MPS MRP

在s-planmng

计算方案

现状

analyze-plan-api，增删改查、计算，包括一组自动运行设置，预览

计划

计算结果

现状

analyze-result-api，增删改查，结果创建计划

计划

即时通信

在p-message，依赖rabbitmq、mongodb

消息网络构建

现状

每个企业+用户占一个queue，sessionId作为topic，按session建exchange按照用户、其他用户bind exchange和queue

机制上不去分1对1还是群聊还是通知，通知的在发送完通知后立即解散所以无法回复

计划

为了实现用户多终端，考虑每用户建一个exchange，exchange到exchange绑定

消息路由过程

现状

由rabbitmq处理，用的是publish subscribe，消息持久化在mongo里，消息格式有前端确定，后端不依赖消息格式及消息内的内容，例外，用md5 checksum做消息id

兰石ERP集成及二次开发

在i-lanshi，依赖i-mongo（集成专用mongodb）

工单获取

ERP端调用

可多次调用

目前只处理了工单成品修改合并，工单工序合并有一些交互上的问题，只保存了第一次调用时过来的工序，后续有需要的话要再与用户讨论

单位处理问题，目前接口只认为Ea是数量单位，ERP中一般用Ea也有“件”，NoMES中Ea转换为“件”，这里会导致回传时单位找不回来，如有问题还需与用户讨论

成品、废品数量回传

通过spring-integration调用，可多次调用，每次传输会在mongo里记，然后统计传输总数，跟系统里的合格品差值后只传差值，可负

热处理工序自动完工

目前处理的是/opers/operstatus/{id}，如果有其他接口需要增加，没有测试

工时归集

展开工单、拆批工单、热处理工单树，加工工序和热处理工序统计，前端合并计算。其中主要涉及产量统计问题，工时使用的是绩效，工单拆批及热处理工单

二次开发

通过nginx代理转发到i-lanshi上做附加处理，具体可以看一下兰石10.10.10.184上enterprise的nginx配置，需要注意跨服务事务问题

采集

参数映射配置在mongodb里

接口调用具体可以找吴桐或者兰石的人

运维

兰石系统

docker-compose.yml

一些其它的想法

1. 附件等文件存储考虑放到一个分布式文件系统容器内，volume都在docker-compose里定义，业务数据不要再用本机文件系统了，便于在docker体系内迁移扩展
2. 企业级docker-compose升级到v3，rolling update需要v3.4