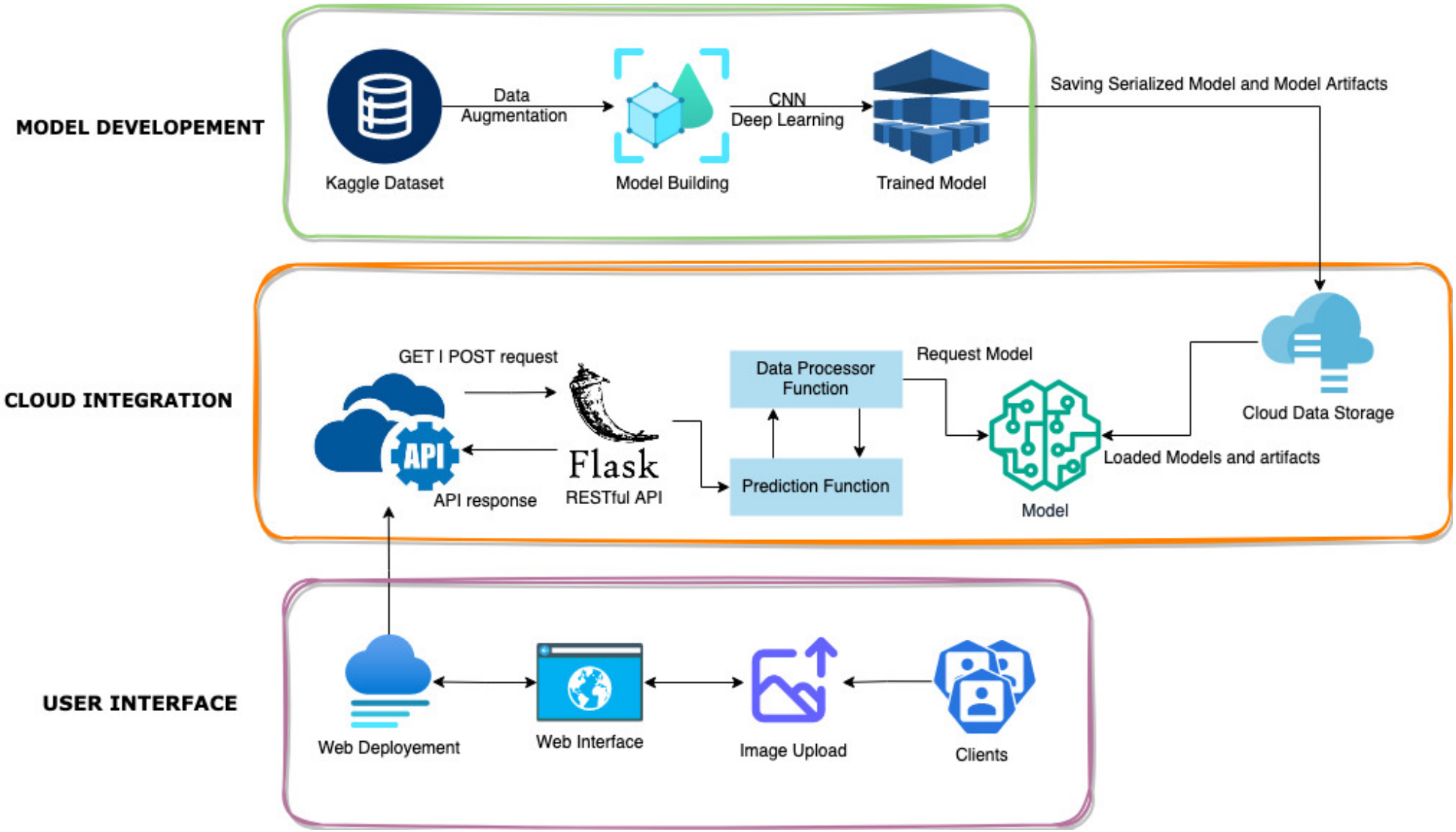
Project Design Phase-III
Technology Stack (Architecture & Stack)

Date	27 October 2023
Team ID	Team - 592706
Project Name	PoxVisio: A Deep Learning Expedition into Monkeypox Skin Lesions
Team Members	1) Atharva Pravin Navghane – 21BCE0083 2) Mrudul Sunil Patil – 21BCE3386 3) Onkar Hule – 21BCE3363 4) Mehul Gupta – 21BCE3897

Technical Architecture Diagram:



A) Technical Architecture:

The technical architecture for the Monkeypox Classification project would include the following components:

1. User Interface: Flask application with a simple web interface for user interaction.
2. Flask Application: This serves as the backend for the application and integrates the ResNet50 model for image analysis.
3. ResNet50 Model: Deep learning model implemented using TensorFlow and Keras for image analysis and classification.
4. Kaggle API: Utilized to fetch the Monkeypox Skin Lesion Dataset (MSLD) from Kaggle.
5. Cloud Storage: Used to store and manage the dataset and trained model.
6. Cloud Computing: Infrastructure for scalable and reliable deployment of the Flask application and ResNet50 model.
7. Web Server: Serving the Flask application and handling requests.

B) Open Source Frameworks:

1. Flask: A micro web framework in Python for building the web application.
2. TensorFlow: Open-source deep learning framework for building and training neural networks.
3. Keras: Open-source deep learning API written in Python, used as an interface for TensorFlow.
4. NumPy: A fundamental package for scientific computing with Python, used for numerical operations on images.
5. Pandas: A powerful data analysis and manipulation library for Python, utilized for handling datasets.

C) Third-party APIs:

1. Kaggle API: Used for accessing the Monkeypox Skin Lesion Dataset (MSLD) from Kaggle.
2. Cloud API

D) Cloud Deployment:

1. Amazon Web Services (AWS): Cloud platform for scalable, reliable, and secure deployment of the Flask application and ResNet50 model. Services like Amazon S3 can be used for cloud storage.
2. Google Cloud Platform (GCP): An alternative cloud platform for deploying and managing the application and model. Google Cloud Storage can be used for storing datasets and models.
3. Microsoft Azure: Another option for deploying the application. Azure Blob Storage can be used for dataset and model storage.

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User interaction with the application	HTML, CSS, JS, Flask
2.	Flask Application	Backend for the application	Python, Flask
3.	ResNet50 Model	Model Model for image analysis and classification	TensorFlow, Keras, Python
4.	Kaggle API	Fetching the Monkeypox Skin Lesion Datase	Kaggle API
5.	Cloud Storage	Data and model storage	AWS S3 / Google Cloud Storage / Azure Blob Storage
6.	Cloud Computing	Infrastructure for deployment	AWS, GCP, Azure
7.	Web Server	Serves the Flask application	AWS, GCP, Azure

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frameworks used for the project	Flask, TensorFlow, PyTorch, NumPy, Pandas
2.	Security Implementations	Security measures implemented	HTTPS, OAuth / AWS IAM / GCP IAM / Azure IAM
3.	Scalable Architecture	Architecture scalability justification	Load Balancers, Microservices
4.	Availability	Ensuring application availability	Failover systems, Disaster Recovery, AWS Availability Zones, GCP Regions, Azure Regions
5.	Performance	Design considerations for performance	Caching, CDNs, High-performance computing