# Html跨平台ui

webkit

Chromium

Electron

Nodejs

Nw.js

与go相关的参考

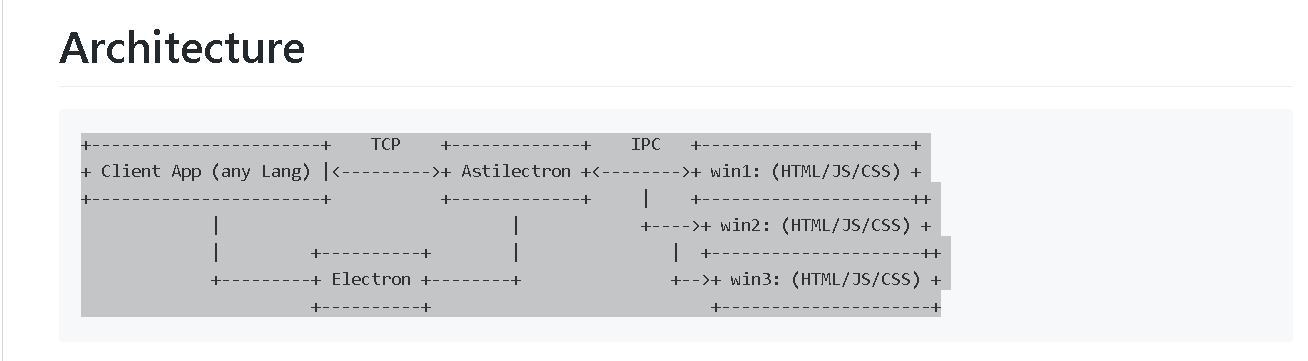
[astilectron](https://github.com/asticode/astilectron): <https://github.com/asticode/astilectron>

[go-astilectron](https://github.com/asticode/go-astilectron): <https://github.com/asticode/go-astilectron>

[starsManager](https://github.com/ZEROKISEKI/starsManager) :

<https://github.com/ZEROKISEKI/starsManager/>,

<https://juejin.im/post/5addcfd76fb9a07aa349df55>



<https://github.com/Equanox/gotron>

<https://github.com/zserge/webview>

<https://github.com/murlokswarm/app>

# RPC微服务选择

Grpc 比较高效，支持多语言，没有服务发现负载等

过一遍 <https://github.com/avelino/awesome-go> Distributed Systems的选出三个项目：

Go-micro, Go-kite，Rpcx

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Go-micro | Go-kite | Rpcx | Hprose-go | Grpc-go |
| 服务发现 | 有 | 有 | 有 | 无 | No(预留接口) |
| 负载 | 有 | 有 | 有 | 无 | No（预留接口） |
| Git更新 | 较多 | 更新少，最多时在2017年 | 较多 | 较多 | 较多 |
| 代码整体感觉 | 1.7 没有加它的插件，如grpc | 没看 | 1.2万 | 1.7万 | 10.7万 |
| websocket | 有 |  | 无 | 有 | 无,有代理实现 |
| Web-js客户端 | 与grpc-go一样 |  | 无 | 有 | 无，有代理实现 |
| 多语言支持 | 有 |  | 不好，只java客户端，其它语言的需要自己实现 | 多语言,不同语言支持协议版本没有完全同步 | 多语言 |
|  | 微服务框架，集成grpc |  | Rpc框架 | Rpc框架 |  |

结论：如果想自己实现多语言的客户端的话rpcx最合适，它的性能好且支持多种协议，方便与现有方案整合在一起，如果要单一一点使用grpc,如果对多语言特别要求的话hprose与grpc比较合适,

最后选择grpc,基于多语言支持良好，社区参考多， 需要两个实现负载（高可用，这个方案可以参考go-micro或者直接使用），websocket方案（go websocket代理）

疑问：1,大部分的rpc都不支持websocket, 是很难吗？ 还是有什么技术问题

2,网上说 js在解析 protocol buffer时，比json团难，这里还是选择使用websocket，因为它能提供非常好的双向通信

# 注入容器

Dot(scry自己写的)，<https://github.com/avelino/awesome-go>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Dot | wire | fx | dig | alice | <https://github.com/facebookarchive/inject> |
| 配置 |  | No | no | no | no | no |
| 依赖关系管理 |  | No | no | no | no | no |
|  |  |  |  |  |  |  |

简单查看代码及项目后，go的inject容器非常少且都处于起步阶段， 这里使用dot功能要完善一些