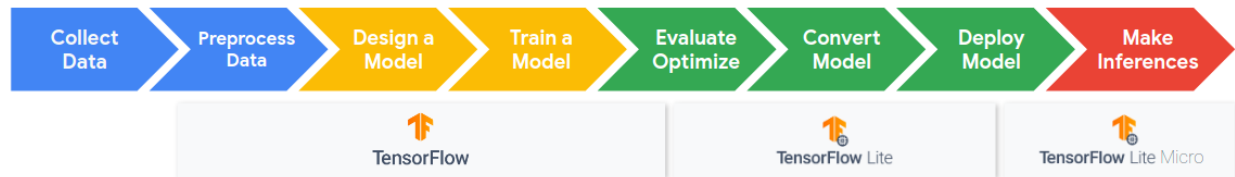# Keyword Spotting Application Architecture

In this reading, we will go over the high-level architecture of the application before we go deep into building the keyword spotting application (KWS).

Step 1 Data collection

For Keyword Spotting we need a dataset **aligned** to individual words that includes thousands of examples which are representative of real world audio (e.g., including background noise).

Step 2: Data Preprocessing

For efficient inference we need to **extract features** from the audio signal and classify them using a NN. To do this we convert analog audio signals collected from microphones into digital signals that we then convert into spectrograms which you can think of as images of sounds.

Step 3: Model Design

In order to deploy a model onto our microcontroller we need it to be **very small**. We explore the tradeoffs of such models and just how small they need to be (hint: it's tiny)!

Step 4: Training

We will train our model using standard training techniques explored in Course 1 and will add new power ways of analyzing your training results, confusion matrices. You will get to train your own keyword spotting model to **recognize your choice of words from our dataset**. You will get to explore just how accurate (or not accurate) your final model can be!

Step 5: Evaluation

We will then explore what it means to have an accurate model and why your training/validation/test error may be different from the **accuracy experienced by users**.

Additional Topics:

We will also consider some additional topics that are unique to keyword spotting: post processing and cascade architectures.