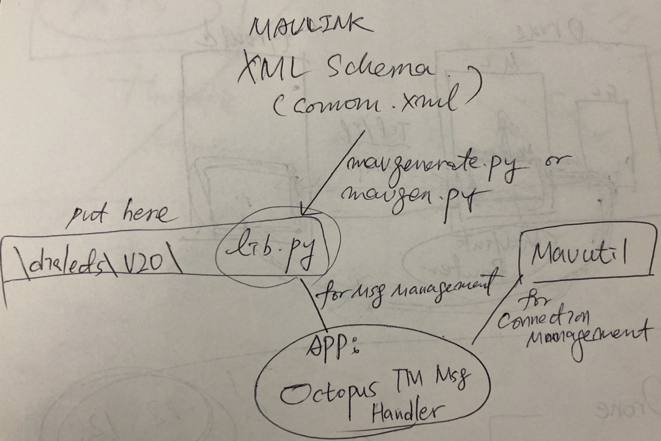
<2021.05.13, Octopus TM Msg Handler 개발 구조>

* TM message 처리 전에 peer (예, Relay Server -> Mavlink TM Message Interpreter SVC”) 간 connection 관리가 필요하며 이는 <https://mavlink.io/en/mavgen_python/> 를 참고
* 상기 내용들은 python 으로 개발되어 있으나 Octopus 전용으로 C# 서비스를 개발하는 것보다, 상기 python 기반 기능들을 활용하는 것이 여러모로 나아보임



<2021.05.10, j2ysoft에 전달사항>

* CPS configuration
  + 일반적으로는 드론→GCS 컴퓨터로의 TM 전달은 양단간 장착된 holybro TM 장비(915MHz)를 통해 direct connection 이 이루어짐
  + 그러나 relay server 를 활용(i.e., through relay server in the middle)하는 구조에서는 {relay server}-{물리 드론, GCS, Octopus} 가 모두 TCP/IP 기반으로 연결되어야 함 (e.g., WiFi, 유선망 등)
    - 아래 그림에서 Ground 는 일반적으로는 GCS 이나, DNA+드론 사업 또는 CPS 과제에서는 relay server 가 될 것임



* + - Ground(i.e., relay server) 내에는 보통 mavlink 의 xml schema 인 common.xml 을 특정 프로그래밍언어로 변환하는 변환기 (예, python의 경우 <https://mavlink.io/en/getting_started/generate_libraries.html>)를 사용하여 mavlink msg handling 프로그램을 만듬
      * 특정 port 에 대하여 대기 (e.g., wait for drone1 at 14550 and drone2 at 14551)
      * Drone 이 켜지면 약속된 relay server (ip, port) 에 붙어서 TM 전송
      * Relay server 는 수신한 TM 을 미리 설정된 GCS(e.g., Octopus) 로 relay 함
        + CPS 과제 내에는 Relay server 에 연결되는 entity 들 정보를 관리하는 연결관리서버가 없으므로, static 하게 relay table 을 관리할 수 있어야 함

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Drone ID | GCS Server IP  SndTMToGCS | Relay Server port  RcvTMFromDrone | GCS Server port  SndTMToGCS | 영상전송  to Drone  [TBD] |
| 1 | 129.x.x.101 | 14550 | 14550 |  |
| 2 | 129.x.x.101 | 14551 | 14551 |  |