tinyML® EMEA

Enabling Ultra-low Power Machine Learning at the Edge

June 26 - 28, 2023

www.tinyML.org

EDGE IMPULSE

`pip install edgeimpulse`

A programmatic approach to automate your MLOps Pipelines

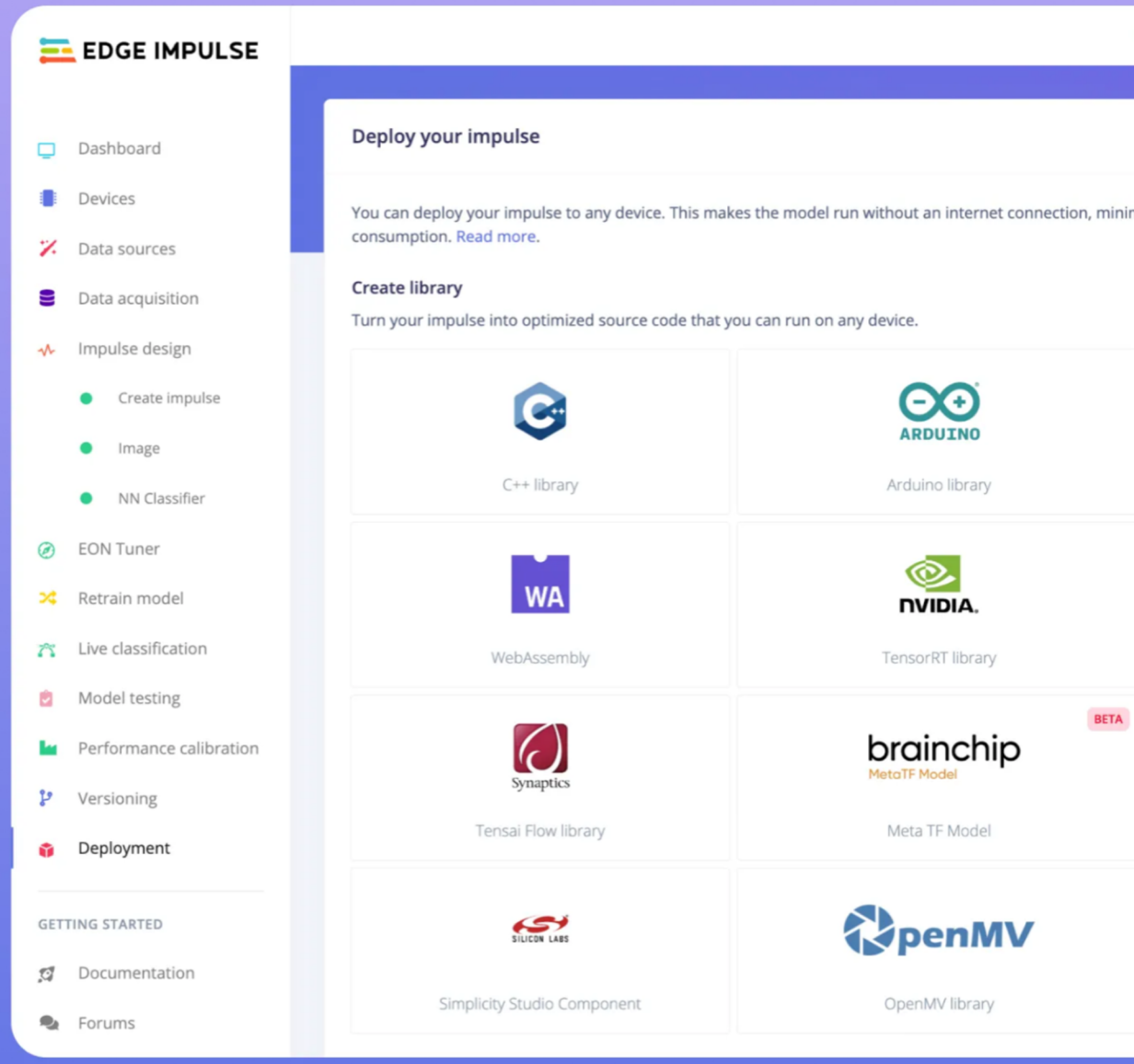
Louis Moreau

Developer Relations Lead Engineer, Edge Impulse

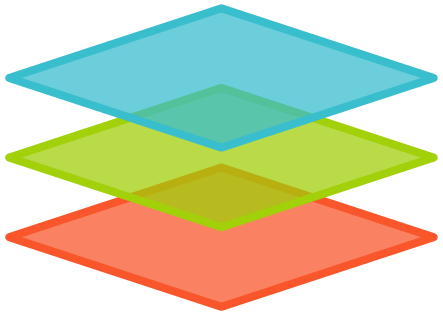
June 27, 2023

Introduction

We started our journey by building tools to ease AI for embedded engineers

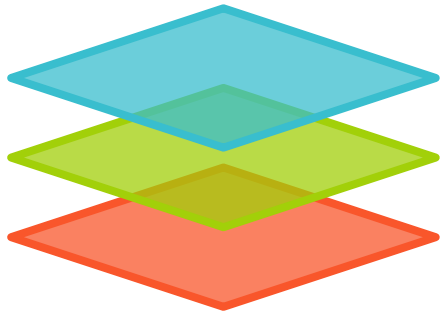


Now we also build tools for domain experts to deploy models on edge devices...



Python SDK

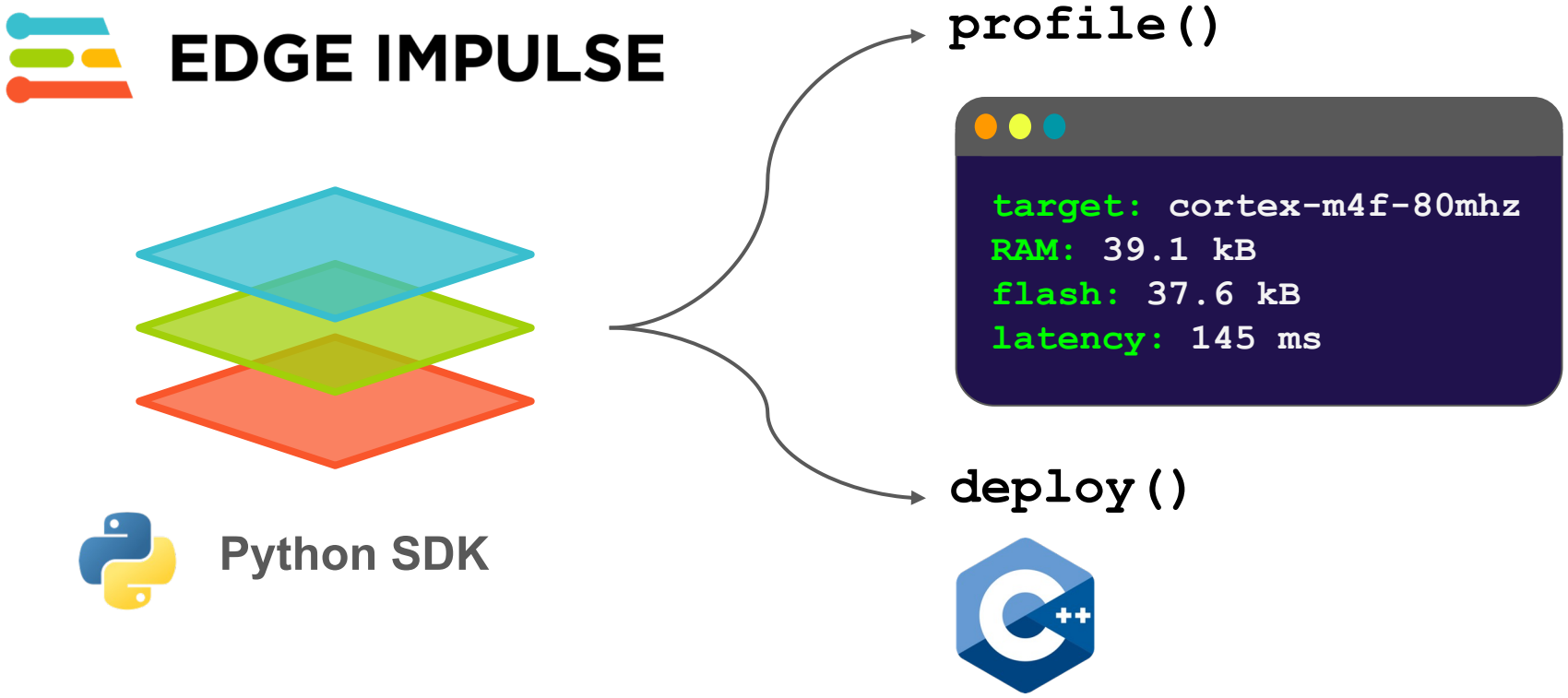
EDGE IMPULSE

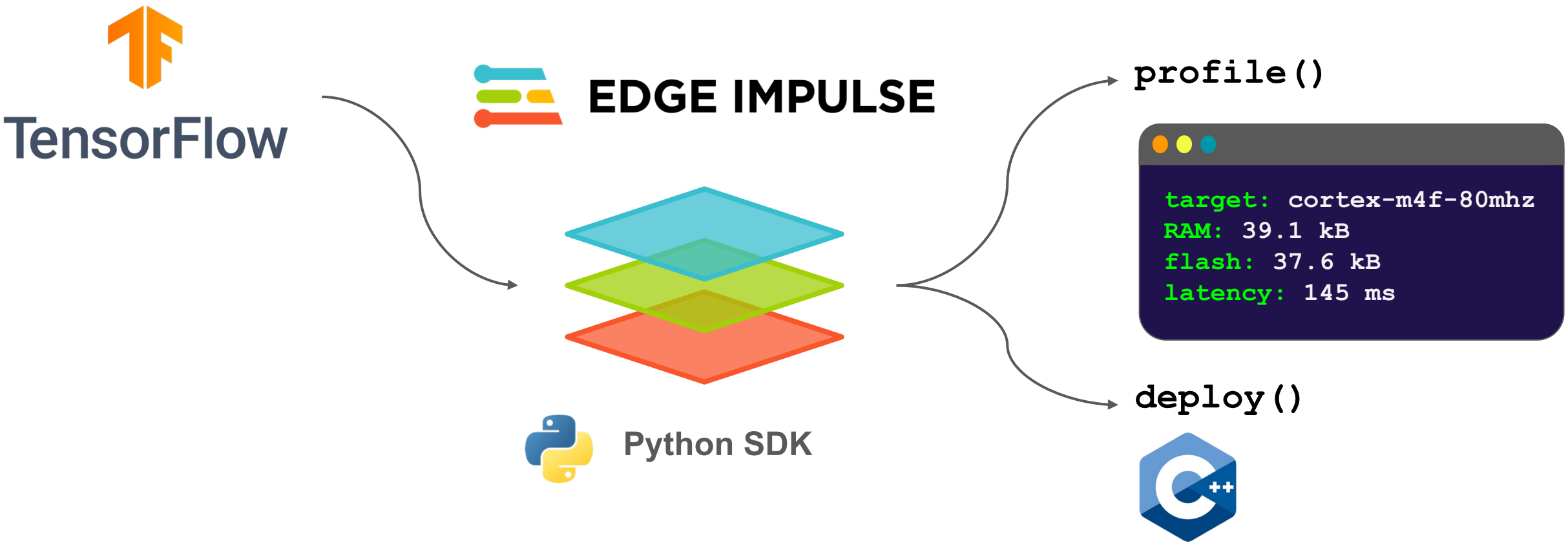


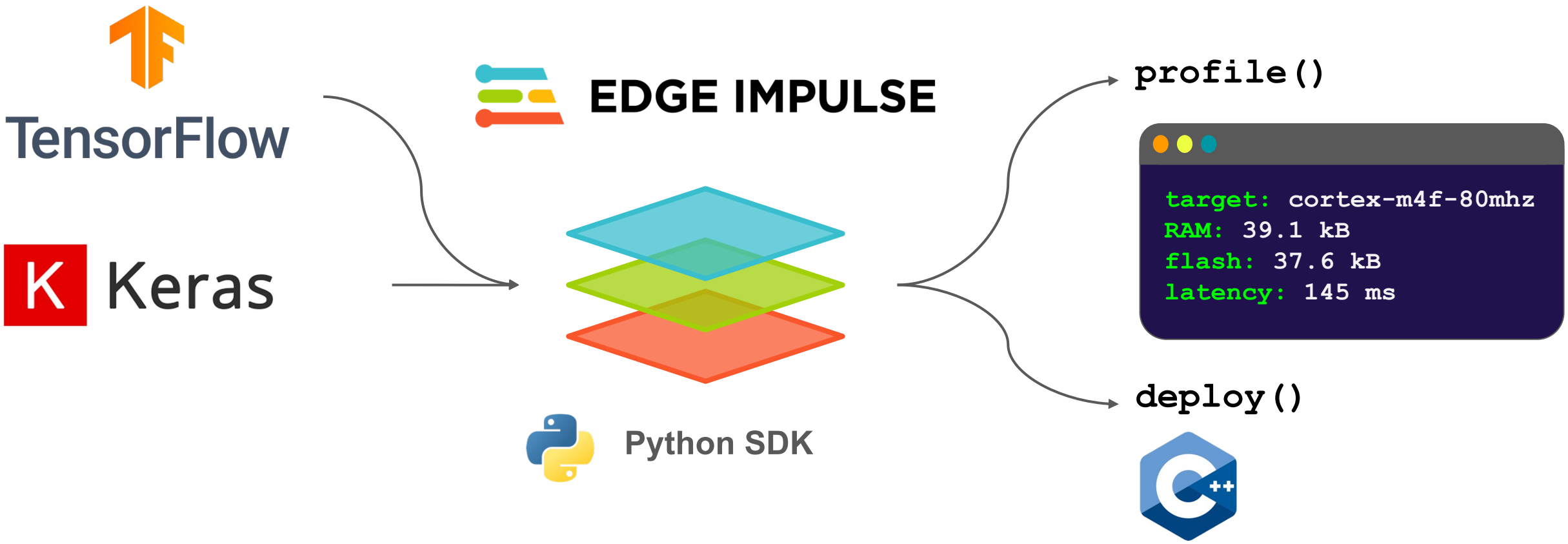
Python SDK

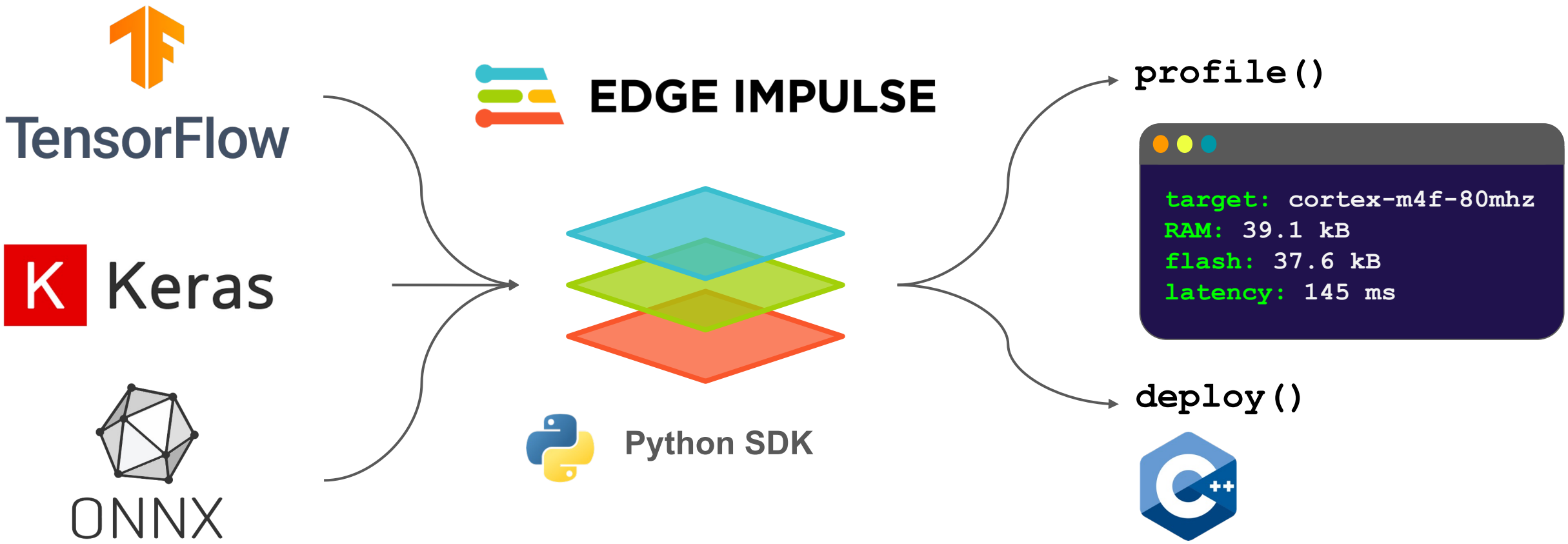
profile()

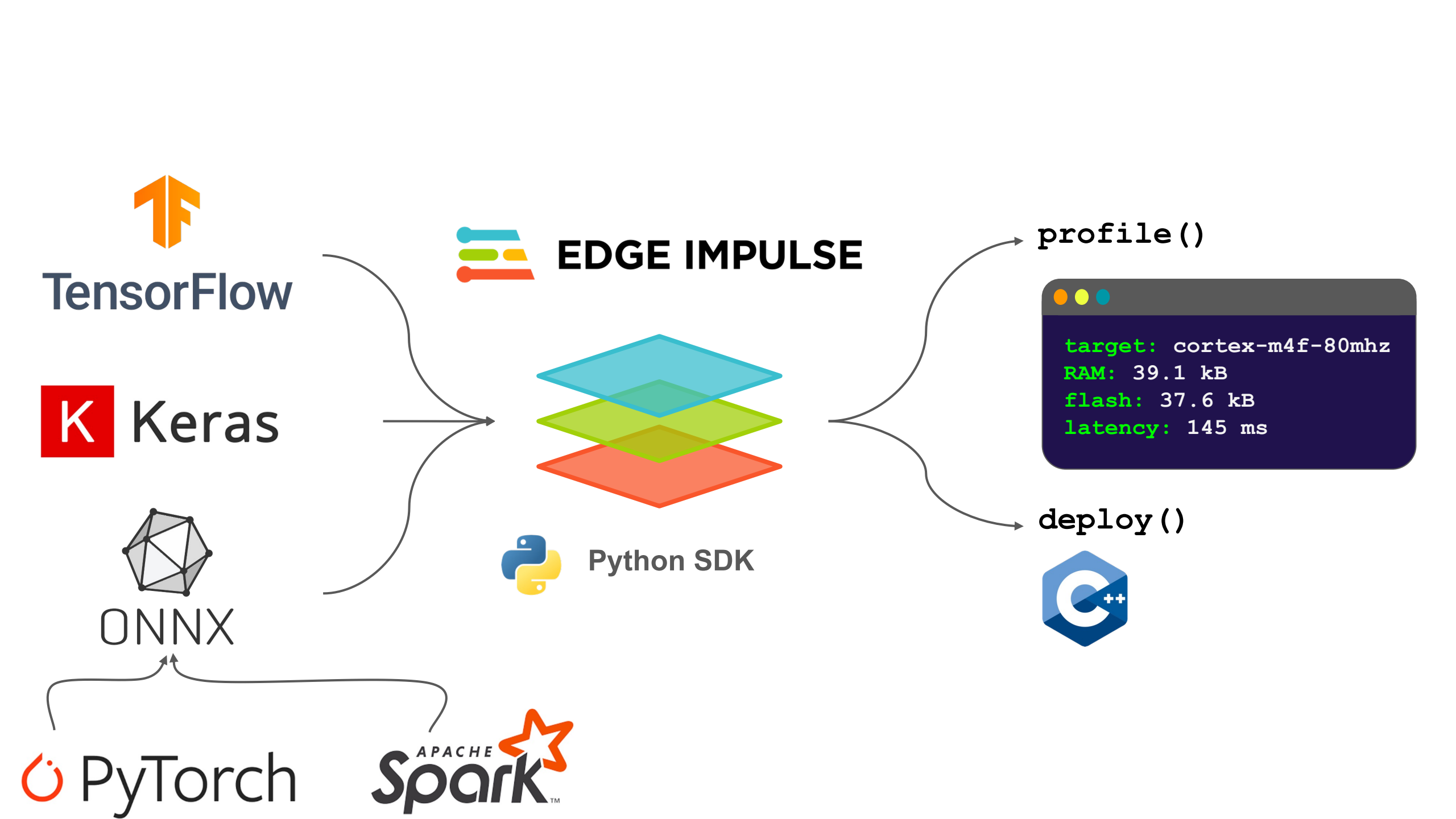
```
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
```


EDGE IMPULSE
profile()
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
deploy()
Python SDK


TensorFlow
EDGE IMPULSE
Python SDK
profile()
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
deploy()


TensorFlow
Keras
EDGE IMPULSE
Python SDK
profile()
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
deploy()


TensorFlow
Keras
ONNX
EDGE IMPULSE
Python SDK
profile()
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
deploy()


TensorFlow
Keras
ONNX
PyTorch
APACHE Spark
EDGE IMPULSE
Python SDK
profile()
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
deploy()

Edge Impulse Python SDK

How it works?

Configure the Edge Impulse Python SDK

```
!pip install edgeimpulse
```

```
[ ] import edgeimpulse as ei

    ei.API_KEY = api_key
```

Profile

```
[ ] profile = ei.model.profile(model="./my_model/",
                               device='cortex-m4f-80mhz')
    print(profile.summary())
```

Deploy

```
[ ] deploy = ei.model.deploy(model="./my_model/",
                             model_output_type=ei.model.output_type.Classification(),
                             deploy_target="zip",
                             output_directory="./")
```

ei-python-sdk.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text Connect

Configure the Edge Impulse Python SDK

```
[ ] !pip install edgeimpulse
```

```
import edgeimpulse as ei

ei.API_KEY = api_key
```

Profile

```
[ ] profile = ei.model.profile(model="./my_model/",
                               device='cortex-m4f-80mhz')
    print(profile.summary())
```

Deploy

```
[ ] deploy = ei.model.deploy(model="./my_model/",
                             model_output_type=ei.model.output_type.Classification(),
                             deploy_target="zip",
                             output_directory="./")
```

ei-python-sdk.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Connect

+ Code + Text

Configure the Edge Impulse Python SDK

```
[ ] !pip install edgeimpulse
```

```
[ ] import edgeimpulse as ei

    ei.API_KEY = api_key
```

Profile

```
profile = ei.model.profile(model="./my_model/",
                           device='cortex-m4f-80mhz')
print(profile.summary())
```

Deploy

```
[ ] deploy = ei.model.deploy(model="./my_model/",
                             model_output_type=ei.model.output_type.Classification(),
                             deploy_target="zip",
                             output_directory="./")
```

Example - Profile

```
[43] profile = ei.model.profile(model="saved_model_float32.zip",
                                device='cortex-m7-216mhz')
     print(profile.summary())
```

```
Target results for float32:
===========================
{
    "device": "cortex-m7-216mhz",
    "tfliteFileSizeBytes": 863312,
    "isSupportedOnMcu": true,
    "memory": {
        "tflite": {
            "ram": 399257,
            "rom": 927576,
            "arenaSize": 398905
        },
        "eon": {
            "ram": 328776,
            "rom": 882432
        }
    },
    "timePerInferenceMs": 75
}
```

ei-python-sdk.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text Connect

Configure the Edge Impulse Python SDK

```
[ ] !pip install edgeimpulse
```

```
[ ] import edgeimpulse as ei

    ei.API_KEY = api_key
```

Profile

```
[ ] profile = ei.model.profile(model="./my_model/",
                               device='cortex-m4f-80mhz')
    print(profile.summary())
```

Deploy

```
deploy = ei.model.deploy(model="./my_model/",
                         model_output_type=ei.model.output_type.Classification(),
                         deploy_target="zip",
                         output_directory="./")
```

Example - Quantize & deploy

Deploy

```
[35] # Generate the representative data for the quantized model
     import glob, cv2
     import numpy as np
     X_data = []
     files = glob.glob ("test-set/*.jpg")
     for myFile in files:
         image = cv2.imread (myFile)
         resized = cv2.resize(image, (96,96), interpolation = cv2.INTER_AREA)
         X_data.append (resized)

     print('X_data shape:', np.array(X_data).shape)
```

```
X_data shape: (36, 96, 96, 3)
```

Example - Quantize & deploy

54s

```
[42] # Set model information, such as your list of labels
     deploy_filename = "generated_cpp.zip"
     labels = ['cotton stem','epidermis onion','housefly legs','unknown','wood stem']
     model_output_type = ei.model.output_type.Classification(labels=labels)

     # Create C++ library with trained model
     deploy_bytes = None
     try:

         deploy_bytes = ei.model.deploy(model="saved_model.zip",
                                        model_output_type=model_output_type,
                                        deploy_model_type="int8",
                                        representative_data_for_quantization=np.array(X_data, dtype="float32"),
                                        engine="tflite-eon",
                                        deploy_target="zip",
                                        output_directory="./")
     except Exception as e:
         print(f"Could not deploy: {e}")

     # Write the downloaded raw bytes to a file
     if deploy_bytes:
         with open(deploy_filename, 'wb') as f:
             f.write(deploy_bytes)
```

0s

```
[44] !ls

     generated_cpp.zip  sample_data  saved_model_float32.zip  test-set
```

Need more?

The Python SDK is built on top of the Edge Impulse Python API bindings, the `edgeimpulse-api` package.

These are Python wrappers for all of the web API calls available to interact with Edge Impulse projects programmatically
(i.e. without needing to use the Studio graphical interface).

Need more?

```
[ ]  python -m pip install edgeimpulse-api
```

```
[ ]  from edgeimpulse_api import Configuration, ApiClient, ProjectsApi

     # Settings
     host = "https://studio.edgeimpulse.com/v1"
     api_key = "ei_dae2..."

     # Create a client object that can connect to our project
     config = Configuration(host=host, api_key={"ApiKeyAuthentication": api_key})
     client = ApiClient(config)

     # Get info about the project
     projects = ProjectsApi(client)
     project_list = projects.list_projects()
     print(project_list.projects[0])
```

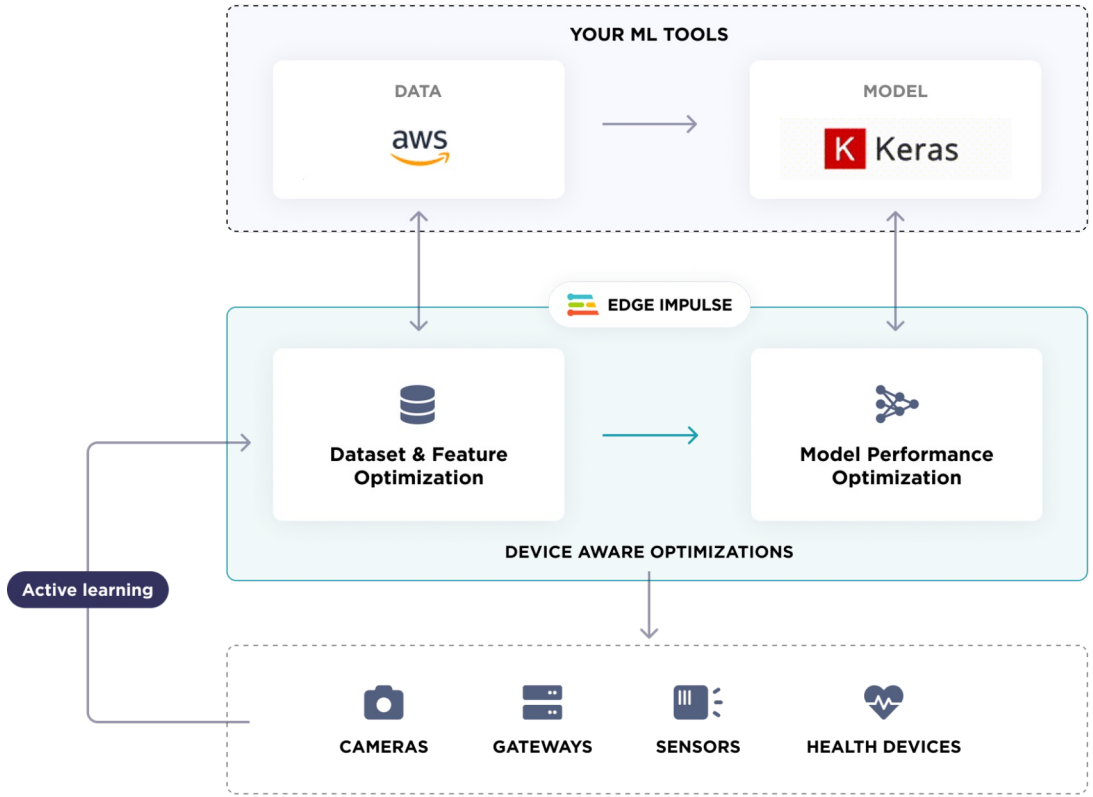
Edge Impulse Python SDK

Integrations and automation

Integrations & automation

Designed to help ML practitioners with every stage of their workflow

- Integrates easily in **existing ML workflows**
- Unlocks **feature engineering** and fasten feedback loops
- Empowers **model optimization** and deployment to any device

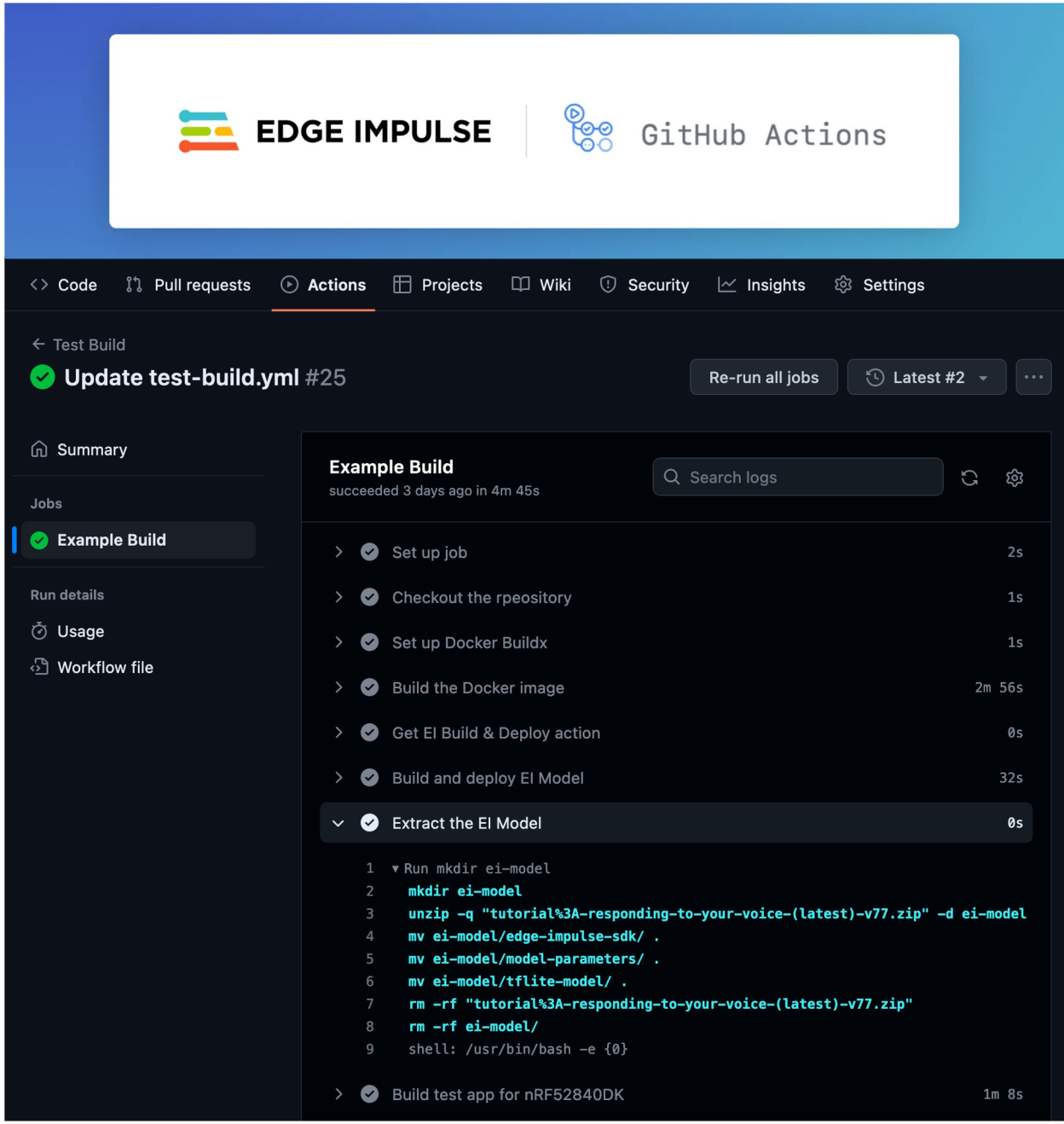


Integrations & automation

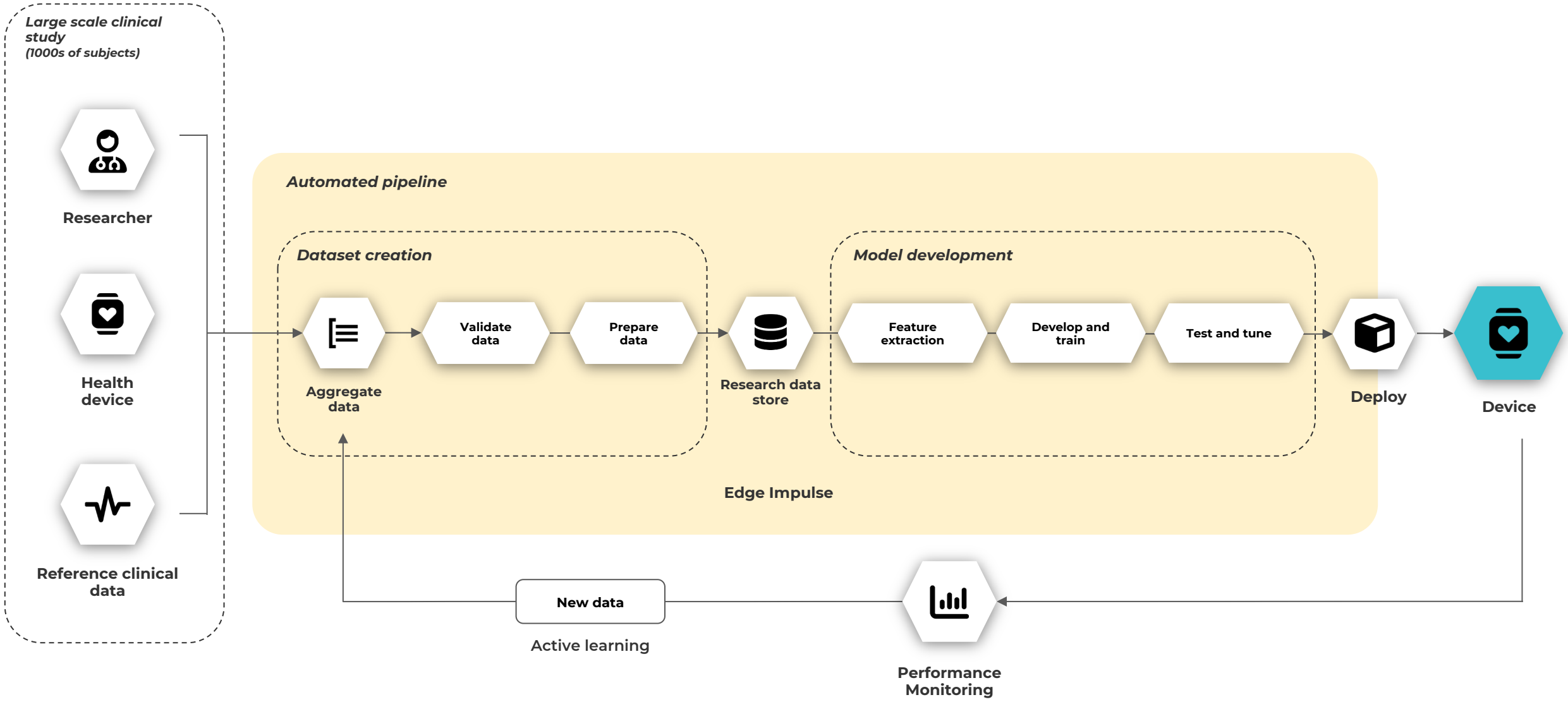
CI/CD pipelines

- CI/CD is one of the critical factors for delivering fully tested and up-to-date software or firmware.

- We developed a **GitHub Action** to easily profile, build and deploy your Edge Impulse model



Health ML automation example

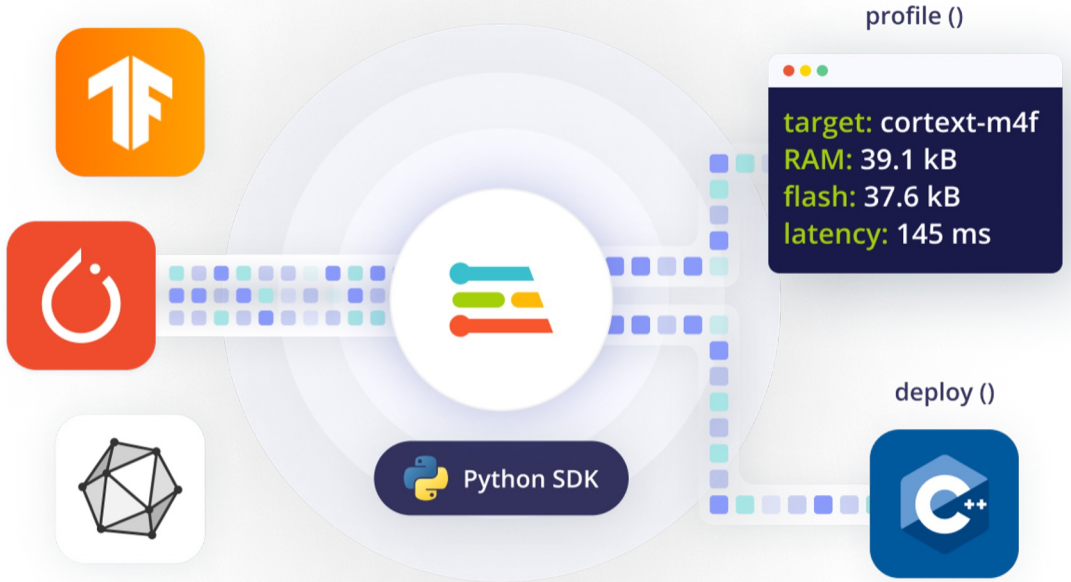


Edge Impulse Python SDK

Recap

Convert Python Models into Optimized C++

- Profile on-device performance of any trained model
- Analyze the impact of architectural decisions
- Generate optimized C++ libraries
- Deploy to edge devices



Copyright Notice

www.tinyml.org