**Usage of Surround**

We need to understand the goals that surround is trying to achieve in order to understand under which circumstances does one need surround to use. The problems that surround is aiming to overcome are

* The same changes were required again and again to refactor code written by data scientists to make it ready for serving e.g. no standard way to run scripts, no standard way to handle configuration and no standard pipeline architecture.
* Existing model serving solutions focus on serving the model rather than serving an end-to-end solution. Our machine learning projects require multiple models and glue code to tie these models together.
* Existing serving approaches do not allow for the evolution of a machine learning pipeline without re-engineering the solution i.e. using a cloud API for the first release before training a custom model much later on.
* Code was commonly being commented out to run other branches as experimentation was not a first class citizen in the code being written.

And under any of the following circumstances one should consider using surround

* You want a flexible way to serve a pipeline in Python without writing C/C++ code.
* You have multiple models (custom or pre-trained) from different frameworks that need to be combined into a single Surround solution.
* You want to use existing intelligent APIs (AWS Rekognition, Google Cloud AI, Cognitive Services) as part of your Surround implementation.
* You have pre or post processing steps that aren't part of your models but need to be deployed as part of your Surround implementation.
* You need to package up your dependencies for running Surround as an offline solution on another machine.

In other words the goal of surround is simply to make the life of data scientists easier and at the same time give them the opportunity to work on their tasks without worrying about the codes they are leaving behind.