1. 报文是如何设计的：考虑数据加密，协议id占的字节（4字节：2^32），协议头（协议id+协议长度）+协议体（详细数据）

报文样例：

帧头+协议头+详细数据+帧尾

1. Netty对接卫星：自定义编码解码器（粘报拆包），解析自定义的报文
2. 智能卫星报文解析引擎：
   1. 数据源文件存储到MinIO（备份数据）
   2. 解密报文
   3. 协议信息加索引（协议id索引）。根据协议id从协议项表中获取完整的协议信息
   4. 解析报文按照key-value的形式对照解析（异步解析，提高吞吐量）
   5. 存储（MySQL（元数据）+MongoDB（具体数据，组合索引，主从集群，分片））+传输（MQ，消息不丢失，幂等性、持久化，高可用）
3. 设备报警：
   1. 通过MQ接收协议数据（考虑消息积压）
   2. 通过MySQL查询报警规则（数据库加索引（gliderId，协议id）），规则是包括等级的。
   3. 包含超时规则：
      1. Redis的过期回调
   4. 其他规则
      1. SpEL表达式
      2. 动态窗口（一定时间范围的窗口）的报警次数判断
         1. Redis（value：list）
         2. 延时队列+全局变量（byte[]）
   5. 报警：邮件（3级）+短信（2级）+电话（1级）
4. 设备拍摄的图片，通过卫星发回，分多包数据，解析后拼成完整的图片。
5. 前端页面展示报告，用户可选择下报告文件。

前端页面同一时间段的页面可以使用freemarker或thymeleaf存minio，步骤如下。

海事报告下载：大数据量情况下

* 1. 判断该时间段报告是否生成过（MySQL）
  2. 生成过：
     1. 直接查minio获取
  3. 未生成过：
     1. 从MongoDB查询设备数据（组合索引+分片）
     2. 使用异步生成报告
     3. 将报告采用分片的形式上传minio（minio存的是分片）
        1. 文件分片的地址以list返回给前端，前端从minio获取报告分片后拼成整个文件，展示给用户
        2. 使用邮箱发送给用户，考虑压缩。