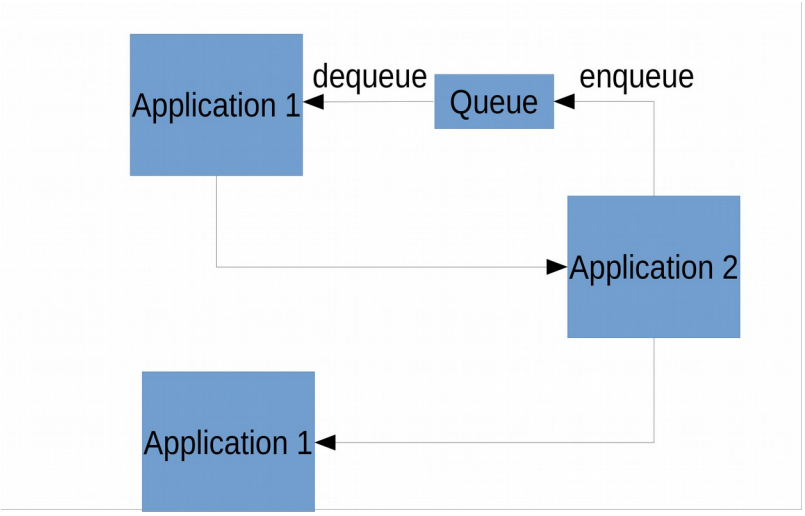
7. This is an architecture problem, please send solutions as an image (can be hand drawn) or a word art.

1. **Please see the flow chart given below**
2. **Application 2 is a blocking process**
3. **Problem/Desired Solution:**
    1. **The Application 1 has no information on how far Application 2 has reached.**
    2. **We need a solution in a way in which Application 1 can track and display how far Application 2 has reached.**
4. **Constraints:**
    1. **Application 1 and Application 2 can't exchange variables/information directly once Application 2 starts**
    2. **Application 1 can exchange any number of variables when it is initializing Application 2 and vice versa at the end when Application 2 ends**
    3. **We control what code goes in Application 2 and do anything we want**
5. **Assumptions**
    1. **We can assume Application 1 is a python GUI framework**
    2. **Application 2 can be anything, like COM object, another python/C/C++/java application**
6. **Note:**
    1. **Please justify your solution and why you think it would be the best.**
    2. **Also, please elaborate your solution sufficiently.**

This is a classic problem that requires **inter process communications**. This can be achieved by **Publish-Subscribe pattern**. There are various ways this can be achieved. For this application which does not share complex information a **simple queue** between from **multiprocessing** module can be used between Application1 and Application2. Application 2 can publish the task completion percentage to the queue and Application 1 can read from it and publish on the UI. This solution is simple and elegant. However, the classic problem with this is of **synchronization**. Both applications needs to be synchronized to access the queue. The synchronization can be achieved easily by the use of **locks**. Please refer the following figure for the above solution.



However, for complex information distribution, there are many message brokers available with various functionalities. Some of the famous ones are Kafka, ActiveMQ, Redis, RabbitMQ.