APG3039F
Spatial Data Infrastructures (SDI) Project

Case Study: Chief Directorate: National Geospatial Information

Report Prepared By:

Emily Wood, WDXEMI002

3rd Year UCT Geomatics Geoinformatics Stream Engineering

Links:

Website:

CDNGI Geoportal

GitHub Repository:

emilywood-uct/cdngi-geoportal: Metadata catalogue for my Cape Town WMS datasets

Date of Submission:

28th May 2025

Table of Contents

Table of Contents **2**

1. Introduction **4**

2. System Design **5**

2.1. Context Diagram 5

2.2. Data Flow Diagrams (DFDs) 6

2.1.1 Top-level Diagram *6*

2.1.2 First-Level Diagram *7*

2.3. Package Diagram 8

2.4. Decomposition Diagram 10

3. **System Implementation - Prototype *Geoportal*** **12**

3.1. Geospatial Data 12

3.1.1. Choice of Data *12*

3.1.2. Sources of Data *13*

3.1.3. Refinement of Data *13*

3.2. Geo-Server 14

3.2.1. Design *14*

3.2.2. Implementation *14*

3.2.3. Motivation *15*

3.3. WMS and WFS 16

3.3.1. Design *16*

3.3.2. Implementation *16*

3.3.3. Motivation *16*

3.4. Metadata Catalogue 17

3.4.1. Design *17*

3.4.2. Implementation 17

3.4.3. Motivation 18

3.5. Interactive Web Map 18

3.5.1. Design 18

3.5.2. Implementation 19

3.5.3. Motivation 21

3.6. Website 21

3.6.1. Design 21

3.6.2. Implementation 22

3.6.3. Motivation 22

4. Conclusion 23

5. Appendix - Data Dictionary 24

6. Appendix – Diagrams 29

Disclaimer:

If any diagrams appear unclear due to size constraints, please refer to the Diagrams Appendix to view them in full size for better clarity.

1. Introduction

This project was developed as part of a university course on Spatial Infrastructures. The goal of the assignment was to apply the principles of SDI design to a real-world scenario. For this project, our team was tasked with assisting the Chief Directorate: National Geospatial Information (CDNGI) in improving how they manage, store, and share spatial data.

CDNGI has offices spread across multiple geographic locations and currently faces several challenges with their existing geoportal. These include frequent downtime of the portal, difficulties in downloading spatial datasets, a web map that is not user-friendly, and the absence of a proper metadata catalogue. These issues limit the usefulness of the geoportal for both internal and public users.

Our assignment was divided into two main parts:

- To design a new SDI system that improves data management, sharing, and accessibility.
- To implement a working web geoportal that demonstrates how this new SDI could function.

Our proposed solution involves creating a cloud-based server to replace the outdated server and reduce downtime and reliability issues. The new geoportal is more accessible and includes an interactive, user-friendly web map. It is connected to a metadata catalogue, allowing users to easily view details about each dataset and download data using services like WMS, WFS, or directly as shapefiles where available.

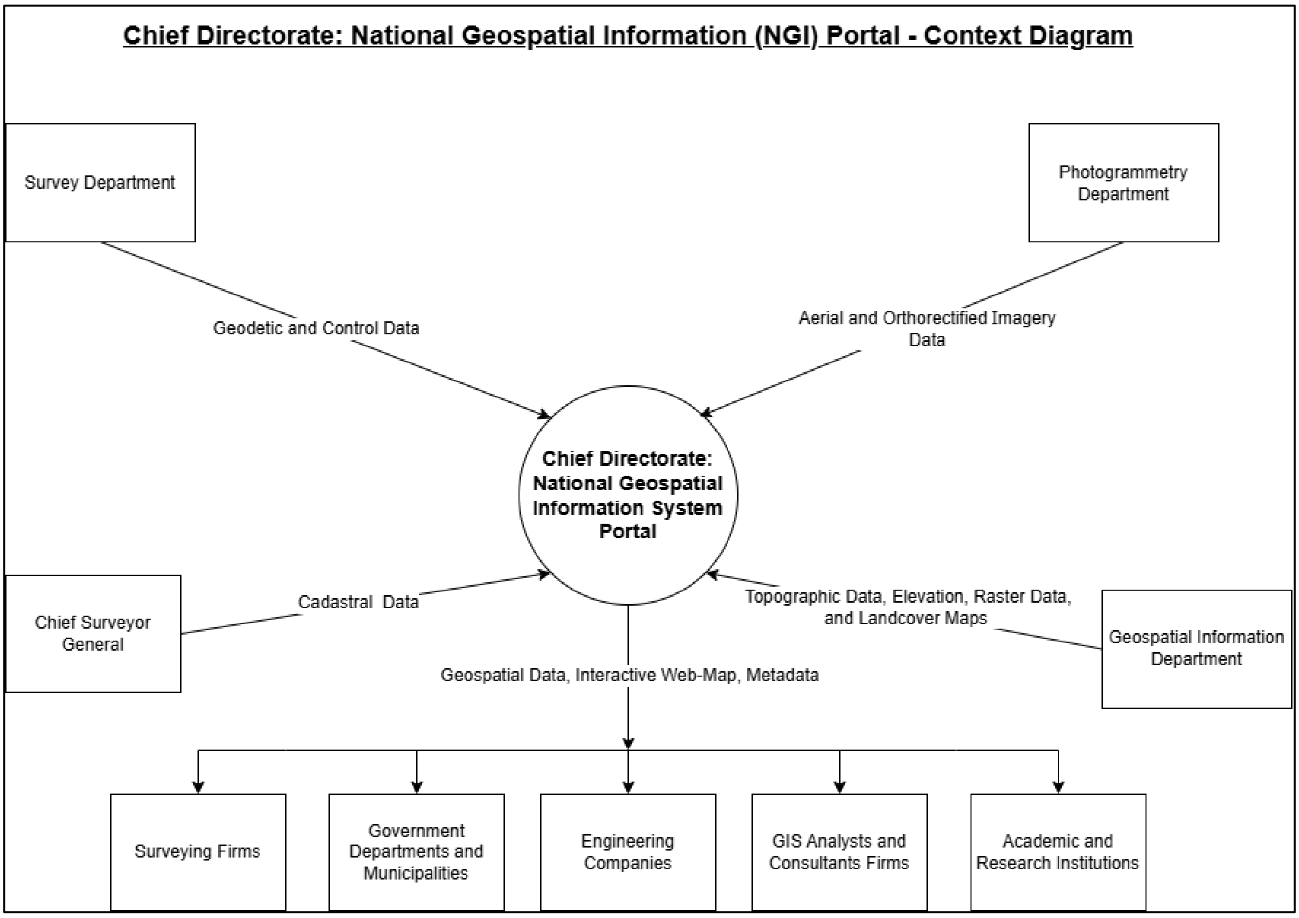
Although the system was designed to work at a national scale, the prototype implementation is limited to the Cape Town area due to the scope and constraints of this being a university project. However, the overall design was approached as if it were to be implemented fully across CDNGI.

This project shows how modern technology and good SDI practices can help CDNGI deliver spatial data more effectively and support better decision-making and public access to geographic information.

2. System Design

2.1. Context Diagram

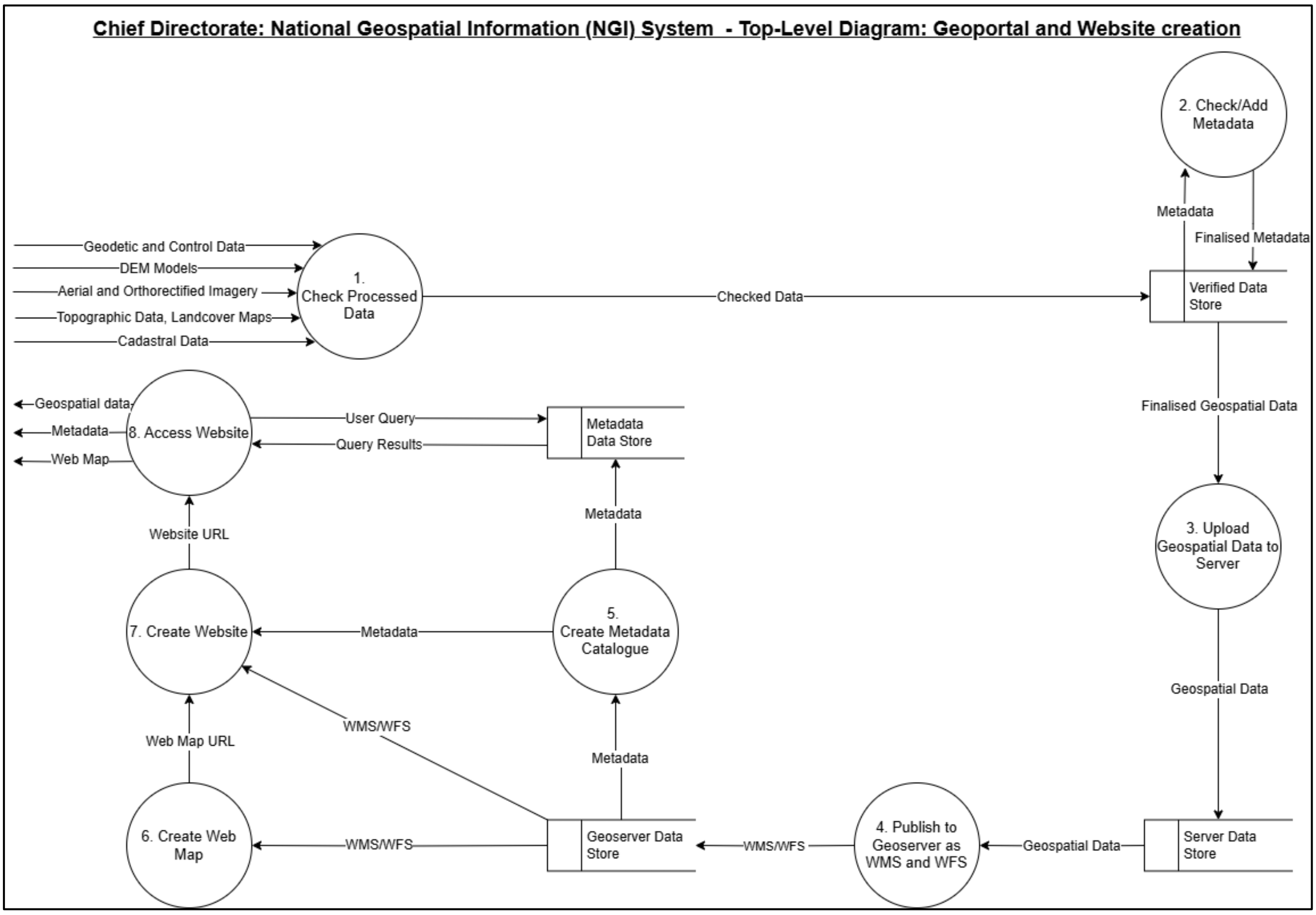
The context diagram is a crucial first step in system design, as it provides a clear, high-level view of how the proposed SDI system interacts with external entities. It defines the system boundaries and illustrates the flow of information between the geoportal and its data providers and end users. This helps stakeholders understand the system's role within its broader environment, identify key interfaces, and ensure that all relevant parties are considered in the design. By visualising these interactions early on, the context diagram lays the foundation for more detailed system analysis and development.



2.2. Data Flow Diagrams (DFDs)

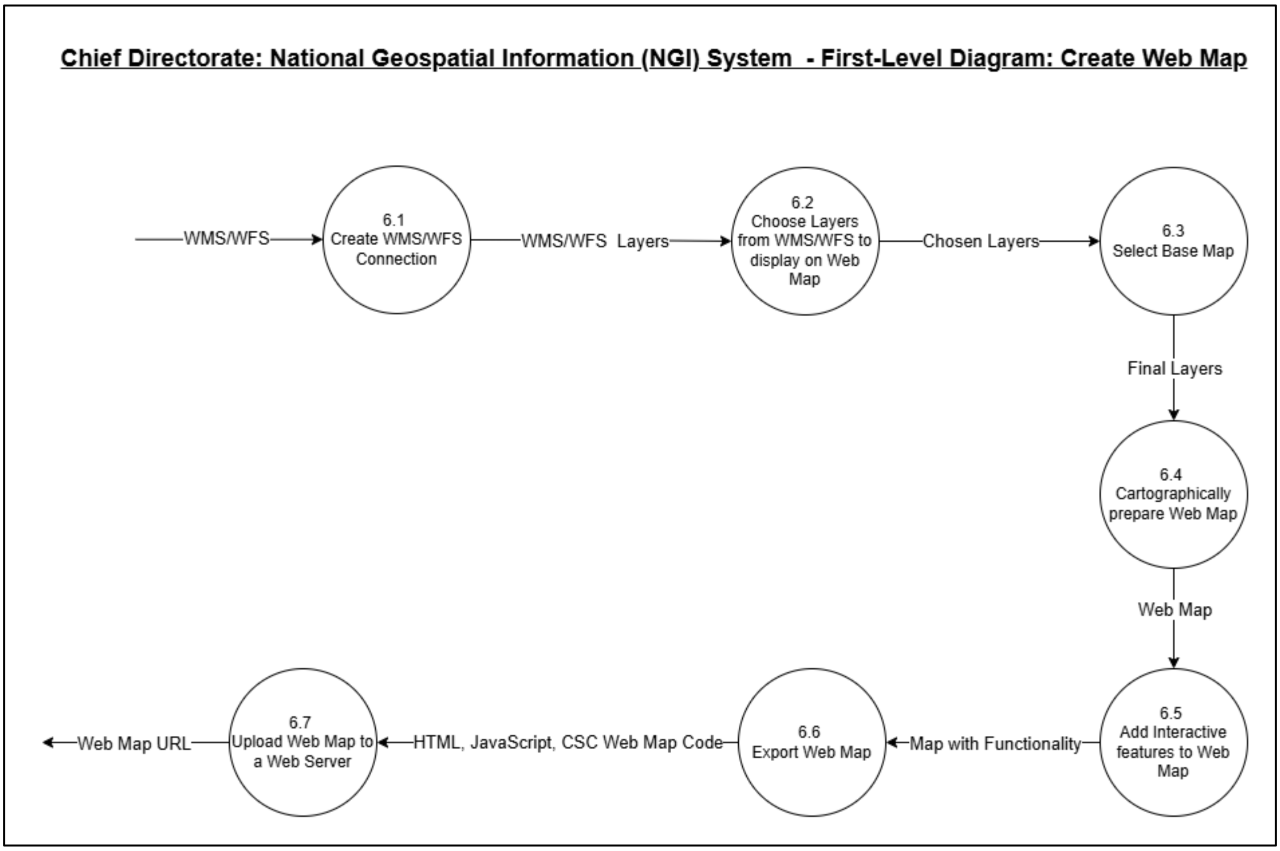
2.1.1 Top-level Diagram

The top-level Data Flow Diagram (DFD) provides an overview of the main processes involved in developing the geoportal system. It breaks the system design down into eight core processes, showing how data moves between each process and the external entities involved. This diagram is important because it offers a structured view of the system's functionality without going into technical detail. It helps identify the major components of the workflow, their relationships, and how data is transformed throughout the system. This high-level perspective supports better planning, communication, and coordination during the system design and development stages.



2.1.2 First-Level Diagram

The first-level DFD zooms in on one of the key processes from the top-level diagram, creating the interactive web map. It details the internal steps, data inputs, and outputs involved in building and integrating the map component of the geoportal. This more detailed view is essential for understanding specific data interactions, such as establishing WMS/WFS connections to the web map application, cartographically preparing layers for web display, and finally exporting and hosting the map on a web server. By focusing on one critical process, the first-level DFD helps developers and stakeholders analyse the technical requirements and ensure that this component functions correctly within the larger system.



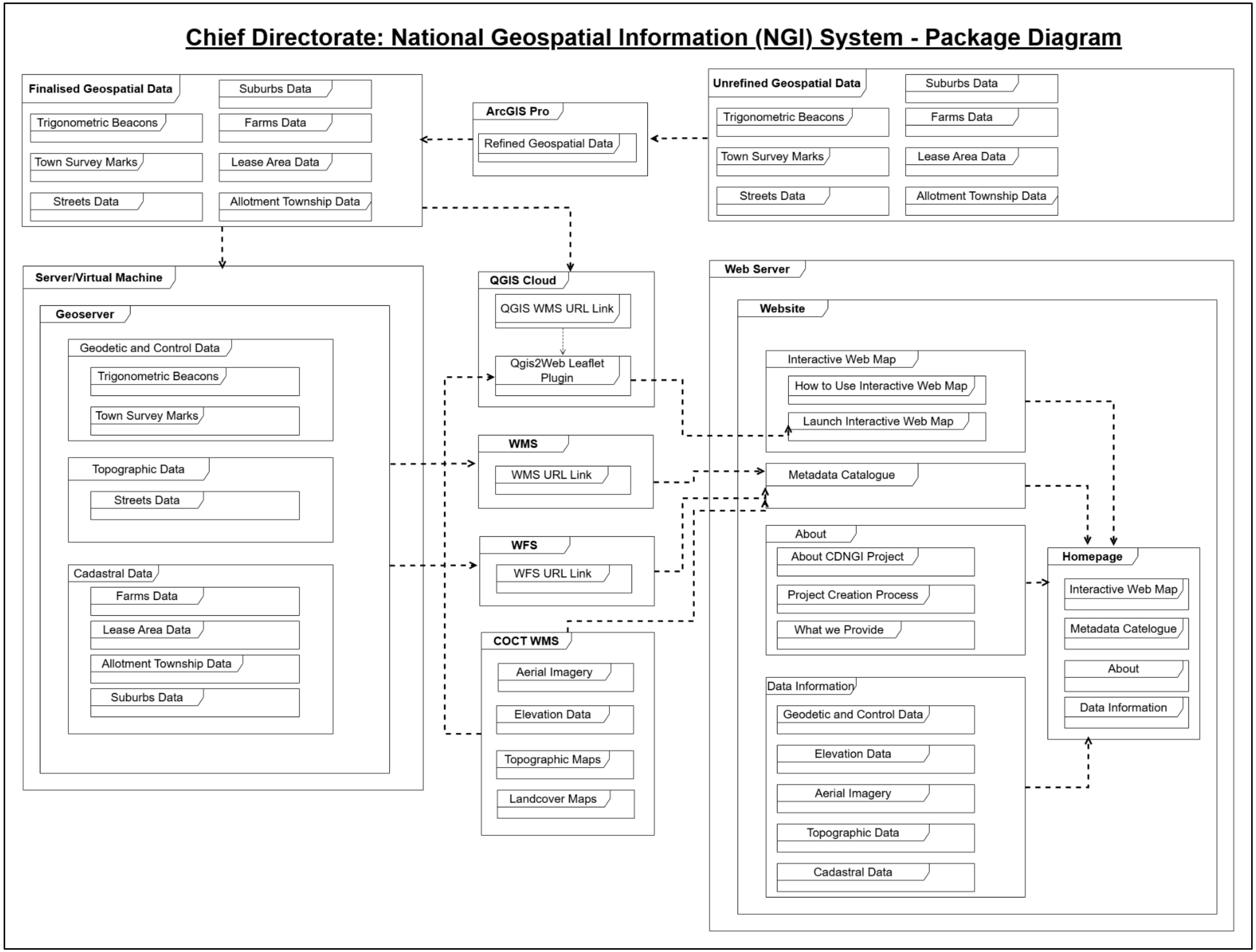
2.3. Package Diagram

The package diagram is used to represent the internal structure of the spatial data infrastructure by grouping related components, such as the server, website, and spatial data services, into logical packages. It is particularly well-suited for showing how spatial data is stored, accessed, and communicated between various parts of the system. This makes it an effective tool for visualising the architectural design of a geoportal.

In our project, the diagram illustrates how spatial data is hosted on a cloud-based server, how WMS and WFS services are used to provide access to this data, and how these services are connected to the metadata catalogue. It also shows how QGIS Cloud was used to create a hosted spatial database and generate WMS links, which were then used to build an interactive Leaflet map via the qgis2web plugin. The internal structure of the website is also displayed, showing how the map, metadata, and download tools are linked and hosted on a central web server.

We chose a package diagram for this purpose because it allows us to clearly represent the system's modular structure and communication flow without unnecessary technical complexity. Unlike an Entity-Relationship Diagram (ERD), which is best suited for relational databases and does not represent spatial data or web services effectively, the package diagram accommodates both spatial data and service-oriented architecture. Similarly, a Class Diagram, while useful for object-oriented design, includes detailed attributes and methods that are too specific for simply showing how spatial data is stored and accessed. The package diagram strikes the right balance, offering a clear, system-level view ideal for communicating the structure of our spatial data infrastructure.

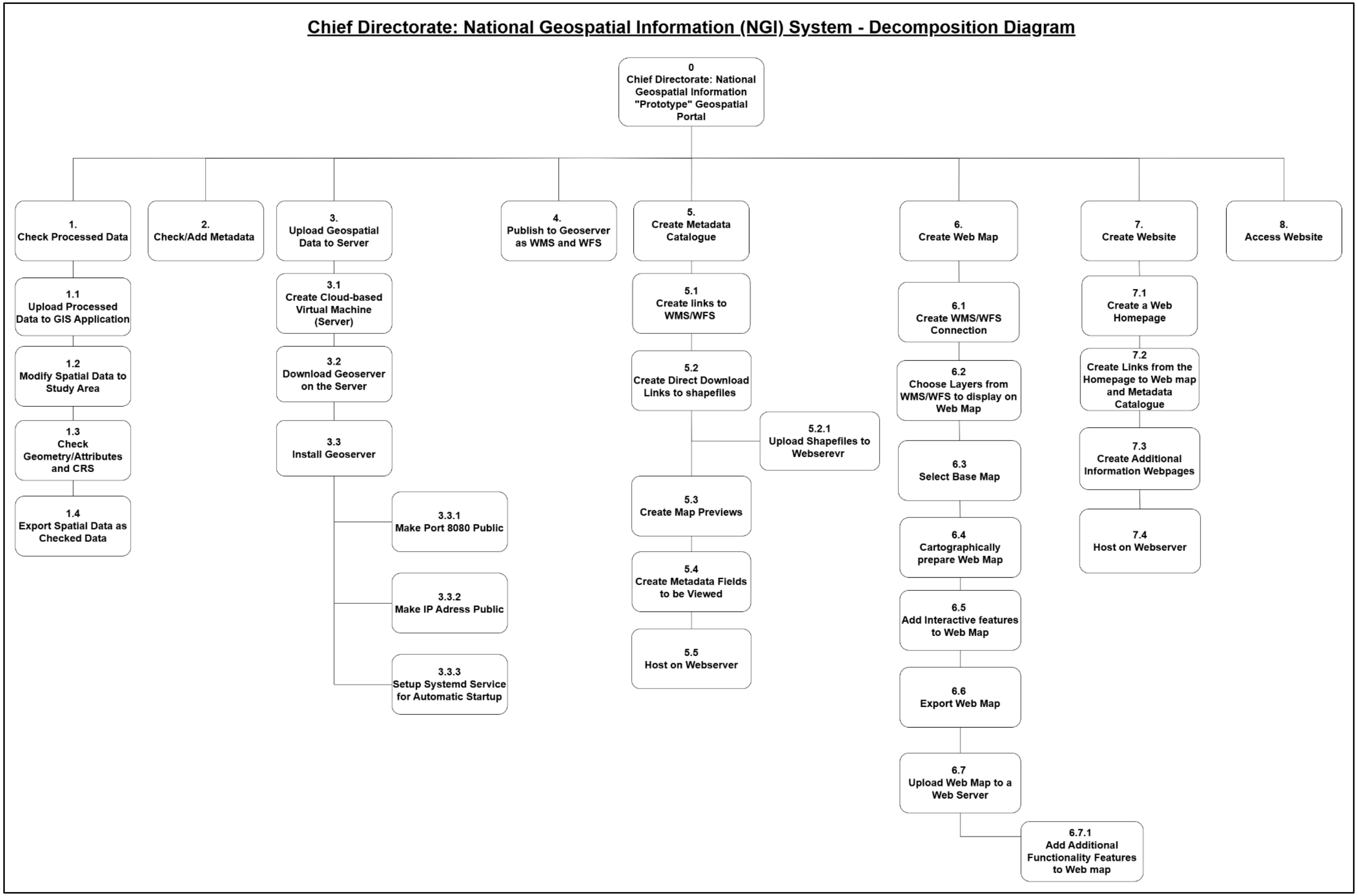
.


Chief Directorate: National Geospatial Information (NGI) System - Package Diagram
Finalised Geospatial Data
Suburbs Data
Trigonometric Beacons
Farms Data
Town Survey Marks
Lease Area Data
Streets Data
Allotment Township Data
ArcGIS Pro
Refined Geospatial Data
Unrefined Geospatial Data
Suburbs Data
Trigonometric Beacons
Farms Data
Town Survey Marks
Lease Area Data
Streets Data
Allotment Township Data
Server/Virtual Machine
Geoserver
Geodetic and Control Data
Trigonometric Beacons
Town Survey Marks
Topographic Data
Streets Data
Cadastral Data
Farms Data
Lease Area Data
Allotment Township Data
Suburbs Data
QGIS Cloud
QGIS WMS URL Link
Qgis2Web Leaflet Plugin
WMS
WMS URL Link
WFS
WFS URL Link
COCT WMS
Aerial Imagery
Elevation Data
Topographic Maps
Landcover Maps
Web Server
Website
Interactive Web Map
How to Use Interactive Web Map
Launch Interactive Web Map
Metadata Catalogue
About
About CDNGI Project
Project Creation Process
What we Provide
Data Information
Geodetic and Control Data
Elevation Data
Aerial Imagery
Topographic Data
Cadastral Data
Homepage
Interactive Web Map
Metadata Catalogue
About
Data Information

2.4. Decomposition Diagram

The decomposition diagram is used to break down the overall system into smaller, well-defined sub-processes, providing a clear hierarchical view of how the system functions. It plays a key role in system design by helping organise complex tasks into manageable parts, clarify the scope of each function, and guide the transition from conceptual design to implementation.

In this project, the decomposition diagram serves as a combined overview of the top-level and first-level data flow diagrams, offering a unified view of the entire process involved in developing the geoportal with some additional sub-processed added. It captures both high-level processes, such as data refinement, metadata creation, and web development, and their more detailed sub-processes, like establishing WMS/WFS connections and configuring the interactive map. This structured layout supports better system planning, highlights the logical flow of development, and ensures that no essential functional process is overlooked during implement


Chief Directorate: National Geospatial Information (NGI) System - Decomposition Diagram
0 Chief Directorate: National Geospatial Information "Prototype" Geospatial Portal
1. Check Processed Data
1.1 Upload Processed Data to GIS Application
1.2 Modify Spatial Data to Study Area
1.3 Check Geometry/Attributes and CRS
1.4 Export Spatial Data as Checked Data
2. Check/Add Metadata
3. Upload Geospatial Data to Server
3.1 Create Cloud-based Virtual Machine (Server)
3.2 Download Geoserver on the Server
3.3 Install Geoserver
3.3.1 Make Port 8080 Public
3.3.2 Make IP Adress Public
3.3.3 Setup Systemd Service for Automatic Startup
4. Publish to Geoserver as WMS and WFS
5. Create Metadata Catalogue
5.1 Create links to WMS/WFS
5.2 Create Direct Download Links to shapefiles
5.2.1 Upload Shapefiles to Webserevr
5.3 Create Map Previews
5.4 Create Metadata Fields to be Viewed
5.5 Host on Webserver
6. Create Web Map
6.1 Create WMS/WFS Connection
6.2 Choose Layers from WMS/WFS to display on Web Map
6.3 Select Base Map
6.4 Cartographically prepare Web Map
6.5 Add Interactive features to Web Map
6.6 Export Web Map
6.7 Upload Web Map to a Web Server
6.7.1 Add Additional Functionality Features to Web map
7. Create Website
7.1 Create a Web Homepage
7.2 Create Links from the Homepage to Web map and Metadata Catalogue
7.3 Create Additional Information Webpages
7.4 Host on Webserver
8. Access Website

3. System Implementation - Prototype *Geoportal*

3.1. Geospatial Data

3.1.1. Choice of Data

Given the focus of the case study on the CDNGI, the selection of spatial data was guided by the types of datasets typically managed by national mapping and geospatial authorities. As such, the geoportal was designed to incorporate the following key data themes:

- **Geodetic and Control** datasets, including survey beacons and coordinate reference frameworks.
- **Topographic** layers representing natural and built features as well as Topological Maps.
- **Aerial Imagery,** to provide high-resolution, real-world visual context.
- **Elevation Data**, in the form of Digital Elevation Models (DEMs), supporting terrain analysis.
- **Land Cover maps**, to represent vegetation, urban development, and other surface types.
- **Cadastral Data**, such as farms, allotments, and lease areas, to depict legal land divisions.

These categories were selected to reflect the core responsibilities of CDNGI and to serve a wide range of potential users including surveyors, urban planners, and environmental analysts.

3.1.2. Sources of Data

The spatial datasets were sourced from two primary repositories:

- **The University of Cape Town (UCT) Geomatics File Server**, which provided high-quality base data such as survey control networks and topographic layers.
- **The City of Cape Town (COCT) Open Data Portal,** which offered access to authoritative municipal datasets including cadastral boundaries, aerial imagery, elevation models, and land cover maps.

Due to project limitations regarding file size and storage constraints, particularly when hosting on platforms such as GitHub and QGIS Cloud, the COCT Imagery Web Map Service (WMS) was used in place of downloading full datasets for certain large layers. Specifically, the Elevation Data (DEMs), Aerial Imagery, and Land Cover Maps were accessed directly via WMS. This approach allowed real-time visualization of these layers in the geoportal without the need to host or serve the large files independently.

These sources were selected for their credibility, completeness, and relevance to the Cape Town study area, ensuring a reliable and efficient foundation for the spatial components of the project.

3.1.3. Refinement of Data

Once all required datasets were acquired, they were imported into ArcGIS Pro for preprocessing. The first step involved standardising coordinate systems and inspecting attribute fields for consistency. Subsequently, each dataset was clipped to the Cape Town municipal boundary using the Batch Clip tool.

This spatial restriction was necessary to:

- Keep the scope of the project focused on a defined area,
- Ensure manageable file sizes compatible with the limitations of GitHub and QGIS Cloud, which were later used for hosting and publishing the data.

This refinement process ensured that only relevant spatial content was included, maintaining both performance efficiency and clarity in the final geoportal product.

3.2. Geo-Server

3.2.1. Design

The goal behind the Geoserver component was to create a system that can host spatial datasets securely while remaining publicly accessible through WMS and WFS services. A key requirement was to set it up on a remote, cloud-based server to ensure easy access from any location and platform, while maintaining system stability and performance.

3.2.2. Implementation

To achieve this, A cloud-based Virtual Machine (VM) was created through Microsoft Azure's free student subscription. The VM was configured to run on an Ubuntu interface, and port 8080, being the port Geoserver runs on, was made publicly accessible. Additionally, it was ensured that the IP address was public, so that all WMS and WFS links referencing that address could directly fetch data from the server.

The VM was accessed from the local computer using SSH with the following command:

```
ssh -i C:\Users\emily\.ssh\cdngi-vm_key.pem azureuser@4.222.216.109
```

From the Ubuntu console, Geoserver was installed and manually started it using the startup script. Initially, Geoserver was only accessible via browser (at http://4.222.216.109:8080/geoserver) while the Ubuntu console session remained open, which posed a limitation.

To overcome this and ensure Geoserver runs continuously, a systemd service file (geoserver.service) was created on the virtual machine. This service automatically starts Geoserver on boot and keeps it running in the background.

The service was configured with the appropriate Java command and `GEOSERVER_DATA_DIR`, then enabled using:

```
sudo systemctl enable geoserver

sudo systemctl start Geoserver
```

This setup ensures Geoserver remains accessible 24/7, even after a VM restart. Once Geoserver was running reliably, all required spatial data was uploaded from the local machine to the VM. The Geoserver web interface was used to register the data directory, publish each layer, apply appropriate styling, and complete the metadata setup for accessibility via WMS and WFS services.

3.2.3. Motivation

A cloud-based virtual machine was chosen to host Geoserver in order to make the spatial data fully publicly accessible and to ensure that the server could be managed remotely from any location or computer. This approach is more practical in a real-world workplace setting, where members of a GIS team may need to work on the server both from the office and from home.

Geoserver was selected for its open-source flexibility and dedicated support for OGC-compliant services like WMS and WFS. Hosting it in the cloud provided scalability and simulated a real production environment.

Using Microsoft Azure gave the ability to monitor resource usage, compare speeds of different VM types, and track the actual cost of running a VM continuously. The one hundred credits provided through the Azure Student subscription allowed the server to run 24/7 for approximately 3.5 months, without incurring any personal costs.

Importantly, this setup also allowed our team to gain valuable direct experience in configuring and managing a virtual machine, covering everything from SSH access and port setup to installing Geoserver and creating automated system services to ensure continuous availability.

3.3. WMS and WFS

3.3.1. Design

The use of Web Map Services (WMS) and Web Feature Services (WFS) was integral to making the spatial data both accessible and usable across various GIS platforms. These services enable users to access up-to-date geospatial data directly from the server without needing to manually download files. WMS provides rendered map images for visualisation, while WFS offers direct access to raw vector data for analysis and editing. This design allows external users, such as GIS analysts, municipal planners, and engineers, to easily integrate the data into applications like ArcGIS Pro and QGIS by simply adding a WMS or WFS connection linked to the server. This not only improves accessibility but also ensures that users always interact with the most current datasets.

3.3.2. Implementation

The WMS and WFS services were implemented through the cloud-hosted Geoserver, which was configured to expose spatial layers using its public IP address. After uploading and registering the spatial datasets within Geoserver, the team enabled WMS and WFS capabilities for each relevant layer. These services were then evaluated by adding the WMS and WFS URLs into QGIS and ArcGIS Pro, confirming that the data could be viewed and queried directly from the server.

The public IP and open port 8080 allowed for seamless access to these services from any remote location, enabling external users to fetch live data without interacting with the raw files.

3.3.3. Motivation

Geoserver was chosen for publishing WMS and WFS services due to its open-source nature, robust support for OGC standards, and ease of configuration. It offers flexibility in managing layer styling, metadata, and service parameters, all through a user-friendly web interface.

Geoserver compatibility with multiple data formats and its ability to serve both raster and vector data made it an ideal solution for deploying interoperable services to meet the needs of making the CDNGI spatial data accessible. Additionally, its seamless integration with cloud-based server supported the overall goal of building a scalable, accessible spatial data infrastructure.

3.4. Metadata Catalogue

3.4.1. Design

The metadata catalogue, commonly referred to as a data clearinghouse, was designed to provide users with comprehensive access to spatial dataset information as well as facilitate interaction with the data itself. Its primary function is to present key metadata details for each dataset, such as title, description, keywords, and coordinate reference system. In addition, the catalogue enables users to connect directly to spatial data hosted on the cloud-based Geoserver through embedded WMS and WFS links. For smaller datasets, it also provides an option to download shapefiles directly, supporting quick and easy access to raw data when needed. To enhance usability, map previews were included for each dataset, allowing users to visualise the data before accessing or downloading it.

3.4.2. Implementation

The metadata catalogue was developed using HTML and hosted on a GitHub web server. Metadata fields for each dataset were manually populated within the HTML structure, with embedded WMS and WFS URLs linking back to the Geoserver-hosted spatial data. For datasets with downloadable shapefiles, each zipped shapefile folder was uploaded to the same GitHub repository, and the download button was linked directly to these files.

Map previews were implemented using Leaflet.js, allowing each dataset to be visualised in a mini-interactive map.

The shapefiles used for the previews were stored within the GitHub repository and rendered using Leaflet's GeoJSON functionality. For datasets sourced from the COCT WMS, the layers were accessed in real time using the appropriate WMS URLs and layer names.

Custom styling, including dataset-specific icons and colour schemes, was added through inline CSS and JavaScript. Additionally, a search functionality was implemented to allow users to filter and locate datasets more efficiently.

3.4.3. Motivation

A custom metadata catalogue was chosen over existing solutions due to practical and technical constraints. The team originally considered GeoNode, a widely used open-source platform for metadata management, but its installation and operation on a cloud-based VM presented challenges. GeoNode was primarily designed for deployment on locally installed servers with graphical interfaces and adapting it for a headless cloud-based server proved complex and impractical within the project timeline.

Instead, developing a custom solution using HTML provided greater control over the appearance, structure, and functionality of the metadata catalogue. This approach ensured that the catalogue could be tailored to the project's needs, integrated seamlessly with the GitHub-hosted web portal, and remained lightweight and easily accessible from any browser.

3.5. Interactive Web Map

3.5.1. Design

The interactive web map was designed to serve as a user-friendly interface that visualises all spatial data layers overlaid on an OpenStreetMap base layer. A grid-based map index is overlaid on the data, enabling users to navigate and identify datasets associated with specific spatial regions.

The interface is designed to be highly interactive, allowing users to toggle individual layers on and off, search for data by suburb, geolocate their current position, and search grid cells by number.

A key feature of the interactive map is its seamless integration with the metadata catalogue. Users can either open the metadata catalogue directly or interactively access it through the grid system. By selecting a specific grid cell, a side pane opens showing dataset categories relevant to that area. From there, users can view detailed metadata and download spatial data specific to the selected grid cell.

3.5.2. Implementation

The implementation of the interactive web map took a slightly different approach from the initial design plan. Rather than using the WMS generated by the cloud-hosted Geoserver, the team opted to use QGIS and QGIS Cloud for greater flexibility and integration. Spatial data was first imported into QGIS, where it was then uploaded and published to a cloud database using the QGIS Cloud plugin, which automatically generated a QGIS WMS service. This WMS was reconnected back into QGIS, where the final map composition was created.

To view the QGIS WMS, add the following link to QGIS or ArcGIS Pro:
https://wms.qgiscloud.com/wdxemi002/CDNGI_GeoPortal/

Once all layers were cartographically styled and the base map configured, the team used the qgis2web plugin to export the project as a Leaflet-based web map. This plugin allowed for the creation of a fully functional interactive map with built-in features such as layer control, search tools, and geolocation.

The exported map was hosted using GitHub Pages, and further customisations were made to the HTML and JavaScript code. These modifications enabled additional features such as:

- Search functionality for locating data by grid number or suburb name;
- User geolocation feature that identifies and zooms to the user's current position;
- Dynamic map index highlighting with cell number popups to indicate and identify the selected grid cell;
- Scale-dependent map index that displays a broader grid at zoomed-out levels and a finer grid when zoomed in;
- Interactive metadata side pane, triggered by clicking a grid cell, to display relevant dataset categories;
- Direct linking to specific metadata entries within the catalogue based on the selected grid and layer;
- Clickable project logo that redirects users to the geoportal’s homepage
- Loading screen displayed while spatial layers and components are initialising;
- Layer control panel to toggle visibility of specific spatial datasets;
- Zoom extent restriction to confining map navigation within the defined study area.

This level of customisation provided a polished and intuitive map interface tailored to the needs of end users.

3.5.3. Motivation

The decision to create a second WMS through QGIS Cloud, despite having an existing WMS on the cloud-based Geoserver, was driven by the advantages offered by QGIS and its plugin ecosystem. Using QGIS allowed for a streamlined, end-to-end workflow in a single environment, where data upload, styling, map creation, interactivity, and export could all be performed efficiently.

Additionally, the qgis2web plugin and QGIS Cloud services provided a smoother path for integrating the data with Leaflet, reducing the need to code a standalone Leaflet application from scratch. This approach not only saved time but also provided valuable learning opportunities, expanding the team's experience with QGIS, web publishing tools, and cloud-based spatial data management, skills highly relevant in real-world GIS practice.

3.6. Website

3.6.1. Design

The website functions as the central hub of the geoportal system, providing users with streamlined access to all available services, including the interactive map and metadata catalogue, through a user-friendly interface. It is designed to ensure seamless navigation via a clearly structured navigation bar, as well as intuitive launch buttons placed on the homepage. In addition to connecting system components, the website also presents detailed information on the datasets made available through the portal, the overall project scope, and the data development process. A resolute "How to Use" section is also included to guide users through the various functionalities of the interactive map and help them make full use of the portal's features.

The **link** to the **website**:

CDNGI Geoportal

3.6.2. Implementation

The website was fully custom-built, with each individual page created using tailored HTML code. All webpages are hosted on a GitHub web server, enabling free and accessible public access. The homepage integrates direct links to both the Metadata catalogue and the interactive web map, accessible either via the navigation bar or through prominent launch buttons. This modular design ensures users can move easily between components. The HTML source code for the website is openly available in the **GitHub repository**:

https://github.com/emilywood-uct/cdngi-geoportal,

Where the complete implementation details and structure of each custom page can be reviewed.

3.6.3. Motivation

The decision to develop the website using HTML code rather than relying on a template-based website builder like Wix was intentional. This approach offered full control over the site's layout, design, and embedded functionality, something not easily achievable through pre-built themes or constrained platforms. Moreover, hand-coding the website provided valuable real-world experience in front-end development, aligning with the types of skills often required in professional GIS and web-mapping roles. This method ensured that the resulting website was purpose-built, free from unnecessary features, and capable of delivering a smooth and tailored user experience without compromise.

4. Conclusion

As a Geoinformatics stream major, I found this project to be an incredibly important and relevant exercise that offered real insight into the kinds of tasks and responsibilities one might face in the professional GIS workplace, particularly in managing and maintaining spatial data infrastructures. It provided a practical experience in setting up a cloud-based virtual machine, configuring Geoserver, publishing WMS/WFS services, building a metadata catalogue, and designing an interactive web map and custom website using HTML and GitHub. I thoroughly enjoyed learning every component, especially the integration of spatial data services with web-based platforms and gaining direct experience with tools like QGIS Cloud, Leaflet, and the intricacies of setting up a Microsoft Azure Virtual Machine. The project was both valuable and rewarding, and my partner and I were very satisfied with the final outcome. Despite initial hurdles, we were able to overcome each challenge, which made the end result even more fulfilling and gave us a sense of confidence in applying these skills in real-world GIS environments.

5. Appendix - Data Dictionary

ArcGIS Pro – A professional desktop GIS application developed by Esri. It is used for advanced spatial analysis, data visualisation, and integration of geospatial data through services like WMS and WFS.

Batch Clip – A geoprocessing tool in GIS software that allows users to clip multiple spatial datasets at once to a defined boundary or extent, streamlining the process of spatial refinement.

Cartographically – Relating to the design and production of maps, particularly in terms of how spatial data is visually represented.

CDNGI – Chief Directorate: National Geospatial Information

Cloud-based server – A virtual server hosted on the internet (cloud), enabling remote access, improved scalability, and reduced downtime compared to traditional physical servers.

COCT – City of Cape Town

Context diagram – A high-level system diagram that defines the boundaries of a system and shows its interactions with external entities, such as users and data sources.

CSS – Cascading Style Sheets

Data refinement – The process of cleaning, transforming, and preparing raw spatial data to meet quality standards for storage, visualisation, and analysis.

Decomposition diagram – A hierarchical model that breaks down a system into smaller sub-processes or functions. It helps visualise the structure and scope of a system for better understanding, development, and management.

DFD – Data Flow Diagram

DEM – A Digital Elevation Model is a raster representation of the earth's surface that depicts elevation values, used for terrain analysis, 3D modelling, and hydrological applications.

ERD – Entity-Relationship Diagram

First-level Data Flow Diagram – A more detailed breakdown of a specific process from the top-level DFD, illustrating the internal data flows, inputs, and outputs involved.

GeoJSON – A format for encoding a variety of geographic data structures using JavaScript Object Notation such as JSON. It is widely used for representing simple vector geometries such as points, lines, and polygons.

Geoportal – A web-based platform that provides access to geospatial data and related services such as visualisation, discovery, and download.

Geoserver – Geoserver is an open-source server application designed to share, publish, and edit geospatial data. It allows users to serve maps and data from a variety of formats using standard web services such as WMS and WFS.

Geolocation – A technology that identifies a user's real-world geographic location through GPS, Wi-Fi, or IP address, and integrates this position into a map or web application.

GitHub – A web-based platform for version control and hosting code repositories. It can also be used to host static websites and downloadable content, such as spatial datasets, when paired with HTML and JavaScript.

GIS – Geographic Information Systems

Grid Index – A spatial reference system that divides a geographic area into square or rectangular cells, allowing datasets to be organised and accessed based on location within the grid.

GPS – Global Positioning System

HTML – HyperText Markup Language is the standard language used to create and structure content on the web. It defines the elements of a webpage such as text, links, images, and interactive forms.

Interactive map configuration – The customisation of a web map's appearance and behaviour, such as defining which layers are included, styling features, setting zoom levels, and enabling user interactions like clicking or searching.

Interactive Web Map – A digital map displayed in a browser that allows users to interact with spatial data through zooming, panning, searching, toggling layers, and accessing information about map features in real time.

Interoperability – The ability of different systems, applications, or components to work together seamlessly by using common standards or protocols. In GIS, this often involves enabling cross-platform data access via WMS/WFS services.

IP – Internet Protocol

JavaScript – A scripting language used primarily for adding interactive behaviour and dynamic content to websites. In web mapping, JavaScript is commonly used to enable functions like map zooming, clicking, and data filtering.

Leaflet – An open-source JavaScript library used to build lightweight and interactive web maps. It is commonly used in web GIS applications due to its simplicity and flexibility.

Leaflet.js – An open-source JavaScript library used for building interactive web maps. It allows developers to load and visualise spatial data directly in a browser using formats like GeoJSON or external WMS layers.

Metadata Catalogue – A structured repository of descriptive information about datasets, typically including fields like title, abstract, keywords, spatial extent, and coordinate system. It helps users understand, discover, and access spatial data.

Metadata catalogue – A structured collection of information describing spatial datasets, including details like source, format, coverage, and accessibility.

Metadata creation – The process of generating structured information that describes the content, quality, condition, and other characteristics of spatial datasets to support discovery and use.

Microsoft Azure – A cloud computing platform by Microsoft offering virtual machines, storage, and networking services for scalable and flexible cloud infrastructure.

OpenStreetMap – A collaborative project that provides free, editable geographic data and mapping to the public. It serves as a commonly used base map in web mapping applications.

OGC – Open Geospatial Consortium

Package diagram – A UML diagram used to visually organise and group related parts of a system into packages, making it easier to understand the system's modular structure and relationships.

Navigation Bar – A graphical interface element that contains links or buttons for navigating between different pages or sections of a website, designed to improve user experience and facilitate site navigation.

Port Forwarding – A networking approach that allows external devices to connect to services on a private network by mapping an external port to an internal one. In this project, port 8080 was forwarded to allow public access to Geoserver.

QGIS – A free and open-source desktop GIS software that enables users to generate, visualise, analyse, and publish geographical data.

QGIS Cloud – A cloud-based platform that extends the capabilities of QGIS by allowing users to publish their projects and data online. It enables public or private WMS without the need for a self-managed server.

qgis2web Plugin – A QGIS extension that exports a QGIS map project as a fully interactive web map using either Leaflet.js or OpenLayers. It supports automatic conversion of layers, styles, and widgets into HTML, CSS, and JavaScript code.

qgis2web plugin – A plugin in QGIS that allows users to export their GIS projects as interactive web maps using JavaScript libraries like Leaflet or OpenLayers, facilitating easy sharing of spatial data on the web.

Shapefiles – A widely used geospatial vector data format for GIS software.

SSH – Secure Shell

SDI – Spatial Data Infrastructure is a framework of spatial data, metadata, users, and tools that are interactively connected to enable effective data sharing and usage.

Spatial database – A type of database that is optimised to store and query data representing objects defined in a geometric space, such as points, lines, and polygons.

System boundaries – The defined limits of a system, indicating what is inside the system (internal processes and data) versus what lies outside (external entities or actors).

Systemd service – A system management daemon on Linux-based operating systems that manages the startup and execution of background services. It ensures programs like Geoserver launch automatically on boot and remain running.

Top-level Data Flow Diagram – A simplified visual representation of a system's main processes and how data moves between them and external actors, used to understand overall system functionality.

Ubuntu – A Linux-based open-source operating system commonly used for servers and development environments. It provides a stable interface for installing and running web-based applications such as Geoserver.

UML – Unified Modelling Language

URL – Uniform Resource Locator

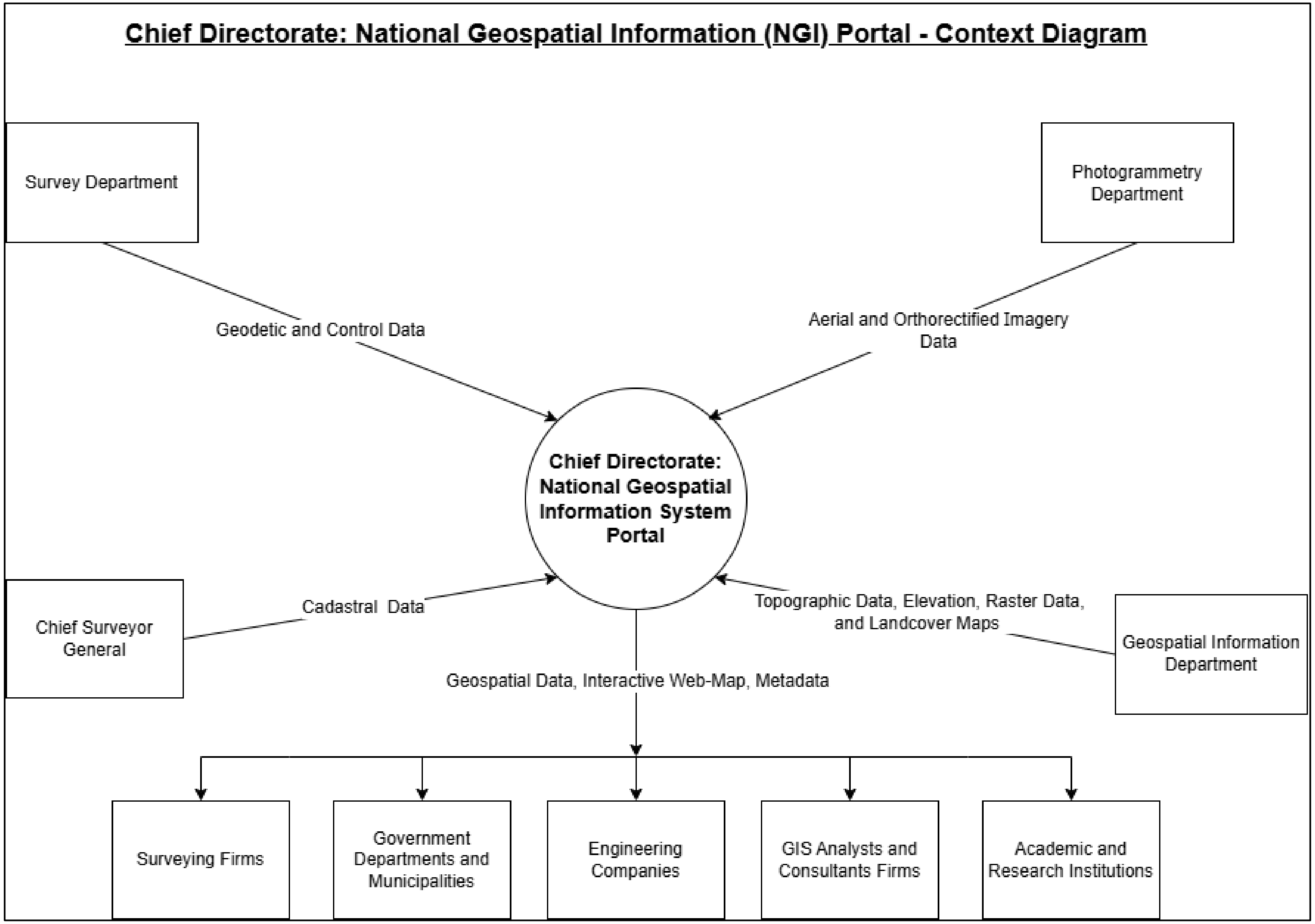
VM – A Virtual Machine is a virtualised computing environment that simulates a physical computer. It enables the installation and running of software (such as Geoserver) on cloud infrastructure, offering flexibility and remote accessibility.

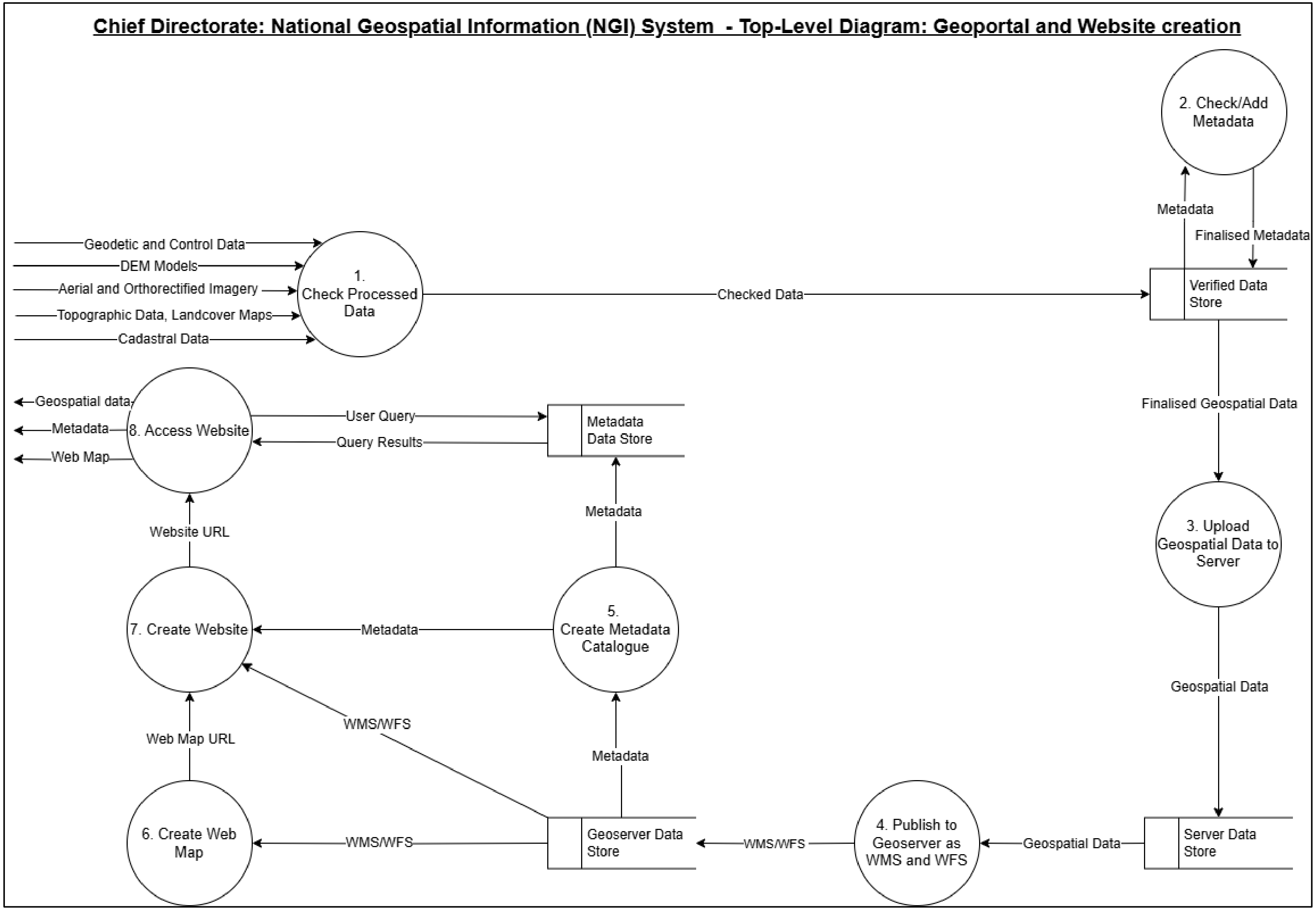
Web development – The creation and maintenance of websites and web-based applications, including coding, design, and functionality integration. In this project, it includes building HTML pages for the website, metadata catalogue, and interactive map.

WFS – A Web Feature Service is an OGC standard for serving raw vector features over the web. Unlike WMS, WFS allows users to query, retrieve, and edit actual geospatial features in applications such as QGIS or ArcGIS Pro.

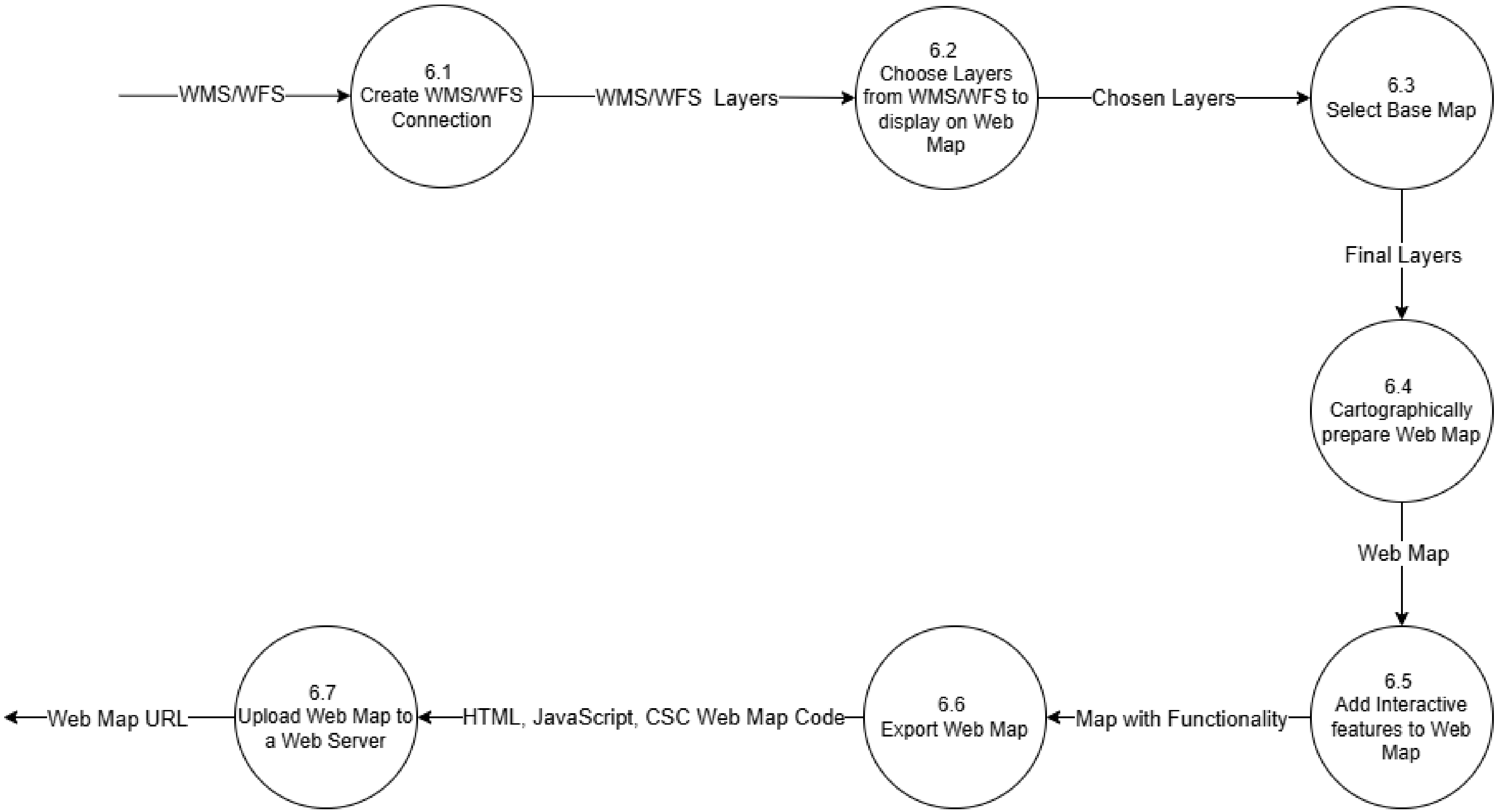
WMS – A Web Map Service standard protocol developed by the OGC that delivers rendered map images from geospatial data stored on a server. It is typically used for visualising spatial data without direct access to the raw data.

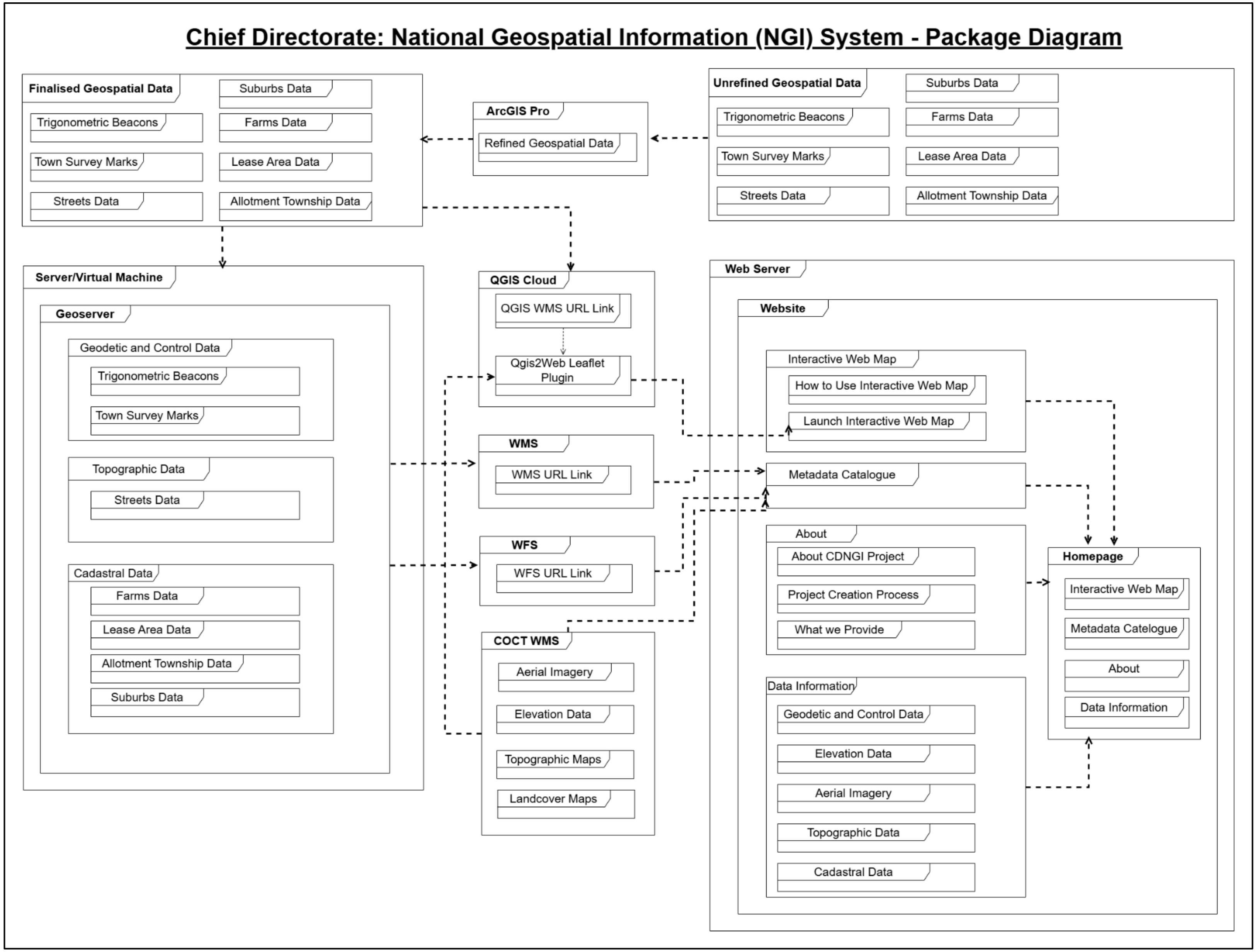
6. Appendix – Diagrams

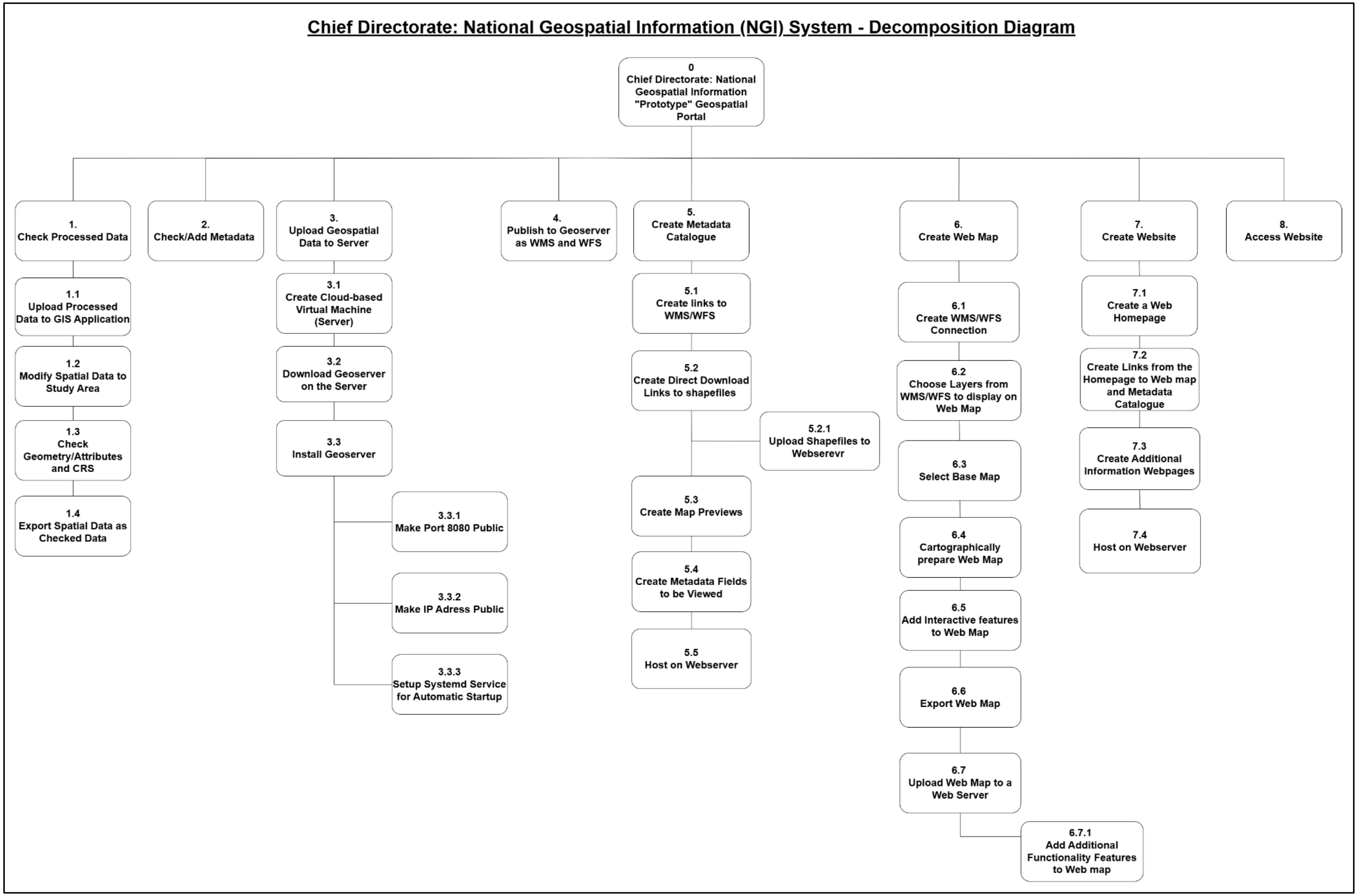



Chief Directorate: National Geospatial Information (NGI) System - Top-Level Diagram: Geoportal and Website creation
2. Check/Add Metadata
Metadata
Finalised Metadata
Geodetic and Control Data
DEM Models
Aerial and Orthorectified Imagery
Topographic Data, Landcover Maps
Cadastral Data
1. Check Processed Data
Checked Data
Verified Data Store
Geospatial data
Metadata
Web Map
8. Access Website
User Query
Query Results
Metadata Data Store
Finalised Geospatial Data
Metadata
Website URL
3. Upload Geospatial Data to Server
7. Create Website
Metadata
5. Create Metadata Catalogue
Geospatial Data
WMS/WFS
Web Map URL
Metadata
6. Create Web Map
WMS/WFS
Geoserver Data Store
WMS/WFS
4. Publish to Geoserver as WMS and WFS
Geospatial Data
Server Data Store

Chief Directorate: National Geospatial Information (NGI) System - First-Level Diagram: Create Web Map




Chief Directorate: National Geospatial Information (NGI) System - Package Diagram
Finalised Geospatial Data
Trigonometric Beacons
Town Survey Marks
Streets Data
Suburbs Data
Farms Data
Lease Area Data
Allotment Township Data
ArcGIS Pro
Refined Geospatial Data
Unrefined Geospatial Data
Trigonometric Beacons
Town Survey Marks
Streets Data
Suburbs Data
Farms Data
Lease Area Data
Allotment Township Data
Server/Virtual Machine
Geoserver
Geodetic and Control Data
Trigonometric Beacons
Town Survey Marks
Topographic Data
Streets Data
Cadastral Data
Farms Data
Lease Area Data
Allotment Township Data
Suburbs Data
QGIS Cloud
QGIS WMS URL Link
Qgis2Web Leaflet Plugin
WMS
WMS URL Link
WFS
WFS URL Link
COCT WMS
Aerial Imagery
Elevation Data
Topographic Maps
Landcover Maps
Web Server
Website
Interactive Web Map
How to Use Interactive Web Map
Launch Interactive Web Map
Metadata Catalogue
About
About CDNGI Project
Project Creation Process
What we Provide
Data Information
Geodetic and Control Data
Elevation Data
Aerial Imagery
Topographic Data
Cadastral Data
Homepage
Interactive Web Map
Metadata Catelogue
About
Data Information


Chief Directorate: National Geospatial Information (NGI) System - Decomposition Diagram
0
Chief Directorate: National Geospatial Information "Prototype" Geospatial Portal
1.
Check Processed Data
1.1
Upload Processed Data to GIS Application
1.2
Modify Spatial Data to Study Area
1.3
Check Geometry/Attributes and CRS
1.4
Export Spatial Data as Checked Data
2.
Check/Add Metadata
3.
Upload Geospatial Data to Server
3.1
Create Cloud-based Virtual Machine (Server)
3.2
Download Geoserver on the Server
3.3
Install Geoserver
3.3.1
Make Port 8080 Public
3.3.2
Make IP Adress Public
3.3.3
Setup Systemd Service for Automatic Startup
4.
Publish to Geoserver as WMS and WFS
5.
Create Metadata Catalogue
5.1
Create links to WMS/WFS
5.2
Create Direct Download Links to shapefiles
5.2.1
Upload Shapefiles to Webserevr
5.3
Create Map Previews
5.4
Create Metadata Fields to be Viewed
5.5
Host on Webserver
6.
Create Web Map
6.1
Create WMS/WFS Connection
6.2
Choose Layers from WMS/WFS to display on Web Map
6.3
Select Base Map
6.4
Cartographically prepare Web Map
6.5
Add Interactive features to Web Map
6.6
Export Web Map
6.7
Upload Web Map to a Web Server
6.7.1
Add Additional Functionality Features to Web map
7.
Create Website
7.1
Create a Web Homepage
7.2
Create Links from the Homepage to Web map and Metadata Catalogue
7.3
Create Additional Information Webpages
7.4
Host on Webserver
8.
Access Website