



Voyis Imaging Inc.
120 Randall Drive, Unit E
Waterloo ON, Canada, N2V 1C6
+1 519.489.0005

Project Brief: Distributed Imaging Services

Objective:

Design and implement a distributed image processing system written in modern C++ capable of generating, processing and storing data. This project is designed to evaluate your ability to design a robust, modular system, utilize Inter-Process Communication (IPC), integrate third-party libraries, and adhere to high standards of C++ development.

Timeline:

- **Project Duration:** 2 weeks from the start date.
-

Notes:

- Candidates are encouraged to use publicly available datasets for testing and demonstration.
- Questions or requests for clarification can be submitted via email during the project period.

System Architecture:

The system should consist of 3 distinct applications: one to generate data, one to process data and one to save data.

Functional Requirements

Each of the three components must run as a separate, isolated process and communicate exclusively via an IPC mechanism of your choosing.

App 1: Image Generator

Purpose: Simulate a data source such as a camera by reading a set of images from disk and publishing them via IPC to other applications.

Requirements:

1. As input, the app must take a folder location containing the images to be sent.



Voyis Imaging Inc.
120 Randall Drive, Unit E
Waterloo ON, Canada, N2V 1C6

+1 519.489.0005

2. Read an arbitrary number of images from a specified location then package and send the image data via IPC to App 2.
3. Image data should be published continuously until the application is stopped. If all images from the input folder have been published, loop over the folder again... forever.
4. The app should be able to handle images of varying sizes and resolutions (e.g. few KB to >30MB)

[App 2: Feature Extractor](#)

Purpose: Receive image data and process it to find key features then publish the original image and the extracted features for other applications to consume.

Requirements:

1. This app should listen for and receive images published by App 1
2. Use the OpenCV library to perform Scale-Invariant Feature Transform (SIFT) detection to extract key points from the received images
3. Once processed, the image data along with the key points (from the SIFT feature extractor) should be sent via IPC to App 3

[App 3: Data Logger](#)

Purpose: Receive processed data (image data + key points) and save the data for future analysis.

Requirements:

1. This app should listen for and receive an arbitrary amount of data published by App 2.
2. Store the received data to a database of your choosing.

[Additional Requirements:](#)

1. The 3 applications should be as loosely coupled as possible and not crash if another application in the system exits or restarts.
2. The applications should be capable of starting in any order
3. The project should include instructions outlining how to build and run the applications. These instructions should be straight forward enough that anyone with basic Software Engineering knowledge can build and run the applications.



Voyis Imaging Inc.
120 Randall Drive, Unit E
Waterloo ON, Canada, N2V 1C6

+1 519.489.0005

4. The apps must build and run in Linux. If you're developing in Windows, you can use WSL

Deliverables:

1. Source code and other necessary files hosted on Github, including a README file with build and run instructions
 2. A brief demo video showing the applications running
 3. Bonus: Basic design document that outlines the software architecture
-

Notes: Candidates are free to use any available tools out there including AI/LLM tools. If your project meets the requirements, we will invite you a face-to-face interview in our office to talk about the project and ask you additional technical questions.