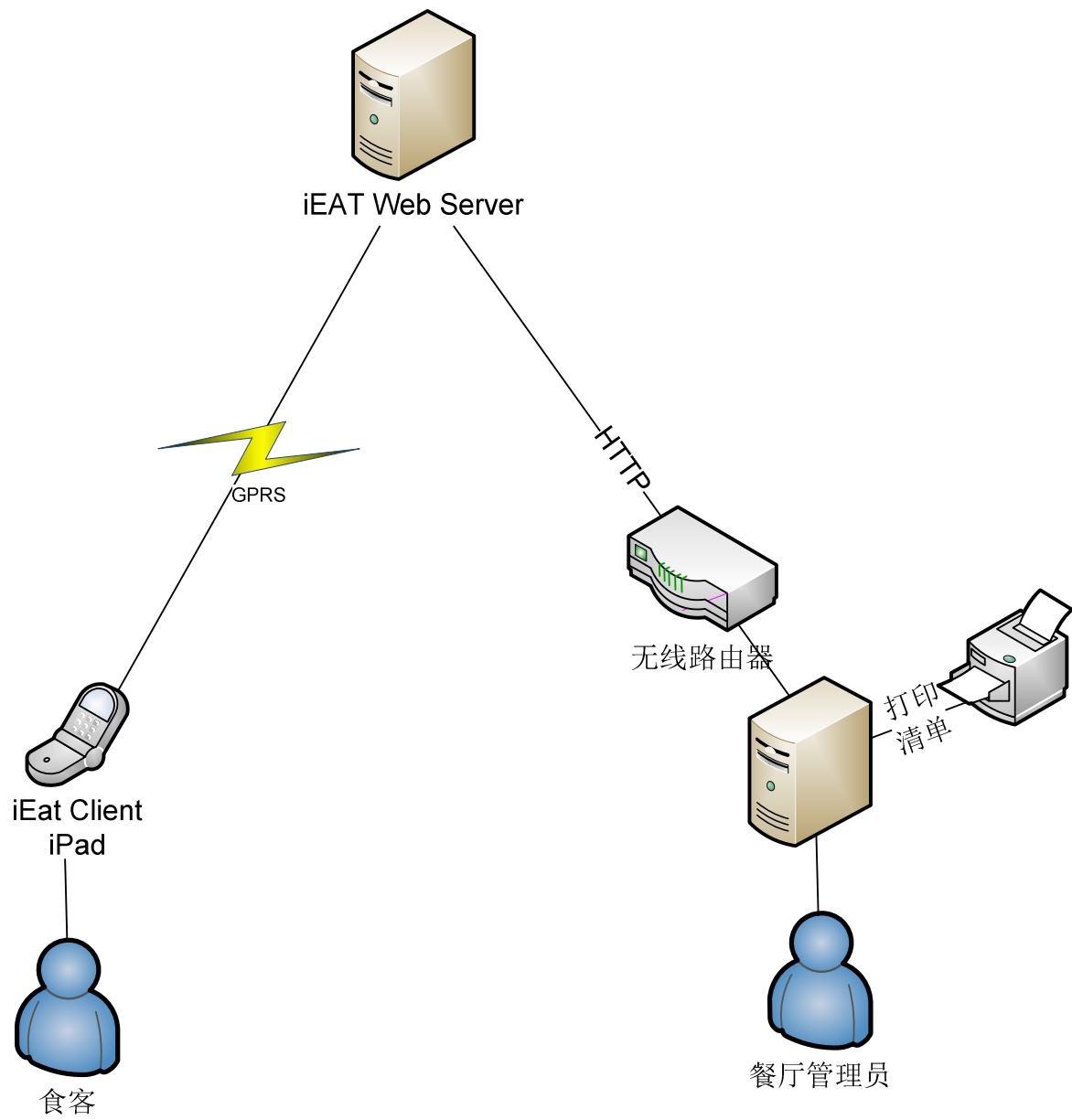


iEat 控制台概要设计

GUI



网络结构



通信协议

/* =====

* 与 Server 通信的 JSON-PDU 格式

* ===== */

1. 数据传输格式

Console 和 Server 信息全部使用 HTTPPOST 传递，数据包使用 JSON 格式序列化，通过 HTTP Request Payload 提交请求，通过 HTTP Response 获取返回信息。

Sample PDU:

Request URL:http://localhost:8888/capi

Request Method:POST

Request Headers:

Connection:keep-alive

User-Agent:Http-client

Request Payload:

["connect","user1","password"]

Resonse PDU:

["000001" , "10001" , "00001" , [1,1],[12,1]]

备注：其中 Connect Request 需要维护一个长期的 HTTP 连接(Connection:keep-alive)，服务器通过该通道传递点菜或者其他消息到 console，该连接在空闲时要定期（没 5 分钟）发送心跳包，确保连接正常，一旦发现连接不正常，立刻发出警告。

数据请求目前包括以下调用：

2.Connect: 建立连接调用

Request Payload:

```
["connect","USER_NAME#user1","TOKEN#password"]
```

字段描述：

1.Action : connect

2.USER_NAME : 用户名

3.TOKEN: 加密后的密码信息

Resonse PDU:

```
["LENGTH(135)","MD5_HASH","000001" ,"10001" ,"00001" ,[1,1],[12,1]]
```

字段描述：

1.整个消息长度，包括这个字段本身，转化成 ascii 表示。

2.整个消息 MD5 Hash 值，用于确保消息发送正确，如果接收到 MD5 不正确的消息，需要丢弃，并发送重发请求，内容使用 BASE64 编码。

3.消息编号

4.服务员编号

5.桌编号

6.点菜信息

6.1 所点的菜编号

6.2 点菜数量

注：每次收到一个完整的 Response PDU，就需要驱动打印机打印出一份点菜记录。

3. Resebd: 重发请求

Request Payload:

["resend", "USER_NAME#user1", "TOKEN#password"]

字段描述:

1. Action : resend

2. USER_NAME : 用户名

3. TOKEN: 加密后的密码信息

Resonse PDU:

["OK|FAIL", "MSG"]

字段描述:

1. 成功或者失败

2. 附加信息

4. SetState: 缺菜调用 (PENDDING)