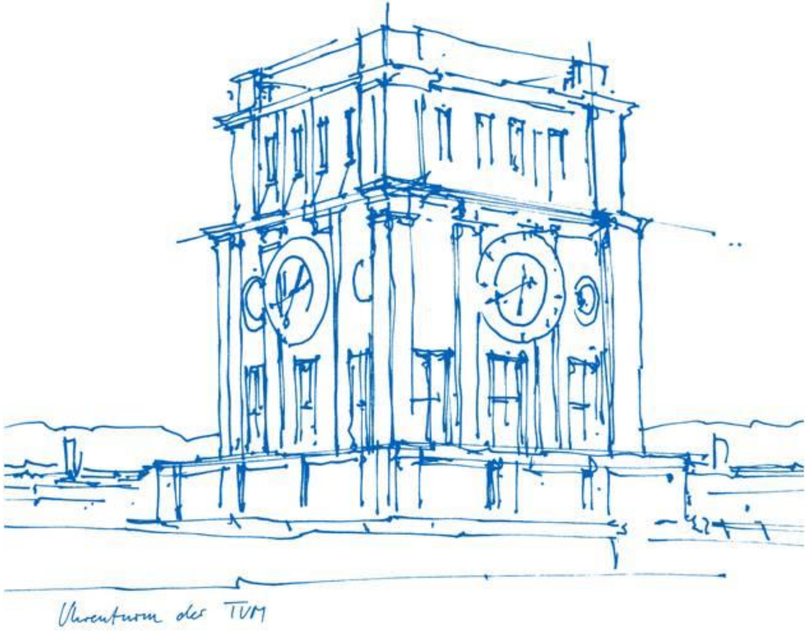
Sprint 2 Presentation

Ahmed Kaddah, Shao Jie Hu Chen, Marlon Müller

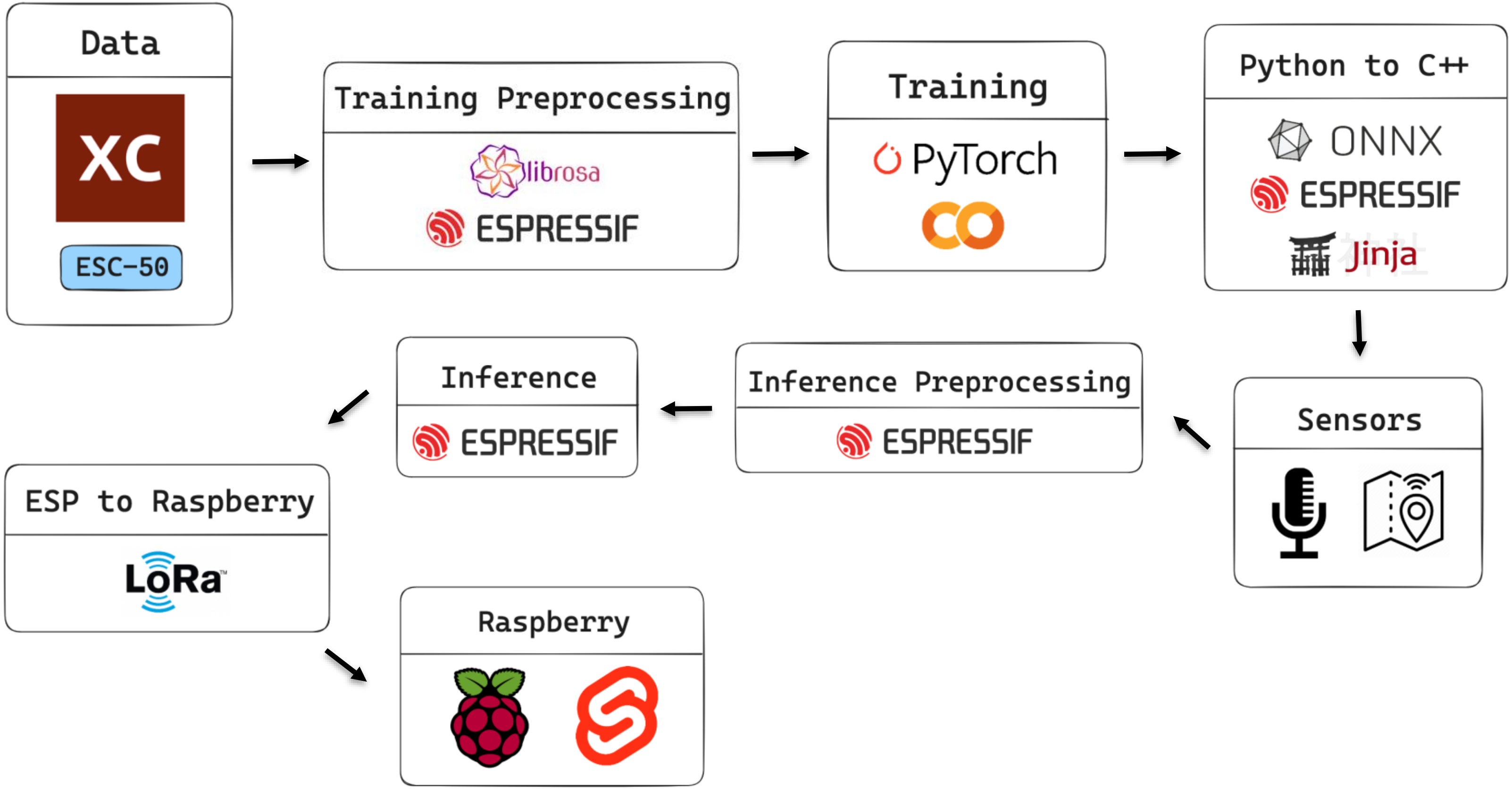
Edge Computing and the Internet of Things

Technische Universität München

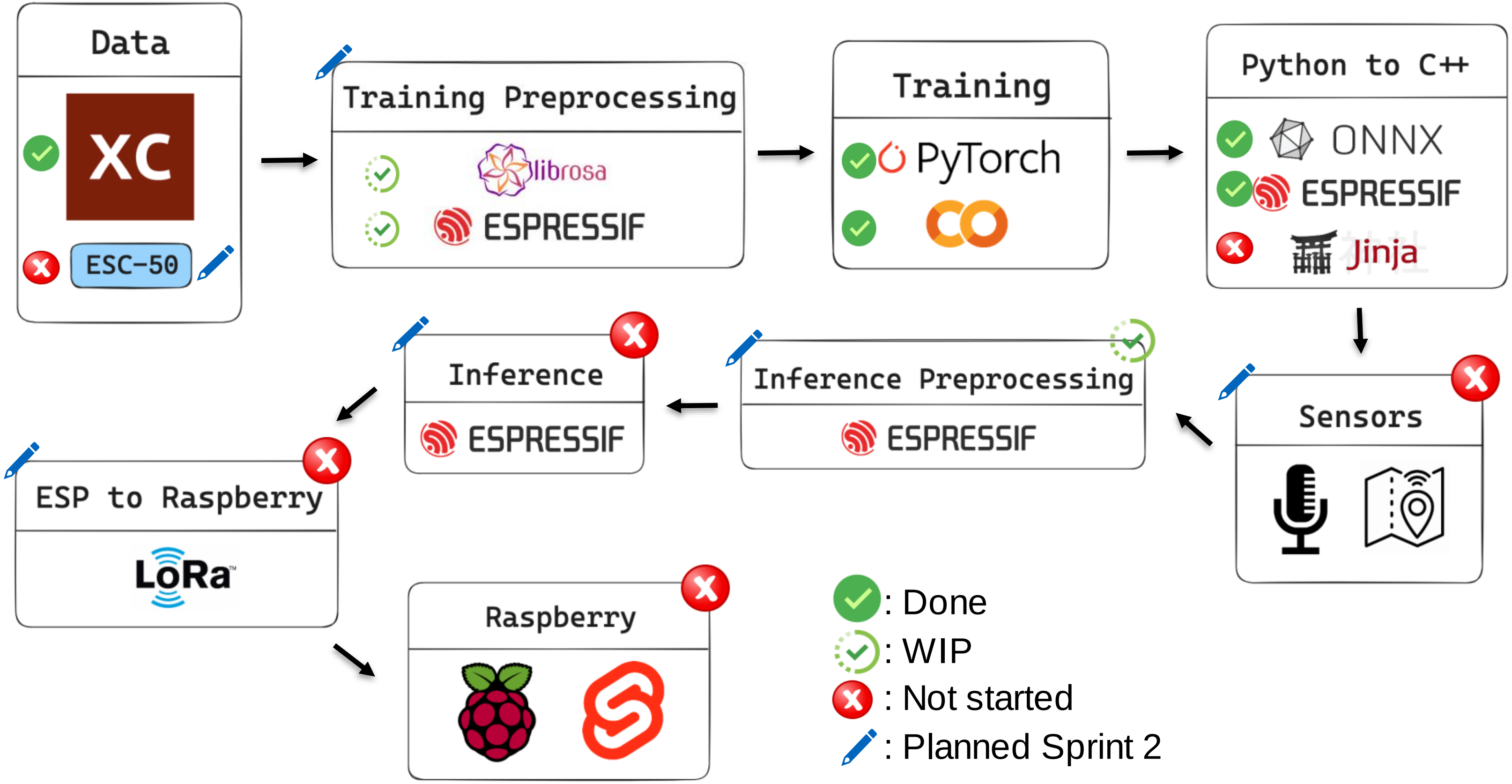
München, 08.12.2023



Functional Prototype Pipeline



Sprint 1



Data
- ✓ XC
- ✗ ESC-50 ✎

Training Preprocessing ✎
- ◌✓ librosa
- ◌✓ ESPRESSIF

Training
- ✓ PyTorch
- ✓ CO

Python to C++
- ✓ ONNX
- ✓ ESPRESSIF
- ✗ Jinja

Sensors ✎ ✗

Inference Preprocessing ✎ ◌✓
- ESPRESSIF

Inference ✎ ✗
- ESPRESSIF

ESP to Raspberry ✎ ✗
- LoRa

Raspberry ✗

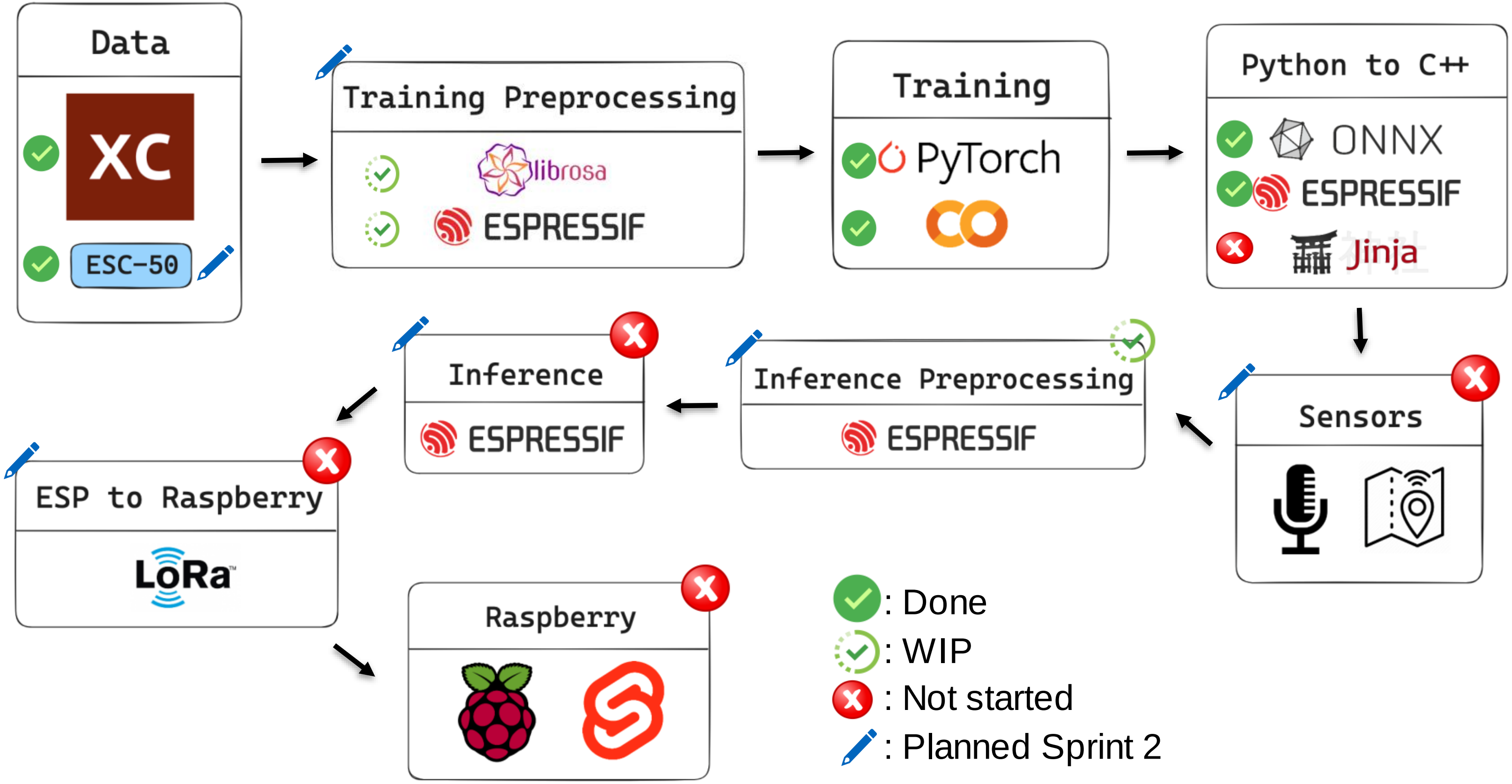
✓ : Done

◌✓ : WIP

✗ : Not started

✎ : Planned Sprint 2

Sprint 2: Integration of ESC50



Data
XC
ESC-50

Training Preprocessing
librosa
ESPRESSIF

Training
PyTorch
CO

Python to C++
ONNX
ESPRESSIF
Jinja

Sensors

Inference Preprocessing
ESPRESSIF

Inference
ESPRESSIF

ESP to Raspberry
LoRa

Raspberry

✓ : Done

✓ : WIP

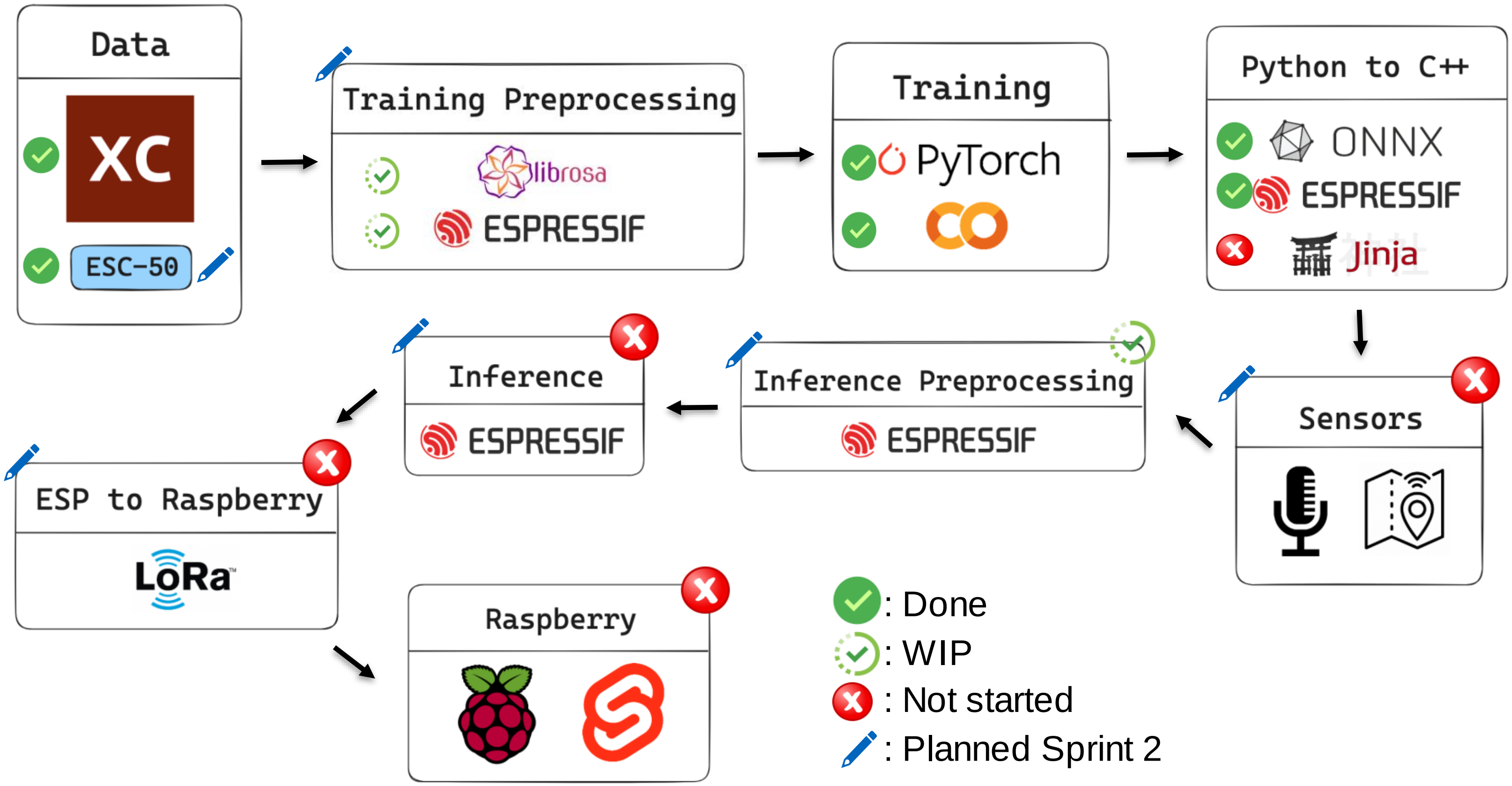
✗ : Not started

✎ : Planned Sprint 2

Sprint 2: Integration of ESC50

- Dataset for Environmental Sound Classification
- E.g., dog, insects, rain, thunderstorm, wind, siren, chainsaw

Sprint 2: Integration of ESC50



Data

Training Preprocessing

Training

Python to C++

Sensors

Inference Preprocessing

Inference

ESP to Raspberry

Raspberry

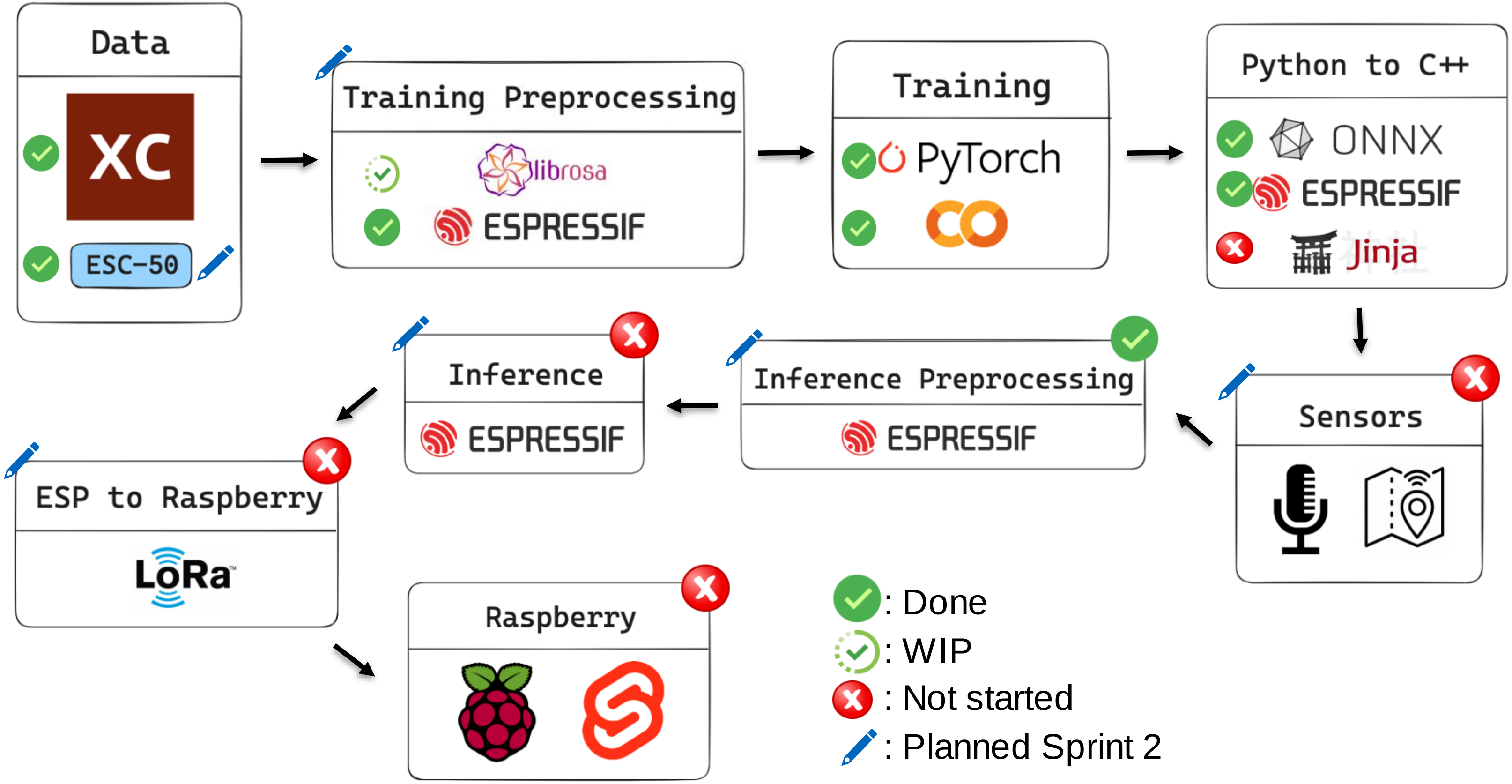
✓ : Done

✓ : WIP

✗ : Not started

✎ : Planned Sprint 2

Sprint 2: Resolve MFCC discrepancies



Data
- ✓ XC
- ✓ ESC-50 ✎

✎ Training Preprocessing
- ◌✓ librosa
- ✓ ESPRESSIF

Training
- ✓ PyTorch
- ✓ CO

Python to C++
- ✓ ONNX
- ✓ ESPRESSIF
- ✗ Jinja

✎ Sensors ✗

✎ Inference Preprocessing ✓
- ESPRESSIF

✎ Inference ✗
- ESPRESSIF

✎ ESP to Raspberry ✗
- LoRa

Raspberry ✗

✓ : Done

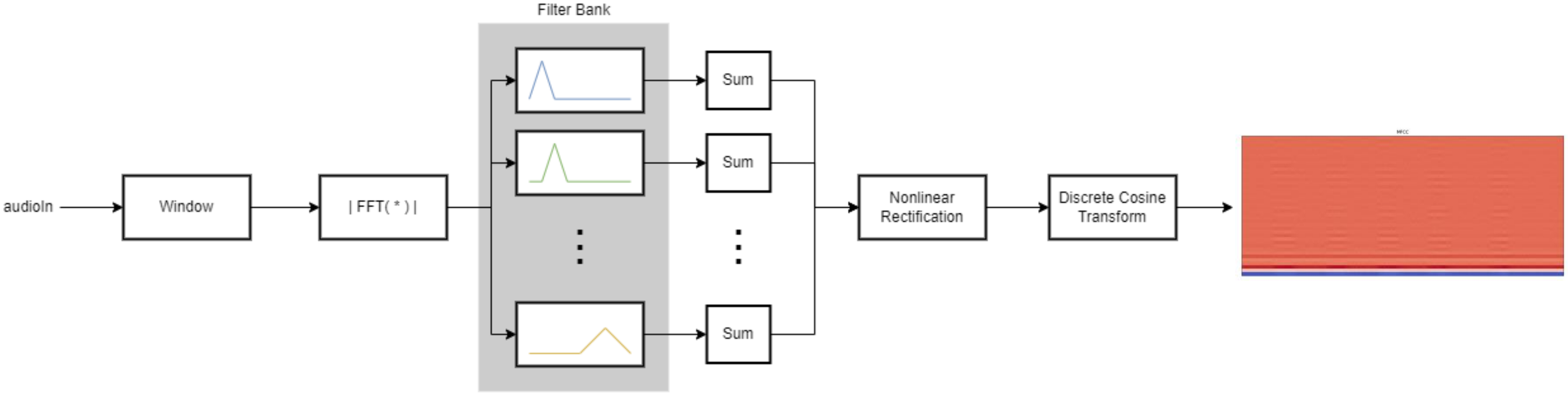
◌✓ : WIP

✗ : Not started

✎ : Planned Sprint 2

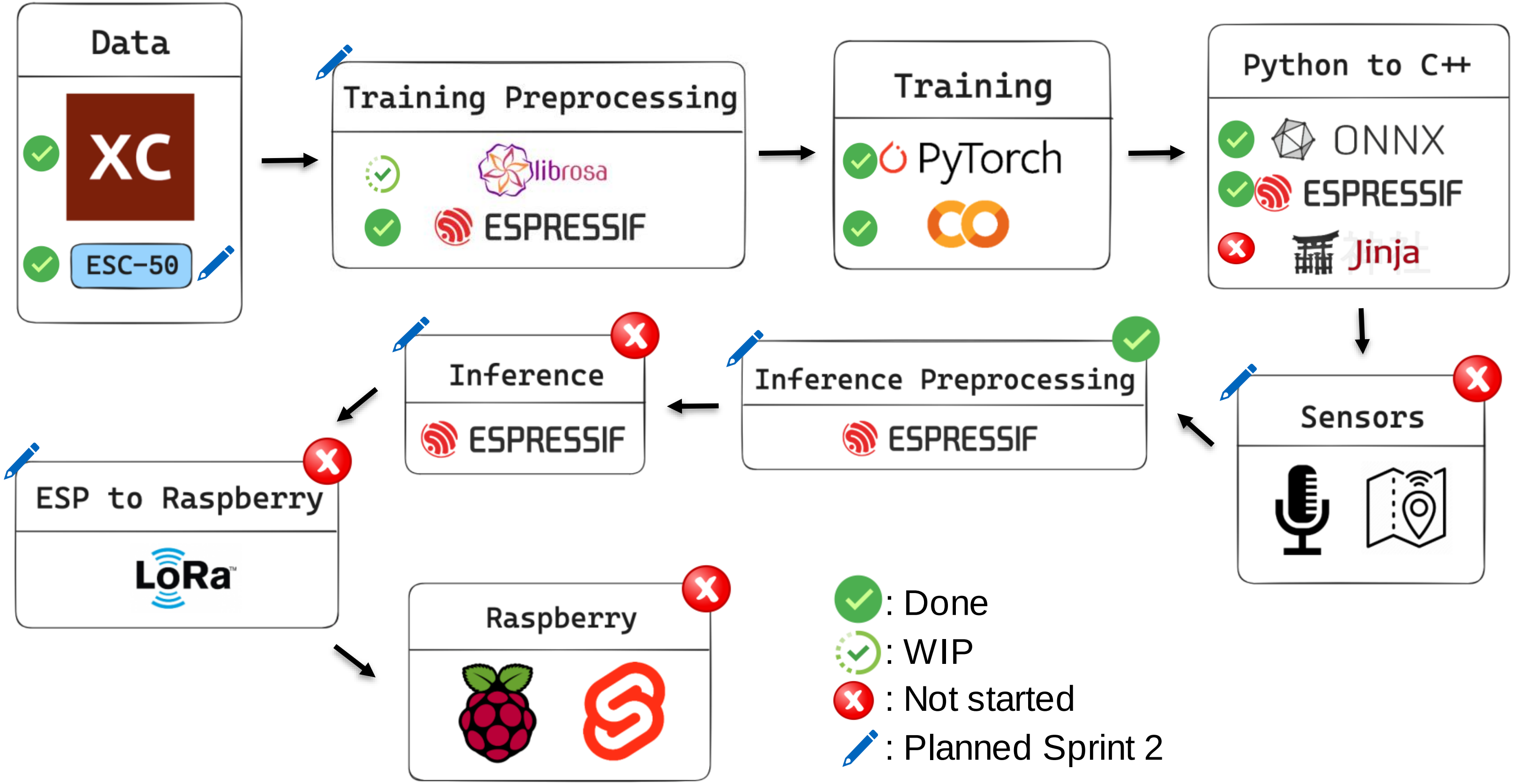
Sprint 2: Resolve MFCC discrepancies

- Custom ESP implementation using ESP-DSP functionality matches Librosa
- Additionally ported implementation to PC



[https://de.mathworks.com/help/audio/ref/mfcc.html]

Sprint 2: Resolve MFCC discrepancies



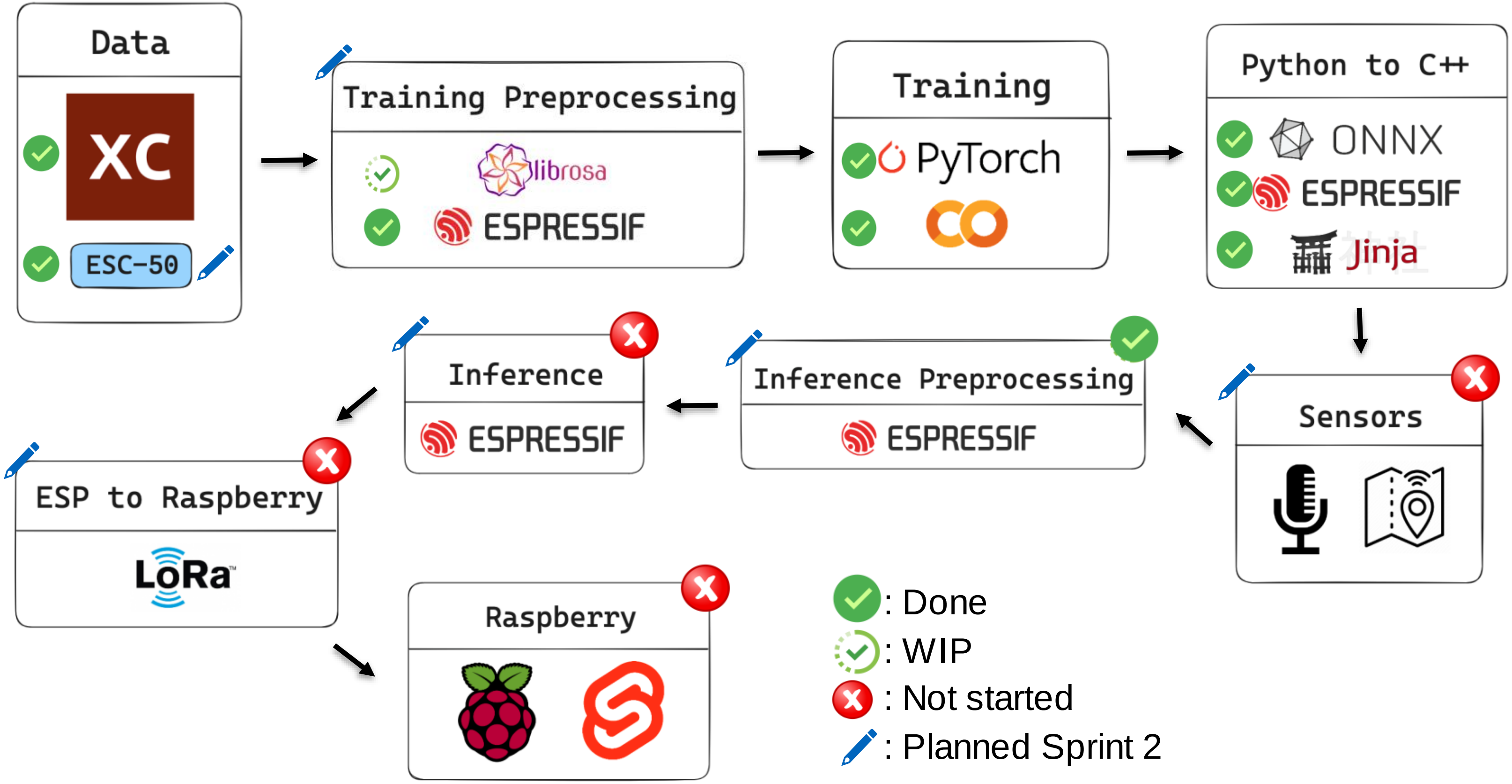
✓: Done

✓: WIP

✗: Not started

✎: Planned Sprint 2

Sprint 2: PyTorch to C++ Template



✓ : Done

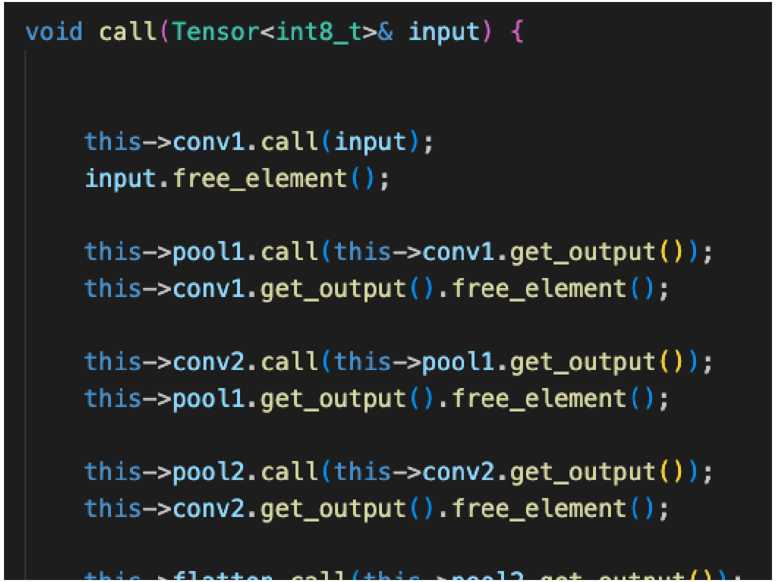
◌✓ : WIP

✗ : Not started

✎ : Planned Sprint 2

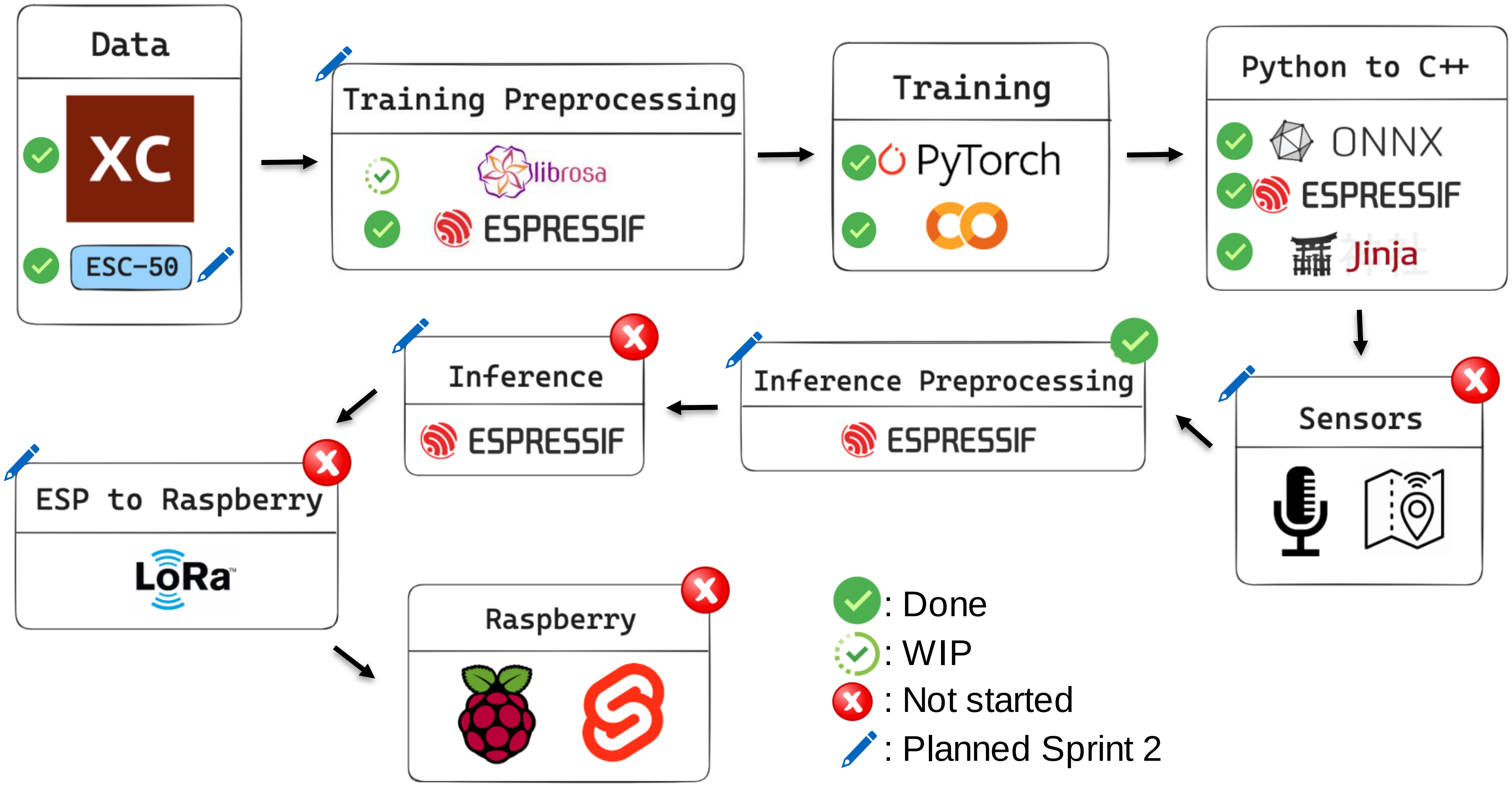
[Sprint 2] PyTorch to C++ Template

- ESP-DL Quantization generates weights but not model itself
- Jinja templates for automatic model.hpp generation



```
void call(Tensor<int8_t>& input) {


    this->conv1.call(input);
    input.free_element();

    this->pool1.call(this->conv1.get_output());
    this->conv1.get_output().free_element();

    this->conv2.call(this->pool1.get_output());
    this->pool1.get_output().free_element();

    this->pool2.call(this->conv2.get_output());
    this->conv2.get_output().free_element();

    this->flatten.call(this->pool2.get_output());
```

Sprint 2: PyTorch to C++ Template



Data

XC

ESC-50

Training Preprocessing

librosa

ESPRESSIF

Training

PyTorch

CO

Python to C++

ONNX

ESPRESSIF

Jinja

Sensors

Inference Preprocessing

ESPRESSIF

Inference

ESPRESSIF

ESP to Raspberry

LoRa

Raspberry

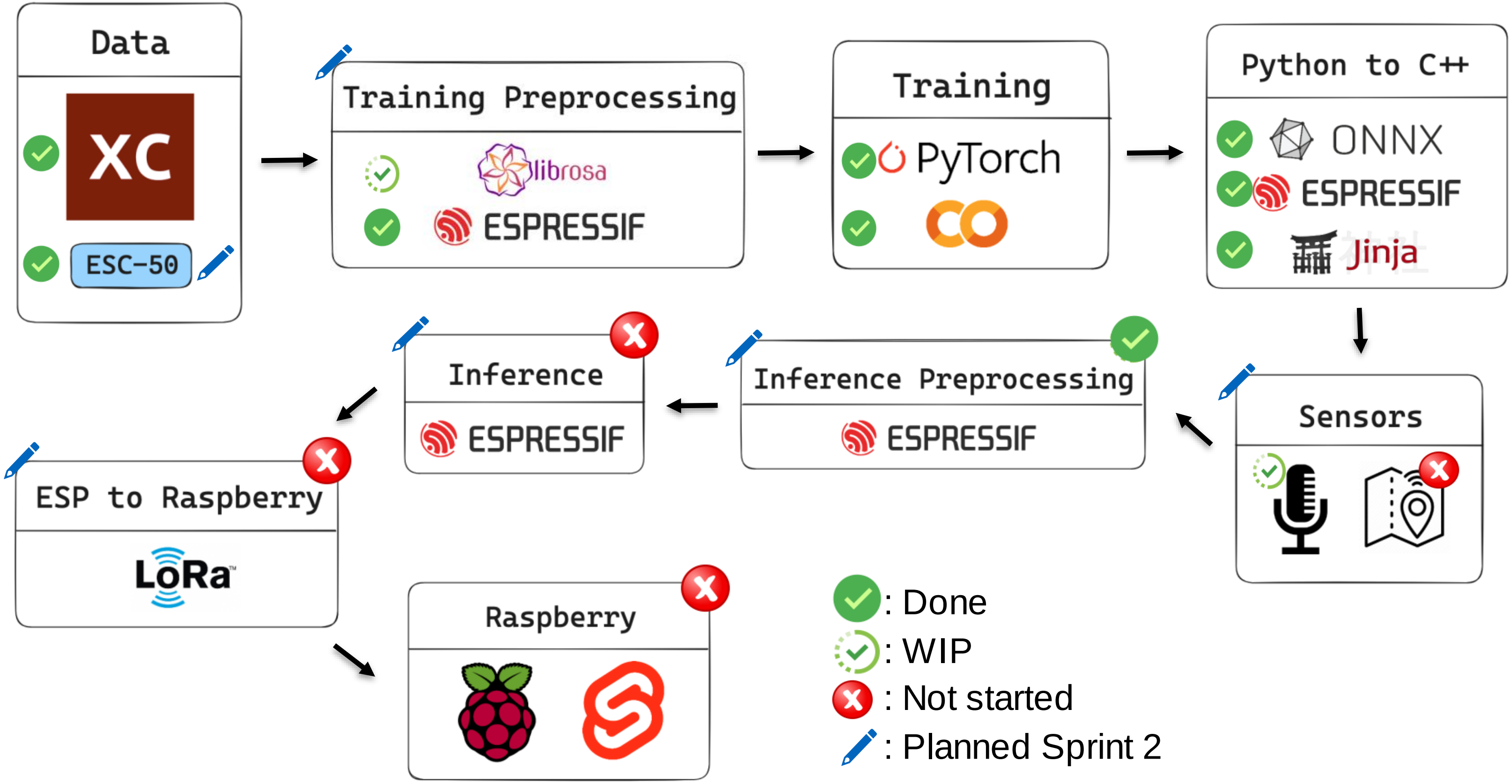
✓ : Done

✓ : WIP

✗ : Not started

✎ : Planned Sprint 2

Sprint 2: Microphone



Data

XC

ESC-50

Training Preprocessing

librosa

ESPRESSIF

Training

PyTorch

CO

Python to C++

ONNX

ESPRESSIF

Jinja

Sensors

Inference Preprocessing

ESPRESSIF

Inference

ESPRESSIF

ESP to Raspberry

LoRa

Raspberry

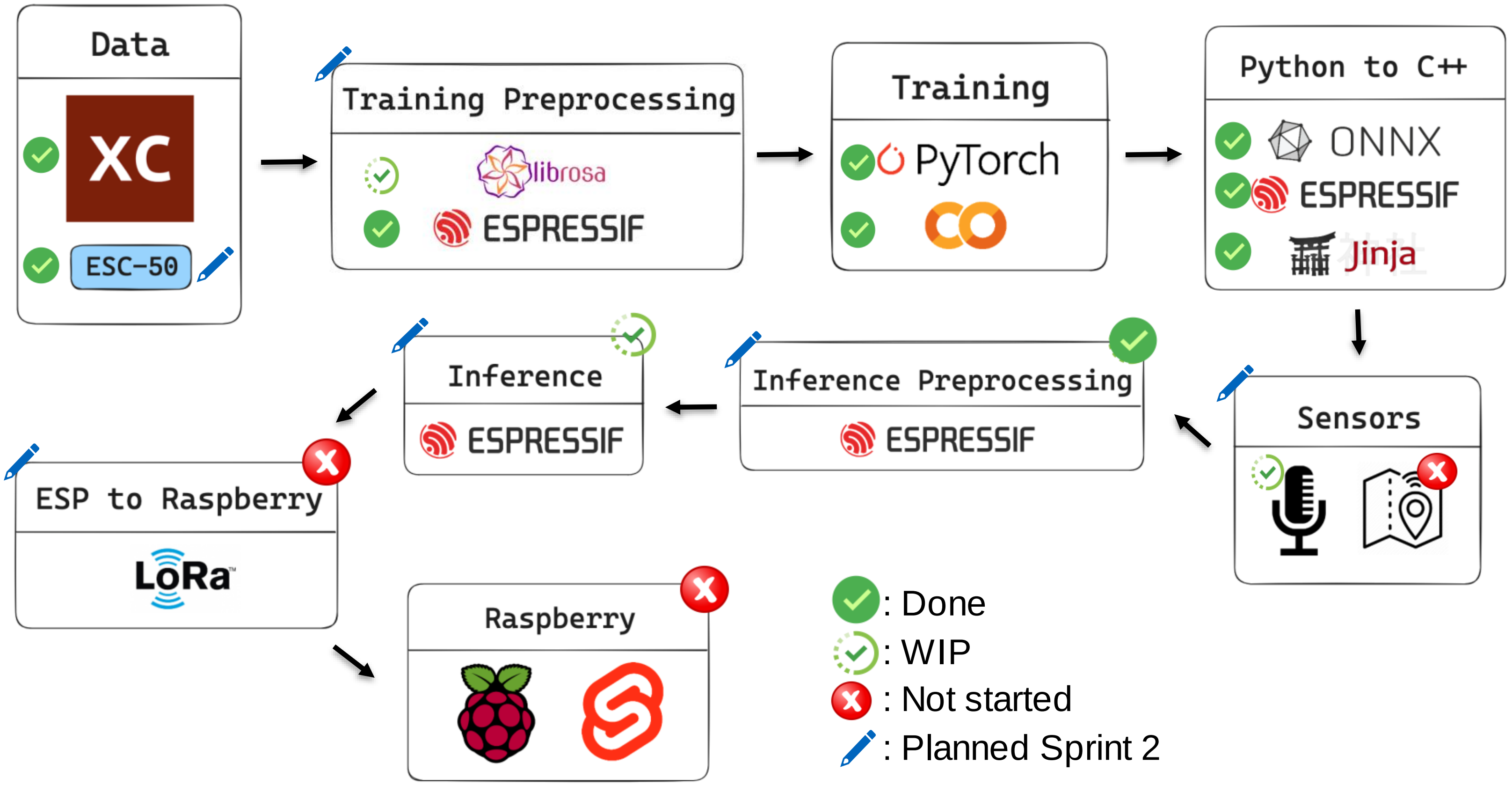
✓ : Done

✓ : WIP

☒ : Not started

✎ : Planned Sprint 2

Sprint 2: Inference (Microphone – MFCC - DL)



Data
✓ XC
✓ ESC-50

Training Preprocessing
librosa
✓ ESPRESSIF

Training
✓ PyTorch
✓ CO

Python to C++
✓ ONNX
✓ ESPRESSIF
✓ Jinja

Sensors

Inference Preprocessing ✓
ESPRESSIF

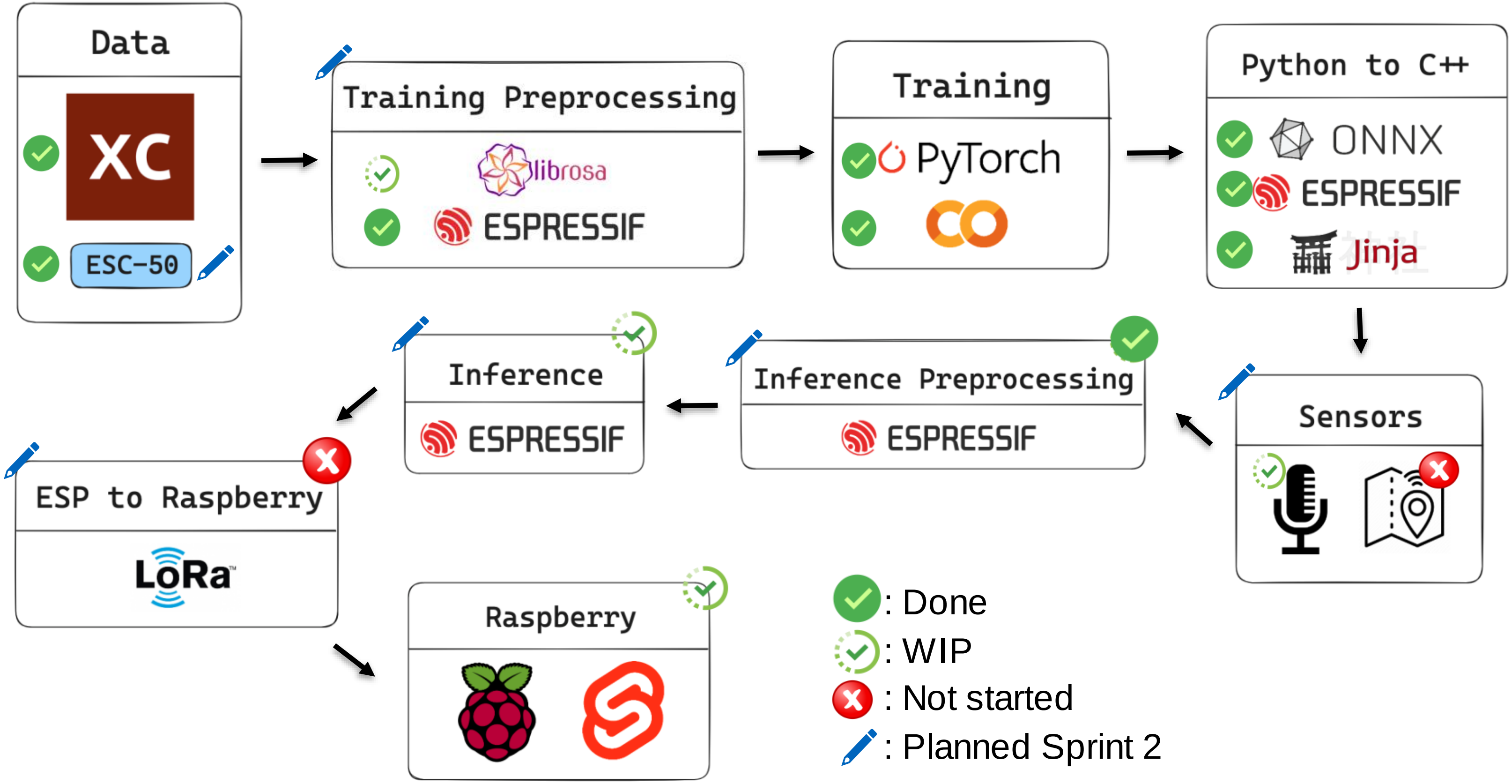
Inference
ESPRESSIF

ESP to Raspberry ☒
LoRa

Raspberry ☒

✓ : Done
✓ : WIP
☒ : Not started
✎ : Planned Sprint 2

Sprint 2: Dashboard



✓ : Done

✓ : WIP

✗ : Not started

✎ : Planned Sprint 2

Sprint 2: Objectives

In general, assemble a **functional prototype**

- Integrate auxiliary data (e.g., ESC-50), resample (and split) audio files
- Build, train, and evaluate a first complete dataset
- Interconnect devices and enable sensor functionality
- Resolve MFCC discrepancies and execute the trained model on the ESP32
- (Optional) Start the development of a website/dashboard for visualization

Sprint 2: Objectives

In general, assemble a **functional prototype**

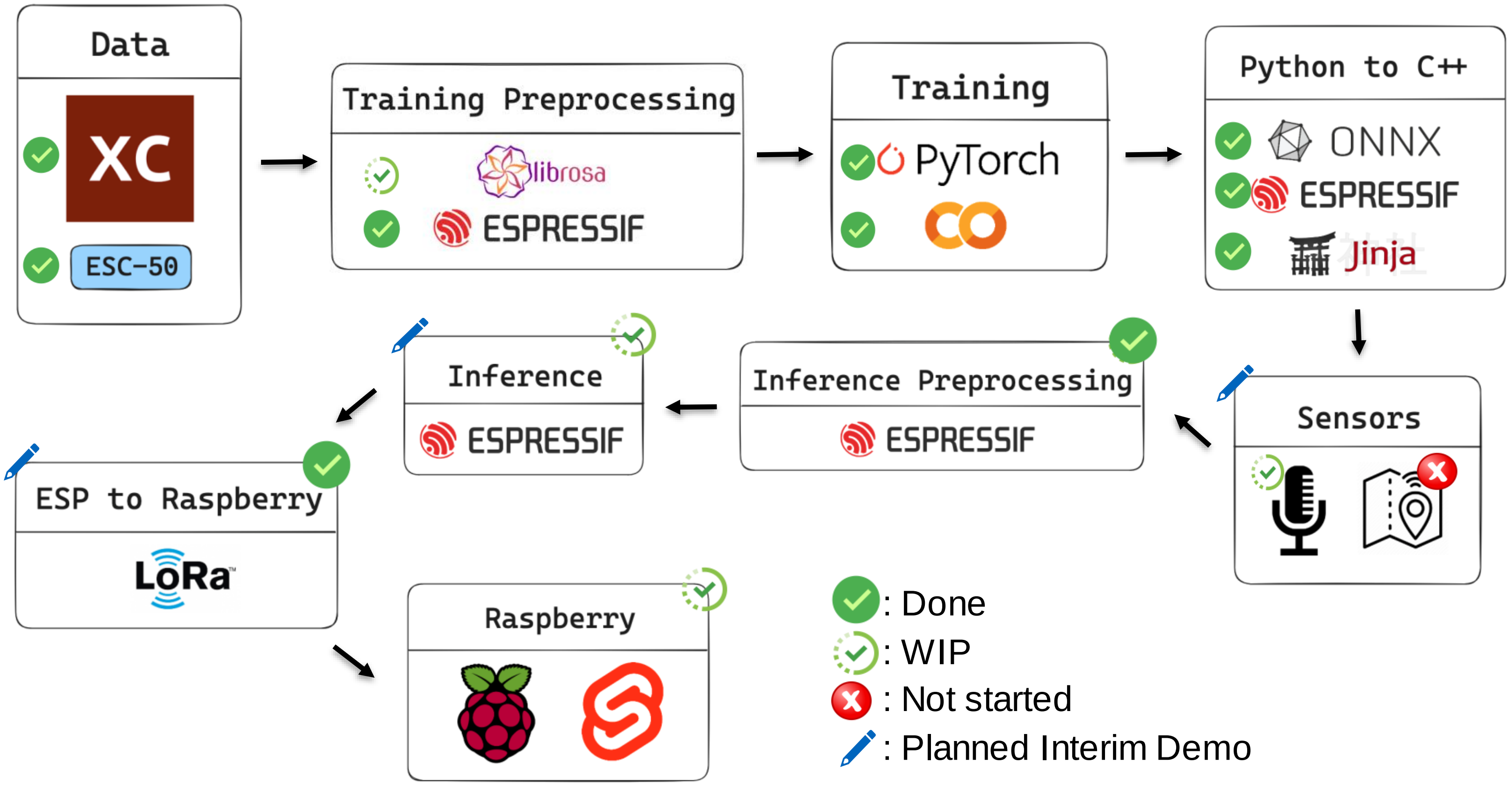
- Integrate auxiliary data (e.g., ESC-50), resample (and split) audio files
- Build, train, and evaluate a first complete dataset
- Interconnect devices and enable sensor functionality
- Resolve MFCC discrepancies and execute the trained model on the ESP32
- (Optional) Start the development of a website/dashboard for visualization

Interim Demo: Objectives

In general, **run** a **functional prototype**

- **Execute the trained model on the ESP32**
- Interconnect devices (LoRa)
- (Optional) GPS
- (Optional) Improve website / visualization
- (Optional) Distinguish salient audio segments

Interim Demo: Objectives



Data

Training Preprocessing

Training

Python to C++

Sensors

Inference Preprocessing

Inference

ESP to Raspberry

Raspberry

✓ : Done

✓ : WIP

✗ : Not started

✎ : Planned Interim Demo