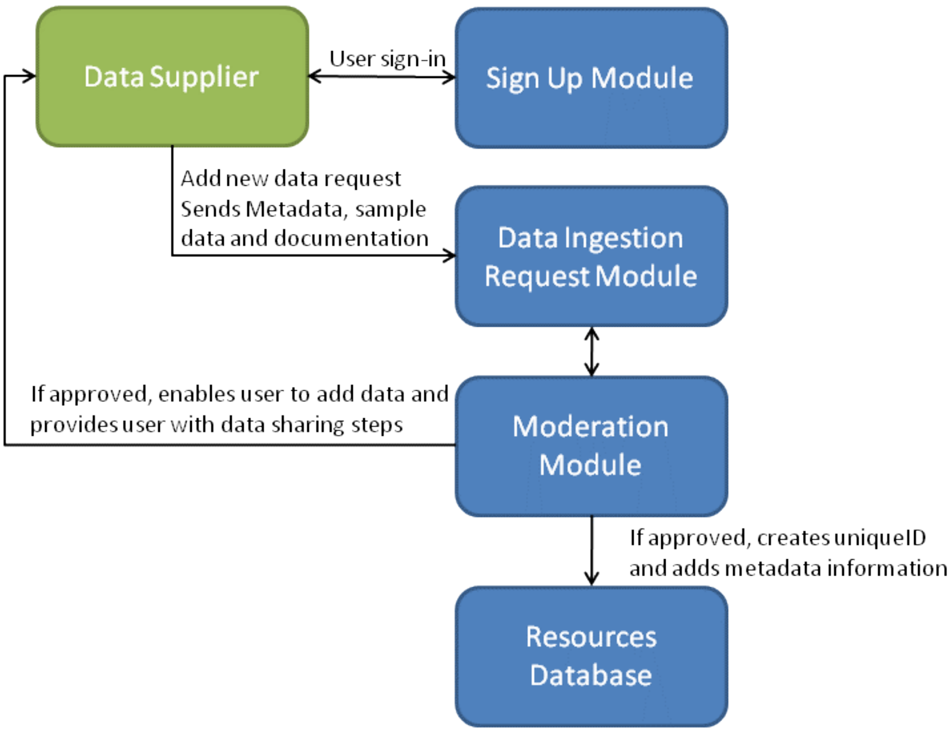

Figure 7: Mechanism to enable data suppliers to Ingest Data

3.2.1.2 Data Catalog

It is a collection of datasets that are available on the platform. The datasets should be published on the platform after thorough validation and should have unique identifiers, metadata, quality data, description, and technical papers associated with them. Major features of this component are search, visualization, data access, and data download, which should be available from web UI and secure REST APIs. All the hosted datasets should have a disclaimer shared by the data supplier along with the ownership details, which should be part of metadata information.

3.2.1.3 Tool Catalog

It is a collection of tools that are developed by the data scientists and domain specific scientists. These tools should be available for usage from a web interface with a code editor and from secure REST APIs to enable users to use them in their choice of programming/GIS tools. Plugins for REST APIs from different platforms should be developed and shared with users. The products/results generated using these tools should be visualized and downloaded. All the tools available in the catalog should have unique identifiers, metadata, tool description, user manuals, and technical papers if applicable.

3.2.1.4 Geospatial web applications

The primary purpose of the data and tool catalog is to enable users who are more proficient in the use case domain, data analysts, and Geographic Information Systems (GIS) experts, to develop their applications with minimum knowledge in computer science. It is essential to host already developed applications and the process involved in making them available over the web platform as tutorials and use cases for the users. These are also helpful for policymakers and decision-makers to make decisions using these applications directly. General Public can also utilize these applications without hands-on involvement with datasets and tools. Apart from the applications developed using data/tool catalog, any other geospatial applications that benefit the users can also be hosted.

3.2.1.5 User Interaction

User interaction is one of the crucial requirements for any website. This provides a way to support users for better utilization of the framework. User interaction in the form of user forums to raise queries, frequently answered questions (FAQs), and Knowledge resources, including user manuals, tutorials, workshops, and webinars. Hosting articles in the field of services domain or articles on applications developed using the tools/datasets is also helpful and motivates users. The tutorials for data ingestion are separated from this component, as data ingestion is for data suppliers. As the tool catalog and data catalog are planned to be designed using REST APIs for data access, tool usage, and data analysis, a developer forum to support the developers and make the portal usage better is needed. A developer forum can benefit these types of applications as it provides an opportunity for community developers to develop tools that, after validation, can be added to the web platform. However, this poses a security threat if not validated thoroughly.

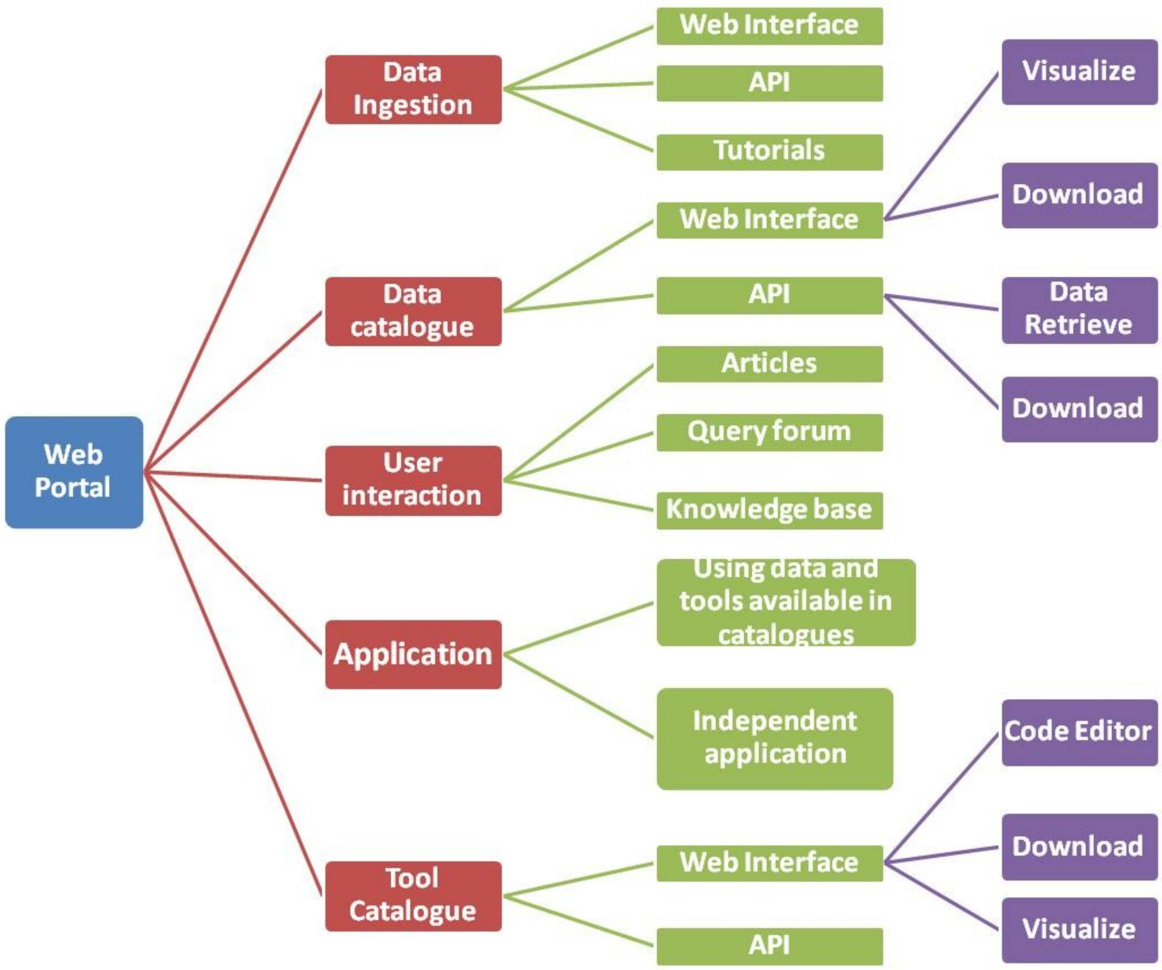

Figure 8: Web Portal features

3.2.2 Medium-level web application layer:

In this layer, we have the modules that implement web applications and application security modules to provide users with secure Web and REST API access.

3.2.2.1 Secure REST APIs

Adding a secure layer to enable secure token/key-based REST API access will ensure proper authentication and authorization along with accountability of the usage. This application module should be added on top of all REST API implementations for secure access.

3.2.2.2 Application Modules

These modules provide the required functionality of the web user interface like user sign-in, search, and user interaction modules like hosting tutorials, FAQs, and developer forum.

3.2.2.3 Visualization

Visualizing the available datasets and tool outputs is one of the significant requirements of the platform. This module allows the platform to view the available datasets and tool outputs as maps. Open layers and Leaflet are some of the widely used opensource map visualization tools.

3.2.3 Low-level application layer:

In this layer, we have the modules that implement the OGC standards and other modules that interact with the data storage layer and provide information to medium and high-level application layers. The following are the major components of this layer: file upload Module, FROST Server, pygeoapi module, GeoServer, and Open Data Cube.

3.2.3.1 File upload module:

This module is for data ingestion purposes. It enables the user to upload files using a web Interface and REST APIs. This is suitable for uploading satellite-derived data products and other spatial datasets available as flat files.

3.2.3.2 FROST Server

This module enables users to ingest data in real-time using STA/STA+ and is suitable for uploading IoT datasets, in-situ instruments datasets, and citizen science datasets. This module can be implemented using FROST Server. FROST server is a rapidly developing opensource STA solution that is planning to implement STA+ in its next version. 52°North STA is another opensource solution that can be considered for this purpose.

Following are the steps involved for publishing datasets using FROST Server.

i. IoT data captured should be converted to STA data model and send to the STA server.

ii. IoT data to be ingested by the client using STA which can be made available using FROST Server.

iii. The FROST server upon data ingestion allows data to be queried using STA.

iv. OGC API – Features supports STA datasets. The STA datasets to be linked with the pygeoapi and published as OGC API – Features by modifying the configuration file.

v. Publish metadata of the datasets as OGC-API Records

vi. IoT data can be visualized using leaflet and Openlayers.

Frost Server currently supports only PostgreSQL database. Considering the volume of sensor data, it may not be the correct choice as relational databases doesn’t perform well for large datasets. This issue and frequent timeouts for queries on large datasets is also raised by few users of FROST server. By using a secondary database for faster queries, this drawback can be overcome.

An Enhancement is proposed, by synchronizing the data between PostgreSQL and secondary database and makes complex time-consuming queries from the secondary database, which results in no change in FROST Server implementation and improves the performance of queries. Elasticsearch, MongoDB, CrateDB and InfluxDB can be considered for this purpose.

3.2.3.3 Pygeoapi

Pygeoapi is an opensource implementation of OGC APIs, which is currently in the development stage. This can be used for implementing the Data catalog, Tool Catalog, and Data Ingestion modules of the framework. This covers OGC API Tiles, OGC API Features, OGC API Records, OGC API Coverages, OGC API Processes, OGC API EDR, and STAC. This also supports STA as data input for OGC API Features. OGC API Maps and OGC API GeoVolumes are in the development stage now and are not implemented in Pygeoapi. Till these features are implemented and tested, OWS WMS and OWS WMTS can be used in parallel for two-dimensional map visualization purposes.

Following are the steps involved for publishing datasets using pygeoapi.

vii. Publish data set as OGC-API Tiles for visualizing as fixed size static tiles.

viii. Publish metadata of the datasets as OGC-API Records

ix. In case of discrete datasets, publish as OGC API Features for Clients to perform queries and use data in their applications.

x. In case of Coverage datasets, publish as OGC API Coverages for Clients to perform queries and use data in their applications.

xi. Develop tools as OGC API Processes, which is a plug-in of pygeoapi for publishing computing tasks that can be executed by the user. This can be used for both Online processing and Offline Processing based on the mode selected (synchronous for Online and Asynchronous for Offline processing)

xii. The data published can be visualized using pygeoapi. Specific libraries in leaflet, Openlayers and QGIS are being developed to support these standards.

3.2.3.4 GeoServer

GeoServer is an opensource implementation of OGC Web Services standards WMS, WFS, WMTS, WCS, and WPS. It also implements CSW. This is one of the widely used opensource software for sharing geospatial datasets over the internet, and the standards supported are widely used. OGC API standards and its opensource implementation are still in the development stage. Hence, utilizing well-tested opensource software and standards provides additional advantages of reusability.

Following are the steps involved for publishing datasets using GeoServer.

i. Publish data set as WMS for visualizing as static files based on client queries related to spatial constraints. **(Visualization)**

ii. Publish data set as WMTS for visualizing as fixed size static tiles. **(Visualization)**

iii. In case of discrete datasets, publish as WFS for Clients to perform queries and use data in their applications. **(Querying and Analysis)**

iv. In case of Coverage datasets, publish as WCS for Clients to perform queries and use data in their applications. **(Querying and Analysis)**

v. Develop processes as WPS, which is a plugin of GeoServer for publishing computing tasks that can be executed by the user. The list of processes available to be uploaded to tool catalogue. **(Processing)**

vi. The data published can be visualized using leaflet, Openlayers or in client-side applications like QGIS. **(Visualization)**

3.2.3.5 Open Data Cube

Considering the large growing datasets and the worldwide users, it is suggested to use Big Data technologies for this framework. All the geospatial datasets, instead of using directly, can be used using Open Data Cube. It is an opensource project for managing large volumes of earth observation datasets. It is one of the data providers of pygeoapi as a plug-in, and hence the data published using open data cube can be disseminated using pygeoapi.

3.2.4 Data Storage layer

The data storage layer should store all the datasets from different sources in the form of flat files and GIS databases. Implementation of open data cube (ODC) is also recommended as we will be dealing with big geospatial datasets. The following are the major components of this layer.

3.2.4.1 Flat files

The core framework should support NetCDF, O&M data model, and geo package. Support for other widely used GIS data formats like GeoTIFF, HDF, and shapefile should be provided as separate modules/plugins based on the requirement.

3.2.4.2 GIS Database

This database hosts GIS datasets. A combination of relation databases like PostgreSQL and NoSQL database that supports GIS operations like MongoDB or Elasticsearch suits this requirement.

3.2.4.3 Resource Database

This database hosts metadata information related to resources, i.e., data and tools. This database serves as a database for OGC API Records. Pygeoapi implementation of OGC API records supports Elasticsearch and TinyDB currently.

3.2.4.4 STA Database

This database is for storing data from STA implementation (FROST server). FROST Server is currently implemented using PostgreSQL database with PostGIS extension. Considering the growing datasets, it is recommended to consider adding a secondary database that supports Big Data storage and query requirement.

3.2.4.5 Data Archive

Data archival is required for data-based web platforms for long-term retention.

3.2.5 Overall high-level architecture

Figure 9 discusses the overall architecture of the proposed platform.

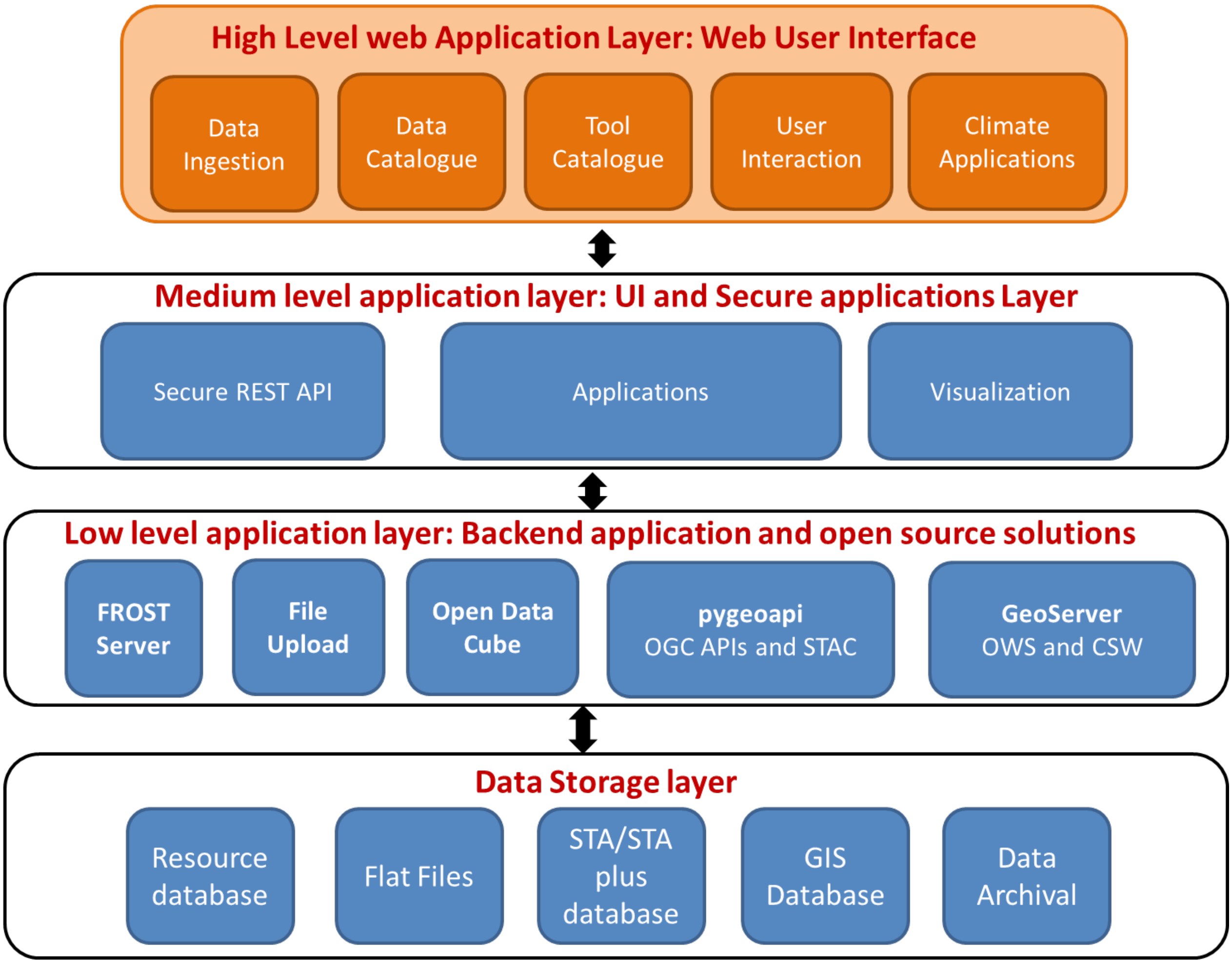

Figure 9: High-Level Architecture diagram of the proposed platform

3.2.5.1 Framework validation

Framework validation is required to be used effectively with accurate and reliable information. For this purpose validation will be carried out at individual module levels. Moderation and validation mechanism for data ingestion will ensure the validation of data hosted along with the metadata. Data correction check and data validation will be done based on the metadata. Before integrating the tools in the framework, each tool will be thoroughly checked and validated along with the tool metadata. The correctness and availability of the tools will be checked periodically based on automated checks. All the applications will be validated before making them available to the users.

3.2.6 Other design considerations

Apart from the functional requirements covered in the architecture discussed, the framework should consider the non-functional requirements inherent to the widely used web services. These include security, performance, and usage analytics.

3.2.6.1 Security

The platform should be robust against all the known security vulnerabilities. For this purpose, the layers of the Open Systems Interconnection (OSI) model should be secured. Frequent vulnerability tests and threat assessments need to be carried out. The detailed design of web security implementation is out of the scope of the current work.

3.2.6.2 Performance

The performance of web interface and REST APIs is vital to attract a larger audience and retain current users. The platform should be able to handle spatial big data effectively. Hence big data technologies should be built in the platform design for 1) Fetching the huge volume of datasets from storage and database in a faster time. 2) Handle large velocity of streaming data from IoT devices and citizen science data streams. 3) Handle many concurrent requests to the platform from users. 4) Manage and handle the complex processing tasks created by users. These challenges should be considered for designing the hardware, network, and software components of the platform. Some of the solutions for handling these challenges are 1) Establishing a cache mechanism for a faster response for frequently used datasets and tools. 2) Using a distributed messaging system for handling streaming data. 3) Well-defined workflow management for handling user requests. 4) Utilization of scalable databases and file systems. 5) Utilizing big data technologies.

3.2.6.3 Analytics

Web usage analytics is another major requirement of web-based services to understand the users and to improve the services based on the analytics information. The design of the platform should consider acquiring and storing website and API usage information, which is beyond the scope of this work.

The components and subcomponents that are part of web-portal are shown in the below figure.

3.2.7 Data Ingestion to Data Catalogue workflow

The following are the steps proposed to be involved between interactions between data supplier and web portal.

1. The data supplier enters the website and chooses data ingestion module.
2. The data supplier will be prompted to register in the website. They

will be registered under any of the four categories: Satellite derived data products, IoT data, in-situ data and citizen data.

3. User registrations are to be manually validated if the data falls under Satellite derived data products/IoT data/in-situ data. Validation is not required for citizen datasets

4. If the user registers as supplier for Satellite derived data products and the registration is validated, the following steps are to be carried out further.

    a. Fill data set details: This includes the dataset name, temporal and spatial resolutions, the ECV to which the dataset belongs to, updation/addition frequency etc (Mandatory)

    b. Provide technical documents including user manual, contact details of supplier (Mandatory), method for deriving product from satellite data (Mandatory) and any other application/articles published using this data.

    c. Required metadata attributes are shared to the user. The supplier prepares a metadata file and uploads the file when prompted. The uploaded file will be validated against requirements.

    d. When the supplier, dataset and metadata are validated, a unique data identifier is created for this data and metadata. The details are ingested into metadata management database (Elasticsearch) which is compatible with OGC Records. By creating this, we ensure that the data is discoverable.

    e. Create a unique API key and link it to the user.

    f. Create an exclusive (physical/virtual) folder in the Satellite derived data products archival with the unique data identifier.

    g. Create an API for user to upload the data to this storage with example using the generated API key. The API should accept the satellite derived data product based on the frequency mentioned by the user in the format given by the user. The API request should be rejected with error if it doesn’t follow the above and pre-agreed terms.

h. The supplier will use this API to upload dataset regularly. The details of upload will be updated in data inventory and the data and metadata will be stored in the folder discussed above. In case of non availability of data or failure of data upload by the user, the data will be marked not available in the data inventory. The user can upload this data at later stage and the inventory will be automatically updated.

i. Perform quality checks everyday to ensure that files are available as agreed. Any discrepancy observed by quality check service can be intimated to the supplier over Email.

5. If the user registers as supplier for IoT data/Citizen data/in-situ and the registration is validated, the following steps are to be carried out further.

   a. Fill data set details: This includes the dataset name, temporal and spatial resolutions (point, multiple points, moving point(s)), the ECV to which the dataset belongs to, updation/addition frequency etc (Mandatory)

   b. Provide technical documents including user manual, contact details of supplier (Mandatory for IoT and Citizen data), and any other application/articles published using this data.

   c. Required metadata attributes are shared to the user. The supplier prepares a metadata file and uploads the file when prompted. The uploaded file will be validated against requirements.

   d. When the supplier, dataset and metadata are validated, a unique data identifier is created for this data and metadata. The details are ingested into metadata management database (Elasticsearch) which is compatible with OGC Records. By creating this, we ensure that the data is discoverable.

   e. Create a unique API key and link it to the user.

   f. Create an exclusive table in the database (STA/STAplus) with the unique data identifier.

   g. Create an API for user to upload the data to this database with example using the generated API key. The API should accept data supplied by user as per STA/STAplus standard. The supplier will use this API to upload dataset regularly.

h. Perform quality checks everyday to ensure that the database is being updated regularly as agreed. Any discrepancy observed by quality check service can be intimated to the supplier over Email.

6. Data Catalogue should be linked to metadata database and data inventory.
7. The links for API access in case of STA/STAplus and download in case of satellite derived products should be mentioned in Data Catalogue

3.2.8 Tool creation to tool publishing workflow

The following are the steps involved in tool creation.

- The tools for spatial data analysis can be categorized into two major categories:
  - Offline tools, that require significant time to execute
  - Online tools that execute in less than 6 seconds
- These tools should be published as WPS using GeoServer or OGC-API Processes using pygeoapi. The record for the tool should be populated to OGC-API records for discoverability.

3.2.9 Geospatial Application workflow

Geospatial applications are the ready to use applications created with or without using the tools and datasets available in the web portal. These applications are web applications that are accessible over web interface for citizens to get required information and decision makers for taking decisions without needing the scientific knowledge and technical expertise of data and domain specific scientists. These applications are meant to be used directly and hence need not require any standards. However, to scale the applications and for faster response, it is better to use big data technologies for such application.

Chapter - 4: Climate Science Research case studies

This chapter discusses how to present different spatial datasets and how different analytics can be done on that. The case study 4.1 demonstrates presenting streaming data from a point like IoT and automating the process from data ingestion to publishing. Case studies 4.2 and 4.3 demonstrate presenting satellite data and developing statistical tools that enable the data to be analyzed. Case study 4.3 demonstrates a use case where the statistical tool is used to perform large scale analytics across different regions over time and space.

4.1 Air pollution monitoring using IoT Case Study

Real time air pollution monitoring is vital in densely populated areas to assess the micro level air pollution. Monitoring the air quality can be useful for providing accurate information to the public and to help policy decision makers to take actions that can improve the quality of air. Monitoring of the pollutants using Internet of Things (IoT) is a cost-effective solution. This has the capability to provide high spatial and temporal air pollutants levels depending on the density of the IoT network.

The idea of this study is to implement an IoT based solution to understand the components involved and to use open-source technologies that are suitable for IoT solutions. For this, we build a prototype solution for monitoring concentrations of various air pollutants along with temperature and humidity. One of the major considerations for this case study is that the solution should be scalable and can handle millions of IoT devices.

LoraWAN sensors CO_2, CO and PM2.5 that also captures temperature and humidity are used.

The data collected from sensors should be collected, processed and stored. A module for data transmission is written in python. This module sends the data it receives to the RESTful APIs that are configured on the Server.

FROST Server, an implementation of OGC SensorThings API (STA), is configured on Server. This server uses PostgreSQL as database. The data sent from the network server is received by FROST server and stored in PostgreSQL database. Elasticsearch, a document-based NoSQL data store and Kibana, visualization platform for Elasticsearch is also configured in this server. The IoT sensor data is stored in Elasticsearch and Kibana is used to create web dashboards of the IoT data.

The major challenges for the IoT data collection module are the ability to handle vast amount of data in real time. Ideally, the data collection module should be distributed in nature with high performance and consistency. Also, the solution proposed should be scalable in nature to cater to the changing requirements. Handling these challenges using FROST Server, an opensource implementation of STA, remains to be seen. FROST Server is written in JavaScript and uses POSTGRESQL database for storing IoT data. Although several demo implementations are implemented using FROST Server and USGS has implemented SensorThings API, the ability to handle huge velocity of data, scalability when the volume of data increases and availability when the user load increases have to be explored. Using scalable database like Elasticsearch and MongoDB and real time streaming services like Apache Kafka can add these features to the STA implementation.

The database used for this solution is Elasticsearch which is a search engine and document based NoSQL database. The data collected and processed by the python script is stored in this database. This database is distributed in nature and has the capability to be scaled with minimal configuration changes. Major requirements of IoT data storage are data integrity, consistency, high availability, and high performance and fault tolerance. Additional requirements like simple use interface for management and data visualization, availability of APIs to multiple programming languages, real time search and analytics, capability to store massive structured and unstructured data sets and capability to perform batch processing can provide additional benefit.

Below figure describes the architecture of the prototype solution in detail consisting of sensors, network server and application.

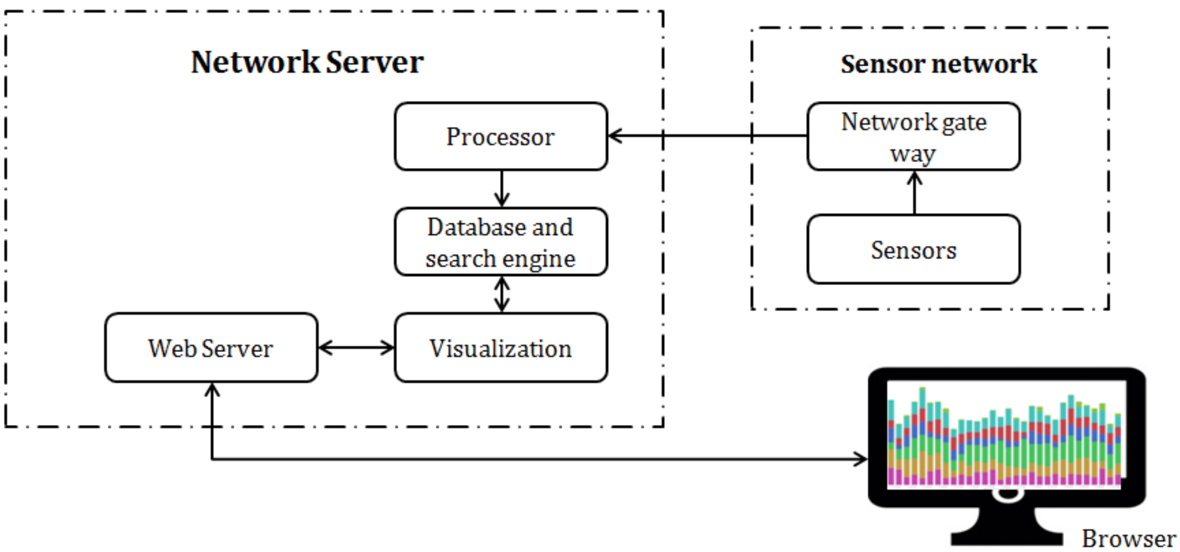

Figure 10: Major components for Air Pollution Monitoring using IoT

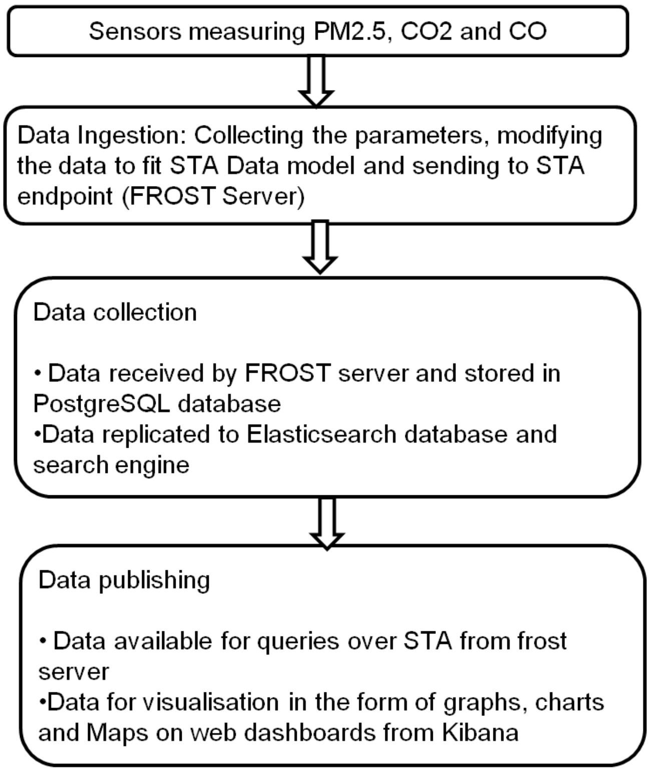

Figure 11: IoT solution for Air Pollution Monitoring -workflow

Nginx is an open source high performance HTTP web server. Kibana dashboards are reverse-proxied through Nginx web server. Nginx uses scalable event-driven architecture and doesn't rely on threads like other traditional web servers and is ideal for handling thousands of simultaneous requests. This makes it very suitable for IoT applications. Other software like Apache web server and HAProxy can also be used for this purpose.

Sample web dashboards that are created using this solution are shown in below figure.

Figure 12: Web Dashboard

Lessons learned:

- FROST Server, which is an open source implementation of SensorThings API is demonstrated to be highly useful for IoT data ingestion
- By combining PostgreSQL, the widely use spatial database with the NoSQL database Elasticsearch, scalability and availability is achieved. Also the performance of the data accessibility is increased due to the faster queries made to the Elasticsearch database.
- Creation of web dashboards on the fly using Kibana software is demonstrated.
- Faster queries
- OGC standards for interoperability and re-usability
- Easy to create web dashboards

In this case study, we demonstrated user visualization and streaming data publish online in an automated way. An automated system is developed that automates data ingestion, data storage and data publication for streaming data. However, the major limitation of this use case is analyzing different types of datasets together, which will be covered in 4.2 and 4.3.

4.2 Data analysis of long-term Ozone data (from 1978 to 2020)

To protect the Ozone layer and sustain the environment from the adverse UV effects, the Montreal Protocol was formed in 1985 to contain the emissions of man-made chlorofluorocarbons (CFCs) and hydro chlorofluorocarbons (HCFCs). Strong evidence exists that the Montreal Protocol has been successful in controlling ozone-depleting substances since its inception. In this study, we analyzed Total column Ozone (TCO) data for 40 years (1979-2020) and calculated the decadal trends over the globe. We focused our research on Antarctica region to understand the area of ozone hole.

Datasets used: For this study, we used National Aeronautics and Space Administration (NASA)'s Nimbus-7/Total Ozone Mapping Spectrometer (TOMS), TOMS-Earth Probe and Aura/Ozone Monitoring Instrument (OMI) sensors data.

Results of the study:

For this study, we analysed 40 years of daily datasets of 1deg x 1 deg spatial resolution. We calculated Mann-kendall slope for the daily values over any region to understand the trend of Tropospheric Ozone over that area. Tools for calculating average TCO concentration for the selected time period, mann-kendall test and Sen's slope is also developed.

The algorithm used for this study, can be queried based on time period (day, month, year) and spatial extent (global, India, bounding box, shapefile). This helped to carryout studies over different locations at different time periods.

To understand the ozone hole recovery, average total columnar ozone (TCO) values and mann kendall test is performed over Antarctic region.

Using daily satellite measurements with a spatial resolution of $1^{\circ}\times1^{\circ}$, the below figure illustrates the TCO concentration averaged during the period of mid-September to mid-October for a decade.

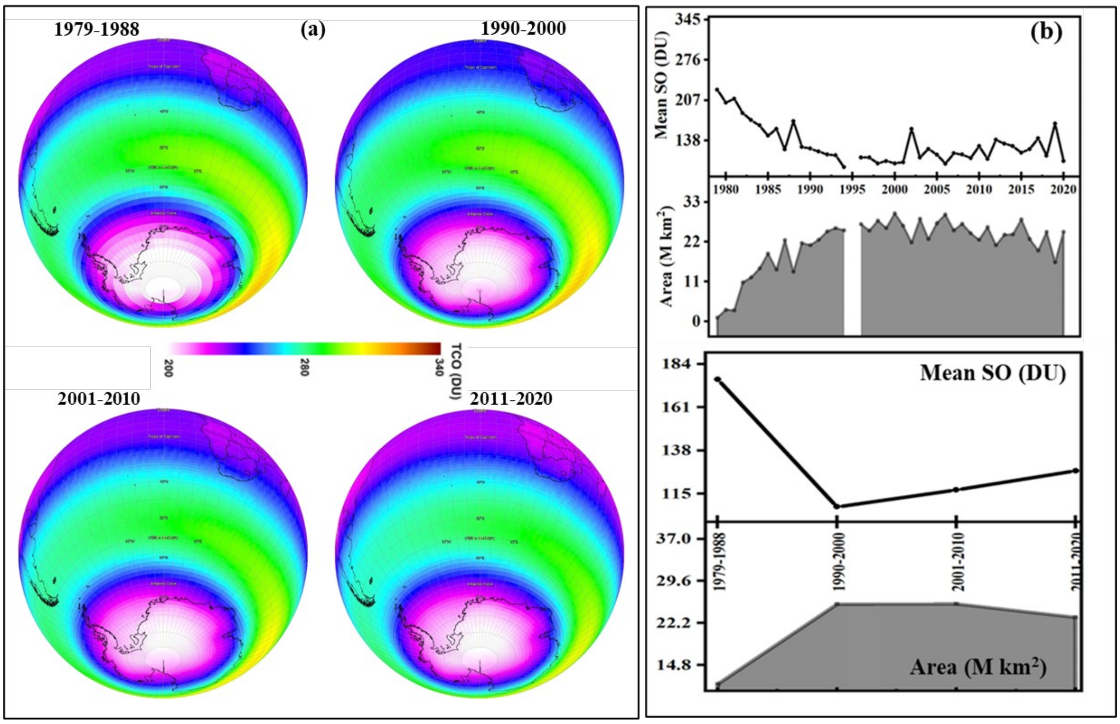

Figure 13: Daily Averaged decadal TCO over the Antarctica region and mean yearly stratosphere Ozone (SO) and Area (million-M km2) along with the decadal mean during 1979-1988, 1990-2000, 2001-2010 and 2011-2020 [Data accessed from https://ozonewatch.gsfc.nasa.gov/ on 15th June 2022).

In the present study, we chose the TCO threshold value < 220 DU for assessing the Ozone hole recovery. According to the TCO concentrations in the present study, the Antarctic Ozone hole's extent has decreased considerably over the past decade compared to that of the period from 1978 to 1988. Areas of the TCO over the Antarctica region during different decadal time windows are 11.47 million km2, 25.52 million km2, 25.56 million km2 and 23.19 million km2 respectively. The results of this study demonstrate how well the Montreal Protocol functions in the recovery of ozone hole.

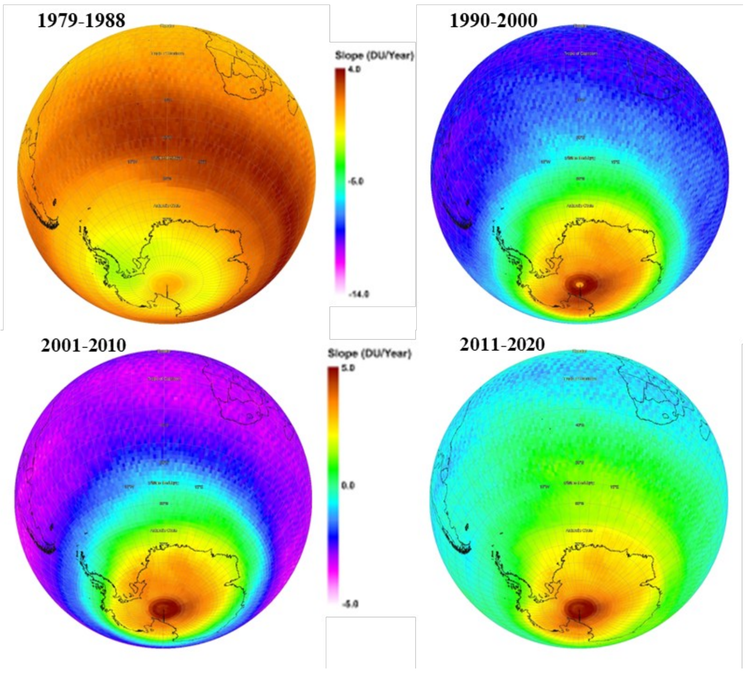

Figure 14: The Global Mann Kendal slope (Smk) during a) 1979-1988 b) 1990-2000 c) 2001-2010 and d) 2011-2020 time periods.

Further, as shown in the above figure, we determined the yearly Mann Kendall (MK) slopes (Smk) of the global Ozone values throughout various decadal time periods. The MK test is used to statistically determine whether the TCO data exhibits a monotonic increasing or decrease trend over the span of space and time. This accurately depicts the Ozone value's rising or falling trend at the pixel level. The positive (negative) slope indicates an upward (downward) tendency trend. We have fixed separate scale bar for the 1979-1988 period, due to the large variability in the slope (-14 to +4). The Smk over the Antarctica is -4 DU year-1 in the 1979–1988 decade, as shown in the above figure. It shows that Ozone values across this area have decreased during the decade 1979–1988. This was a driving force for the creation of the Montreal Protocol. The Smk values, which are +4, +3.5, and +3 DU year-1 for the decades 1990-2000, 2001-2010, and 2011-2020 respectively, show that this trend levelled off over the ensuing decades and then slowly reversed, leading to the observed increase in Ozone. Although, the Ozone values have increased in the last 3 decades after the formation of Montreal Protocol, it has been noticed that the rate of change is retreating slowly every decade (+4 DU year-1 in 1990-2000 and +3 DU year-1h in 2011-2020).

This could be due to three reasons, according to recent research and the findings of the current study:

i. The global emission of prohibited long-lived CFCs from unidentified sources.

ii. CFCs that were exempted as a result of industry requests, as permitted by the Montreal Protocol, which allows industries to request exclusions (UNEP).

iii. The existence of Ozone-depleting compounds with a short half-life, such as dichloromethane, which are not covered by the Montreal Protocol.

Lessons learned:

- Reusable software modules (tools) reduced the overall data analysis time as same tools are used for daily Ozone data.
- The functionalities of these tools are given below.
    - Re-sampling to given resolution
    - Average pixel value for calculating average total columnar ozone
    - Mann-Kendall test to study the trends
- These tools support queries on spatial and temporal extents, using which specific study for Antarctic region is carried out.

In this case study, we analyzed 40 years datasets and provide analytical outcomes. This is an Extension of 4.1 as it demonstrates combining multiple datasets and application of statistical methods to it.

4.3 Assessment of the Impact of COVID-19 Nationwide Lockdown on Air Pollution across India

The nationwide lockdown was imposed over India from 25 March to 31 May 2020 with varied relaxations from phase I to phase IV to contain the spread of COVID-19. Thus, emissions from industrial and transport sectors were halted during lockdown (LD), which has resulted in a significant reduction of anthropogenic pollutants. In this research, we study the impact of first two lockdown phases (LD1 and phase LD2), which were strictly implemented on the Air pollution across India.

Data considered: Satellite-based troposphere columnar nitrogen dioxide (TCN) from the years 2015 to 2020, troposphere columnar carbon monoxide (TCC) during 2019/20, and aerosol optical depth (AOD550) from the years 2014 to 2020 during phase I and phase II LD and pre-LD periods were investigated with observations from Aura OMI, Sentinel-5P TROPOMI, and Aqua and Terra MODIS.

Methodology:

- Parameters **NO_2, AOD and CO** are selected & processed
- Total three time periods are defined: Pre-Lockdown (**PLD**) (1.03.2020 to 21.03.2020), Lockdown Phase-I (**LD1**) (25.03.2020 to 14.04.2020) and Lockdown Phase-II(**LD2**) (15.04.2020 to 03.05.2020)
- **Region of Interest** for this study: India, State level, Indo Gangateic Plains, Central India and North East India
- **Time Averaged Maps** generated for the PLD, LD1 and LD2 time periods.
- Time Averaged Maps generated for the same dates for the year 2019 for the 3 parameters and for the years 2014-2019 for AOD (6 year mean) & 2015-2019 for NO2 (5 year mean)
- **Difference Maps** generated to compare the changes w.r.t 2019 for all 3 parameters and w.r.t 6 year mean and 5 year mean for AOD and NO2 respectively
- **Metrics used:** Mean value of the difference maps for the RoI, % of positive pixels (> 1SD) and % of negative pixels (< -1SD)
- Student's T-Test is carried out for studying the **statistical significance**

The workflow diagram is given in detail in below figure.

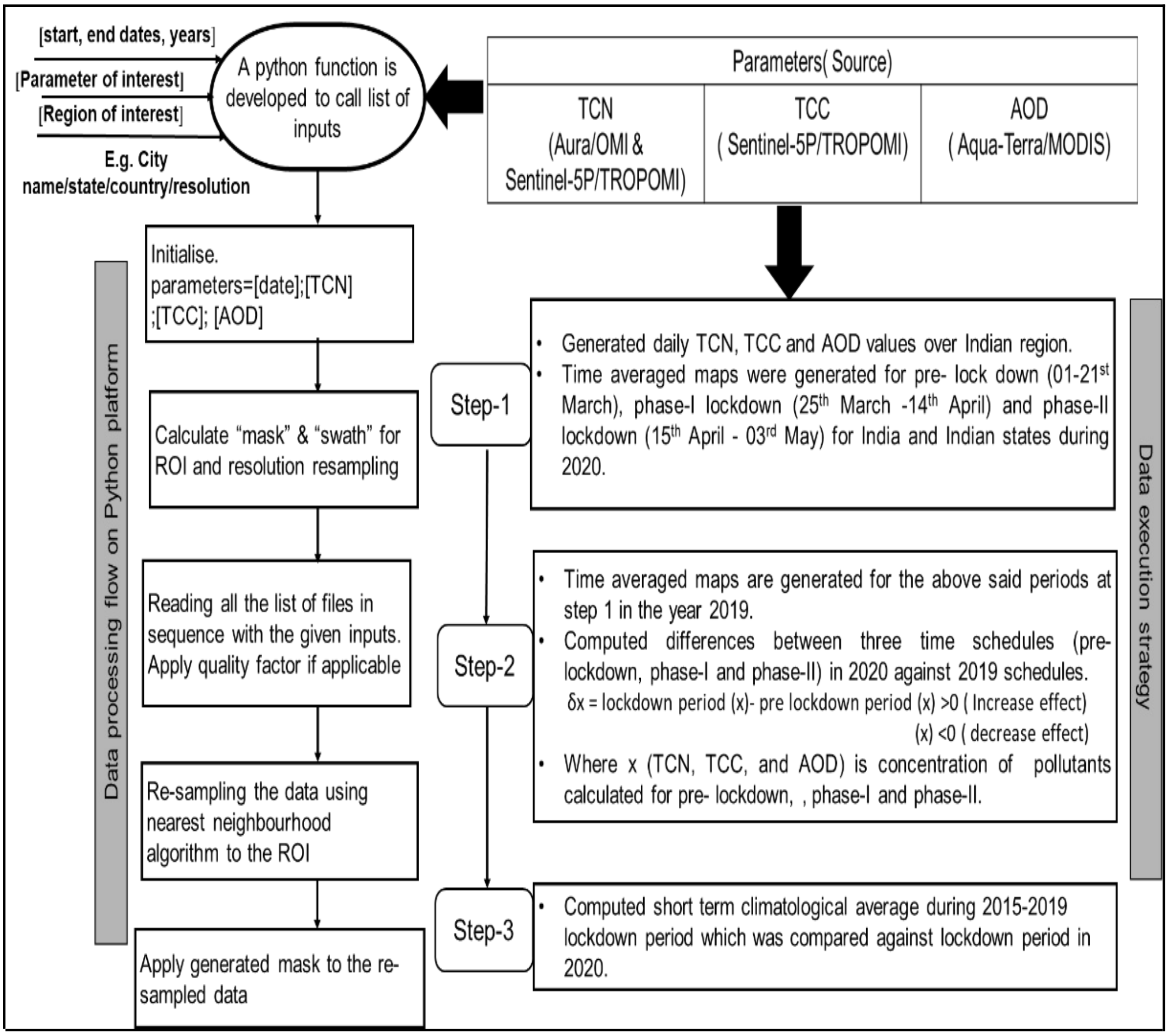

Figure 15: Workflow of the study

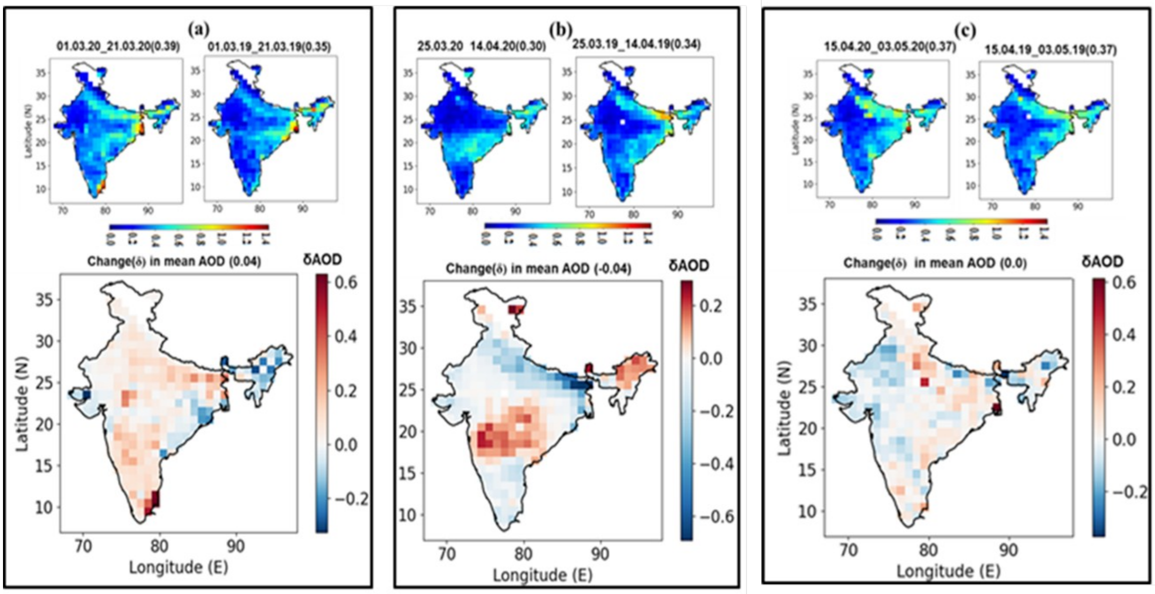

Figure 16: Mean AOD Values and change in AOD values during PLD, LD1 and LD2 time periods

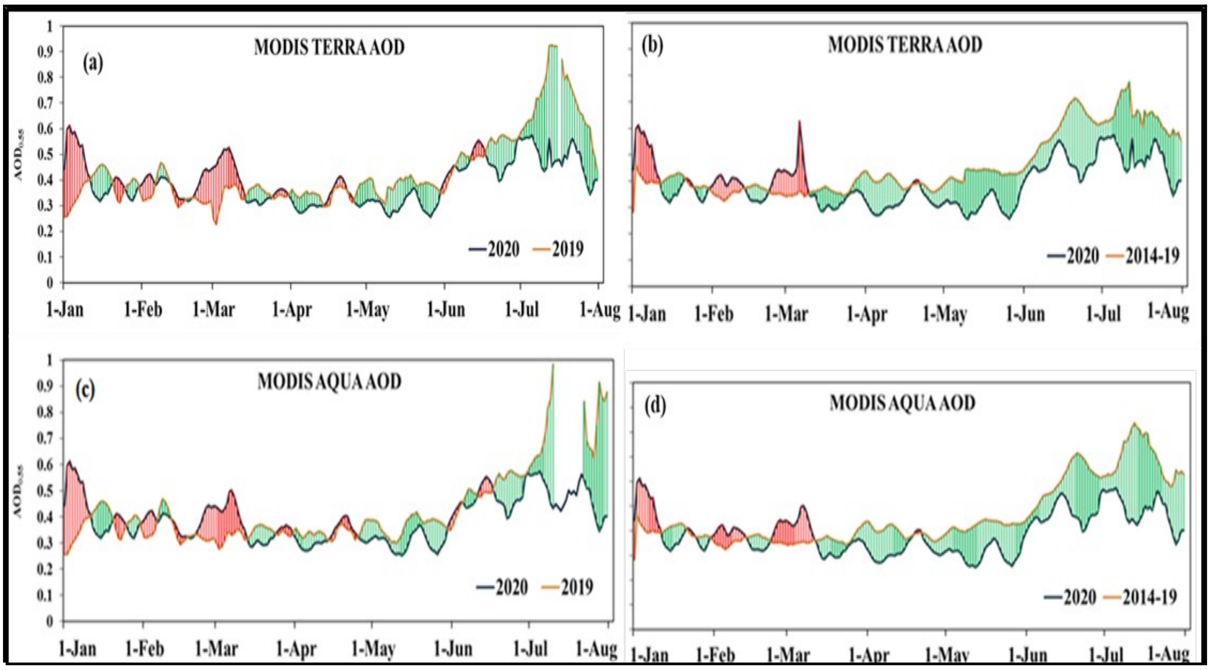

Figure 17: 7 Day Average of AOD values

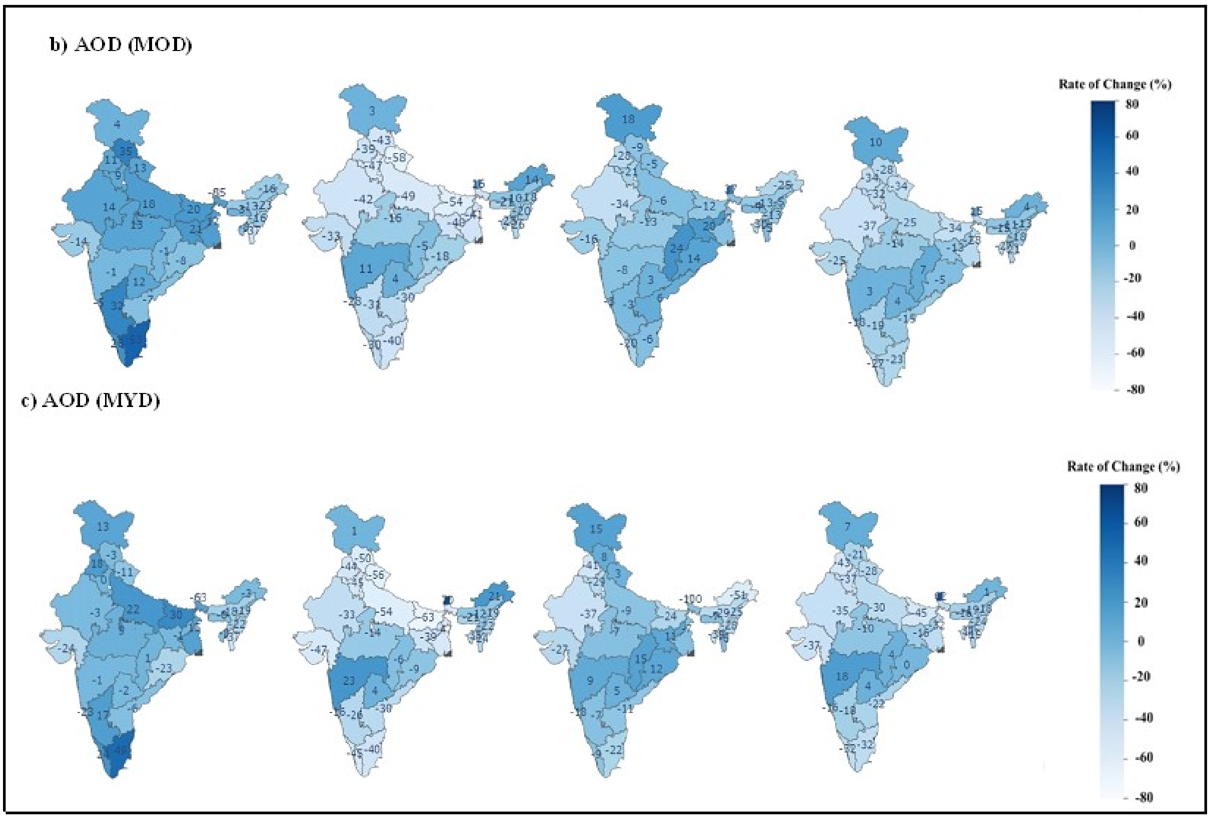

Figure 18: State-wise rate of change of AOD values

Although three parameters and 7 different datasets are considered for this study, we used reusable software modules for processing these datasets based on the user selected parameters on spatial and temporal extents. For visualizing the outputs, we used reusable software that is same for all the seven datasets. This reduced our work as no re-work is required to process different datasets as the processing steps don’t change. Thus, having a common processing engine where algorithms for processing and visualizing can be reusable is very important for climate studies. Having to use these reusable blocks over web provides the additional advantage of remote execution and availability to multiple users.

Lessons learned:

.

Chapter-5: Conclusion and Future work

In this research work, we discuss the need for web based Geospatial Services framework. By studying the available frameworks, current implementations, and the growing need for the geospatial services to empower citizens with geo intelligence, help researchers to carry out faster research and enable policy makers to take informed decisions, we arrived at the functional and non-functional requirements of the framework. The current solutions are evolving provides for these kind of functionalities and there is a need to integrate multiple datasets for analysis for a particulate use case. Keeping this in view, requirements and design of the framework is carried out with data ingestion, data catalogue, tool catalogue and ready to use geospatial applications as major functionalities.

Data ingestion module enables data suppliers to ingest different types of spatial data into the platform. Data Catalog provides datasets for visualization, access and downloads. Tool Catalog helps process the datasets with readily available published tools with FAIR (Findable, Accessible, Interoperable and Re-usable) principles. Apart from functional requirements, the need for scalability, availability and analytics is also discussed.

We found that the OGC Web Services standards and OGC API standards and SensorThings API standard are suitable for the framework based on the services they provide, user base, compliance with current web technologies and recommendations, and the availability of the open-source implementations. GeoServer, pygeoapi and FROST Server are some of the open-source software that implement the OGC standards mentioned above. Further, we discuss the need for the extension of SensorThings API to accommodate large amounts of citizen data by adding additional entities related to the data ownership. We also discuss the need for utilizing Big Data solutions for the platform and we incorporated some of the Big Data Solutions that can be used like NoSQL databases and open data cube.

Further, we implement three use cases in climate science domain using the components of the framework design using which we demonstrated that the processes from data ingestion to publication can be automated. In the use case to monitor air pollution using IoT we demonstrated handling of Internet of things data, scalability of IoT data and development of on-the-fly web dashboards that provide insights into the IoT streaming data. In this use case, we combined SensorThings API implementation with Elasticsearch and Kibana stack and demonstrated the additional advantages this combination provides in terms of scalability, availability and also provides the ability to create and query web dashboards.

From the other two use cases, we provided tools to analyze long term Ozone data and air pollutants data obtained from satellite derived products. Case studies 4.2 and 4.3 demonstrate presenting satellite data and developing statistical tools that enable the data to be analyzed. Case study 4.3 demonstrates a use case where the statistical tool is used to perform large scale analytics across different regions over time and space. In both these use cases, we demonstrated the re-usability of the tools, which leads to faster research.

Using the case studies, we demonstrated the framework for some of the features, dataset formats and sources. This work can be extended by implementing a complete end to end suite integrating all the individual components as a single solution that provides all the functionalities of framework.

The need of the end users of this platform change continuously and providing the services the user expects increases the platform usage. Hence, the user behavior needs- to be analyzed continuously, and matched with the platform features. This is possible by acquiring user analytics from user information, website usage information in the form of page visits, visit time, most popular services etc. The need for analytics is mentioned in the design, however it can be further improved by designing the web analytics solution for geospatial services.

In this framework design, we aim to provide tools to the user, which can be used over web interface or REST API, that processes the datasets in both online and offline modes. Although, we discussed in detail how these tools can be hosted, we haven’t incorporated the big data concepts to process large amounts of spatial data. However, further work needs to be carried out to incorporate Big Data concepts for data processing.

Related Publications

1. Aarathi M, Mahesh Pathakoti, Vinod M Bothale, G Biswadip, M.V.R Sesha Sai, V Subramanian, K S Rajan (2019). Design and Implementation of IoT Solution for Air Pollution Monitoring, 2019 IEEE Recent Advances in Geoscience and Remote Sensing: Technologies, Standards and Applications. (TENGARSS), doi: 10.1109/TENGARSS48957.2019.8976041.

2. Mahesh Pathakoti, Aarathi Muppalla, Sayan Hazra, Mahalakshmi D Venkata, Kanchana A. Lakshmi, Vijay K. Sagar, Raja Shekhar, Srinivasulu Jella, Sesha Sai Mullapudi, Uma Vijayasundaram (2021). Measurement report: An assessment of the impact of a nationwide lockdown on air pollution–a remote sensing perspective over India, Atmos. Chem. Phys., 21, 9047–9064, 2021, https://doi.org/10.5194/acp-21-9047-2021.

Bibliography

[1] United Nations. United Nations Sustainable Development Goals. 2015. https://sdgs.un.org/goals, Accessed August 13, 2022.]

[2] IPCC, 2021: Summary for Policymakers. In: Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Masson-Delmotte, V., P. Zhai, A. Pirani, S.L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M.I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J.B.R. Matthews, T.K. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Zhou (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, pp. 3−32, doi:10.1017/9781009157896.001.

[3] Arias, P.A., N. Bellouin, E. Coppola, R.G. Jones, G. Krinner, J. Marotzke, V. Naik, M.D. Palmer, G.-K. Plattner, J. Rogelj, M. Rojas, J. Sillmann, T. Storelvmo, P.W. Thorne, B. Trewin, K. Achuta Rao, B. Adhikary, R.P. Allan, K. Armour, G. Bala, R. Barimalala, S. Berger, J.G. Canadell, C. Cassou, A. Cherchi, W. Collins, W.D. Collins, S.L. Connors, S. Corti, F. Cruz, F.J. Dentener, C. Dereczynski, A. Di Luca, A. Diongue Niang, F.J. Doblas-Reyes, A. Dosio, H. Douville, F. Engelbrecht, V. Eyring, E. Fischer, P. Forster, B. Fox-Kemper, J.S. Fuglestvedt, J.C. Fyfe, N.P. Gillett, L. Goldfarb, I. Gorodetskaya, J.M. Gutierrez, R. Hamdi, E. Hawkins, H.T. Hewitt, P. Hope, A.S. Islam, C. Jones, D.S. Kaufman, R.E. Kopp, Y. Kosaka, J. Kossin, S. Krakovska, J.-Y. Lee, J. Li, T. Mauritsen, T.K. Maycock, M. Meinshausen, S.-K. Min, P.M.S. Monteiro, T. Ngo-Duc, F. Otto, I. Pinto, A. Pirani, K. Raghavan, R. Ranasinghe, A.C. Ruane, L. Ruiz, J.-B. Sallée, B.H. Samset, S. Sathyendranath, S.I. Seneviratne, A.A. Sörensson, S. Szopa, I. Takayabu, A.-M. Tréguier, B. van den Hurk, R. Vautard, K. von Schuckmann, S. Zaehle, X. Zhang, and K. Zickfeld, 2021: Technical Summary. In Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Masson-Delmotte, V., P. Zhai, A. Pirani, S.L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M.I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J.B.R. Matthews, T.K. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Zhou (eds.)].

Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, pp. 33−144. doi:10.1017/9781009157896.002.

[4] GCOS (Global Climate Observing System). 2014. “Submission from the Global Climate Observing System (GCOS) toSBSTA 41 on Agenda Item 8 (b) Research and Systematic Observation.”https://unfccc.int/sites/default/files/175.pdf.

[5] World Meteorological Organization. Climate Knowledge for Action: A Global Framework for Climate Services-Empowering the Most Vulnerable. World Meteorological Organization, 2011.

[6] Giuliani, Gregory & Nativi, Stefano & Obregon, Andre & Beniston, Martin & Lehmann, Anthony. (2017). Spatially enabling the Global Framework for Climate Services: Reviewing geospatial solutions to efficiently share and integrate climate data & information. Climate Services. 8. 10.1016/j.cliser.2017.08.003.

[7] Hewitt, C., Mason, S. & Walland, D. The Global Framework for Climate Services. Nature Clim Change 2, 831–832 (2012). https://doi.org/10.1038/nclimate1745

[8] Maria Teresa Miranda Espinosa, Gregory Giuliani and Nicolas Ray. Reviewing the discoverability and accessibility to data and information products linked to Essential Climate Variables. INTERNATIONAL JOURNAL OF DIGITAL EARTH 2020, VOL. 13, NO. 2, 236-252, https://doi.org/10.1080/17538947.2019.1620882

[9] Guo, Huadong & Wang, Lizhe& Chen, Fang & Liang, Dong. (2014). Scientific big data and Digital Earth. Chinese Science Bulletin (Chinese Version). 59. 1047. 10.1360/972013-1054.

[10] OGC 10-090r3: OGC Network Common Data Form (NetCDF) Core Encoding Standard version 1.0, 2011-04-05, Open Geospatial Consortium, Category: Candidate OpenGIS ® Encoding Standard, http://www.opengis.net/doc/IS/netcdf/1.0 .

[11] OGC 12-128r18: OGC GeoPackage Encoding Standard, 2021-11-16, Open Geospatial Consortium, Category: OGC Encoding Standard, http://www.opengis.net/doc/IS/geopackage/1.3.

[12] OGC 10-129r1: OGC Geography Markup Language (GML) — Extended schemas and encoding rules, 2012-01-16, Open Geospatial Consortium, Category: OpenGIS Implementation Standard, http://www.opengis.net/spec/GML/3.3

[13] OGC 15-100r1: OGC Observations and Measurements – JSON implementation, 2015-12-09, Open Geospatial Consortium, Category: OGC Discussion Paper, http://www.opengis.net/doc/dp/om-json/

[14] OGC 12-080r2: OGC OWS Context Conceptual Model, 2014-01-22, Open Geospatial Consortium, Category: OGC® Implementation Standard, www.opengeospatial.net/doc/IS/owc-conceptual/1.0.

[15] OGC 21-008: Joint OGC OSGeo ASF Code Sprint 2021 Summary Engineering Report, 2021-04-12, Open Geospatial Consortium, Category: OGC Public Engineering Report, http://www.opengis.net/doc/PER/ogc-osgeo-asf-codesprint2021 .

[16] OGC 18-008: OGC SensorThings API Part 1: Sensing Version 1.1, 2021-08-04, Open Geospatial Consortium, Category: OGC Implementation Standard. http://www.opengis.net/doc/is/sensorthings/1.1

[17] OGC 17-079r1: OGC SensorThings API Part 2 – Tasking Core, 2019-01-08, Open Geospatial Consortium, Category: OGC Implementation Standard, http://www.opengis.net/doc/IS/sensorthings-part2-TaskingCore/1.0

[18] OGC 07-006r1: OpenGIS Catalogue Services Specification, 2007-02-23, Open Geospatial Consortium, Category: OGC Implementation Standard.

[19] Kim, M., Kim, M., Lee, E., Joo, I. (2005). Web Services Framework for Geo-spatial Services. In: Kwon, YJ., Bouju, A., Claramunt, C. (eds) Web and Wireless Geographical Information Systems. W2GIS 2004. Lecture Notes in Computer Science, vol 3428. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11427865_1

[20] Gong, J., Shi, L., Du, D., & de By, R. A. (2004). Technologies and standards on spatial data sharing. In *ISPRS 2004 : proceedings of the XXth ISPRS congress : Geo-imagery bridging continents, 12-23 July 2004, Istanbul, Turkey. Comm. IV WG IV/2. pp. 118-128* (pp. 118-128). International Society for Photogrammetry and Remote Sensing (ISPRS). http://www.itc.nl/library/Papers_2004/peer_conf/deby.pdf