Node.js Based IM

by Mofei Zhu
http://zhuwenlong.com

About ME

2010 - 2014

易班 yiban.cn　　Web前端工程师

2014 - Today

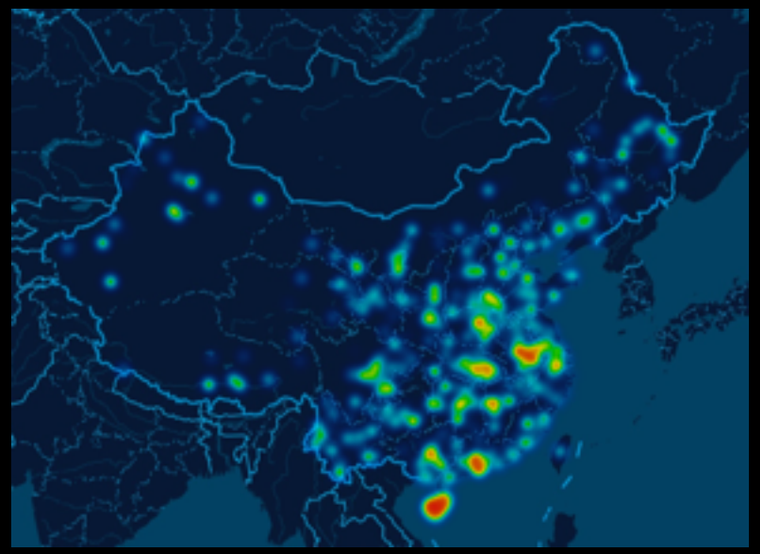
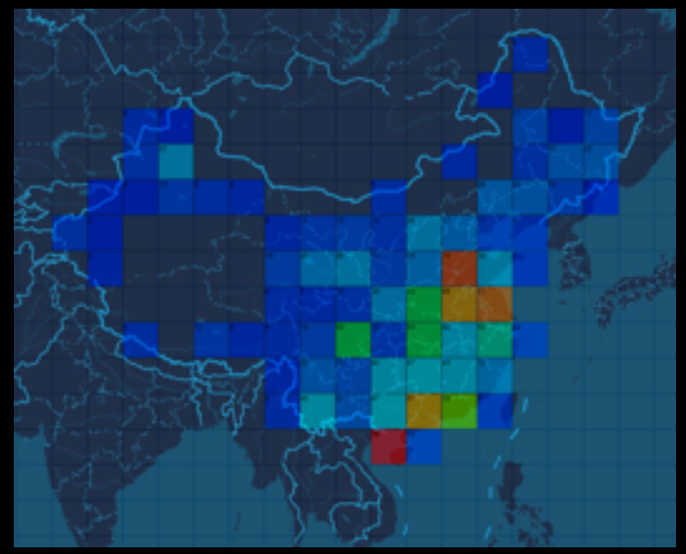
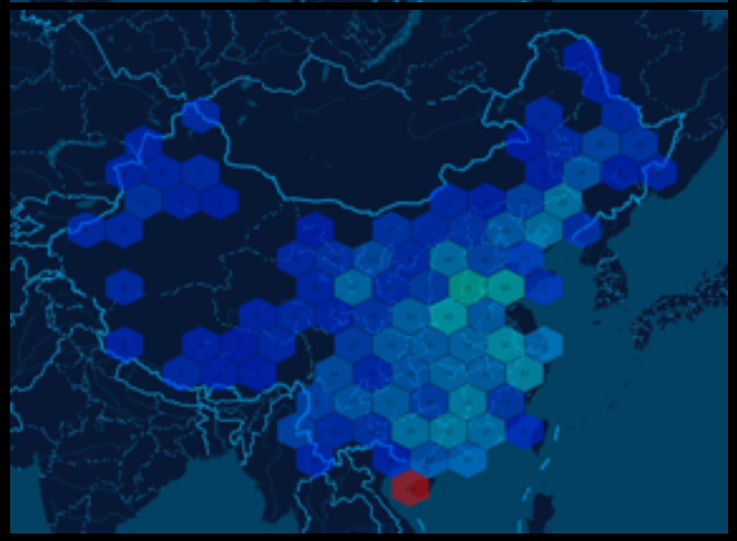
Baidu 百度　　高级研发工程师

Mofei Zhu

朱文龙

http://zhuwenlong.com

https://github.com/zmofei

90
77
75
89
88

Mapv

At First , I’ll show you something.

Chart Room Demo

http://121.40.125.180:8008/

https://github.com/zmofei/socketChatroom

CURRENT: all

All (2)

Mofei [Yourself]

璐

hello world **Mofei**

璐 who are you?

I am Mofei **Mofei**

璐 Nice to meet you

璐 ; 👻 👻 👻

:) **Mofei**

Guest: Hello Word :)

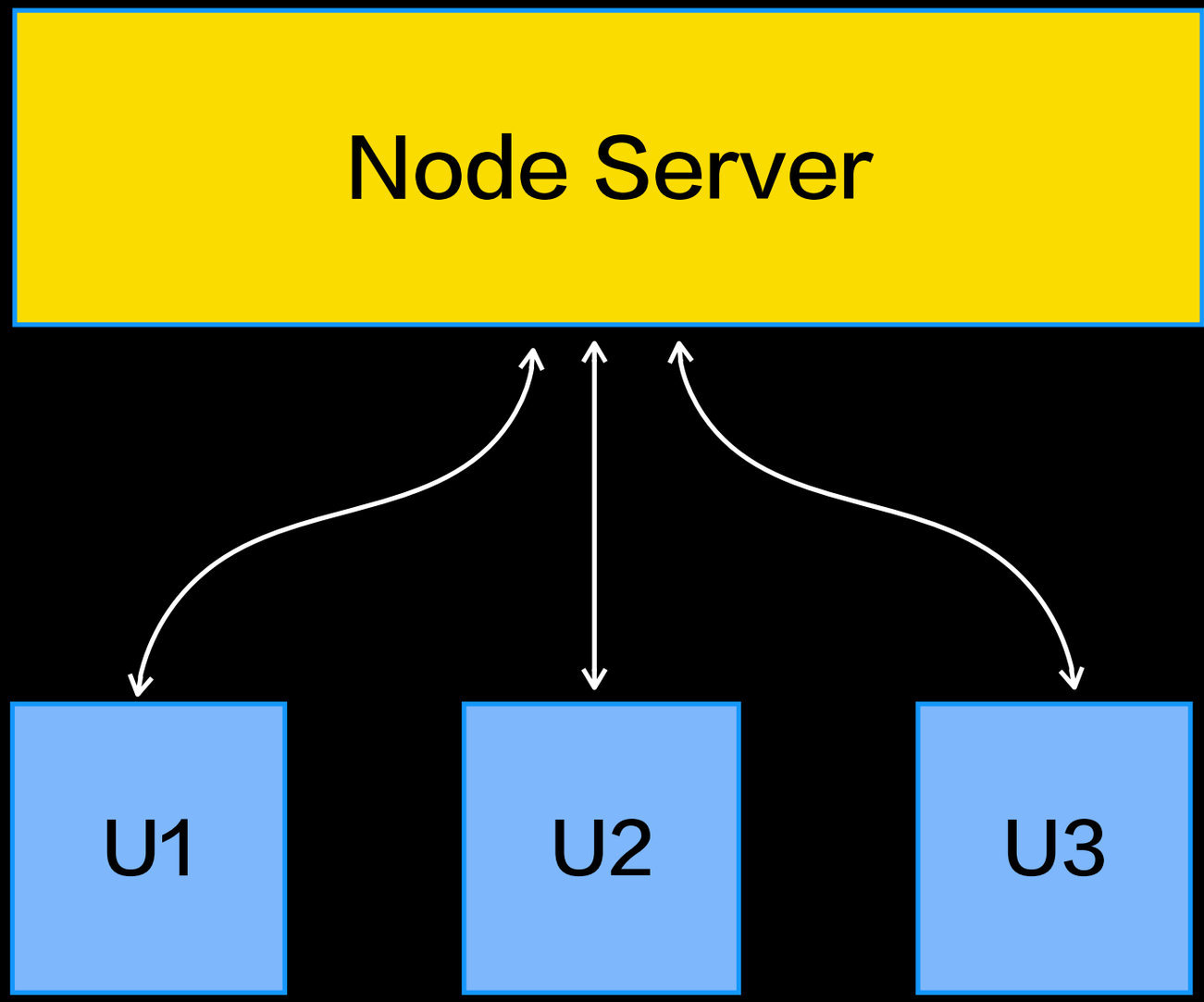
socket.io

It’s **EASY** to use.

test — vim — 80×22

vim

```
 1 var io = require('socket.io')(8280);
 2 var user = {};
 3
 4 io.sockets.on('connection', function(socket) {
 5     socket.store = socket.store || {};
 6     socket.on('reg', function(data) {
 7 +--- 23 lines: if (user[data.username]) {----------------------------------
30     });
31
32     socket.on('publish', function(data) {
33 +--- 24 lines: if (data.touser == 'all') {---------------------------------
57     });
58
59     socket.on('getUsersList', function() {
60 +---  5 lines: var userList = [];--------------------------------------------
65     });
66
67     socket.on('disconnect', function(data) {
68 +---  9 lines: if (user[socket.store.username]) {----------------------------
77     });
78 });
"socket.js" 78L, 2338C                                        5,1           All
```


Node Server
U1
U2
U3

BOSS: Are you **kidding** me?

Database?

Cache?

One Machine ?

Distributed system?

User system?

Security check?

Think & Discuss … …

I Thought...

1. IM System ?
2. What is Distributed System?
3. What if one machine break down?
4. How do machines communicate? Broadcast ?
5. How to deal with big Group message?

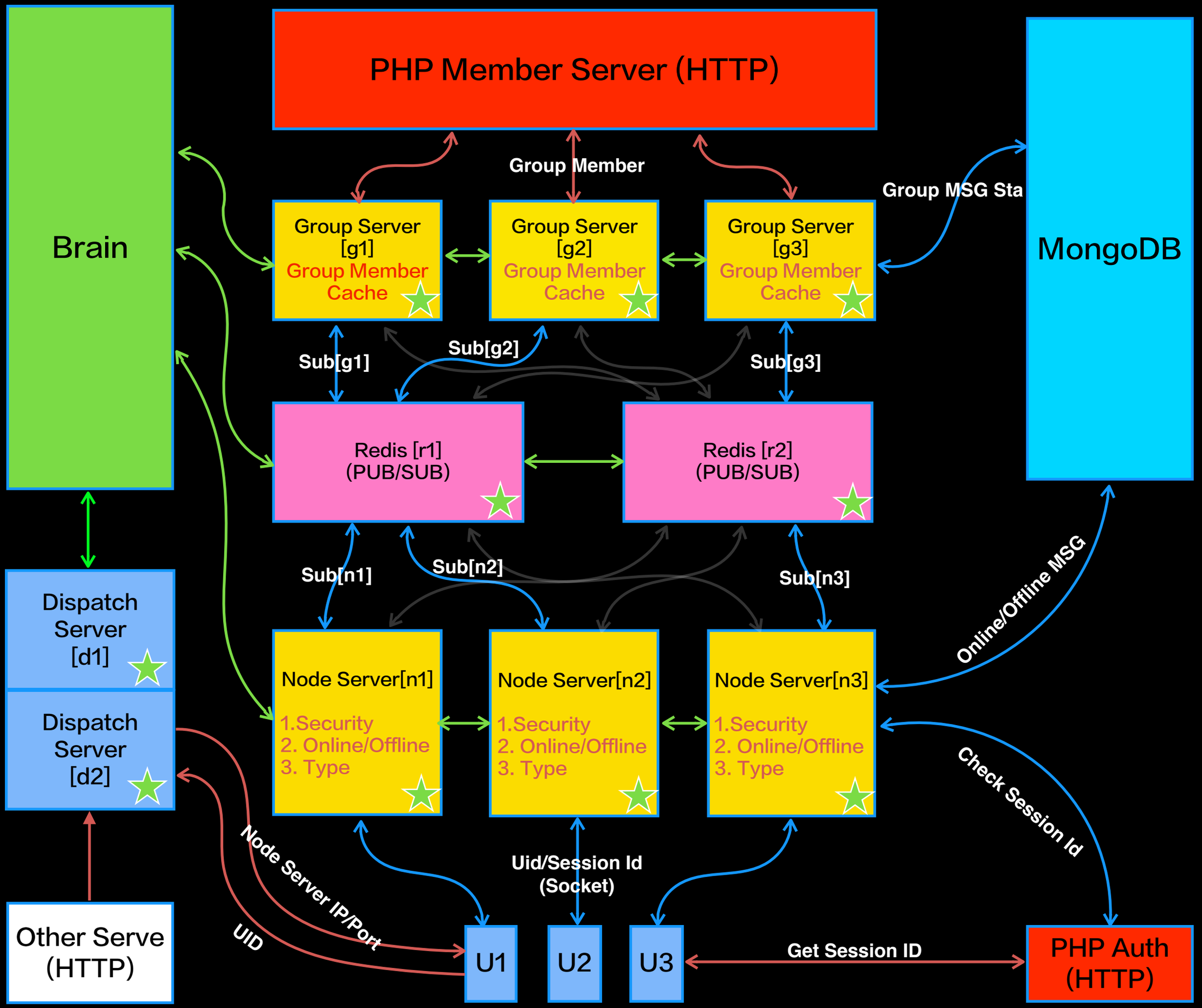
Device Engineer Thought…

1. Which machine should I connect ?
2. How to receive & send message ?
3. How about system message ?
4. 3G/4G Network interruption ?
5. Offline Message ?

At last

At last

- Node.js
- Socket.io
- Mongodb
- Redis
- Security Check
- * Back-end’s back-end(PHP, Mysql,etc.)


PHP Member Server (HTTP)
Brain
MongoDB
Group Member
Group MSG Sta
Group Server [g1] Group Member Cache
Group Server [g2] Group Member Cache
Group Server [g3] Group Member Cache
Sub[g1]
Sub[g2]
Sub[g3]
Redis [r1] (PUB/SUB)
Redis [r2] (PUB/SUB)
Sub[n1]
Sub[n2]
Sub[n3]
Online/Offline MSG
Dispatch Server [d1]
Dispatch Server [d2]
Node Server[n1] 1.Security 2. Online/Offline 3. Type
Node Server[n2] 1.Security 2. Online/Offline 3. Type
Node Server[n3] 1.Security 2. Online/Offline 3. Type
Check Session Id
Node Server IP/Port
UID
Uid/Session Id (Socket)
Other Serve (HTTP)
U1
U2
U3
Get Session ID
PHP Auth (HTTP)

Her Parents

此处内容现场演讲版本可见

Her's performance

此处内容现场演讲版本可见

https://github.com/M6Web/websocket-bench

Result

此处内容现场演讲版本可见

Frequency : CPU & IO

Technical realisation

Brain

PHP Member Server (HTTP)

MongoDB

Group Server
[g1]
Group Member
Cache

Group Server
[g2]
Group Member
Cache

Group Server
[g3]
Group Member
Cache

Redis [r1]
(PUB/SUB)

Redis [r2]
(PUB/SUB)

Dispatch
Server
[d1]

Dispatch
Server
[d2]

Node Server[n1]

1.Security
2. Online/Offline
3. Type

Node Server[n2]

1.Security
2. Online/Offline
3. Type

Node Server[n3]

1.Security
2. Online/Offline
3. Type

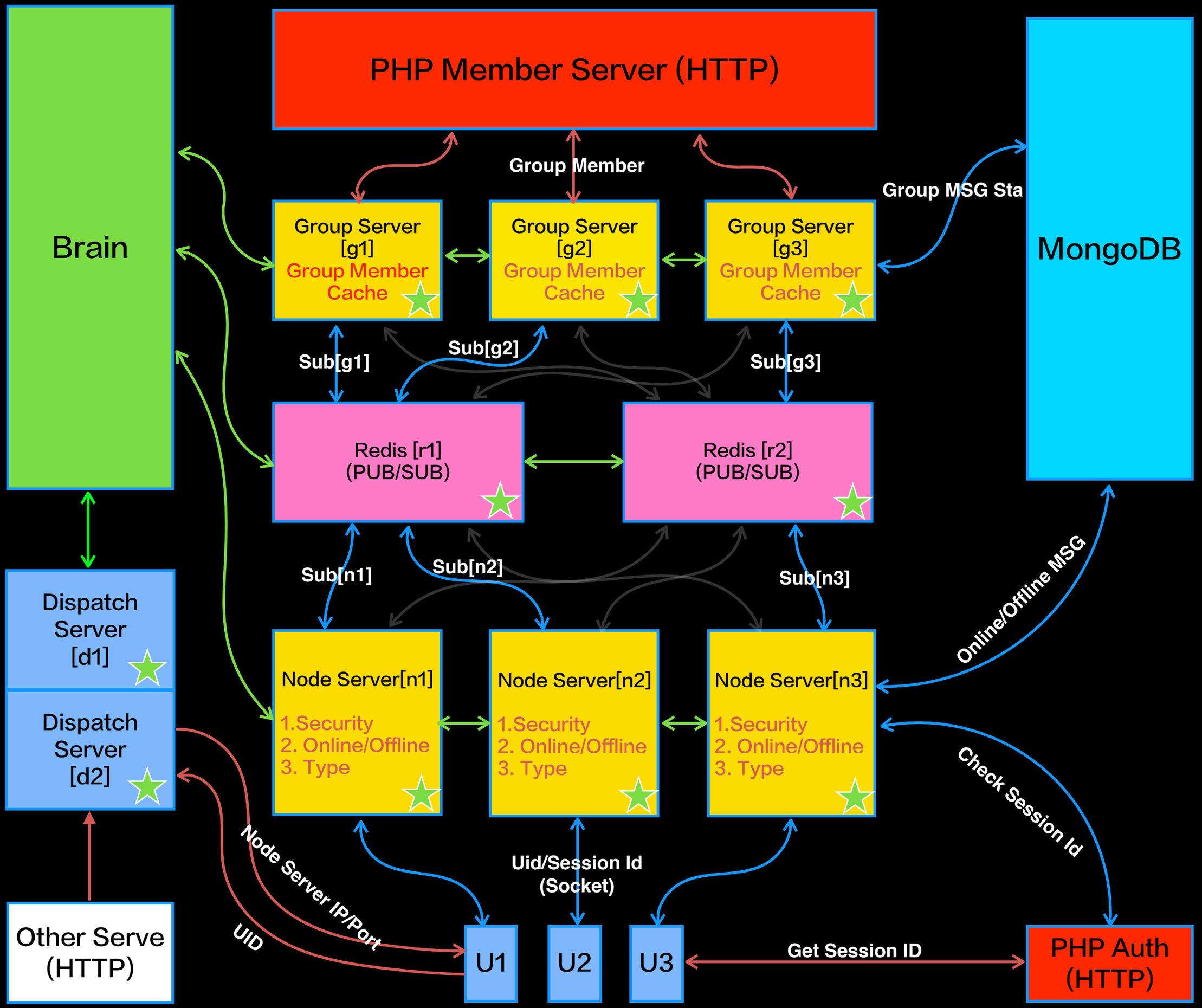
U1

U2

U3

Get Session ID

PHP Auth
(HTTP)


PHP Member Server (HTTP)
Brain
MongoDB
Group Member
Group MSG Sta
Group Server [g1] Group Member Cache
Group Server [g2] Group Member Cache
Group Server [g3] Group Member Cache
Sub[g1]
Sub[g2]
Sub[g3]
Redis [r1] (PUB/SUB)
Redis [r2] (PUB/SUB)
Sub[n1]
Sub[n2]
Sub[n3]
Online/Offline MSG
Dispatch Server [d1]
Dispatch Server [d2]
Node Server[n1] 1.Security 2. Online/Offline 3. Type
Node Server[n2] 1.Security 2. Online/Offline 3. Type
Node Server[n3] 1.Security 2. Online/Offline 3. Type
Check Session Id
Node Server IP/Port
UID
Uid/Session Id (Socket)
Other Serve (HTTP)
U1
U2
U3
Get Session ID
PHP Auth (HTTP)

PHP Member Server (HTTP)

Brain

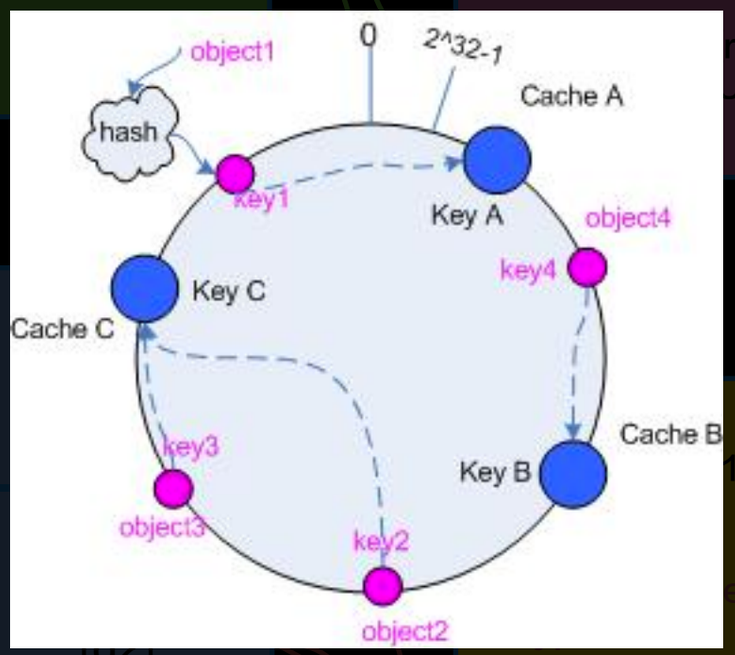
MongoDB

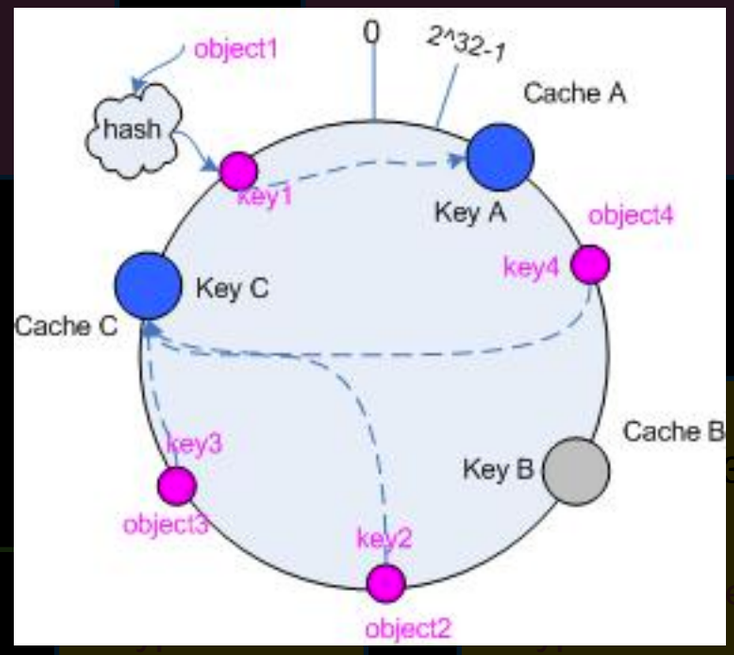
Group Server [g1]
Group Member Cache

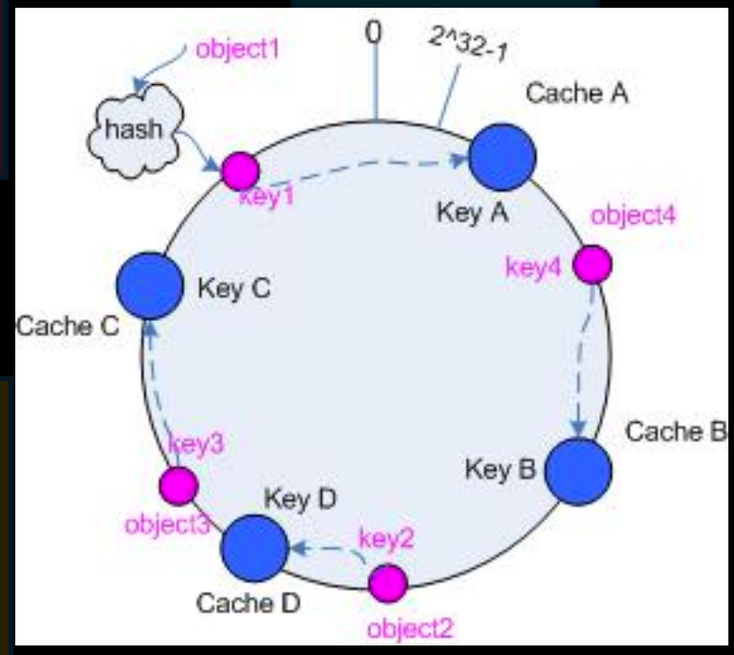
Group Server [g2]

Group Server [g3]

Consistent hashing







Node Server IP/Port

UID

U1

U2

U3

Get Session ID

PHP Auth (HTTP)

Server side

First you should start the pub-sub server

```
mPubsub.createServer('127.0.0.1',4455);   //create a server at ip:127.0.0.1 and port:44
```

Client side

Once you start the server , you can connect to the serve and share the pub-sub

```
//connect to the server (IP:127.0.0.1,port:4455);
mPubsub.connect('127.0.0.1', 4455, function(socket) {

    //SUB
    //sub a channel
    //scoket.sub(channel,callback)
    socket.sub('abc1', function(data) {
        //when somebody publish something in channel 'abc',the following will work
        console.log('sub data', data);
    });

    //UNSUB
    //unsub a channel
    //socket.unsub('channel');
    socket.unsub('abc2');

    //PUBLISH
    //pub something to channel abc
    //scoket.pub(channel,data)
    socket.pub('abc', '123asd');

    //END
    socket.end();

});
```

mPubsub https://github.com/zmofei/mPubsub

That’s All, Thinks