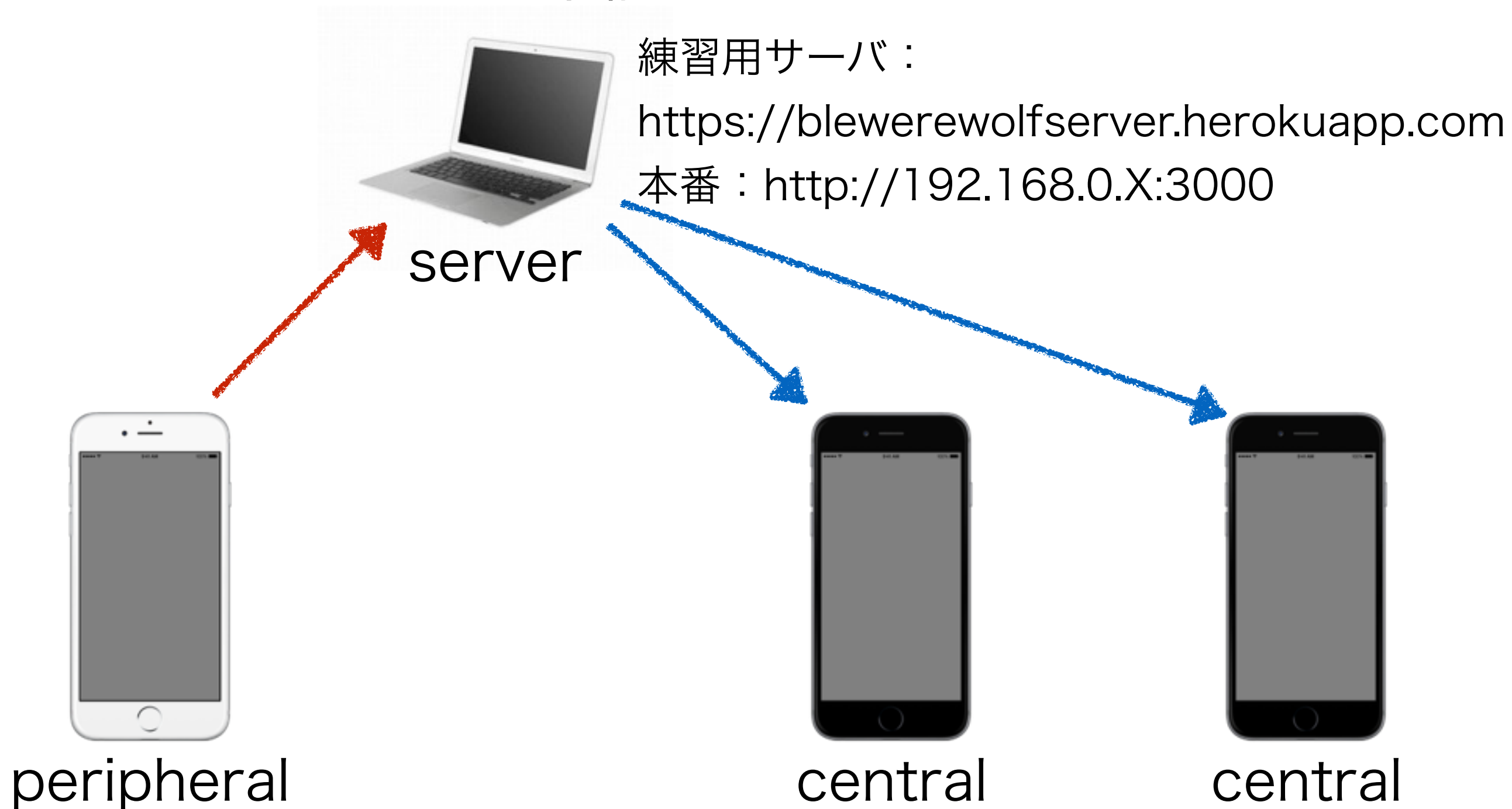


BLEWerewolf

通信の仕組み (概要)

- チャットルームを中継してコマンドをやりとりする



Socket.io (android)

- ・ ライブラリのメソッドについて



```
JSONObject json = new JSONObject();  
json.put("message", "送りたい内容");  
socket.emit("message:send", json);
```



JSONっていうファイル形式にmessageを
入れてemitメソッドを呼ぶ

Socket.io (android)



```
@Override
public void on(String event, IOAcknowledge ack, Object... args)
{
    final JSONObject message = (JSONObject)args[0];
    new Thread(new Runnable() {
        public void run() {
            handler.post(new Runnable() {
                public void run() {
                    try {
                        if(message.getString("message") != null {
                            //message.getString("message")
                        }
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                }
            });
        }
    }).start();
}
```

ここに受信したメッセージ
が流れてくる



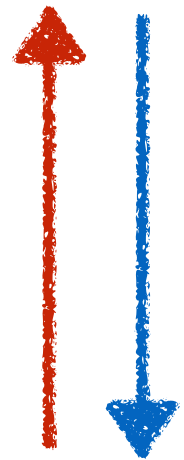
ゲームの流れ



- ・ peripheral
- ・ ゲーム部屋生成
- ・ ルール設定
- ・ ルール確認
- ・ 役職確認
- ・ 夜時間開始

- ・ central
- ・ プレイヤー登録
- ・ 待機
- ・ ルール確認
- ・ 役職確認
- ・ 夜時間開始

プレイヤー登録(1)



- ・ ペリフェラルがゲーム部屋を作ったら、`"advertiseMyDevice:012345:perilD:はるき"`というコマンドが送られてくる

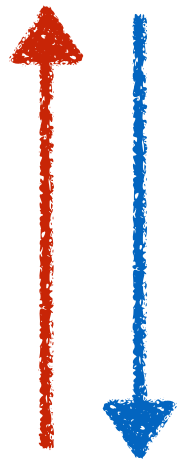
例	パラメータ
012345	ゲームID
perilD	ペリフェラルID
はるき	ペリフェラル名前

- ・ この情報をもとにリストに表示する
- ・ ユーザがリストのセルを選択したら、接続する

- ・ 接続するためには`"mes:100:perilD:centID:participateRequest:012345/centID/はせべ/perilD/0"`というコマンドを送る

例	パラメータ
100	シグナルID
perilD	ペリフェラルID
centID	ペリフェラル名
012345	ゲームID
はせべ	セントラル名前
0	おまじないの0

プレイヤー登録(2)



- ・ 接続できたらペリフェラルから”mes:200:centID:perilD:participateAllow:”というコマンドが送られてくるので、待機画面を表示する
- ・ ちなみにペリフェラルがなんらかの原因でゲーム部屋を閉じた時は”closeMyDevice:012345:perilD>”というコマンドがくられてくるので、これを受け取ったらリストから削除する

例	パラメータ
200	シグナルID
centID	セントラルID
perilD	ペリフェラルID

例	パラメータ
012345	ゲームID
perilD	ペリフェラルID

コマンドの約束事

- ・ 3種類のコマンドについて(A)(B)(C)
- ・ (A)"advertiseMyDevice:<gameId>:<peripheralId>:<peripheralName>"
- ・ (B)"closeMyDevice:<gameId>:<peripheralId>"
 - ・ (A)(B)の二つはペリフェラルのみが送信する
 - ・ 宛先は不定として誰にでも送るし、受信される
 - ・ (A)は2秒間隔で送信し続ける（登録締め切りまで）
 - ・ (B)は部屋を閉じるか、登録を締め切ったら2秒間隔で3回送信する
- ・ (C)"mes:<signalId>:<yourId>:<myId>:<message>"
 - ・ (C)は上以外のすべての通信で使う
 - ・ yourIdが送り先のidentificationID, myIdが送りもとのidentificationID, messageが送る内容
 - ・ signalIdは、2重受信をさけるために各コマンドに付与するIDのことで、各端末は(C)を送るたびにsignalIdを+1していく。（アプリ起動時に0から999999のランダムな値で初期化する）
 - ・ 宛先が自分になっているもののみ受信する
 - ・ ただし宛先がcentralsになっているものはすべてのセントラルで受信する
 - ・ コロンで区切る時に、<message>中にもコロンが含まれることがあるので注意すること（<message>ないのコロンはちゃんとコロンとして処理する必要がある）