# TourRadar

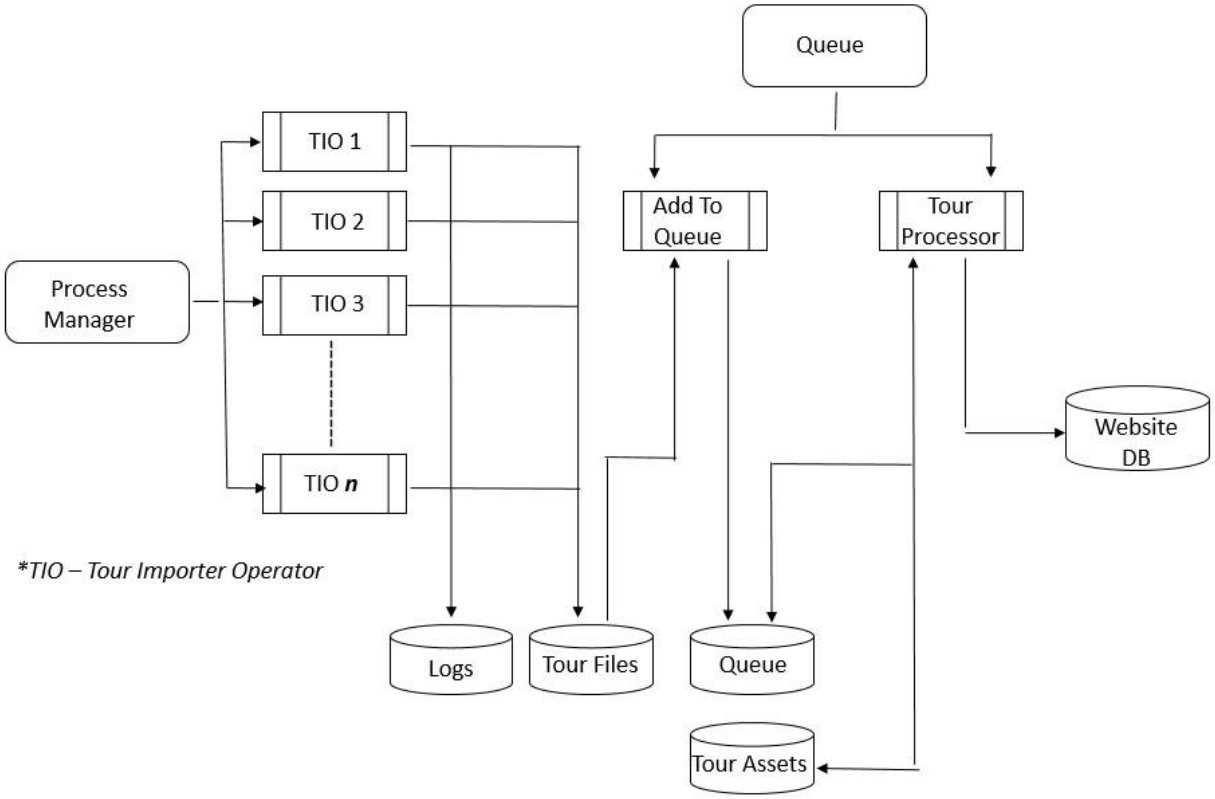
# Backend Engineer Test Case: Scaling Importers

Questions you asked me to answer:

**What problems do you identify in the current setup? Please enumerate them with a brief description of why you believe they are problems, and what risks they carry.**

1. I think that TourFile storage may be overloaded because of multiple queries from Importers and Processor.
2. I didn't understand why we don't store TourAssets in Website DB (may be I didn't understand right the schema, but is this TourAssets storage used anywhere except when it's inserted?)

**What new architecture would you suggest that optimizes for performance, scalability, and reliability?**




I thought about something like on image above. For example, if we will not take from TourFile every time one file by TourProcessor, but we will add a script which will be ran every ***n*** mins and will take more TourFiles by one query and put these files into a Queue storage. And Processor will take one by one files from Queue, will process it and will delete record from Queue.

To not take every time the same files to the Queue, we can add a *is_processed* field in TourFiles storage and when script adds files to the Queue – put 1 for that field and by default it will be 0 and every time we will select all files with *is_processed=0*.

I think that it will help TourFiles storage to work with the multiple queries at the same time from Importers and only once in ***n*** mins to work with the Processor. It also saves data in case when server is stopped (or another exceptional situation happens) and all caches are deleted, we can find the data from queue in Queue storage.

In the practical part I implemented a kind of this my Queue logic.

**How would you ensure your chosen architecture is working as expected?**

I can see any time which of TourFiles was processed. Also, to catch exceptions I can propose to write in Logs a record if the Processor has any errors.

How can I ensure? Difficult to answer, it also depends of many things which I don't know at the moment about the project. Theoretically its always easier than in practice.