SWDD -

**Cloud Base 2D Material Identification**

#### Software Design Document

Name (s): Xuanzhi Zhang

Date: (02/11/2025)

**TABLE OF CONTENTS**

1.0 INTRODUCTION 4

1.1 Purpose 4

1.2 Scope 4

1.3 Overview 4

1.4 Reference Material 4

1.5 Definitions and Acronyms 4

2.0 SYSTEM OVERVIEW 4

3.0 SYSTEM ARCHITECTURE 4

3.1 Architectural Design 4

3.2 Decomposition Description 5

3.3 Design Rationale 5

4.0 DATA DESIGN 5

4.1 Data Description 5

4.2 Data Dictionary 5

5.0 COMPONENT DESIGN 5

6.0 HUMAN INTERFACE DESIGN 5

6.1 Overview of User Interface 5

6.2 Screen Images 6

6.3 Screen Objects and Actions 6

7.0 REQUIREMENTS MATRIX 6

8.0 APPENDICES 6

## INTRODUCTION

## Purpose

Identify the purpose of this SWDD and its intended audience. (e.g. “This software design document describes the architecture and system design of XX. ….”).

## Scope

Provide a description and scope of the software and explain the goals, objectives and benefits of your project. This will provide the basis for the brief description of your product.

## Overview

Provide an overview of this document and its organization.

## Reference Material

*This section is optional.*

List any documents, if any, which were used as sources of information for the test plan.

## Definitions and Acronyms

*This section is optional.*

Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SWDD. These definitions should be items used in the SWDD that are most likely not known to the audience.

## SYSTEM OVERVIEW

1. **General Description**

This software is designed to automate the identification of material layers in 2D materials using cloud-based artificial intelligence and machine learning. In condensed matter physics and materials science, researchers often rely on manual inspection of microscope images to determine the number of layers in materials such as Graphene, MoS₂, and WSe₂. This process is labor-intensive, prone to human error, and lacks standardization.

This software solution leverages deep learning-based image processing to analyze microscopic images, classify material layers, and provide accurate results in real-time. By integrating cloud-based AI models, the system eliminates the need for specialized hardware and makes automated layer identification accessible to a wider range of researchers.

1. **Context and Background**
   1. **Current Challenges in 2D Material Layer Identification**

The identification of material layers in two-dimensional (2D) materials is a critical task in condensed matter physics and materials science. Traditional methods rely on manual inspection of optical microscope images, where researchers visually estimate the number of layers based on contrast variations.

This process is:

* **Labor-intensive:** Requires significant time and expertise.
* **Prone to human error:** Subject to inconsistencies in interpretation.
* **Non-standardized:** Different labs use different calibration techniques, affecting reproducibility.

To address these issues, various image-based AI solutions have been developed, leveraging deep learning and image processing to automate the identification process. These solutions have improved accuracy and efficiency but still face several limitations, particularly in scalability, accessibility, and integration with existing research workflows.

1. **Existing Image-Based and Cloud-Based AI Solutions**

Several AI-powered approaches have been explored to automate 2D material layer detection:

1. Deep Learning-Based Optical Microscopy Analysis
   * Uses convolutional neural networks (CNNs) to detect and classify flakes in microscope images.
   * Achieves high accuracy (up to 95%) in identifying monolayer vs. multilayer regions.
   * Often requires large labeled datasets and retraining for different materials.
2. Cloud-Based Image Processing Platforms (e.g., ZEISS Arivis Cloud, DLgram01)
   * Enables researchers to process images online without needing local computational resources.
   * Provides general-purpose segmentation tools rather than specialized layer identification models.
   * Lacks customization for 2D materials and does not support human-in-the-loop refinement.
3. Traditional Image Processing Techniques
   * Uses contrast enhancement, color thresholding, and edge detection for layer segmentation.
   * Requires manual fine-tuning of parameters for different materials and imaging conditions.
   * Less robust compared to AI-driven methods in handling variable lighting conditions.
4. **Key Differentiators of This Software**

This software addresses the gaps in existing solutions by providing a specialized, cloud-based AI platform tailored for automated 2D material layer identification. The following features distinguish it from current approaches:

* 1. **Fully Cloud-Based AI Processing for Real-Time Analysis**

Existing AI models often require local computation or specialized hardware, making them inaccessible to many researchers. This software provides:

* A web-based interface that allows users to upload optical microscope images and receive instant AI-driven layer identification results.
* Scalable cloud infrastructure, eliminating the need for expensive GPU-based local processing.
* Multi-user access, enabling collaborative research and large-scale data processing.
  1. **Pre-Trained AI Models Specialized for 2D Materials**

Unlike general-purpose AI segmentation tools, this software is specifically optimized for 2D materials, including Graphene, MoS₂, and WSe₂. Advantages include:

* Pre-trained deep learning models that do not require manual retraining.
* Material-specific calibration, improving detection accuracy across different substrates.
* Automatic color correction algorithms to compensate for lighting and microscope variations.
  1. **Human-in-the-Loop Refinement for Increased Accuracy**

Most AI solutions provide static, non-editable results. This software introduces:

* An interactive correction tool, allowing users to manually adjust AI predictions to improve accuracy.
* Adaptive learning, where user corrections refine the model over time.
* Overlaid visualization, enabling users to compare raw images and AI-generated layer maps.
  1. **High-Throughput Batch Processing for Large-Scale Research**

Existing tools often process images one at a time. This software supports:

* Bulk uploads, allowing users to analyze multiple microscope images simultaneously.
* Automated processing pipelines, reducing manual workload.
* Exportable reports, summarizing results in structured formats for research documentation.
  1. **API Integration for Automated Research Workflows**

Current AI-based image analysis solutions lack seamless integration with existing lab workflows. This software provides:

* REST API support, allowing automated image uploads from lab microscopes.
* Programmatic access, enabling researchers to incorporate AI-powered identification into custom research pipelines.
* Integration with cloud storage, ensuring data security and accessibility across research teams.
  1. **Open-Access Model with Community Contribution**

Many commercial platforms are closed-source and require licensing fees. This software promotes accessibility by:

* Offering a freemium model where basic AI models are free, with premium features for large-scale analysis.
* Supporting community dataset contributions to improve model accuracy over time.
* Providing an open dataset repository, allowing researchers to share labeled microscope images.

**Conclusion**

By addressing the limitations of existing AI-based solutions, this software provides a scalable, specialized, and highly accessible platform for automated 2D material layer identification. With cloud-based real-time processing, interactive AI refinement, high-throughput support, and seamless API integration, it offers a comprehensive and innovative solution for researchers in condensed matter physics and materials science.

## SYSTEM ARCHITECTURE

## Architectural Design

The system is designed using a **cloud-native microservices architecture**, incorporating **serverless computing**, **containerization**, and **RESTful APIs** to ensure **scalability, maintainability, and high availability**. The primary goal of this architecture is to provide an **efficient and automated solution for material layer identification** while ensuring **secure user management and image processing capabilities**.

The architecture is divided into two major subsystems:

**New User Service** – Handles all user-related operations, including registration and authentication.

**Images Service –** Manages image processing, including upload, resizing, and AI-based analysis.

These services communicate with each other using RESTful APIs, and the entire system is deployed in a cloud environment using AWS Lambda and Docker-based containers.

## Decomposition Description

Provide a decomposition of the subsystems in the architectural design. Supplement with text as needed. You may choose to give a functional description or an object­oriented (OO) description. For a functional description, put top­level data flow diagram (DFD) and structural decomposition diagrams. For an OO description, put subsystem model, object diagrams, generalization hierarchy diagram(s) (if any), aggregation hierarchy diagram(s) (if any), interface specifications, and sequence diagrams here.

## Design Rationale

Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn’t choose them.

## DATA DESIGN

## Data Description

Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized. List any databases or data storage items.

## Data Dictionary

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.2, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters.

## COMPONENT DESIGN

In this section, we take a closer look at what each component does in a more systematic way. If you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary.

## HUMAN INTERFACE DESIGN

## Overview of User Interface

Describe the functionality of the system from the user’s perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

## Screen Images

Display screenshots showing the interface from the user’s perspective. These can be hand­ drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

## Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

## REQUIREMENTS MATRIX

Provide a cross­reference that traces components and data structures to the requirements in your softwarerequirements specification (SWRS) document.

Use a tabular format to show which system components satisfy each of the functional requirements from the SWRS. Refer to the functional requirements by the numbers/codes that you gave them in the SWRS.

## APPENDICES

*This section is optional.*

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.