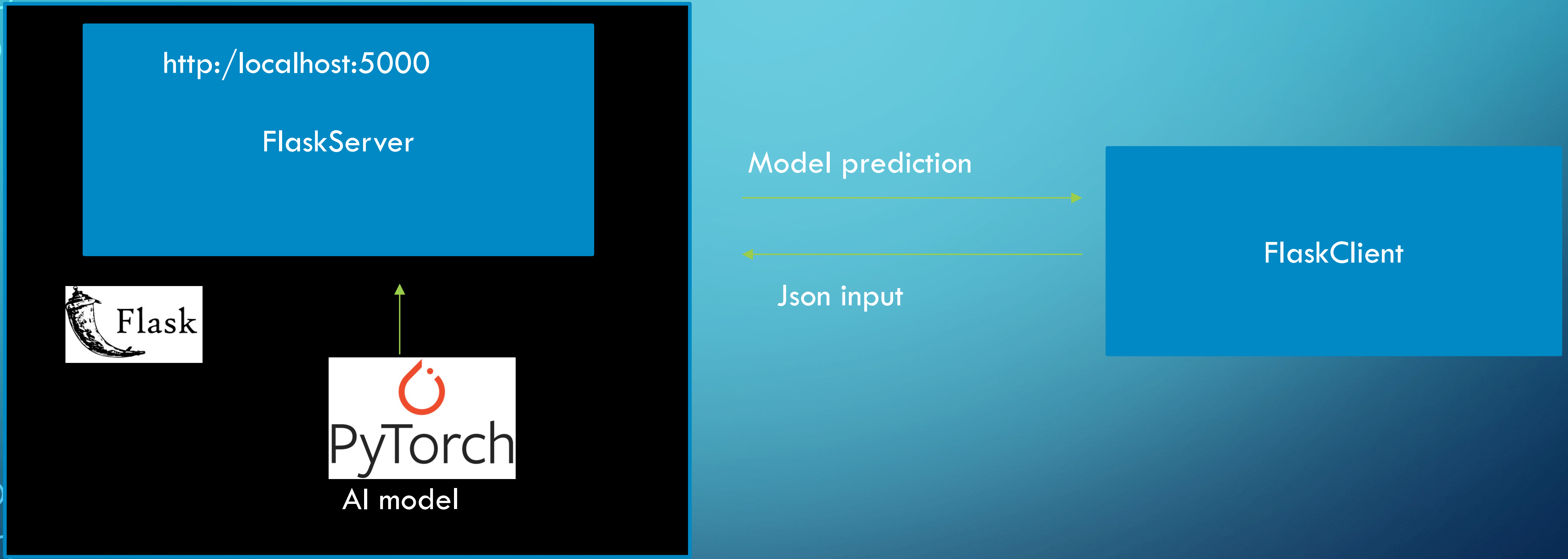
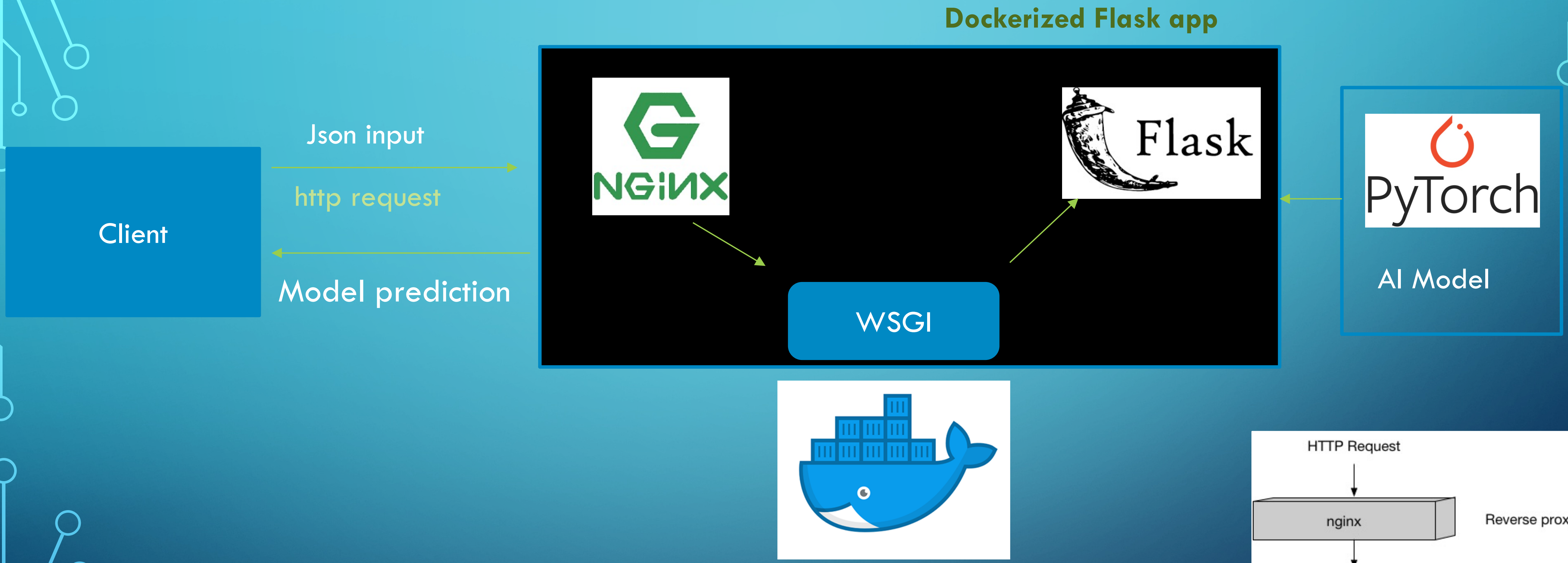
SAGEMAKER INFERENCE PIPELINE

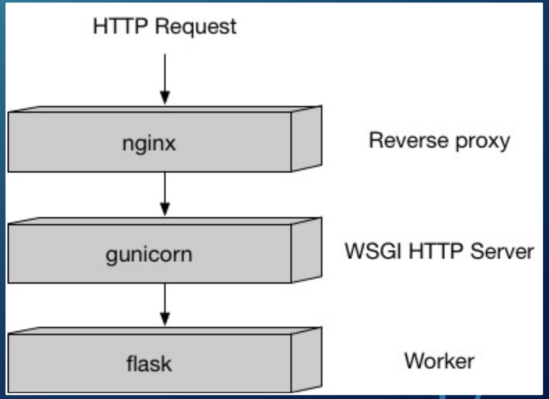
PATH TO AI

ORIGINAL MODEL



MODEL LOCAL SERVER

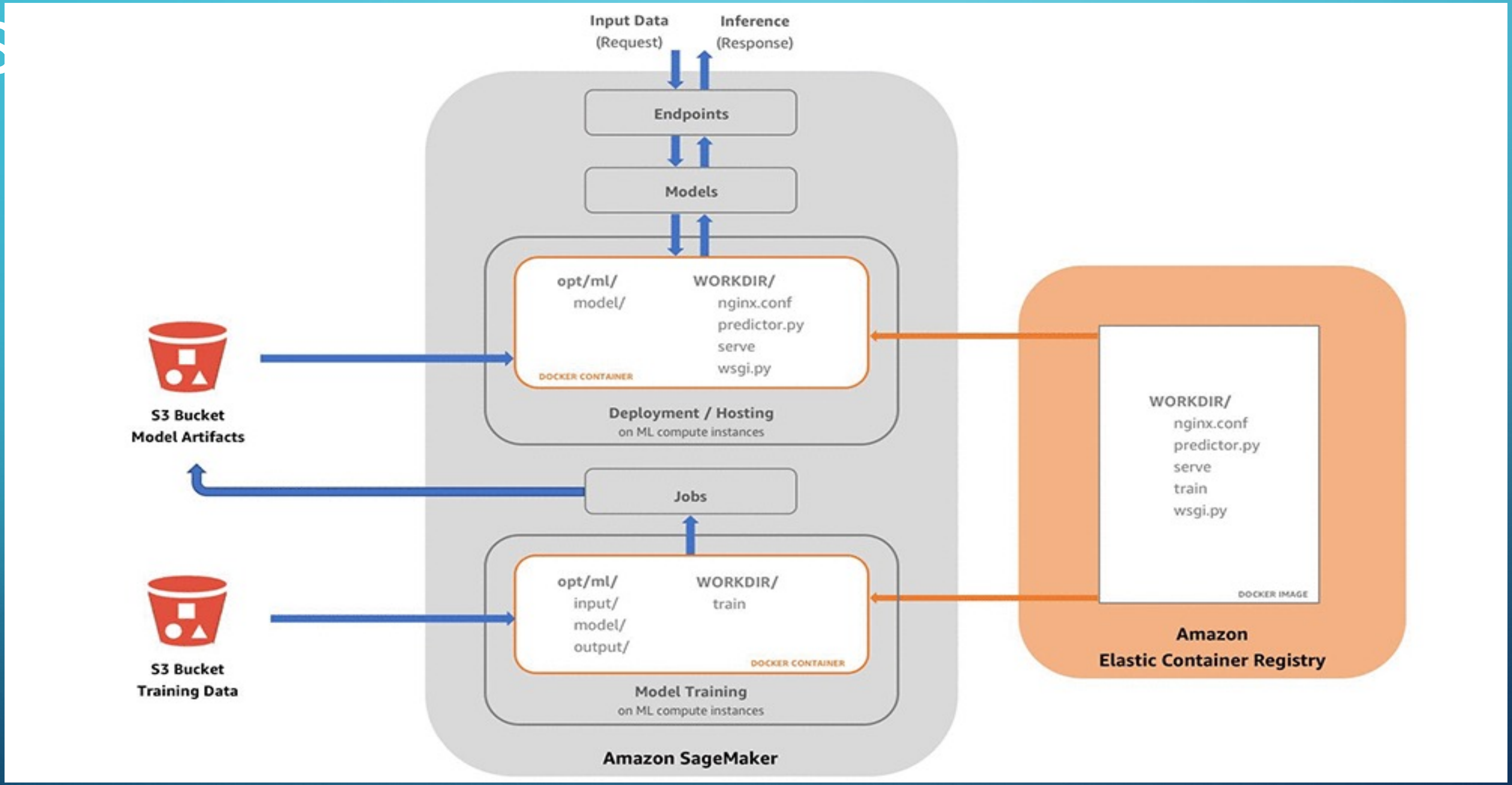




THE STRUCTURE OF THE CODE

- A. **Dockerfile:** This will install all the required dependencies in the docker image and will push 5 applications into the docker image. These 5 applications are:
  - **train:** The main program for training the model. When you build your own algorithm, you'll edit this to include your training code.
  - **serve:** The wrapper that starts the inference server. In most cases, you can use this file as-is.
  - **wsgi.py:** The start-up shell for the individual server workers. This only needs to be changed if you changed where predictor.py is located or is named.
  - **predictor.py:** The algorithm-specific inference server. This is the file that you modify with your own algorithm's code.
  - **nginx.conf:** The configuration for the nginx master server that manages the multiple workers. You can change the parameters (**number of workers and timeout**) based upon your requirements.
- B. **src:** The directory that contains the application to run in the container.
- C. **local-test:** A directory containing scripts and **model weights** and setup for running a simple training and inference jobs locally so that you can test that everything is set up correctly.

PH OHC MODEL DEPLOYMENT TO AWS S



WORKFLOW

- **Dockerized Flask Application:** It is a docker based web-application which will be used to get the predictions of the input incoming from the customer.
- **Deploy the Docker Application to ECR:** The docker containers can be stored in Elastic Container Registry which can then be used to deploy the docker based models on even bigger EC2 machines.
- **Upload model weights to S3:** The pytorch model weights are stored in S3 bucket.
- **AWS Sagemaker endpoint deployment:** The Docker-based web-app designed will be deployed as an endpoint so that it can be used by your project to autoscale the endpoint. There is one more advantage of deploying the model in sagemaker i.e. the Linux machine on which the docker application will be deployed can be controlled by you, thus based upon your requirements you can either use an EC2 machine having high RAM or EC2 machine having low RAM.
- **AutoScale the Sagemaker endpoint:** Once the sagemaker-endpoint is deployed, we will autoscale it to handle concurrent requests. Believe me guys usually autoscaling takes a lot of time, this method won't take more than 5 minutes.