Intermediate Track: Implementing a complex ML pipeline Session 1

Dr. Kerry Donny-Clark

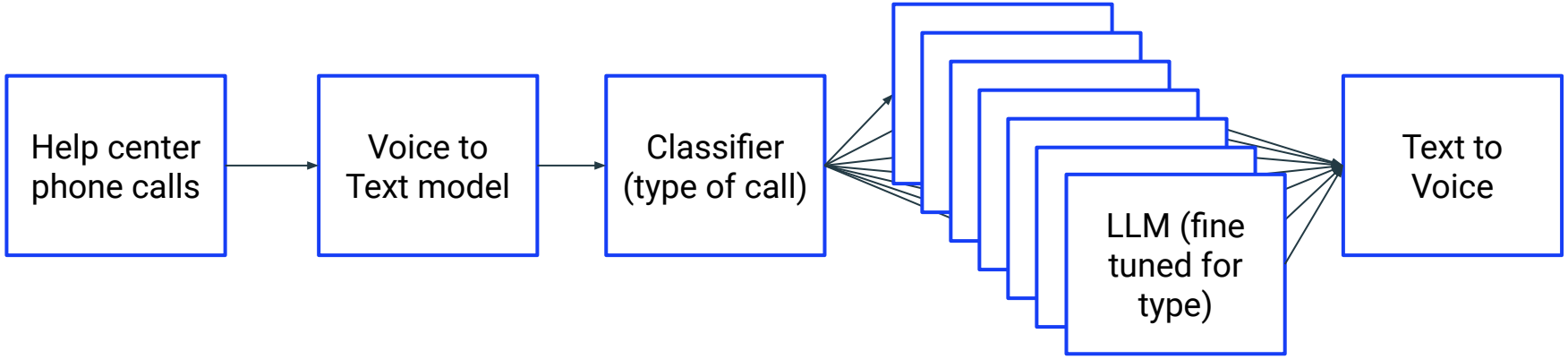
Welcome to Beam College Intermediate Track: Implementing a complex ML pipeline

Hi, I’m Kerry.

Implementing a complex ML pipeline: Overview

1. **Intro to ML in Beam: RunInference and Model Handlers**
2. Choosing Models and how to adapt a model to Beam
3. High level view of the pipeline
4. Deep dive into our example pipeline, Part I
5. Deep dive into our example pipeline Part II
6. Expanding the pipeline to real life use cases

Welcome to Beam College Intermediate Track: Implementing a complex ML pipeline



RunInference

```
class MyComplicatedPridctionStuff(beam.DoFn):
    def setup():
    #Code for loading once
    ...

    def process(self, element):
        #Use model handle to call
        ...
        #Handle errors, do nice error logging
        ...
        #Output useful metrics from the process
        ...
        TODO Oh wait! I need to batch stuff first ...
```

RunInference

```
with beam.Pipeline(options=pipeline_options)  as p:
    (p
    | beam.io.fileio.MatchFiles( gs://my_bucket/images*)
    | beam.io.fileio.ReadMatches()
    | beam.ml.inference.RunInference(model_handler
          .with_preprocess_fn(lambda x: preprocess(x))
          .with_postprocess_fn(lambda x: post_process(x)))
    ...
```

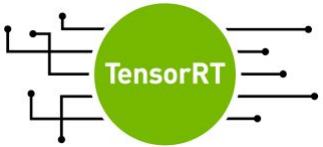
ModelHandlers

```
with beam.Pipeline(options=pipeline_options)  as p:
    (p
    | beam.io.fileio.MatchFiles( gs://my_bucket/images*)
    | beam.io.fileio.ReadMatches()
    | beam.ml.inference.RunInference(model_handler
          .with_preprocess_fn(lambda x: preprocess(x))
          .with_postprocess_fn(lambda x: post_process(x)))
    ...
```

ModelHandlers

PytorchModelHandler**Tensor**

ModelHandlers



ModelHandlers

```
pipeline_handler = HuggingFacePipelineModelHandler(
    task="automatic-speech-recognition",
    model="openai/whisper-small",
    min_batch_size=2,
    load_pipeline_args={'chunk_length_s':30, 'device':'cuda:0'},
    large_model=True)
```

Learn More

Code:

https://github.com/apache/beam/tree/master/sdks/python/apache_beam/ml

RunInference and **ModelHandler** base classes are in **base.py**

Notebooks and examples:

https://github.com/apache/beam/tree/master/sdks/python/apache_beam/examples/inference

https://github.com/apache/beam/tree/master/examples/notebooks/beam-ml

Notebooks can be imported into Google Colab!

Next Session

Choosing Models and how to adapt a model to Beam

Thank you!

Intermediate Track: Implementing a complex ML pipeline Session 2

Dr. Kerry Donny-Clark

Implementing a complex ML pipeline: Overview

1. Intro to ML in Beam: RunInference and Model Handlers
2. **Choosing Models and how to adapt a model to Beam**
3. High level view of the pipeline
4. Deep dive into our example pipeline, Part I
5. Deep dive into our example pipeline Part II
6. Expanding the pipeline to real life use cases

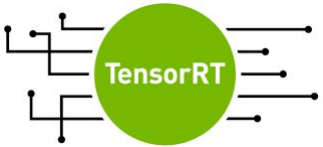
RunInference

```
with beam.Pipeline(options=pipeline_options)  as p:
    (p
    | beam.io.fileio.MatchFiles( gs://my_bucket/images*)
    | beam.io.fileio.ReadMatches()
    | beam.ml.inference.RunInference(model_handler
          .with_preprocess_fn(lambda x: preprocess(x))
          .with_postprocess_fn(lambda x: post_process(x)))
    ...
```

Choosing a model

- Function
- Size
- Framework
- Support in Beam

ModelHandlers



Model Zoos

Model Zoos

HuggingFace
    HuggingFaceModelHandler
    HuggingFacePipelineModelHandler

TensorflowHub
    TFModelHandlerTensor(
        model_uri=CLASSIFIER_URL)

ModelHandlers: Text Example

1. Pick a Model
    a. https://huggingface.co/stevhliu/my_awesome_eli5_mlm_model
2. Choose your model handler
    a. HuggingFaceModelHandler
3. Add preprocessing and/or postprocessing
    a. Encoding, decoding

ModelHandlers: Text Example

Warning! **RunInference** returns a **PredictionResult**:

```
class PredictionResult(NamedTuple('PredictionResult',
                                  [('example', _INPUT_TYPE),
                                   ('inference', _OUTPUT_TYPE),
                                   ('model_id', Optional[str])]))
```

What you usually want is in '**result.inference**'.

ModelHandlers: Text Example

Let's Go To Colab!
https://colab.sandbox.google.com/github/apache/beam/blob/master/examples/notebooks/beam-ml/run_inference_huggingface.ipynb

Next Session

High level overview of a complex ML pipeline

Thank you!

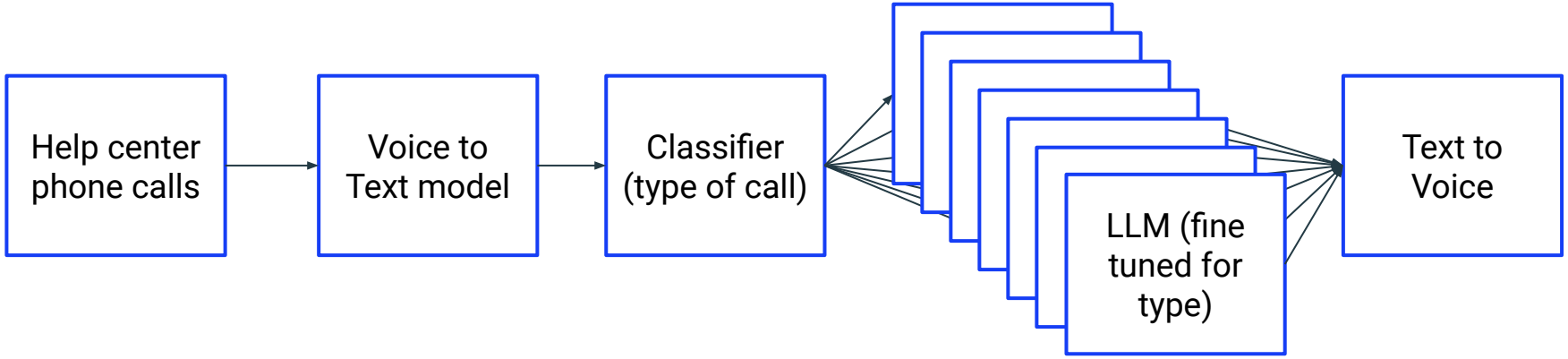
Intermediate Track: Implementing a complex ML pipeline Session 3

Dr. Kerry Donny-Clark

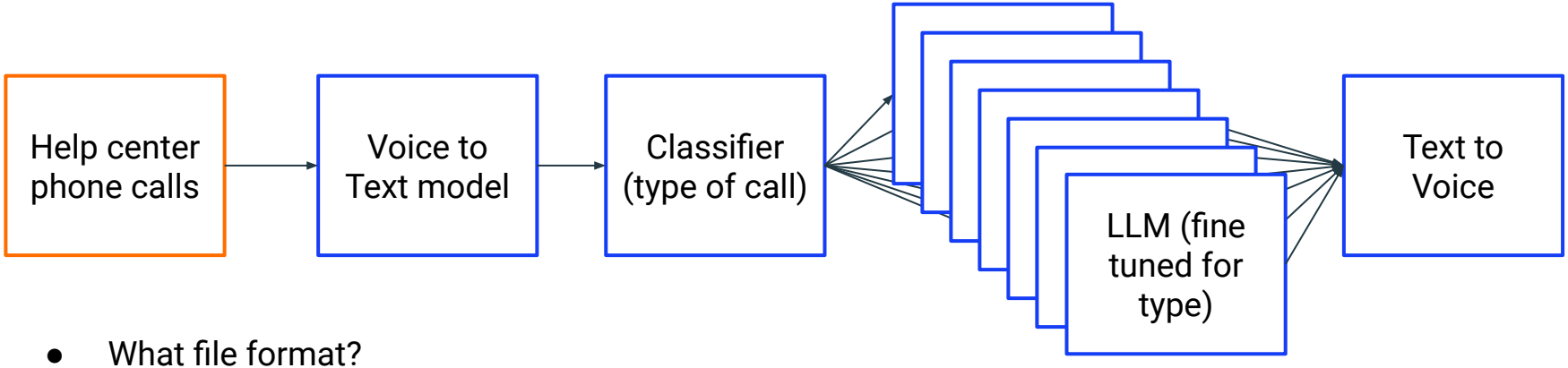
Implementing a complex ML pipeline: Overview

1. Intro to ML in Beam: RunInference and Model Handlers
2. Choosing Models and how to adapt a model to Beam
3. **High level view of the pipeline**
4. Deep dive into our example pipeline, Part I
5. Deep dive into our example pipeline Part II
6. Expanding the pipeline to real life use cases

Our Pipeline

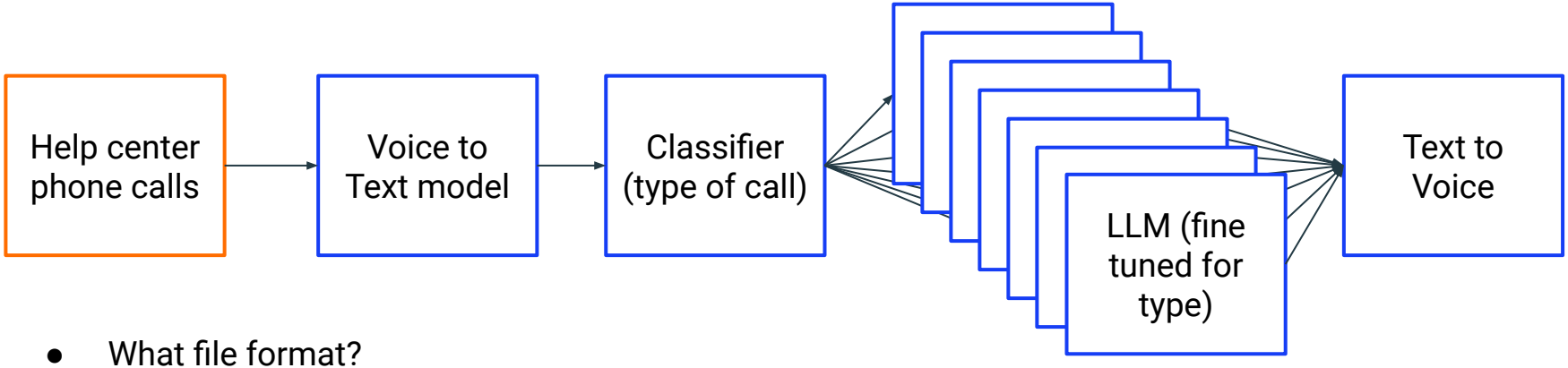


Our Pipeline



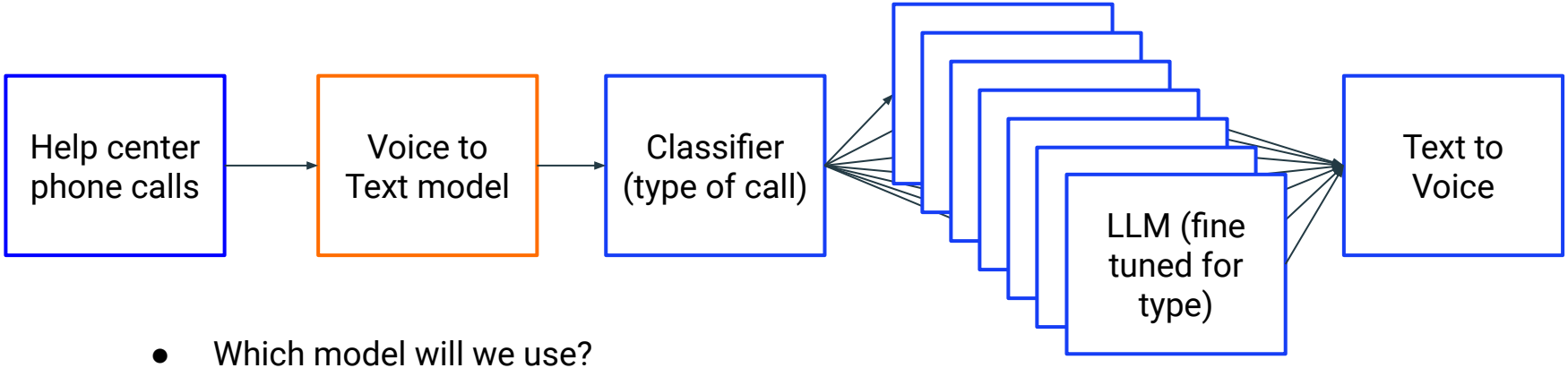
- What file format?
- Where are they stored?
- How long are the audio clips?

Our Pipeline



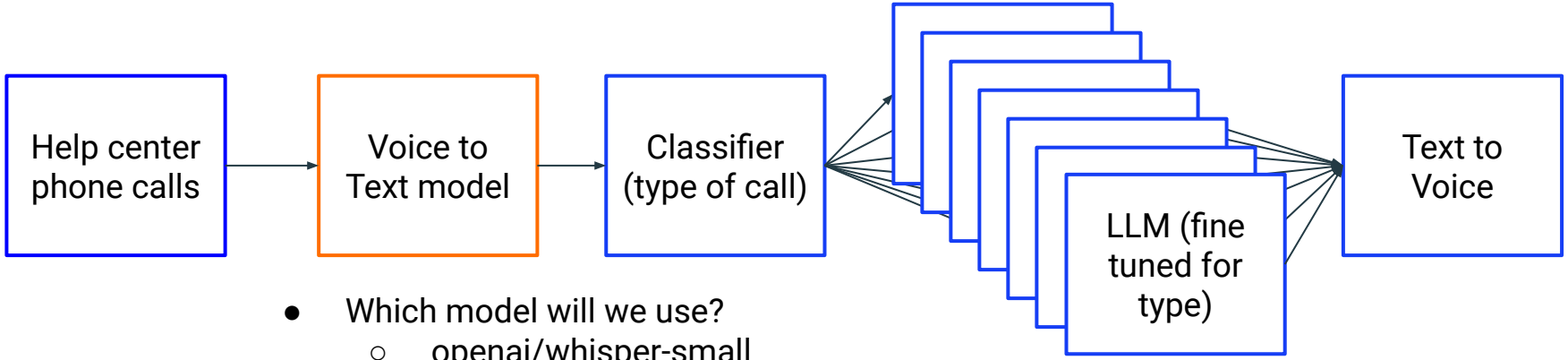
- What file format?
  - .wav
- Where are they stored?
  - GCS bucket
- How long are the audio clips?
  - 1-2 minutes

Our Pipeline



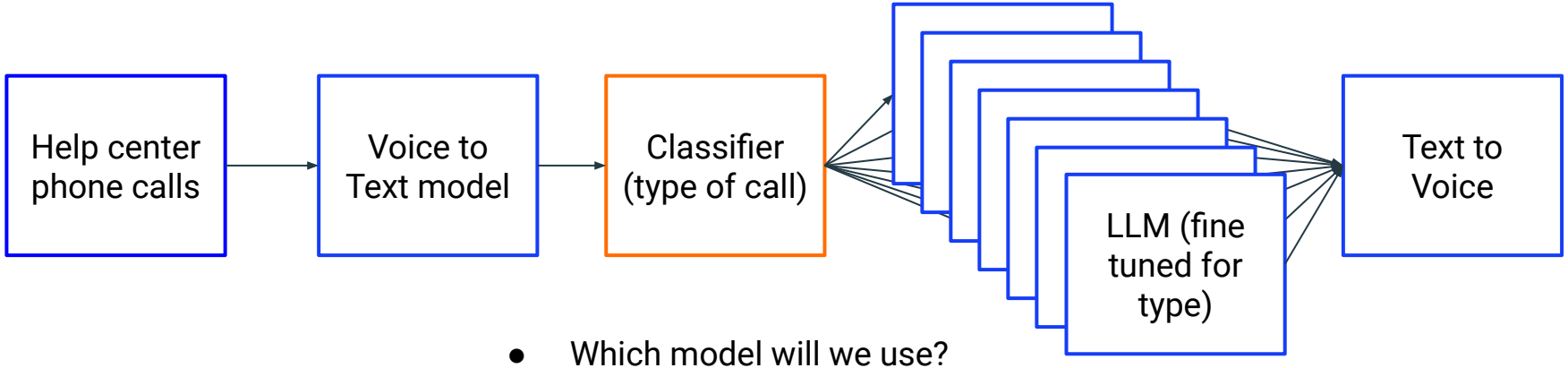
- Which model will we use?
- Where will we get the model?
- What framework?
- Which model handler?
- How we will preprocess the data?

Our Pipeline



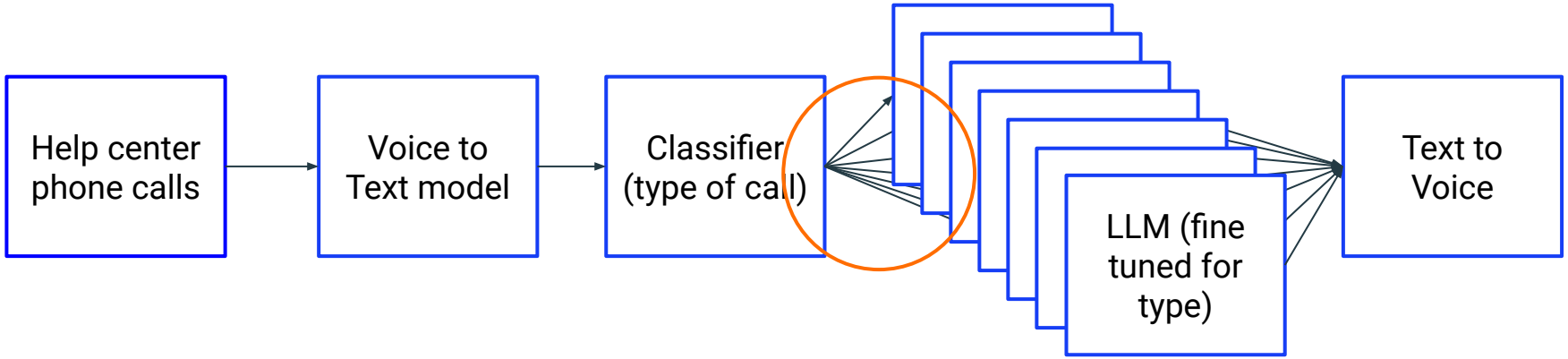
- Which model will we use?
  - openai/whisper-small
- Where will we get the model?
  - HuggingFace
- What framework?
  - pytorch
- Which model handler?
  - HuggingFacePipelineModelHandler
- How we will preprocess the data?
  - HuggingFace Pipelines will do it automatically

Our Pipeline



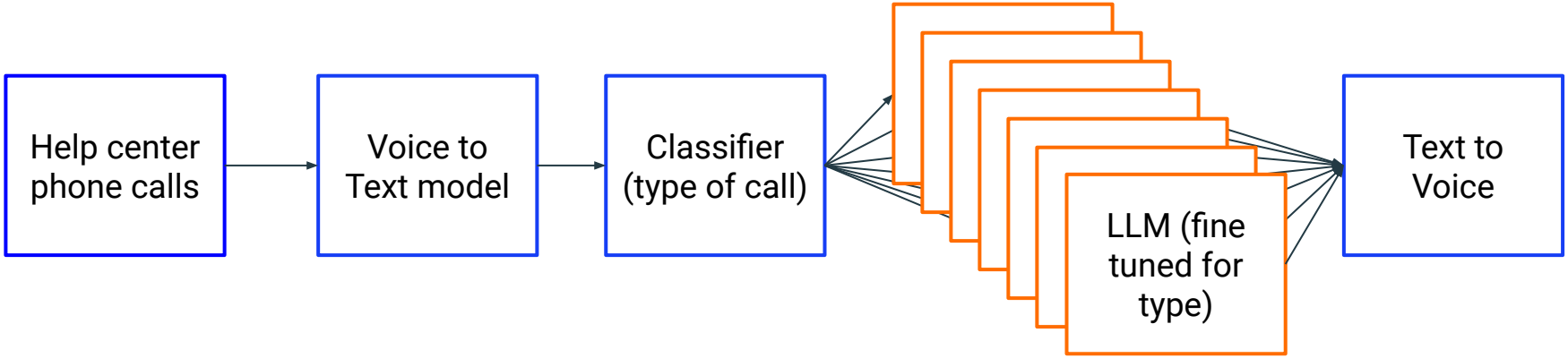
- Which model will we use?
  - XGBoost classifier
- Where will we get the model?
  - We will train it
- What framework?
  - XGBoost
- Which model handler?
  - XGBoostModelHandlerPandas

Our Pipeline



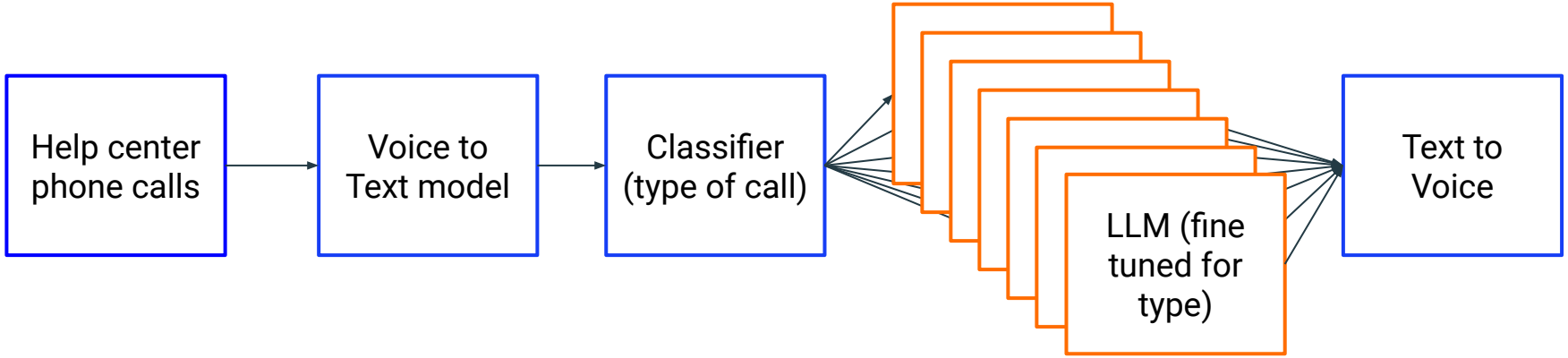
- How do we match class to model?
  - Use a KeyedModelHandler
  - https://cloud.google.com/dataflow/docs/notebooks/per_key_models

Our Pipeline



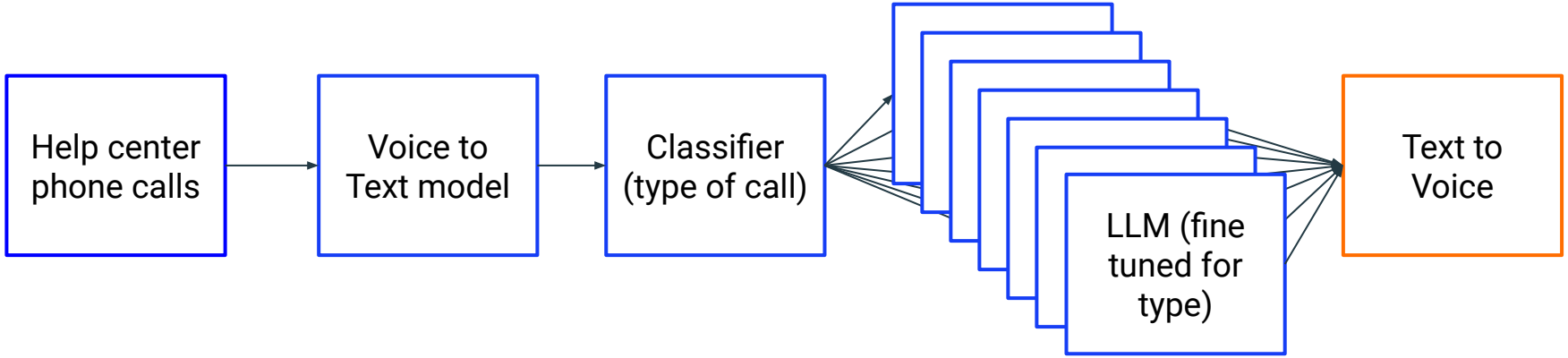
- Which models will we use?
- Where will we get the model?
- What framework?
- Which model handler?

Our Pipeline



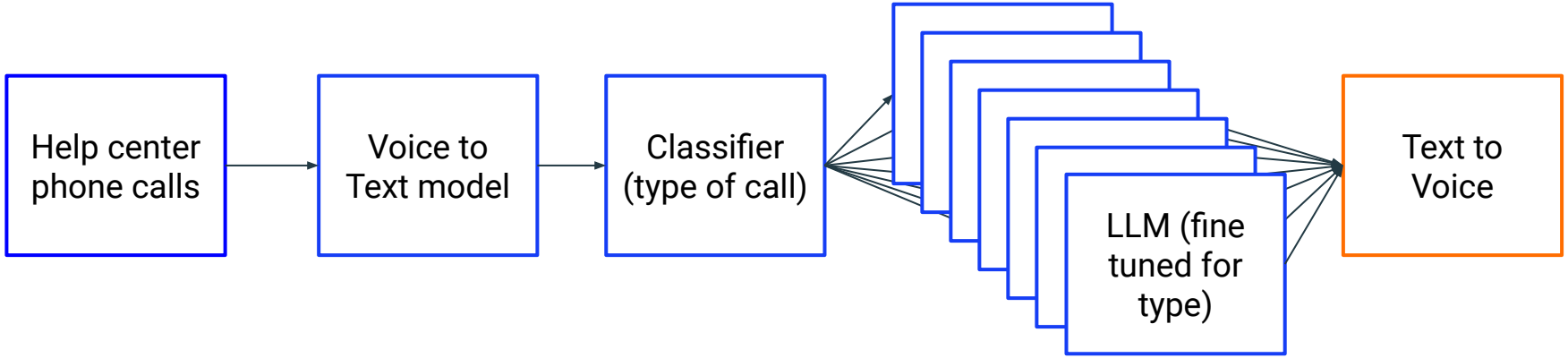
- Which models will we use?
  - Small LLMs: GPT2, Bert, etc
- Where will we get the model?
  - HuggingFace
- What framework?
  - PyTorch
- Which model handler?
  - HuggingFaceModelHandler

Our Pipeline



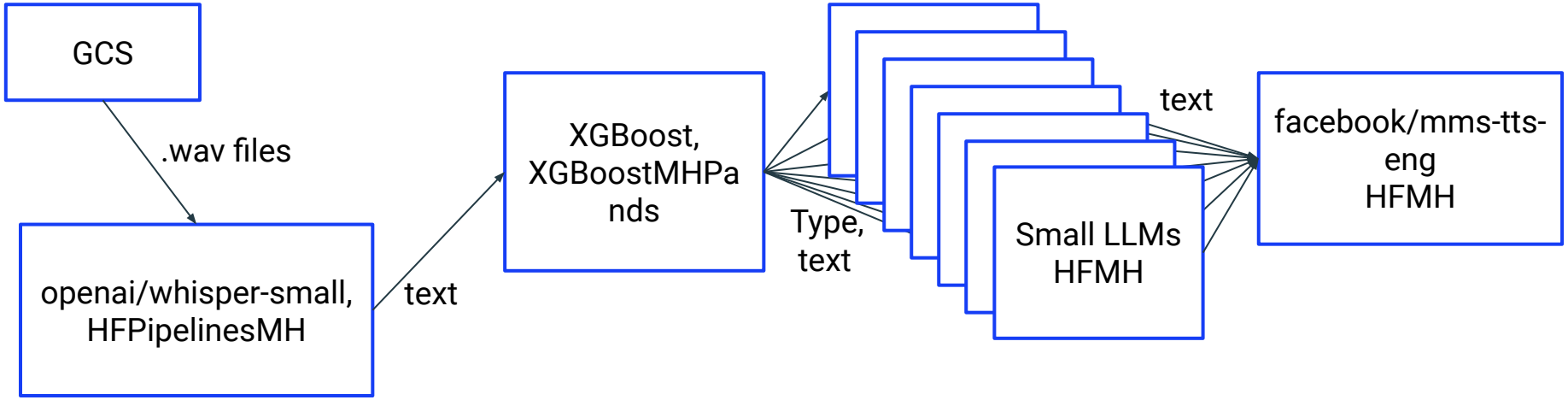
- Which model will we use?
- Where will we get the model?
- What framework?
- Which model handler?

Our Pipeline



- Which model will we use?
  - facebook/mms-tts-eng
- Where will we get the model?
  - HuggingFace
- What framework?
  - PyTorch
- Which model handler?
  - HuggingFaceModelHandler

Our Pipeline, detail view



HF = HuggingFace
MH= ModelHandler

Links

- Speech to Text: https://huggingface.co/openai/whisper-small
- XGBoost: https://xgboost.readthedocs.io/en/stable/
  - Beam notebook: https://github.com/apache/beam/blob/master/examples/notebooks/beam-ml/run_inference_xgboost.ipynb
- Text to Speech: https://huggingface.co/facebook/mms-tts-eng
- Beam KeyedModelHandler notebook: https://cloud.google.com/dataflow/docs/notebooks/per_key_models

Next Session

Deep dive into our example pipeline, Part I

DataflowWorkerAcceleratorTypes

Thank you!

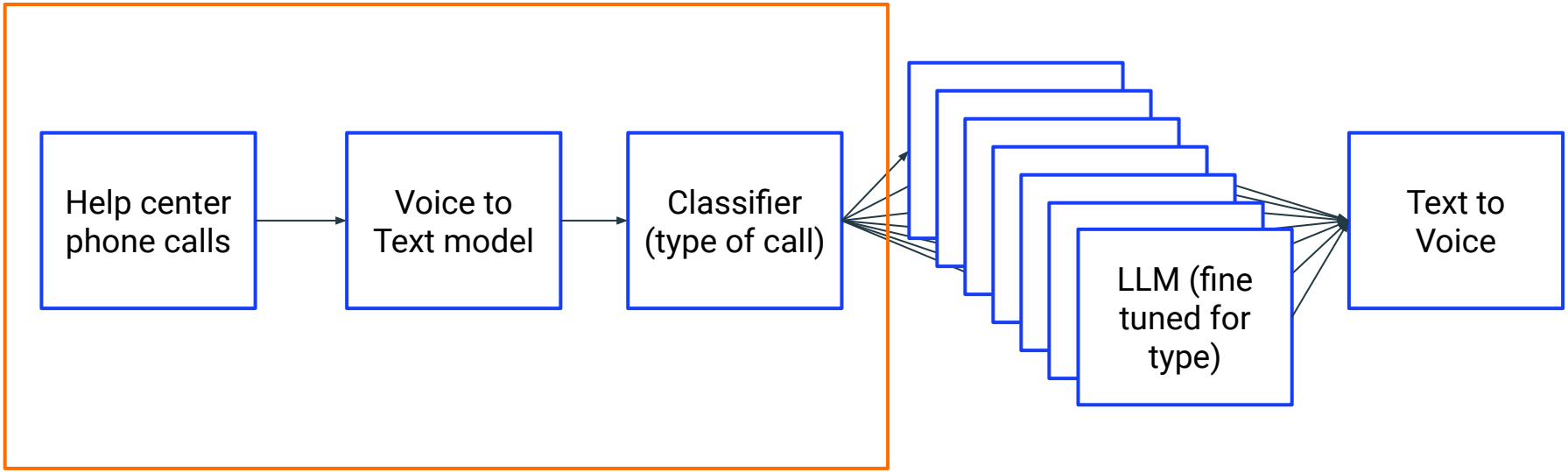
Intermediate Track: Implementing a complex ML pipeline Session 4

Dr. Kerry Donny-Clark

Implementing a complex ML pipeline: Overview

1. Intro to ML in Beam: RunInference and Model Handlers
2. Choosing Models and how to adapt a model to Beam
3. High level view of the pipeline
4. **Deep dive into our example pipeline, Part I**
5. Deep dive into our example pipeline Part II
6. Expanding the pipeline to real life use cases

Our Pipeline
We will cover the first half



To Colab!

Next Session

Deep dive into our example pipeline, Part II

DataflowWorkerAcceleratorTypes

Thank you!

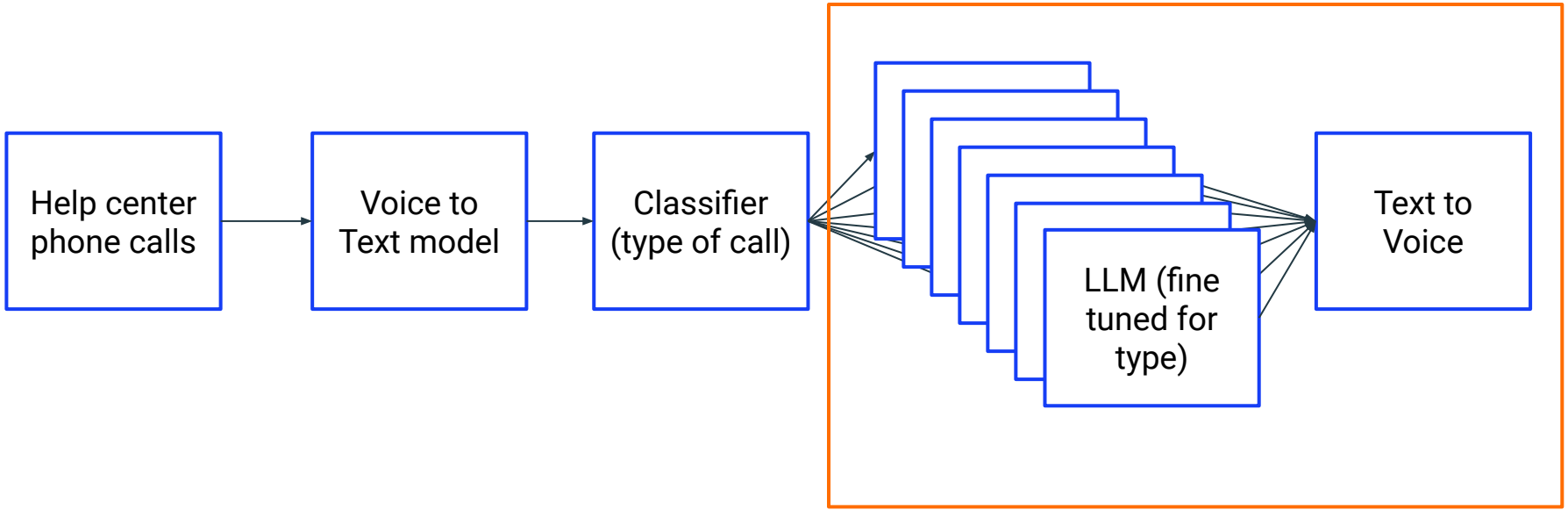
Intermediate Track: Implementing a complex ML pipeline Session 5

Dr. Kerry Donny-Clark

Implementing a complex ML pipeline: Overview

1. Intro to ML in Beam: RunInference and Model Handlers
2. Choosing Models and how to adapt a model to Beam
3. High level view of the pipeline
4. Deep dive into our example pipeline, Part I
5. **Deep dive into our example pipeline Part II**
6. Expanding the pipeline to real life use cases

Our Pipeline
We will cover the second half



To Colab!

Next Session

Expanding the pipeline to real life use cases

DataflowWorkerAcceleratorTypes

Thank you!

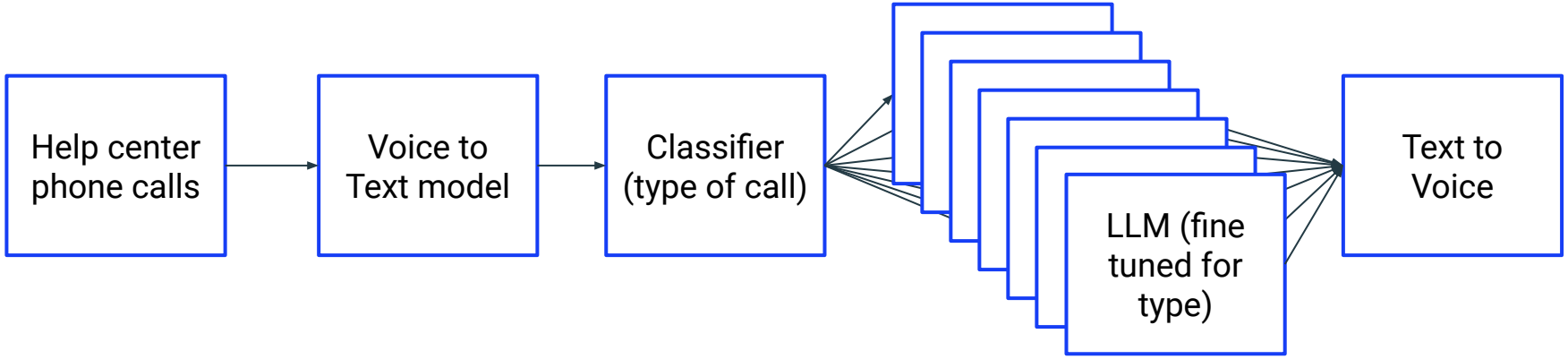
Intermediate Track: Implementing a complex ML pipeline Session 6

Dr. Kerry Donny-Clark

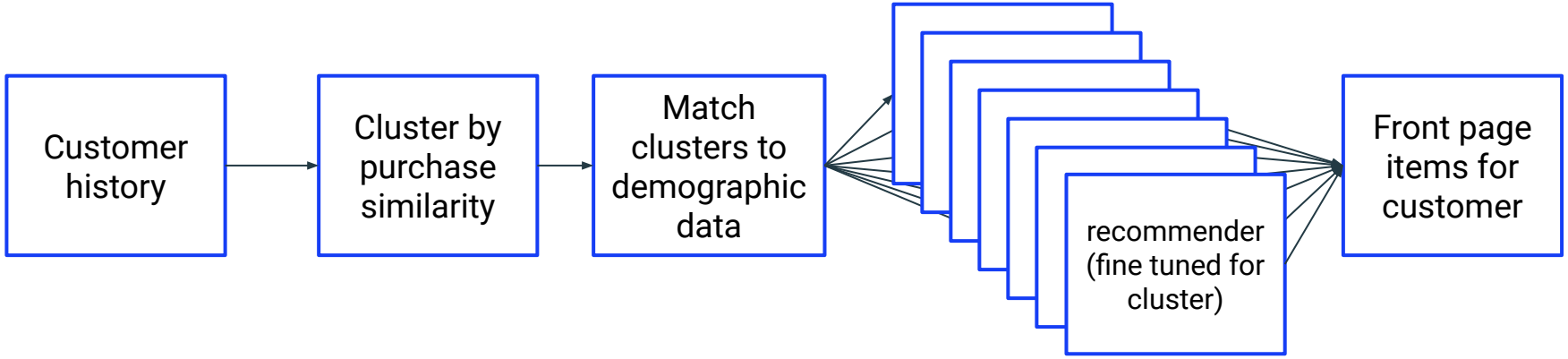
Implementing a complex ML pipeline: Overview

1. Intro to ML in Beam: RunInference and Model Handlers
2. Choosing Models and how to adapt a model to Beam
3. High level view of the pipeline
4. Deep dive into our example pipeline, Part I
5. Deep dive into our example pipeline Part II
6. **Expanding the pipeline to real life use cases**

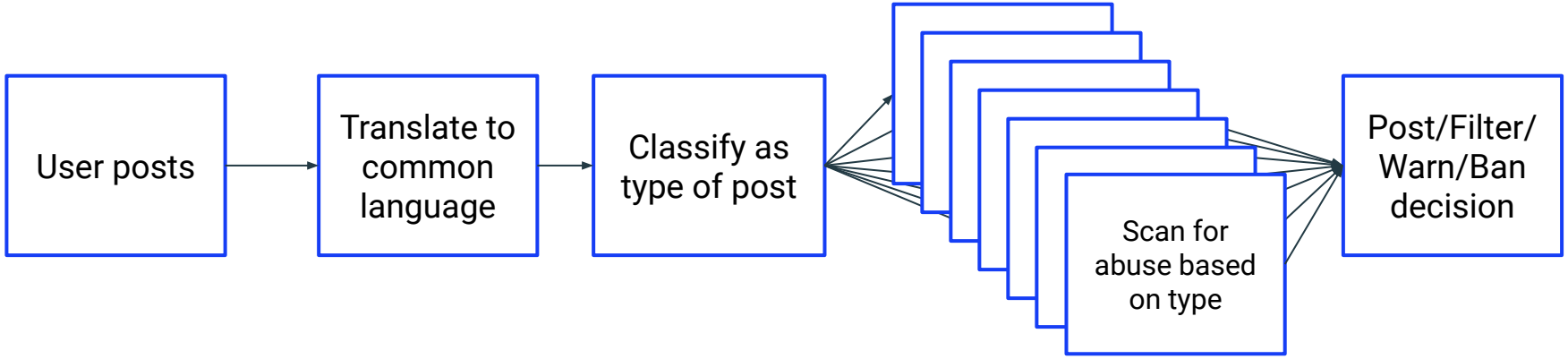
Our Pipeline



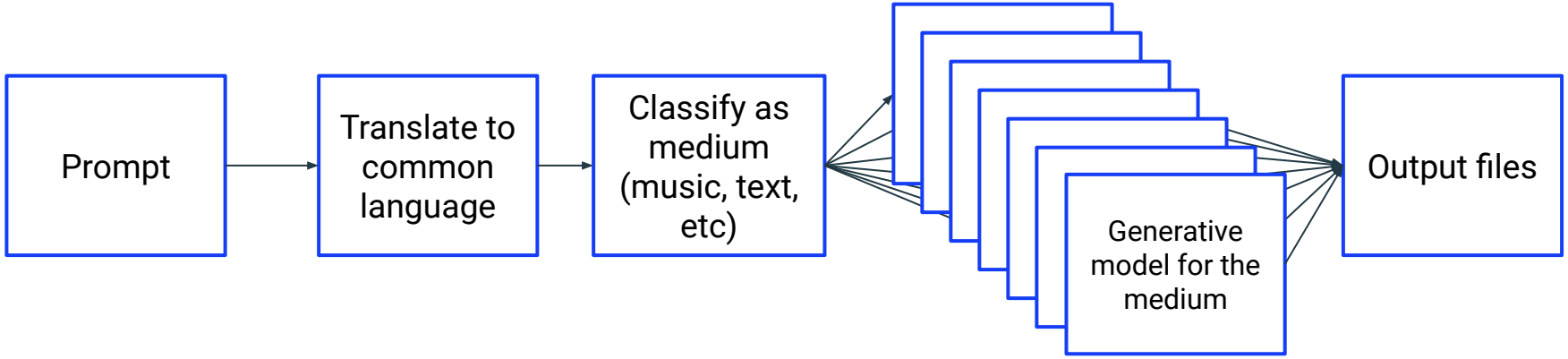
Same Pipeline, different business problem



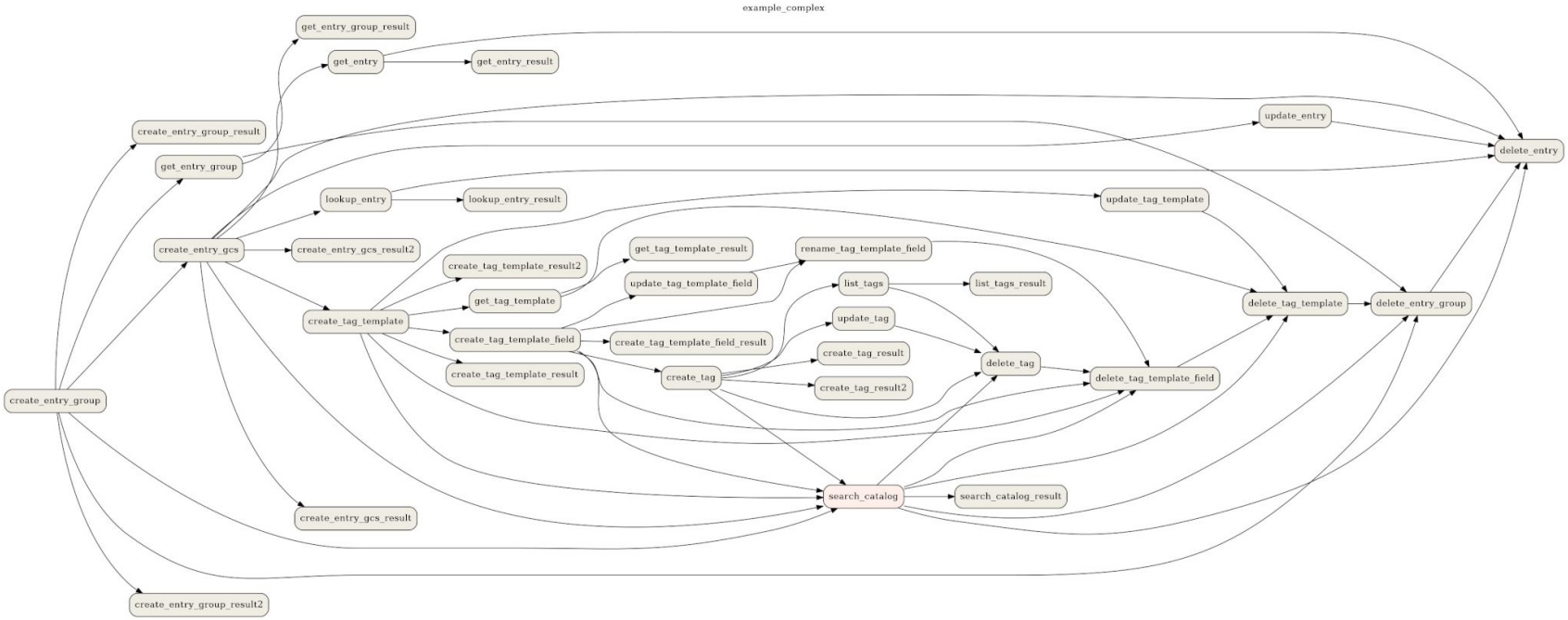
Same Pipeline, different business problem



Same Pipeline, different business problem



And of course it can get crazier



Source: Apache Software Foundation

DataflowWorkerAcceleratorTypes

Thank you!