**Tbox 上位机 GUI 界面构建**

1、界面交互架构对比

* 1. B/S和C/S交互架构介绍
  2. 软硬件环境要求
  3. 各自优缺点

1. BS和C/S 交互架构开发语言环境
2. 基于tbox架构，对接GUI界面
3. 界面交互架构对比
   1. 1.1 B/S和C/S交互架构介绍

关于软件GUI交互界面构建可选形式，有C/S架构和 B/S架构两种方式：

C/S架构即Client和Server，客户使用软件，需要安装客户端软件，才能使用，但可单机使用，也可以联网使用。如果如果联网，程序需要搭建服务器server，后期可以对客户端进行升级维护, 方便功能上的额外拓展。

B/S架构即Brower和Server，客户无需安装客户端，只需通过浏览器访问web地址即可使用软件界面，但需要搭建server端，但优点是跨平台，软件升级维护，功能拓展等，不需要用户做啥工作。

* 1. 软硬件环境要求

B/S架构对于客户端用户来说，只需要一个浏览器即可使用，但是C/S架构用户，如果处理的是较为复杂的，数据量大的业务时，对客户电脑内存，cpu等会有最低要求。

* 1. C/S和B/S优缺点对比：

CS优点：

A.能充分发挥客户端PC的处理能力，很多工作可以在客 户端处理后再提交给服务器，所以CS客户端响应速度快。

B.操作界面漂亮、形式多样，可以充分满足客户自身的个 性化要求。

C. C/S结构的管理信息系统具有较强的事务处理能力， 能实现复杂的业务流程，如果对稳定性有要求，C/S是比 较有优势。

D. 安全性能可以很容易保证，C/S一般面向相对固定的用 户群，程序更加注重流程，它可以对权限进行多层次校验， 提供了更安全的存取模式，对信息安全的控制能力很强。 一般高度机密的信息系统采用C/S结构适宜。

C/S缺点：

A. 需要专门的客户端安装程序，分布功能弱，针对点多 面广且不具备网络条件的用户群体，不能够实现快速部署 安装和配置。

B.兼容性差，对于不同的使用环境，比如windows, linux， mac，需要针对不同环境，开发不同版本，具有较大的局 限性。开发、维护成本较高，发生一次升级，则所有客户 端的程序都需要改变。

C. 用户群固定。由于程序需要安装才可使用，因此不适 合面向一些不可知的用户，所以适用面窄。

BS优点：

A.分布性强，客户端零维护。只要有网络、浏览器，可以 随时随地进行查询、浏览等业务处理。

B. 业务扩展简单方便，通过增加网页即可增加服务器功 能。

C. 维护简单方便，只需要改变网页，即可实现所有用户 的同步更新。

D. 开发简单，共享性强。

B/S缺点：

A. 个性化特点明显降低，无法实现具有个性化的功能要 求。

B. 客户端服务器端的交互是请求-响应模式，通常动态刷 新页面，响应速度明显降低（Ajax可以一定程度上解决这 个问题）。无法实现分页显示，给数据库访问造成较大的 压力。

C. 在速度和安全性上需要花费巨大的设计成本。

D. 功能弱化，难以实现传统模式下的特殊功能要求。

小结：CS响应速度快，安全性和稳定性强，一般应用处理复杂业务需求中，但是开发维护成本高；BS可以实现跨平台，客户端零维护，但是个性化能力低，响应速度较慢。所以有些单位日常办公应用BS，在实际生产中使用CS结构。

1. BS和C/S 交互架构开发语言环境

tbox由c++和python构建。

CS架构语言环境：

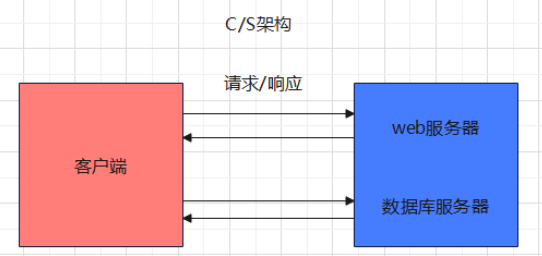
C++可选的方式有win32，mfc，qt等等，前两者只针对windows环境使用，qt可以跨平台。Python图形界面可选方式比较多，但是实际生产优选pyqt5。

所以比较好的搭配：

C++、qt、qtdesigner

Python、pyqt5、qtdesigner

整体对比而言，c++运行效率会比python高很多，但是python开发效率高很多，额外拓展库也很多。



BS架构语言环境：

BS架构环境语言选择非常多，因为需要闭环，涉及环节也特别多，也需要搭建后端服务器。后端服务器软件常见的比如Apache，IIS，Nginx，Tomcat等，后端服务器脚本语言比如java，php，node.js，python等等。数据管理也需要数据库比如mysql，oracle等等。前端web用户的 GUI界面搭配html+css+js(js 也有很多成熟框架)。所以可见，服务器+后端语言+数据库+前端语言才能成为构成一套完整的BS架构。

比较好的搭配：

Linux+apache+mysql+php+html+css+js

Linux+nginx+mysql+python+html+css+js

Linux+tomcat+oracle+java+html+css+js

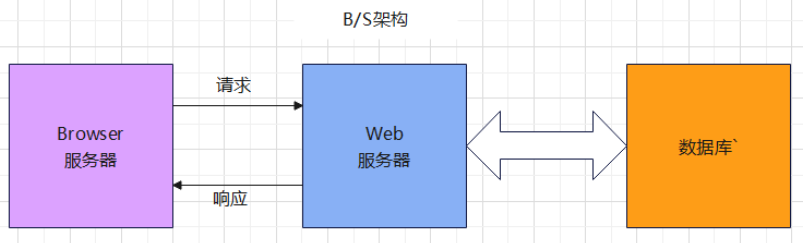
### 对于语言选择，不管是php，python还是java，都有比较成熟的web框架，比如php有thinkphp, Laravel,Yii等，python有Django， Flask，Tornado等，java有[hibernate](https://so.csdn.net/so/search?q=hibernate&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/lelouch_e/article/details/_blank)，struts2，MyBatis，Spring等等。

整体对比而言：

Java适合较大型应用，用户量大，并发大时，是比较好的选择，但是开发效率低。

PHP便捷开发，但只针对web应用，如果有其他额外需求和拓展，不能很好满足，就比较局限。

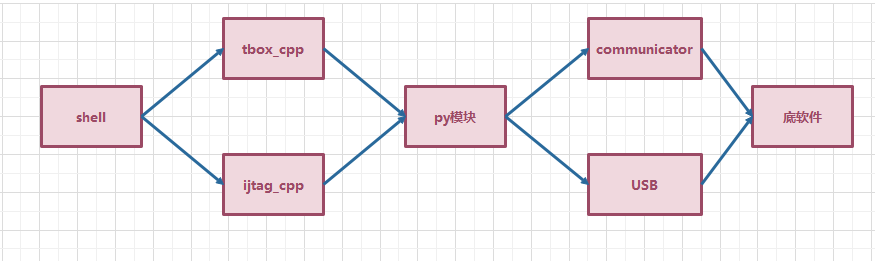
Python 便捷开发，不仅可以开发web应用，也能同时处理额外其他业务需求，提供的技术方案比较多。



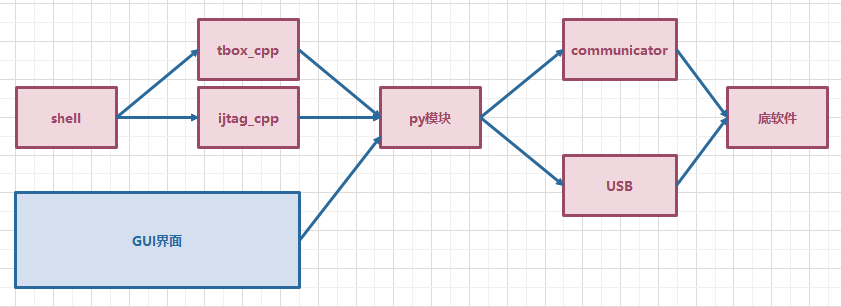
小结：基于现在tbox已经是一套成型的架构程序，在此基础上开发用户界面，C/S架构是比较优选的，能稳定处理流程复杂的，且数据量大的业务。开发结构相对简单，而B/S架构开发结构涉及面特别广，需要额外引入许多新技术形成闭环，并且对数据量大、单次长时业务处理交互时，web网络可能会存在问题。

1. 基于tbox架构，对接GUI用户界面

无论选择哪种架构，都会涉及交互界面与上位机数据对接，目前Tbox现有数据交互结构图：



在现有架构中，shell+cpp是控制台形式进行交互，cpp起到解析shell指令，并调用py模块处理业务，然后通过communicator或者usb两渠道与底软交互。如果在此基础上加入GUI用户界面交互。交互界面和tbox对接形式需要重点关注。如果cpp板块只是解析指令，调用py模块处理业务，那么可以并行加入GUI界面，直接GUI界面与py模块对接数据，如下图：



如果cpp板块除了处理解析指令，还有额外的其他业务处理，那么单独并行GUI的时候，依然需要对接cpp，是否可以考虑GUI界面与shell并行，数据和cpp对接，但是这样接受并显示cpp返回的信息时，处理起来稍微复杂。

