Summary

Our project is developed following a dependency structure, typical of the audit chain module.
The main idea is that events of any kind, like IoT, file systems, and measurement loggers should be transmitted securely via the network. There is a central event queue, which creates and records these events (producer). After the producer successfully generates an event, it goes to rabbitMQ (event queue) and uses them for further steps that would involve third parties (consumer). Events are recorded serially and the central event queue is transmitted via the network queue.
To achieve this goal we used a variety of technology stack such as Java as a main programming language and Python for the implementation of GUI. Also, we installed Docker, so our system resources to be used efficiently (consume less main memory, easy to port etc) and JUnittests to make sure that nothing in the code is broke.