框架设计-2017/3/11

一、微服务部分

1. 暂时不考虑服务注册和发现，服务地址和端口都是硬编码，初始化时就要设定好，并且服务调用方已知服务地址。
2. 服务与消息代理通过REST API通信
3. 服务通信层面先只考虑http和mqtt两个协议，一个是请求/响应类型，一个是发布/订阅类型

* 基于http协议通信的服务，消息代理只需转发请求响应即可

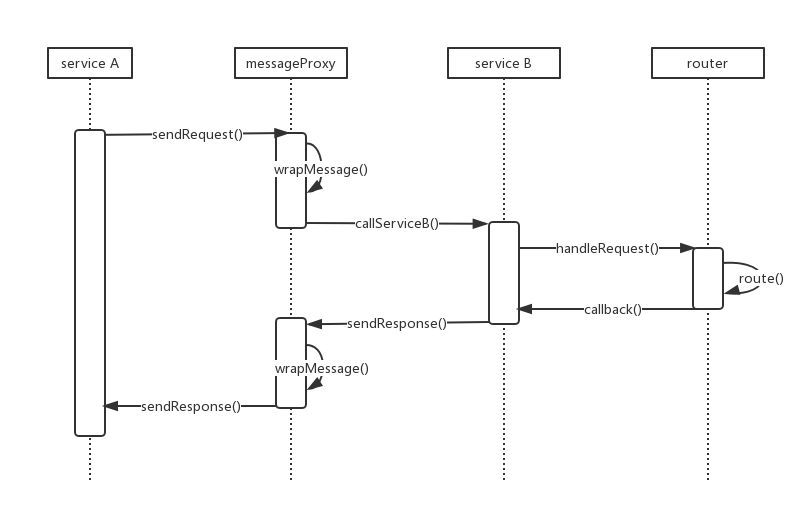


图1 http请求/响应消息代理

* 基于mqtt协议通信的服务，消息代理订阅服务，对于服务发布者，它的订阅者是消息代理，因此消息代理要维护该服务的真正订阅者列表，以便在代理收到服务推送时将其转发给真正的服务订阅者。

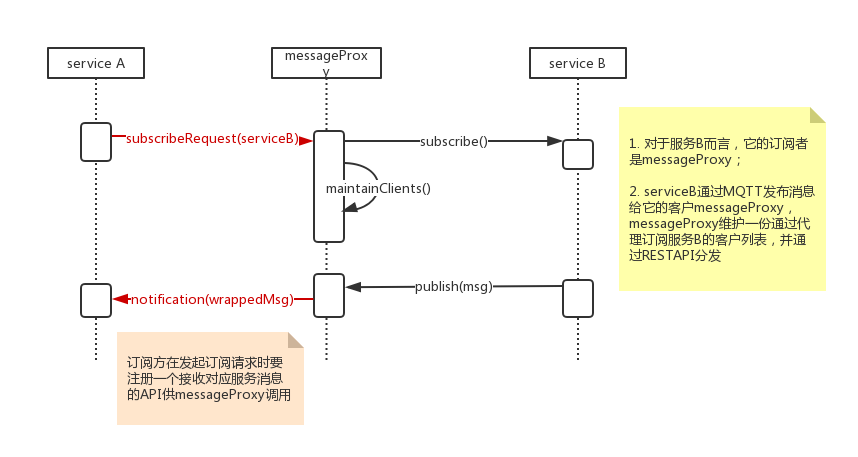


图2 mqtt发布/订阅消息代理

1. 硬件抽象部分

底层和设备、硬件平台相关的抽象就按照cylon.js，还是把元件和平台抽象成两层，这样组合更灵活一点。

* 1. 抽象类包括robot, driver, adaptor要确定抽象接口和组装规则
  2. Platform(Adaptor的具体实现)先挑几个目前最流行的实现，我们可以先实现对Arduino和Raspberry Pi的支持，正好这两个设备实验室也有，之后测试也方便一些
  3. Device(Driver的具体实现，与平台相关)同上，基于Platform挑基础的几个先实现

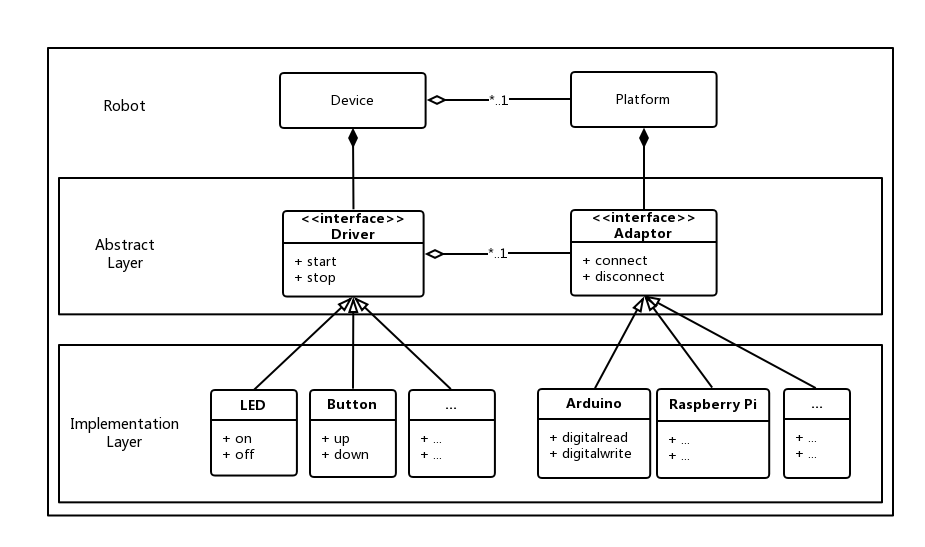


图3 底层分层架构

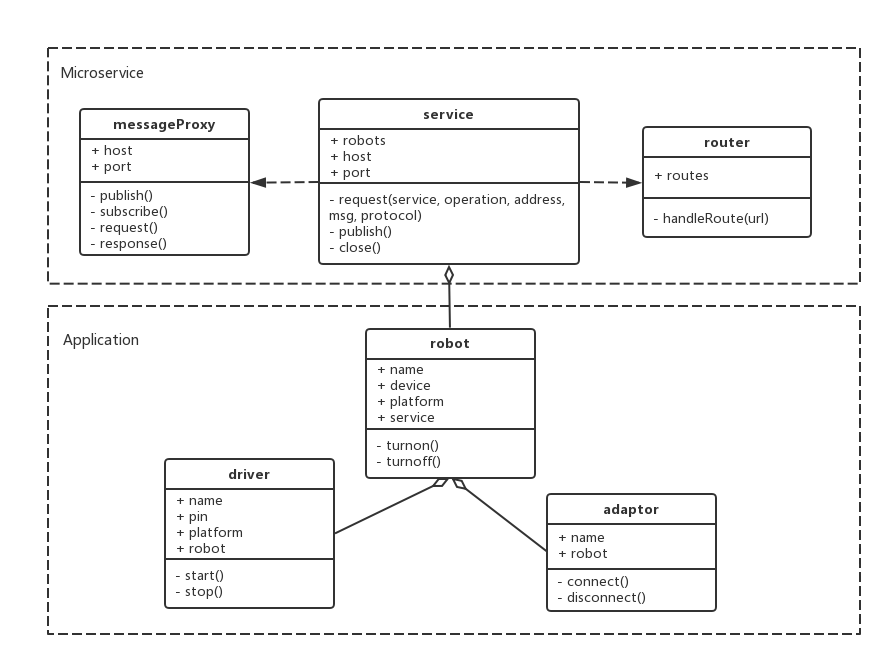
1. 把底层和服务层结合

图4 关键模块

以上这些是比较关键的模块，在实现的时候肯定还会有细分的情况，比如

1. messageProxy下可能会要分出分别对http协议和mqtt协议封装处理的辅助模块；
2. 应用启动统一入口，完成初始化等工作
3. 通用工具模块
4. 等等