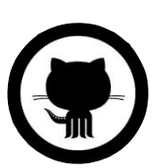
Node.js 與多方服務串接實務

Server Connect To API / 3rd Service Pattern

Caesar Chi @clonncd



@clonncd


JSDC.tw


Hiiir
行動×商務
社群×媒體
Qbon


MiTAC
MiCloud


Node.js
Taiwan


D-Link
Building Networks for People


miiCasa

Pantone 2345U

Node.js Taiwan 社群協作中文電子書

授權

作者

本書由 Node.js Taiwan 社群成員協作，以下名單依照字母排序。

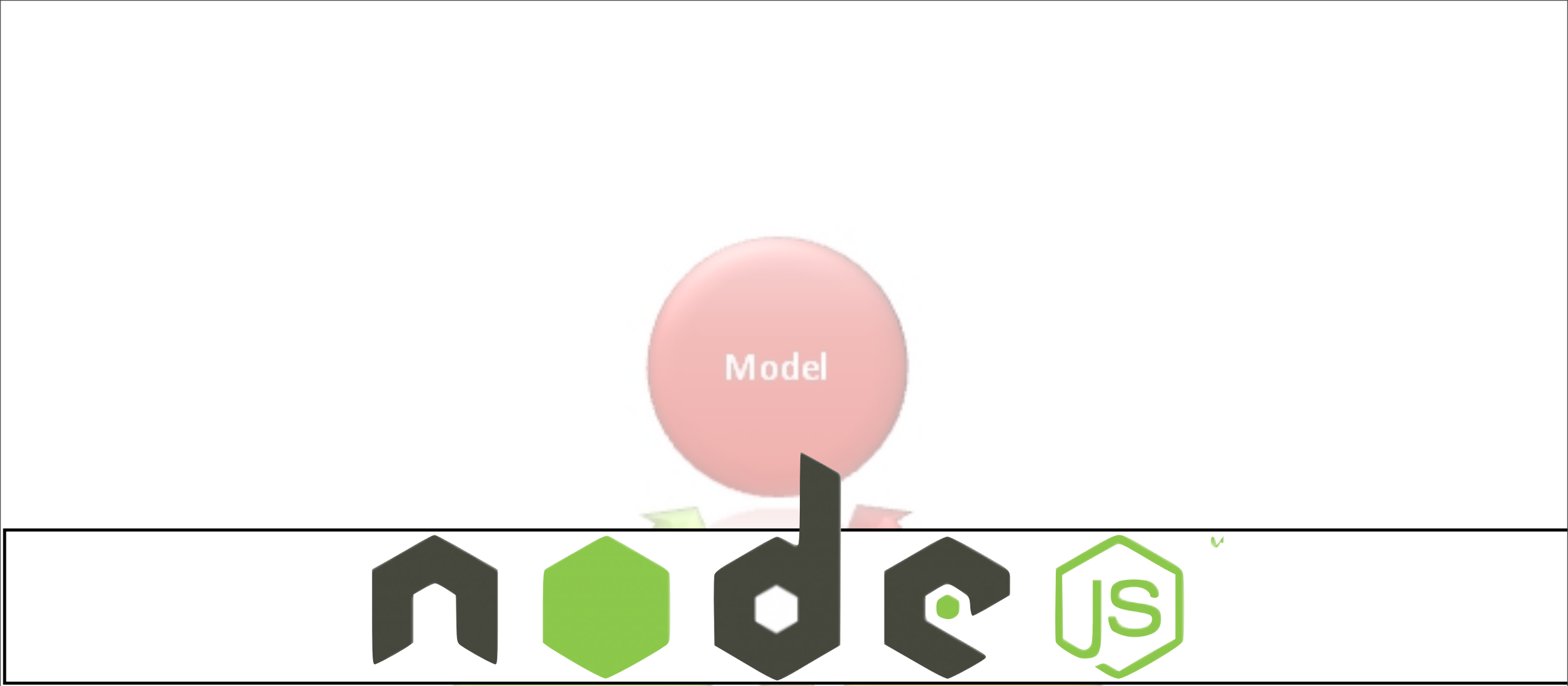
- Caesar Chi (clonn)
- Fillano Feng (fillano)
- Kevin Shu (Kevin)
- lyhcode http://about.me/lyhcode

Node.js Taiwan 是一個自由開放的技術學習社群，我們歡迎您加入一起學習、研究及分享。

https://github.com/nodejs-tw/nodejs-wiki-book

Pantone 2345U

Pantone Bright Red U


Model
View
Controller
node
JS

Structure

Clients

Server

That means we thought done with something in Node.js

But actually we did nothing special.

```
doCallback(function () {
  doCallback(function () {
    doCallback(function () {
      doCallback(function () {
        doCallback(function () {
          doCallback(function () {
            doCallback(function () {
              doCallback(function () {
                doCallback(function () {
                  doCallback(function () {
                    doCallback(function () {
                      doCallback(function () {
                        doCallback(function () {
                          doCallback(function () {
                            doCallback(function () {
                              return "I am done";
                            });
                          });
                        });
                      });
                    });
                  });
                });
              });
            });
          });
        });
      });
    });
  });
});
```

Request of

As a Front end Server

Structure

front Server

send Request

Services

secret

information

cache

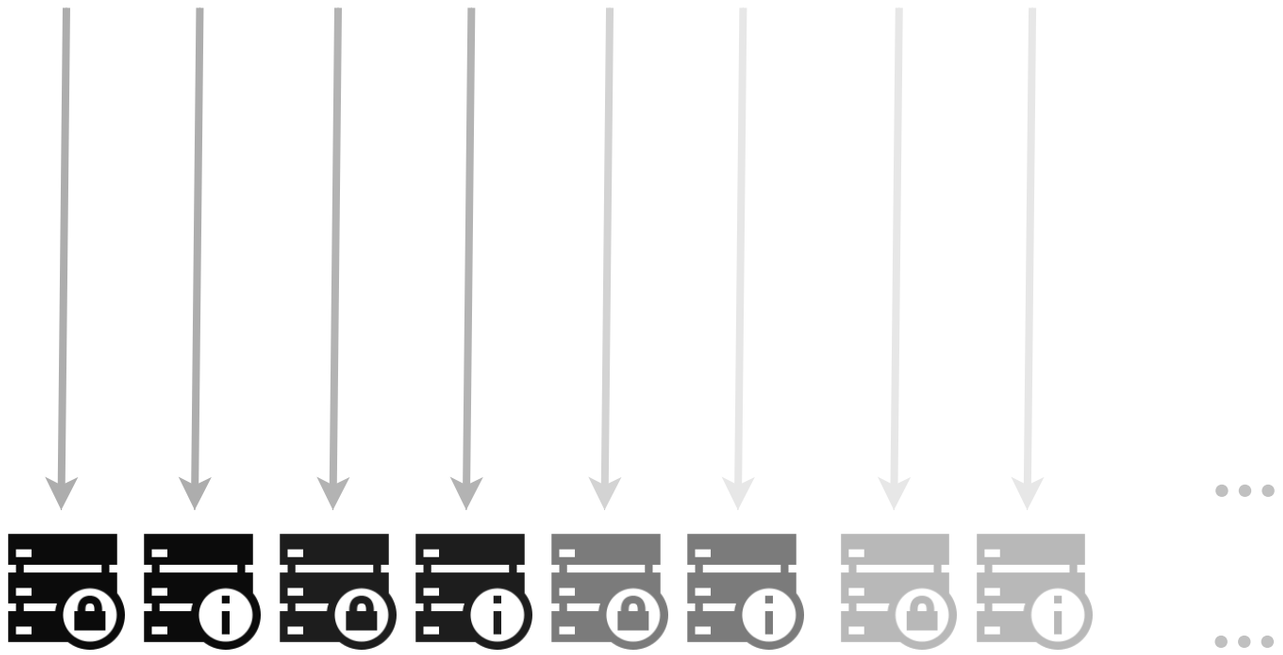
When Node.js has Request

WWW are APIs

Structure

front Server

send Request



Services

But When

Service requirement

Grows Up

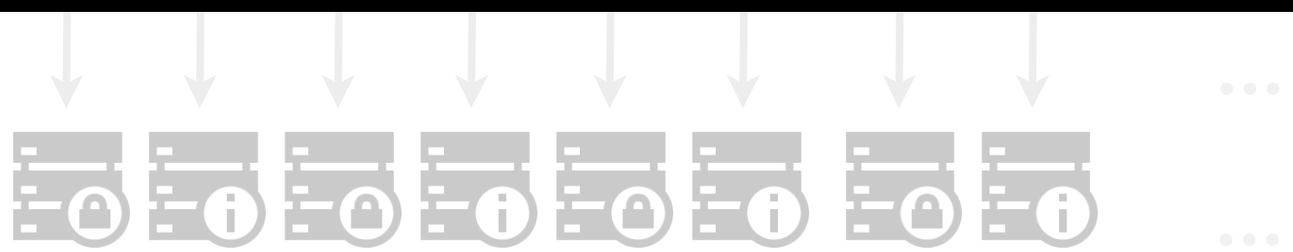
Structure

front Server

send Request

Too Much Code To Handle

Internal server



Service Definition

- Front server can be flexible to connect
- Support Restful API for internal connect
- Same Router rule for client
- Modules install / uninstall be easy
- Light way feedback data

Server Definenation

Tips

Service as a module, feature as a apps

- https://github.com/clonn/module-loader
- Modules install / uninstall be easy
- Light way feedback data

```
modules
└── app1
    ├── index.js
    └── routes
        └── member.js
```

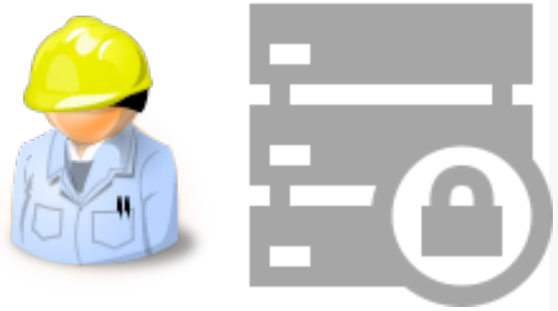
```
modules
└── app2
    ├── index.js
    └── routes
        └── member.js
```

```
modules
└── app3
    ├── index.js
    └── routes
        └── member.js
```

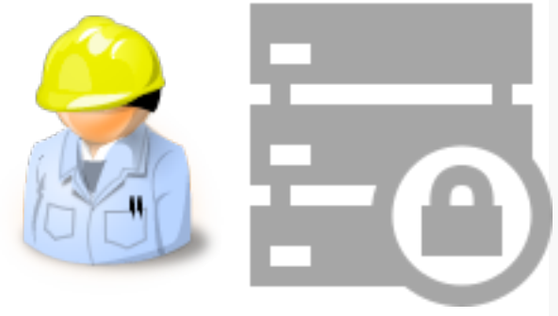
```
modules
└── app4
    ├── index.js
    └── routes
        └── member.js
```

Routers as apps, auto route

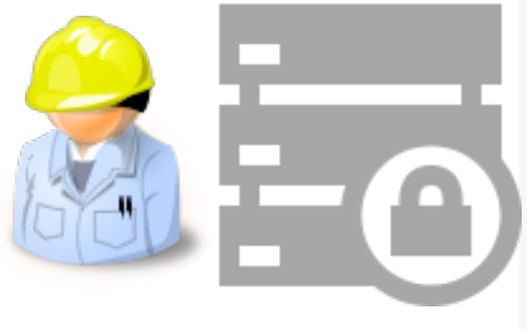
```
modules
└── app1
    ├── index.js
    └── routes
        └── member.js
```



```
modules
└── app2
    ├── index.js
    └── routes
        └── member.js
```

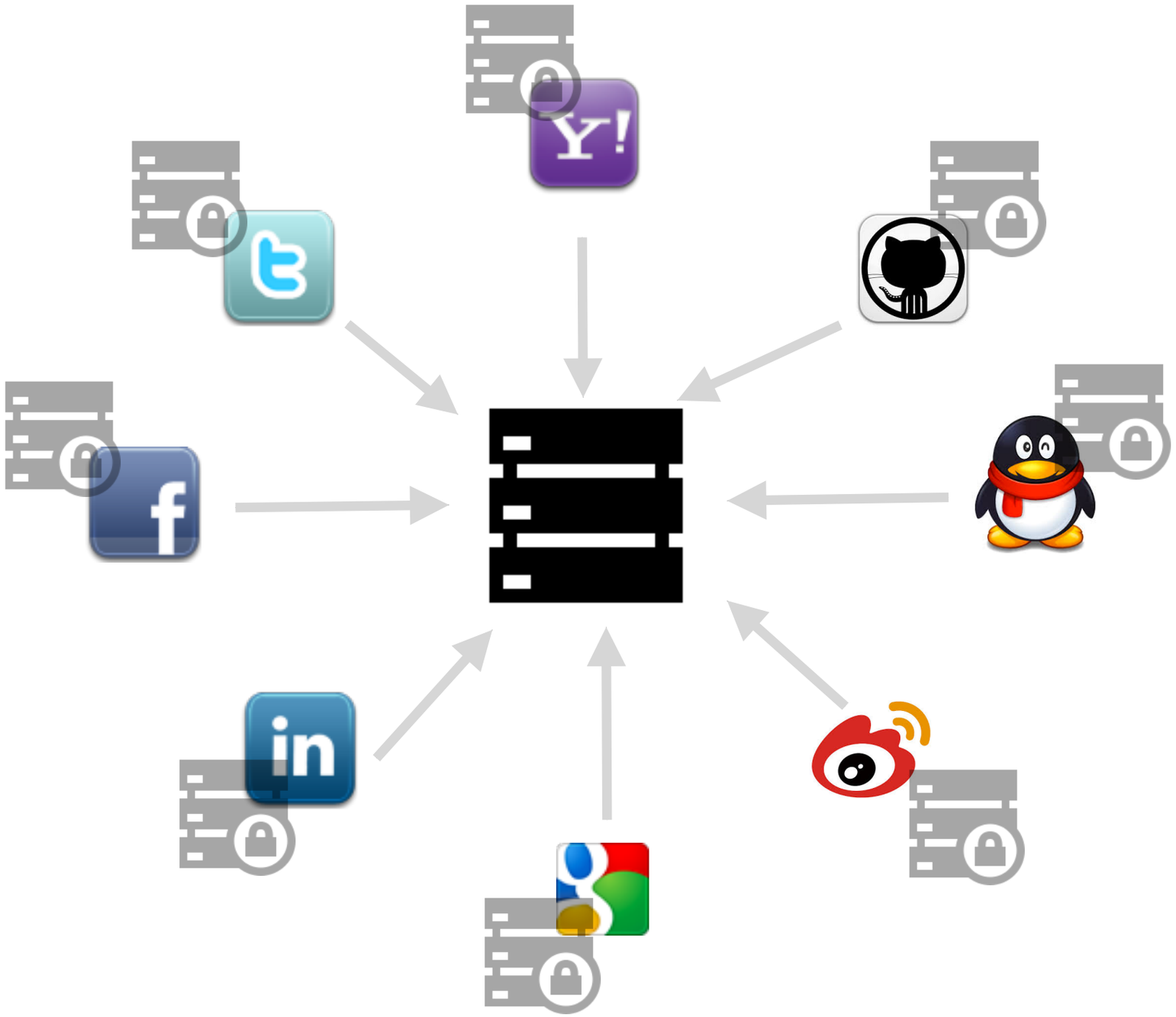


```
modules
└── app3
    ├── index.js
    └── routes
        └── member.js
```

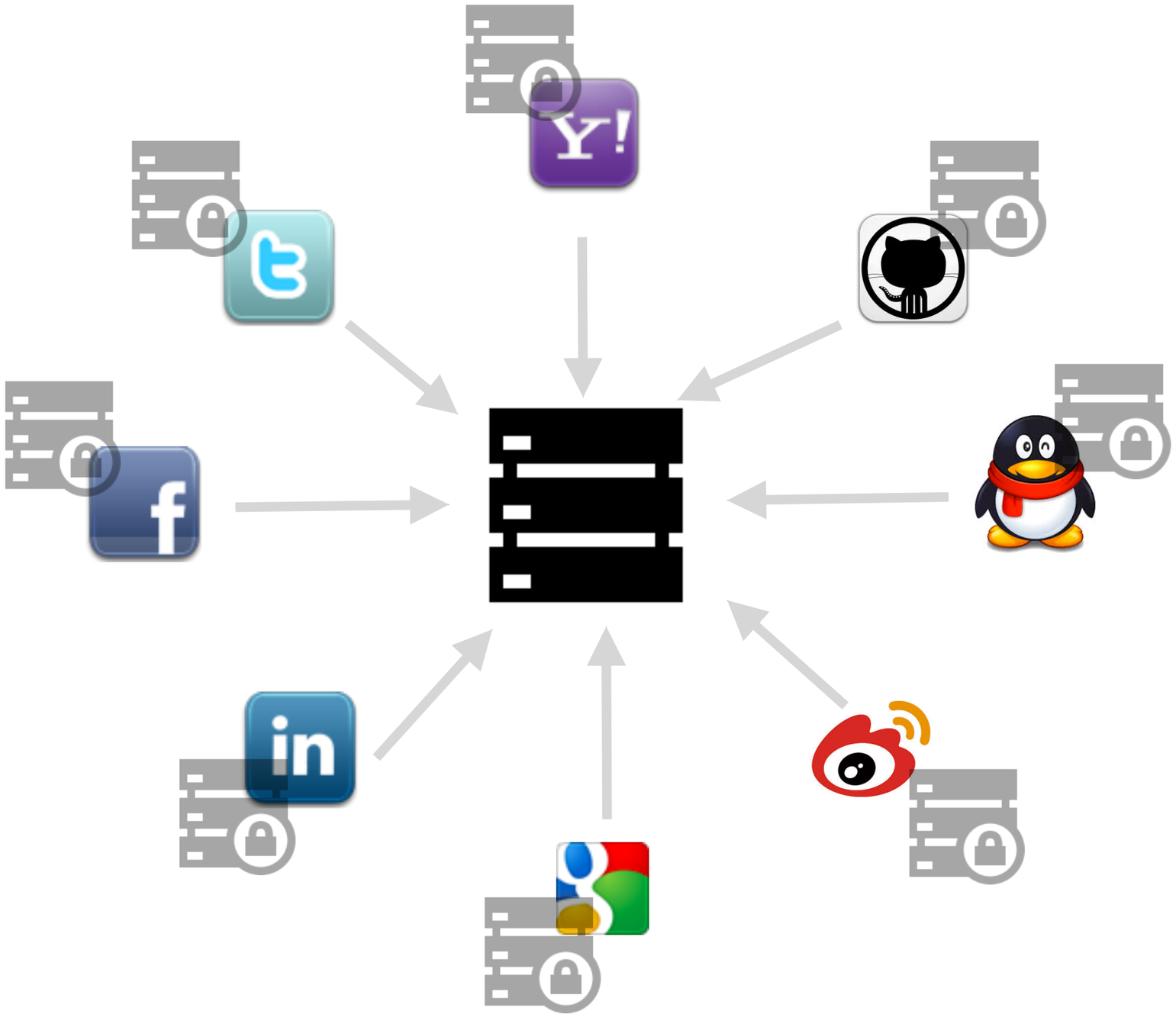


```
modules
└── app4
    ├── index.js
    └── routes
        └── member.js
```

Every developers can handle by themselves



Make connect easy,
A tiny distribution server



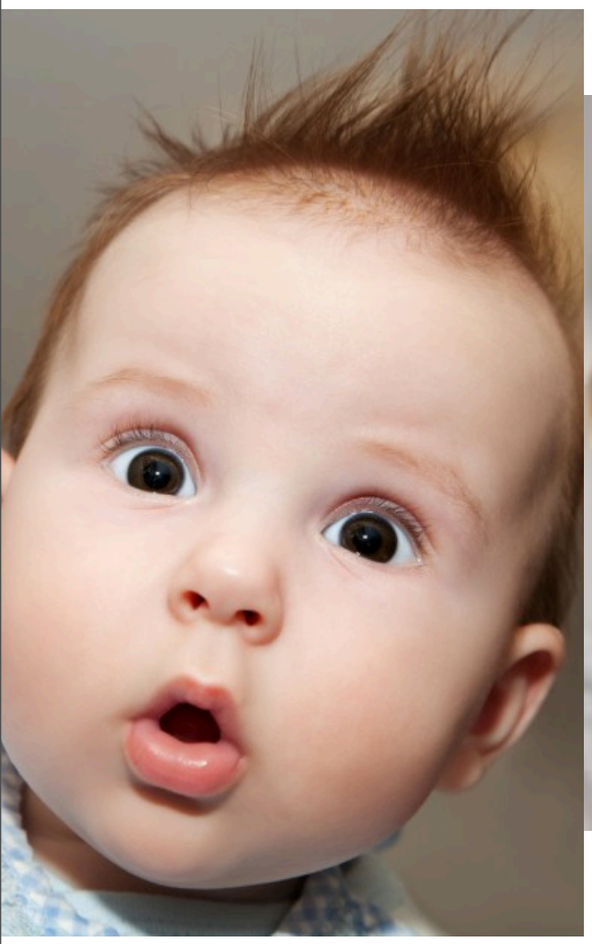
Face Response Issue ...

Wait ... Wait ... Wait ...

https://www.flickr.com/photos/tormodspictures/

伺服器反應很慢

http://chinese.engadget.com/2014/06/21/facebook-android-update-africa/




Let's refactoring code

```
exports.jenkins = function(req, res){
  var body = req.body || {};
  var redmine = config.redmine;
  var title = req.query.buildName || "";

  res.send({
    status: 200,
    message: "it is webhook"
  });

  if ( ! req.query.buildName) {…

  data = {…

  slack.webhook({…
    console.log(response);
  });
};
```

```
exports.jenkins = function(req, res){
  var body = req.body || {};
  var redmine = config.redmine;
  var title = req.query.buildName || "";

  res.send({
    status: 200,
    message: "it is webhook"
  });

  if ( ! req.query.buildName) {…

    data = {…

    slack.webhook({…
      console.log(response);
    });
};
```

Response back as soon as posible

```
exports.jenkins = function(req, res){
  var body = req.body || {};
  var redmine = config.redmine;
  var title = req.query.buildName || "";

  res.send({
    status: 200,
    message: "it is webhook"
  });

  if ( ! req.query.buildName) {…

  data = {…

  slack.webhook({…
    console.log(response);
  });
};
```

Heavy processing or request, put late

Response as short as posible

https://www.flickr.com/photos/calliope/

what can NOT be late

- real time date, have to be light (geo, data)
- robust data, we could not avoid (but save in cached, redis and process them later)

What can late

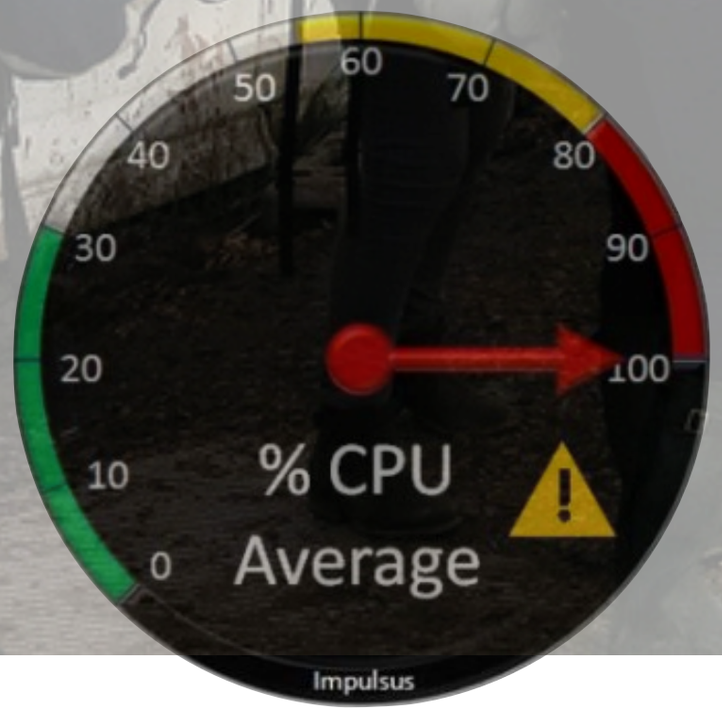
- log data writing
- data saving (not for real time response)
- data caching

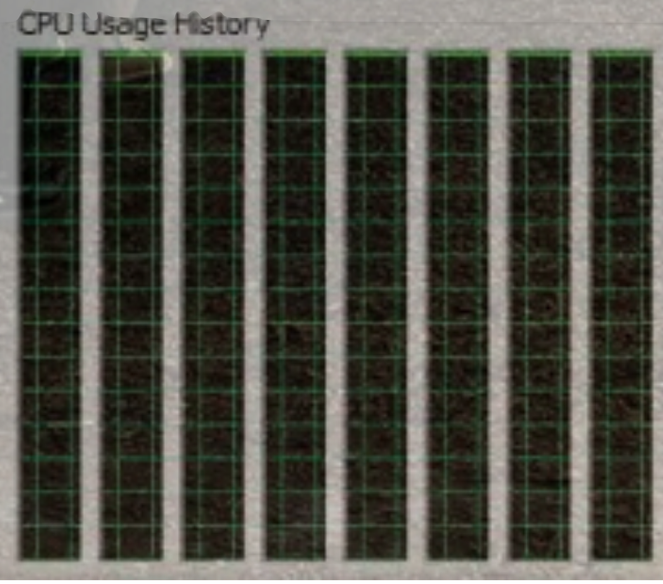
Tips

Response First, Process Later

Too many callback event in waiting queue

https://www.flickr.com/photos/peter_curb/14318113011/sizes/c/in/photostream/





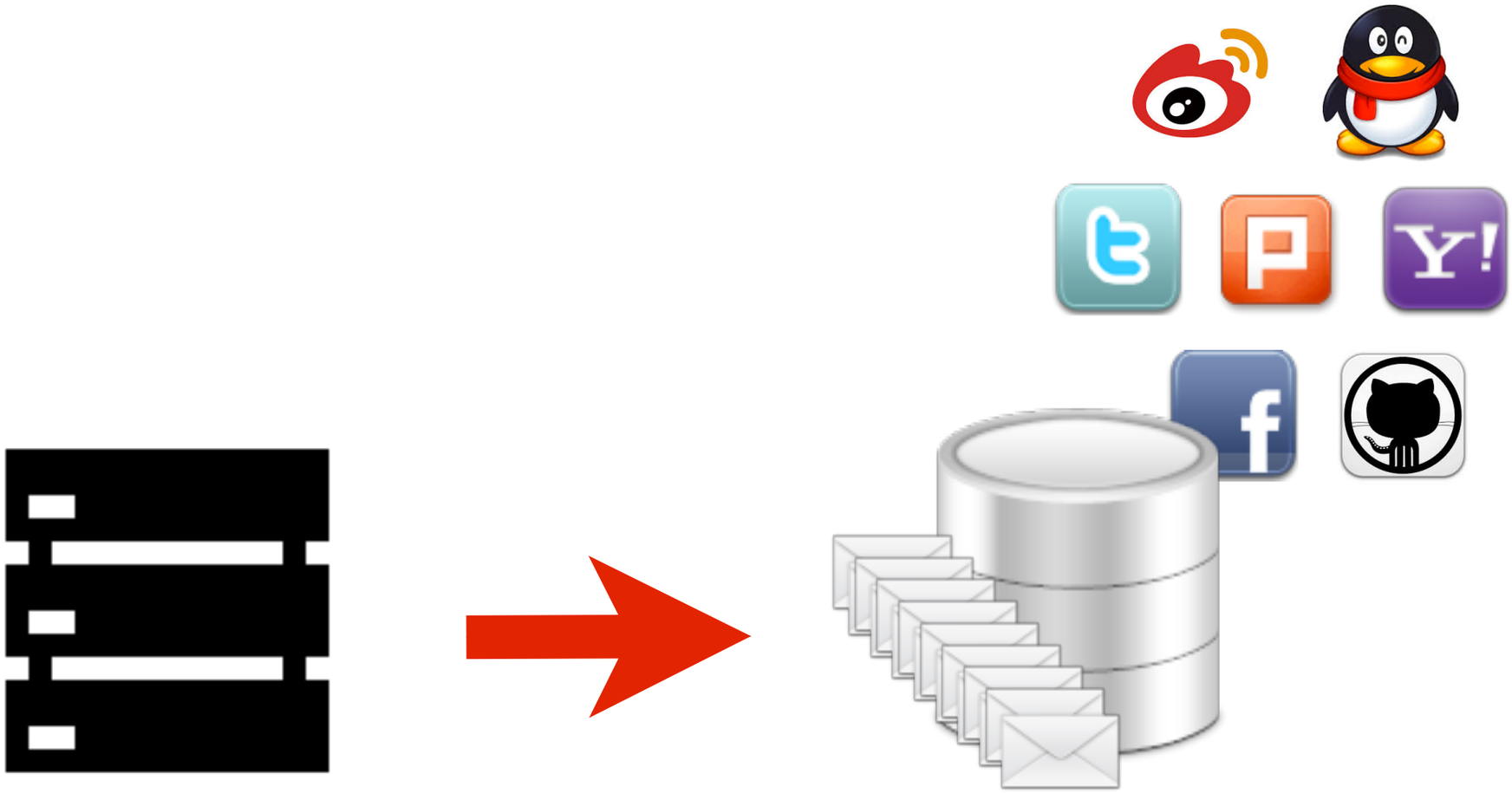
WARNNING, It Cause High CPU Usage

https://www.flickr.com/photos/peter_curb/14318113011/sizes/c/in/photostream/

We put process in

Task Queue.

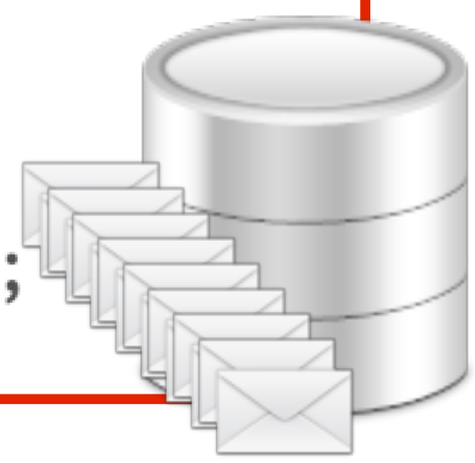
http://learnboost.github.io/kue/



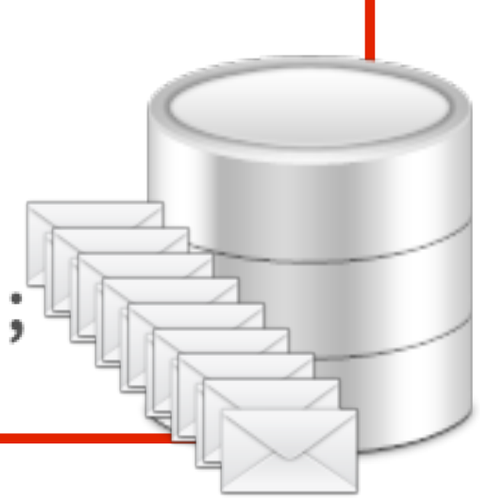
Save process, praramsin Task Queue

```
exports.jenkins = function(req, res){
  var body = req.body || {};
  var redmine = config.redmine;
  var title = req.query.buildName || "";

  res.send({
    status: 200,
    message: "it is webhook"
  });

};
```

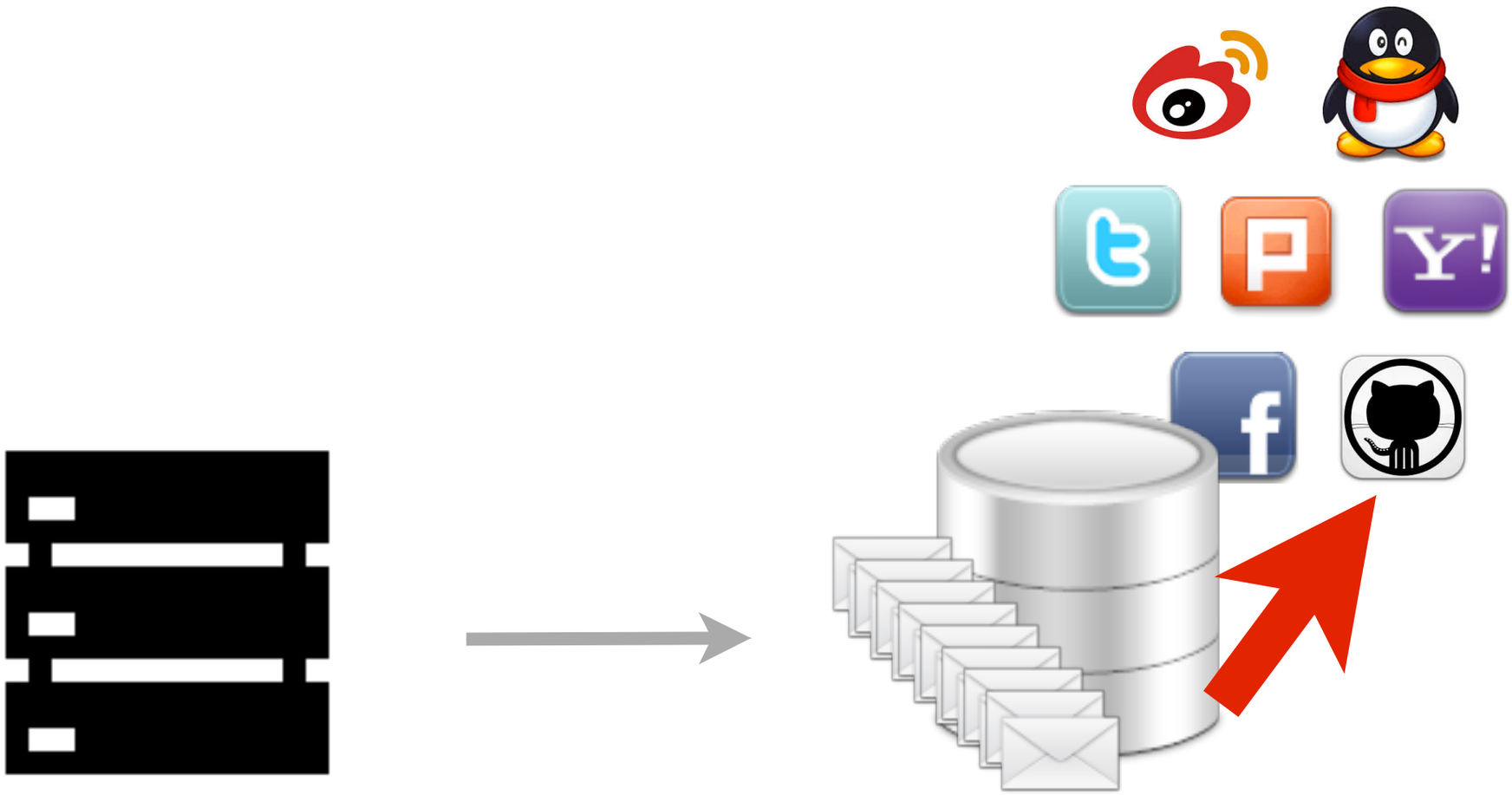
```
if ( ! req.query.buildName) {…

data = {…

slack.webhook({…
  console.log(response);
});
```



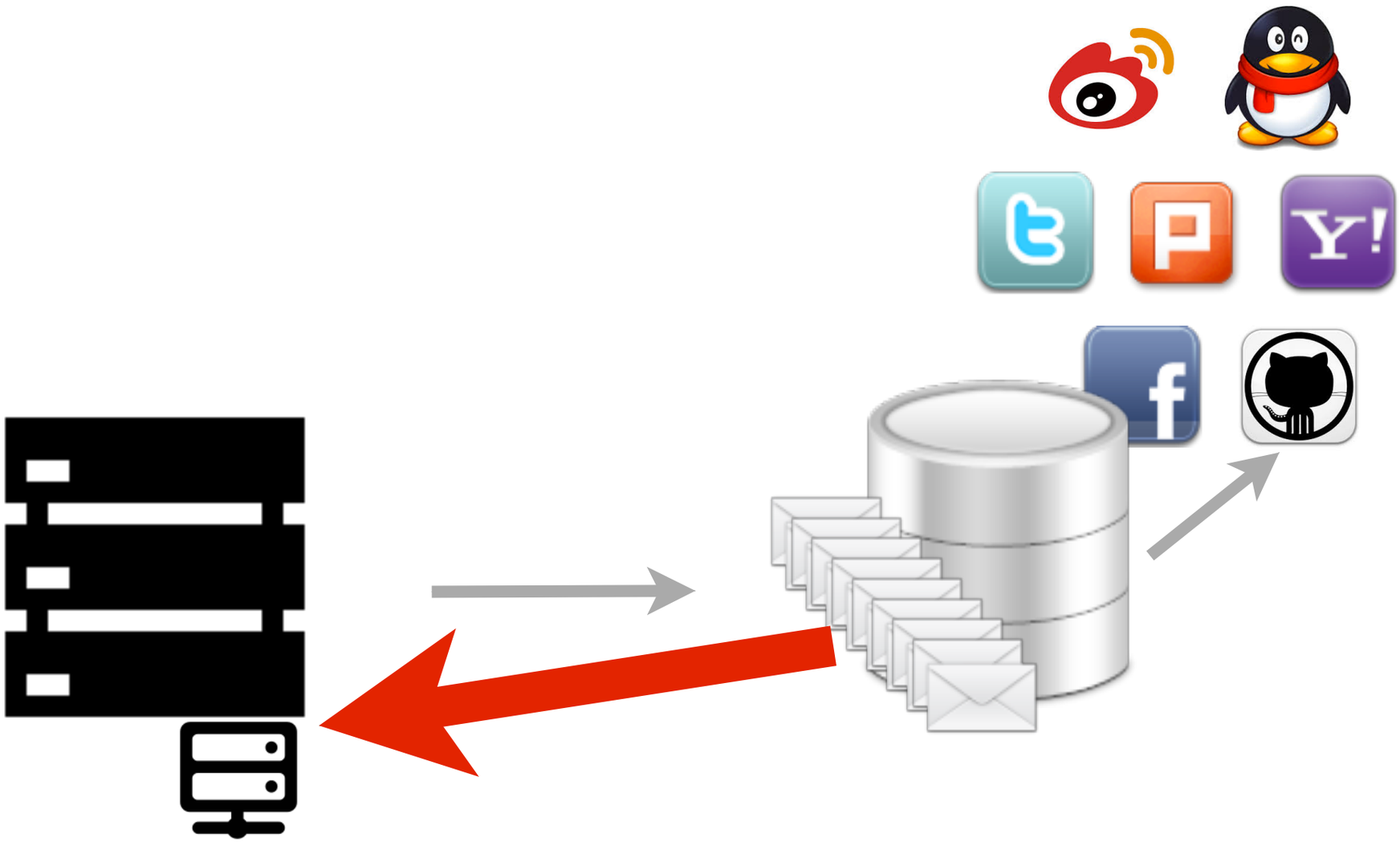
```
exports.jenkins = function(req, res){
  var body = req.body || {};
  var redmine = config.redmine;
  var title = req.query.buildName || "";

  res.send({
    status: 200,
    message: "it is webhook"
  });

if ( ! req.query.buildName) {…

data = {…

slack.webhook({…
  console.log(response);
});

};
```



Set callback event when things done



Send request to 3rd party library / API / URL



Feedback to Server or Storage

trigger callback event to Server


THAT WAS...
AWESOME
memegenerator.net

How about Client?

How feedback to clients, when task queue process task / jobs done, how could we notify clients?

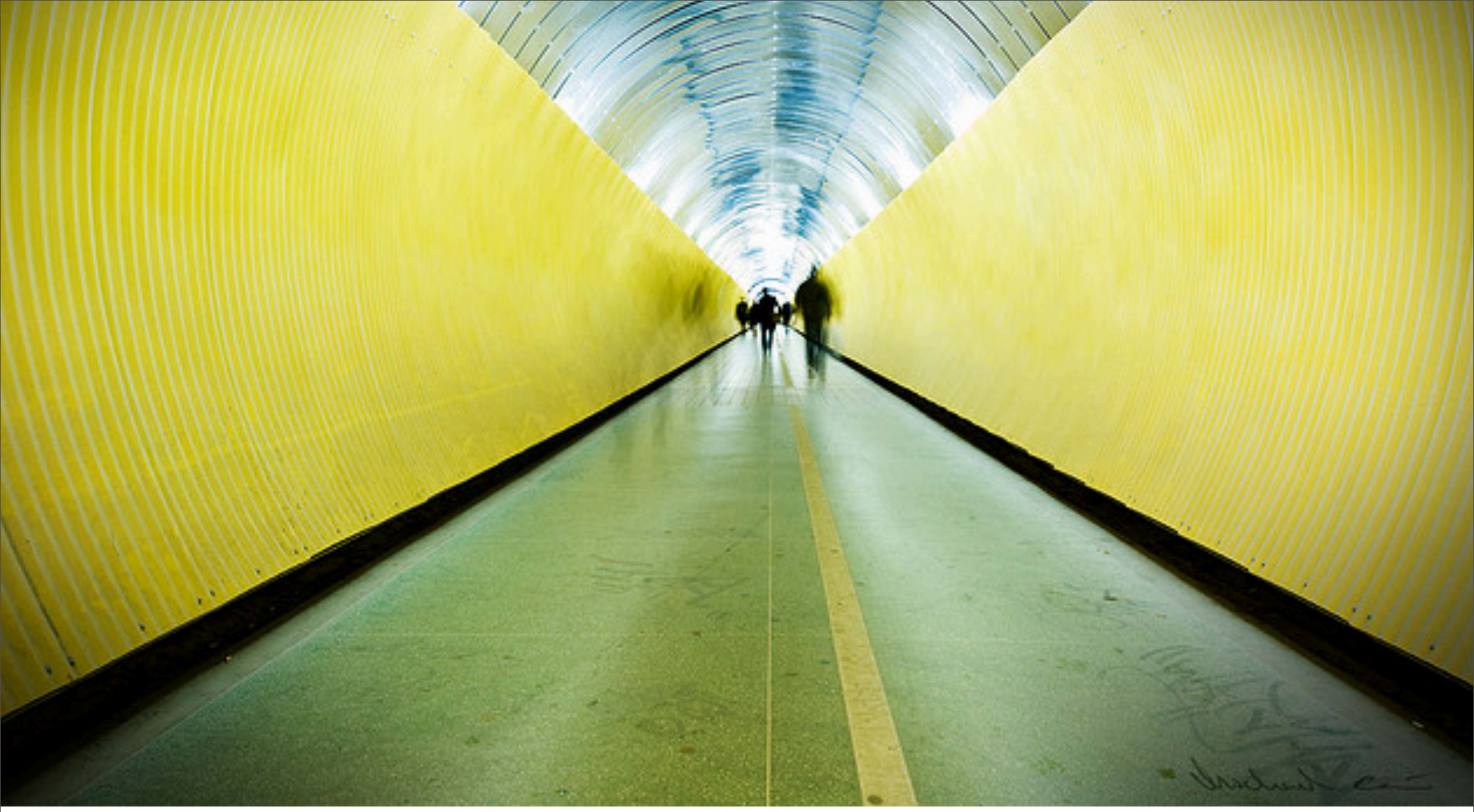
Time meet

socket.io

http://socket.io/

or

Socket



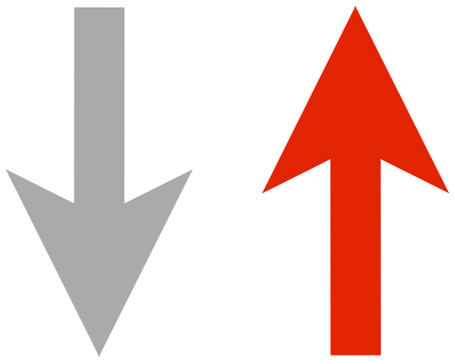
Client and Server connect via a Tunnel

With handshake

Structure

Clients

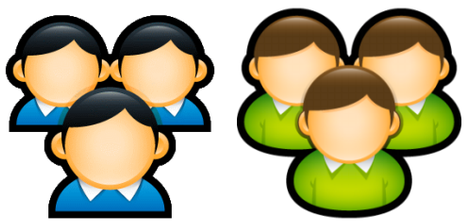
Server push



Server

Structure

Clients



Server push

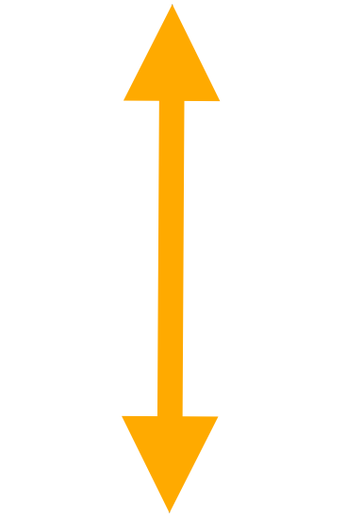
Server

From single way to double ways

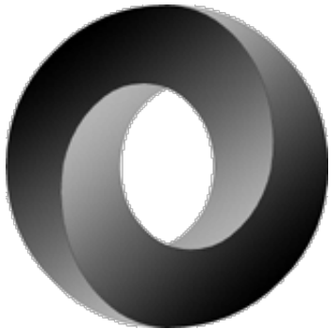
Structure

Clients

Server push



Server



JSON

Ads

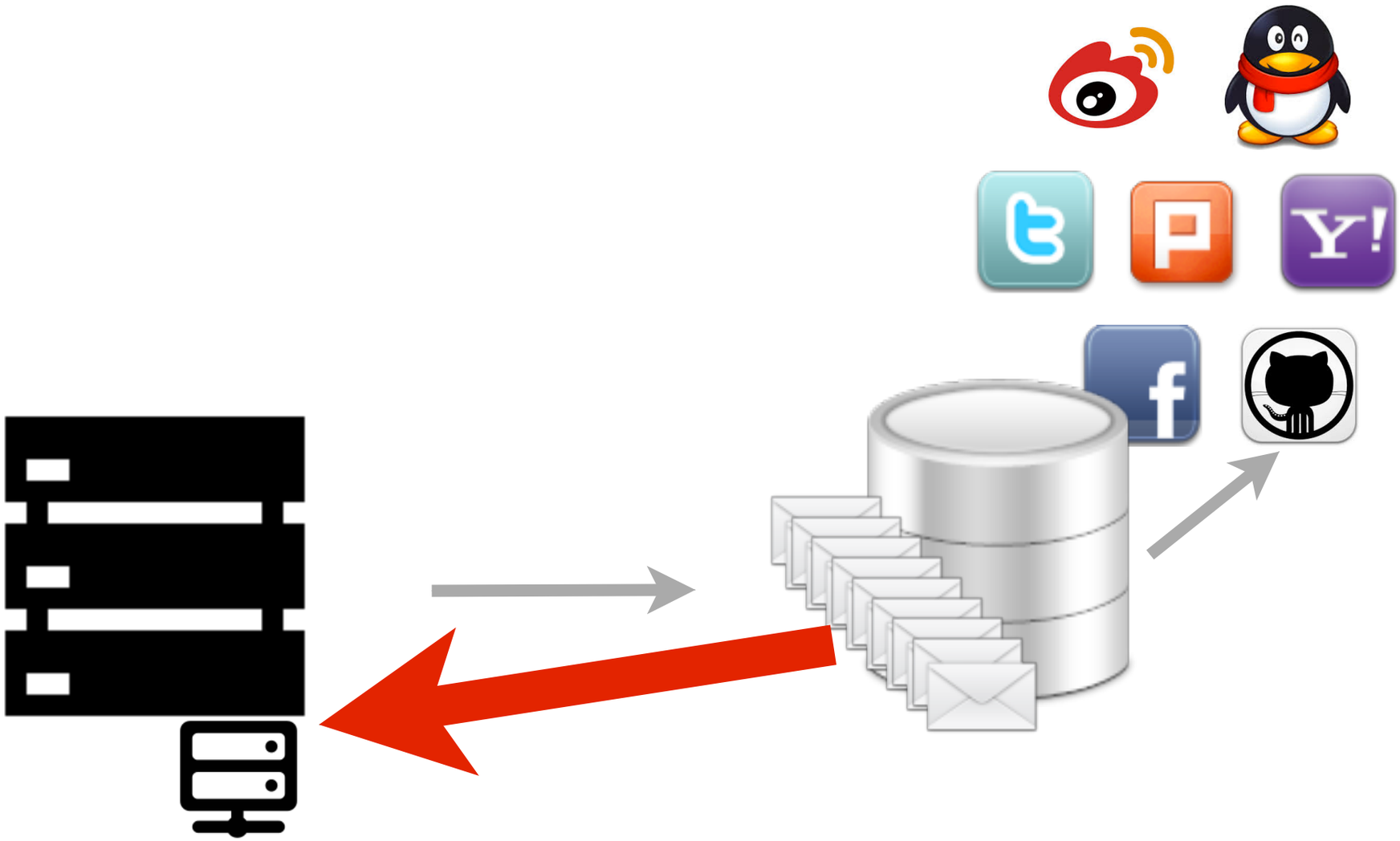
Habit

From single way to double ways

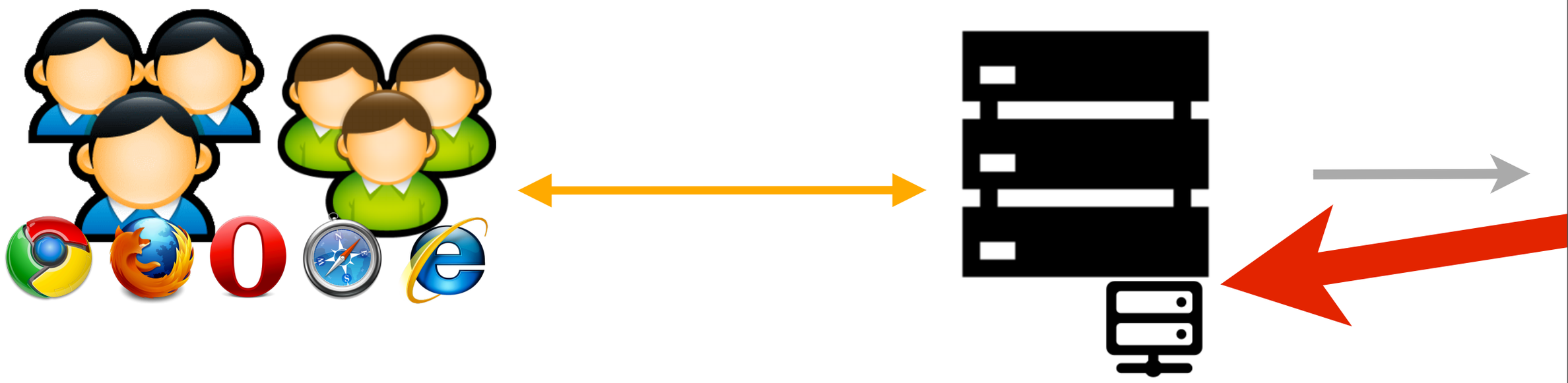
Server push, that means

- Ajax mode changed
- Browser can do more things
- User request is getting more
- Server has to afford more currency
- Server has opportunity feedback data

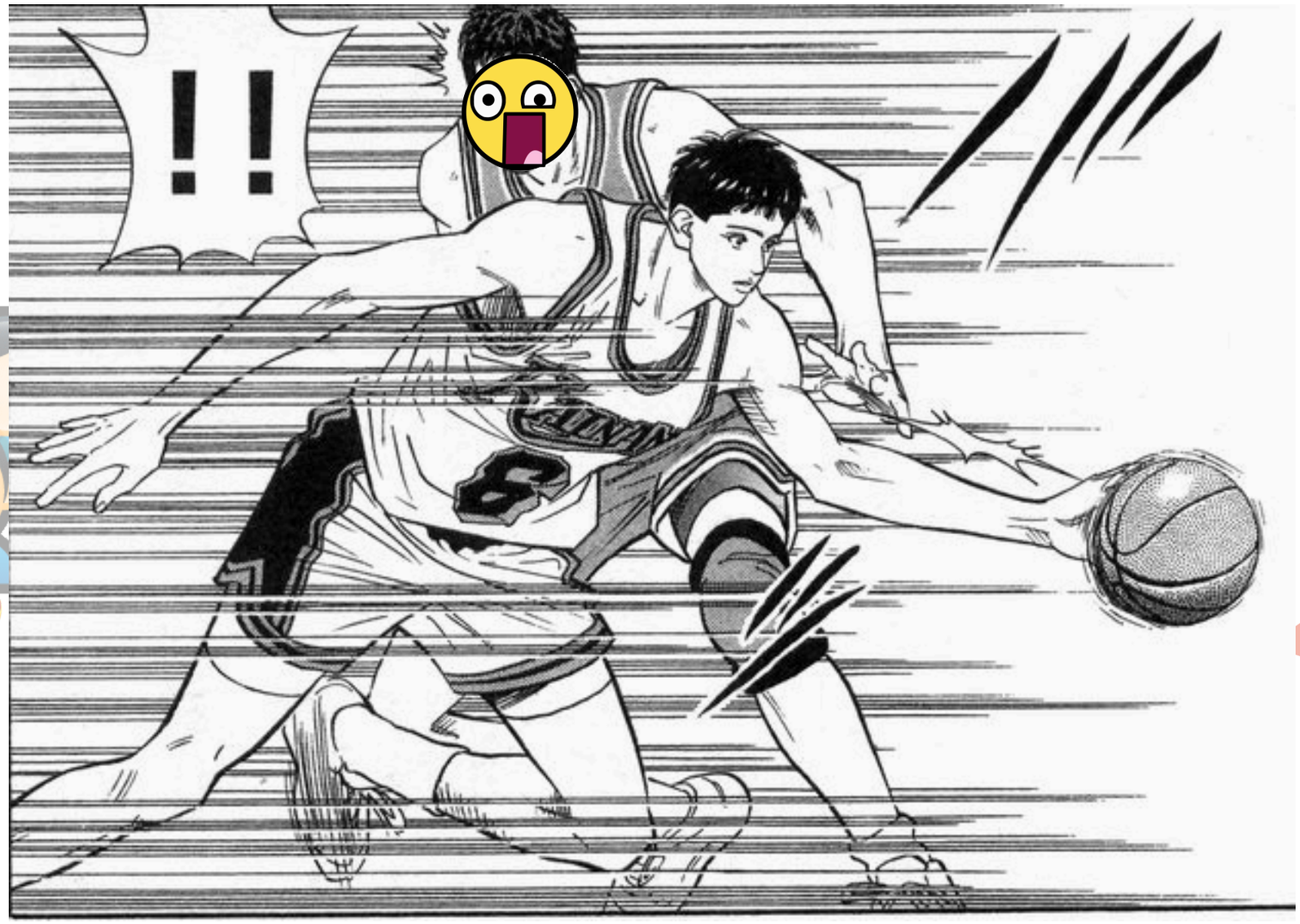
Let's integrate with Socket.io / Socket and Task Queue



Feedback to Server or Storage



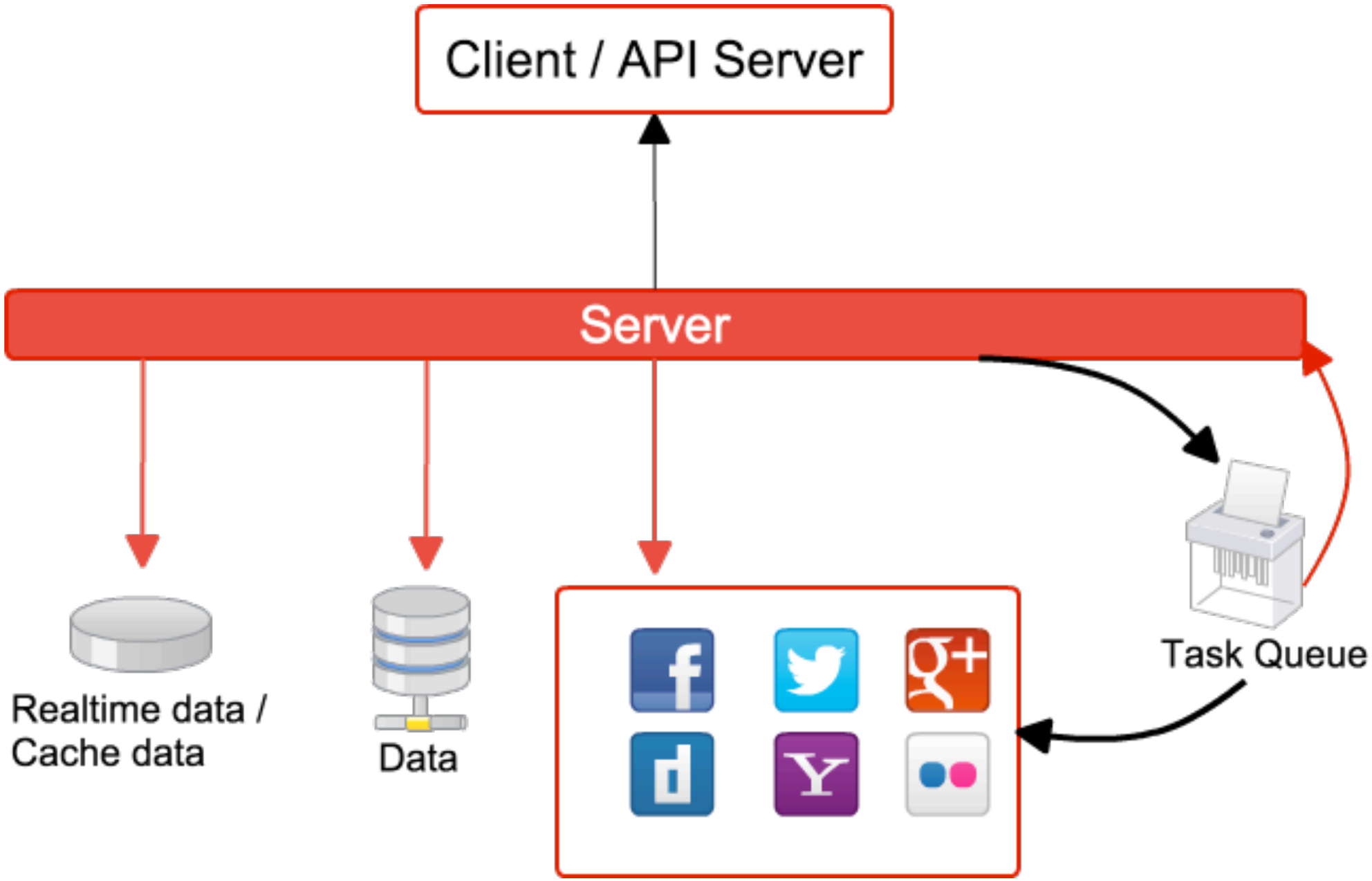
Return data to User in a few minutes

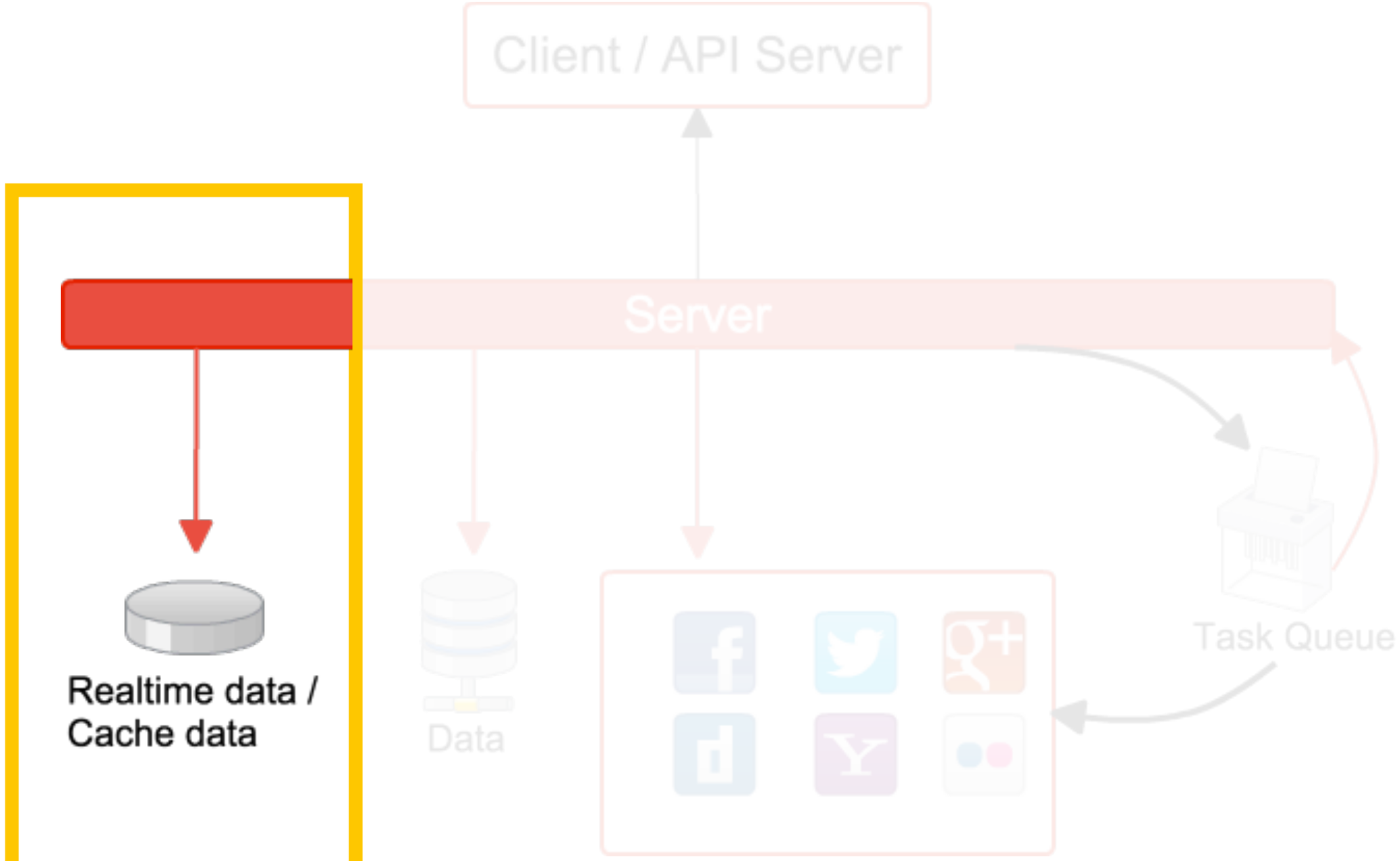


Return data to User in a few seconds

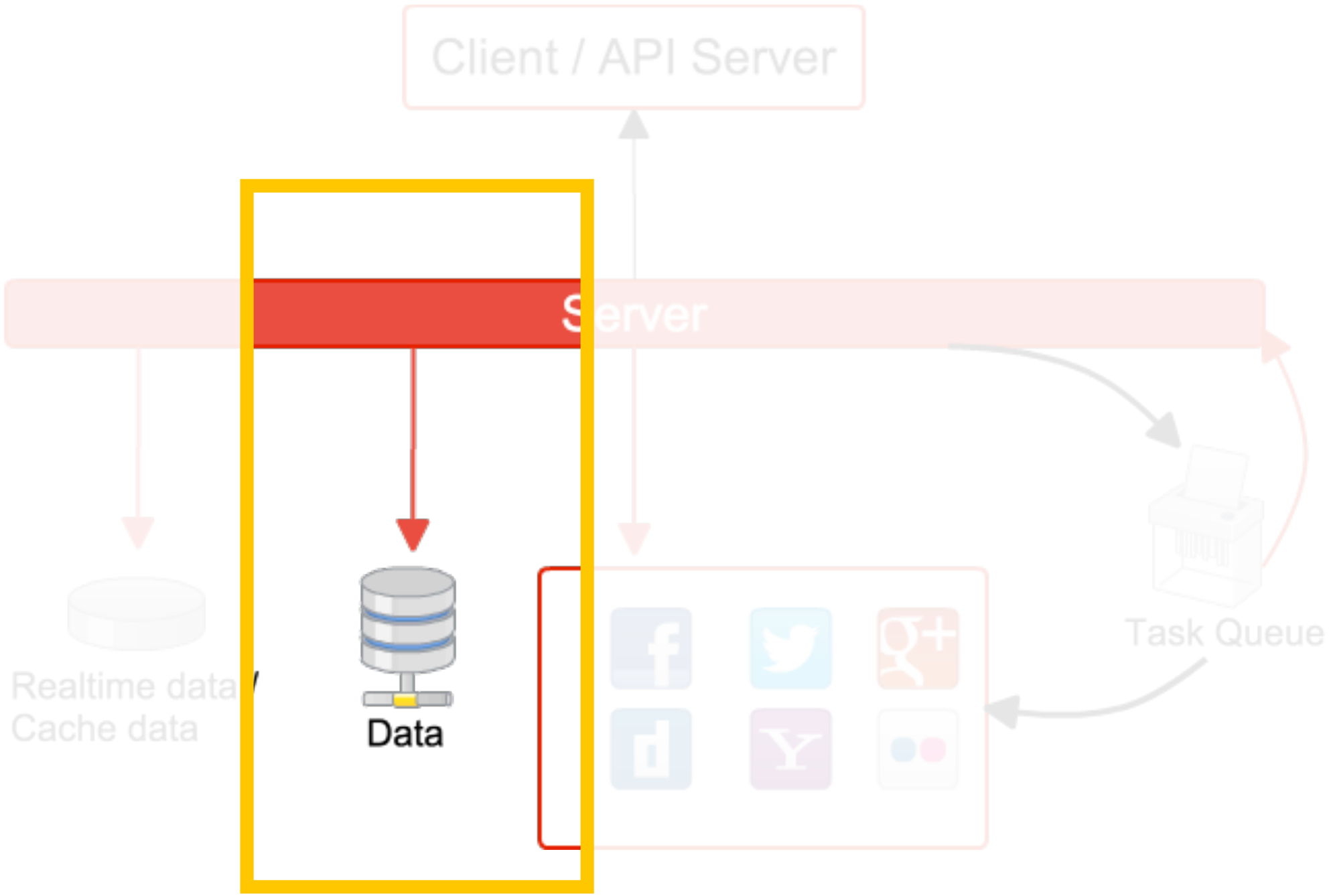
Increase User Experience

Not Make Them Wait

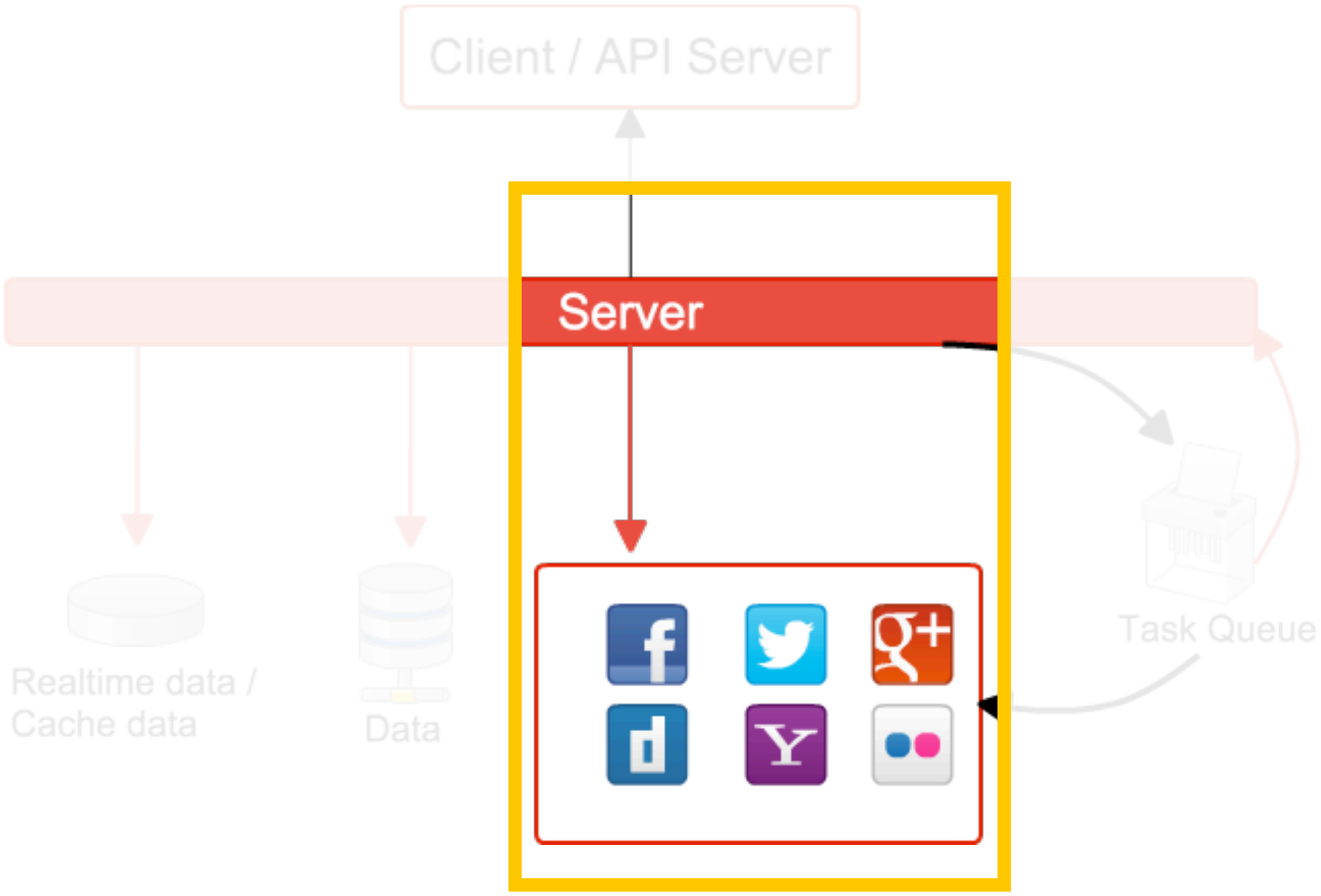

Client / API Server
Server
Realtime data /
Cache data
Data
Task Queue


Client / API Server
Server
Realtime data /
Cache data
Data
Task Queue

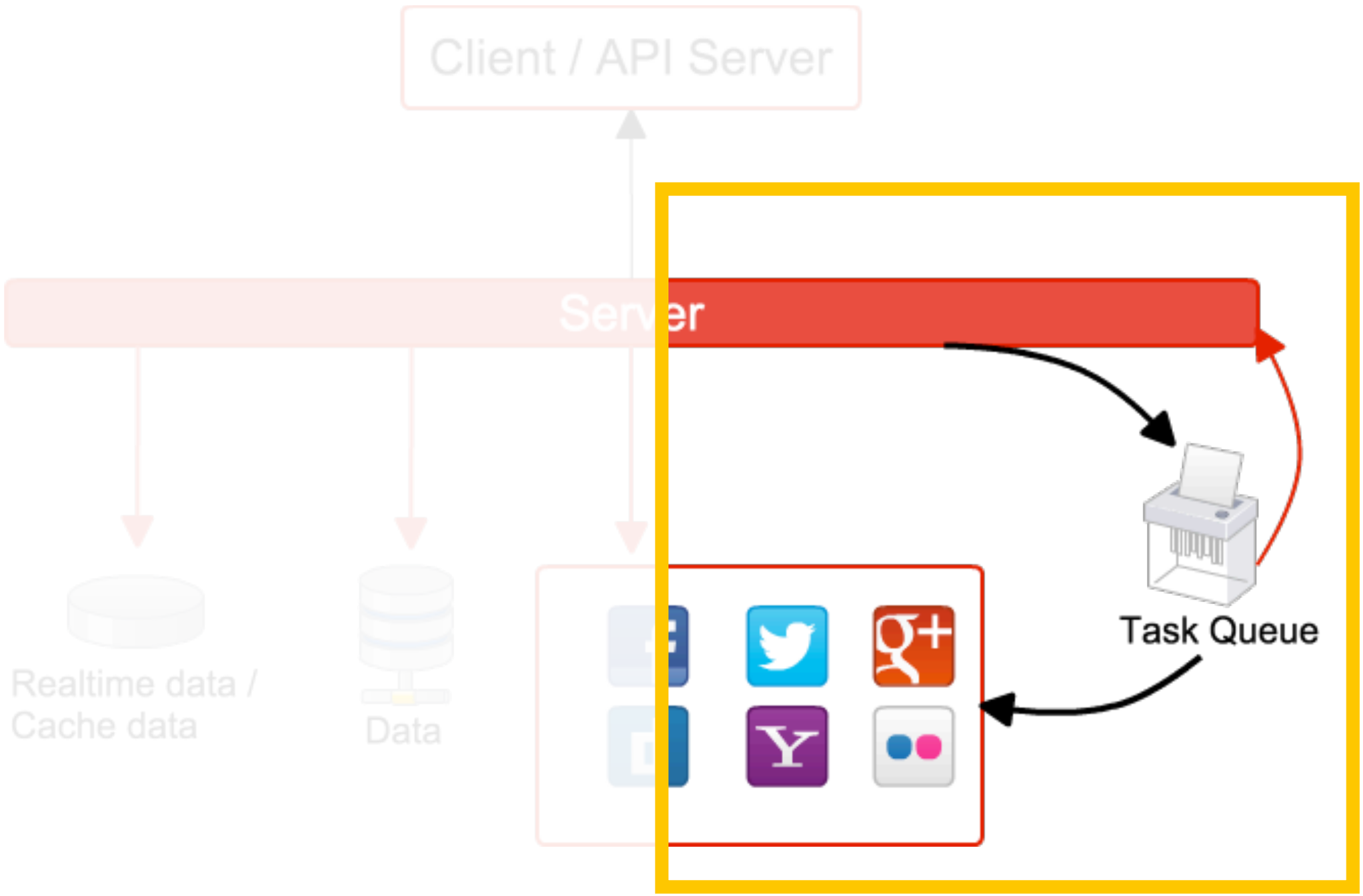
Cache Layer



Storage Layer

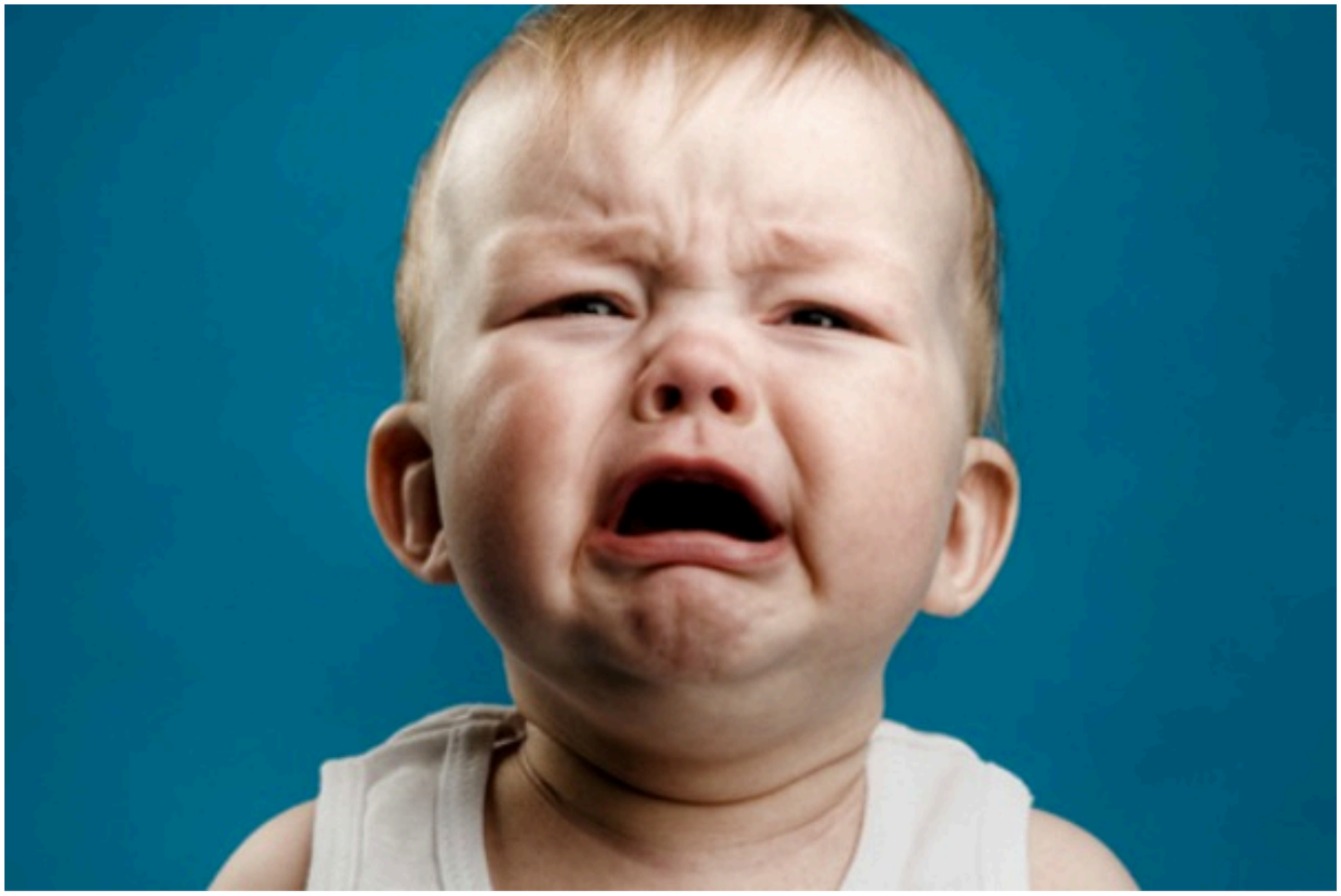


Part of access



Part of heavy access

Not finished yet



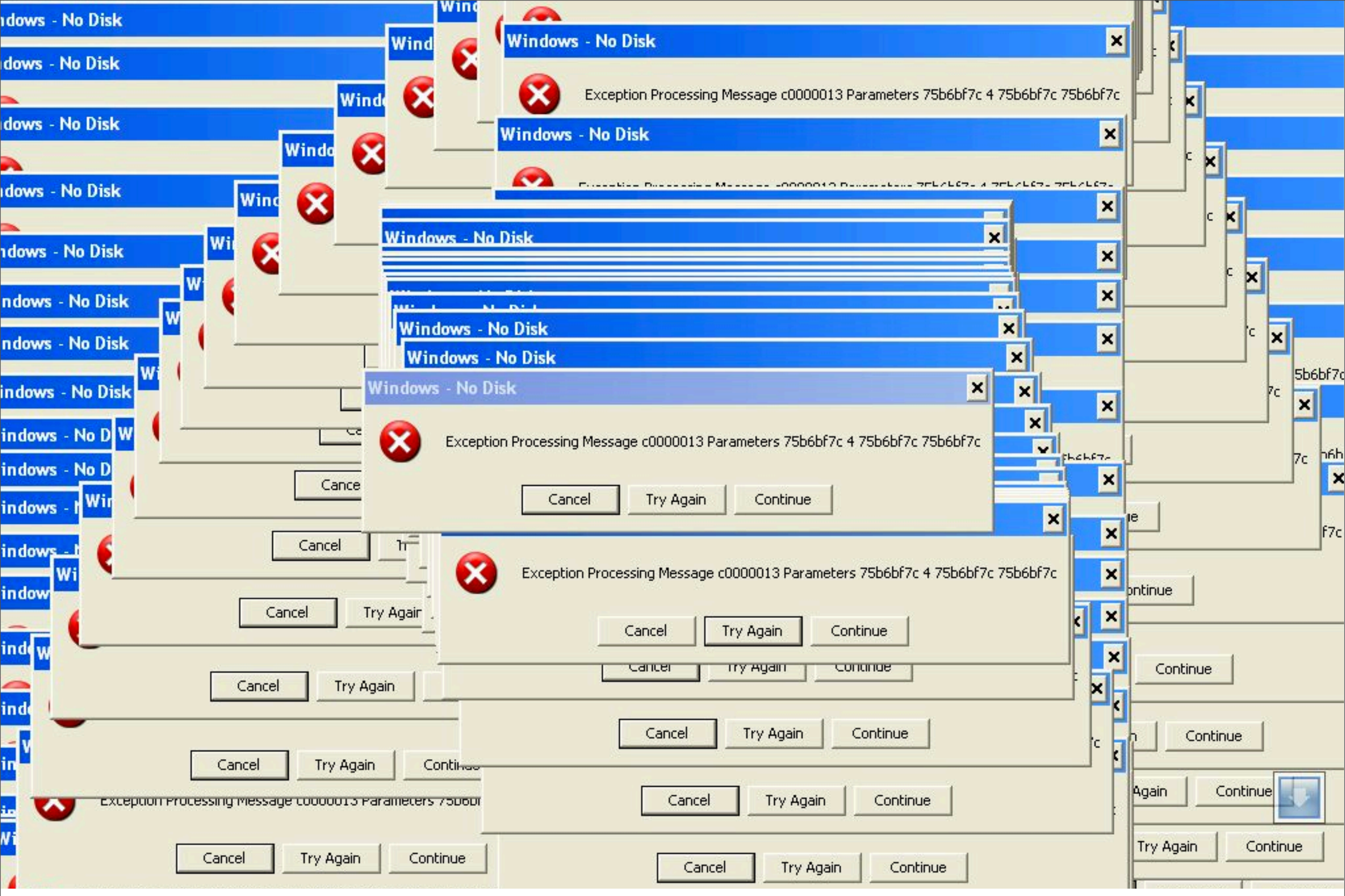
chmen

21 + 1 + 3

r by host name

Ping type	Service	AVG res. (ms)	outages	last outage	warnings	last warning	uptime	
http	express / home 20 7616 last check: 12:55 a few seconds ago	330	65	11:25 2 hours ago			98.376% started 2 hours ago	detail
http	Idibay / home 20 4000 7074 last check: 12:55 a few seconds ago	1727	44	12:17 39 minutes ago	147	11:31 an hour ago	98.758% started 38 minutes ago	detail
http	iloire.com / english version 20 1500 7610 last check: 12:55 a few seconds ago	535	12	07:11 6 hours ago	37	11:31 an hour ago	99.481% started 6 hours ago	detail
http	iloire.com / home 20 1500 7580 last check: 12:55 a few seconds ago	651	11	07:11 6 hours ago	37	11:31 an hour ago	99.443% started 6 hours ago	detail
http	form post test / post 20 7631 last check: 12:55 a few seconds ago	290	6	11:31 an hour ago			99.192% started an hour ago	detail
http	Idibay direct / home 20 4000 7318 last check: 12:55 a few seconds ago	1247	4	12:16 39 minutes ago	8	11:31 an hour ago	99.788% started 38 minutes ago	detail
http	ASP Photo Gallery / demomvc 20 1300 7594 last check: 12:55 a few seconds ago	645	3	03:47 9 hours ago	44	11:31 an hour ago	99.929% started 9 hours ago	detail
http	ASP Photo Gallery / mvcphotogallery 20 800 7683 last check: 12:55 a few seconds ago	345	2	Aug 21st, 11:27:29 a day ago	38	12:00 an hour ago	99.942% started a day ago	detail
http	CachiruloValley / home 20 3000 7688 last check: 12:55 a few seconds ago	274	2	12:17 38 minutes ago	8	12:17 38 minutes ago	99.962% started 37 minutes ago	detail
http	VitaminasDev / home 20 1500 7637 last check: 12:55 a few seconds ago	583	1	Aug 21st, 11:27:06 a day ago	29	11:31 an hour ago	99.942% started a day ago	detail
http	ASP Photo Gallery / home 20 1300 7720	230	1	Aug 21st, 11:27:28 a day ago	18	11:31 an hour ago	99.962% started a day ago	detail

http://letsnode.com/example-of-what-node-is-really-good-at



http://mortarnpistol.com/2011/12/16/big-rigs-over-the-road-racing/

then Promises/A+


```
ApiData(token, user).then(function(newData) {
  return that.updateDB(newData.id).then(function(params) {
    return params;
  }).fail(function(error) {
    throw error;
  });
}).then(function(result) {
  return cb(null, result);
}).fail(function(error) {
  return cb(error);
});
```

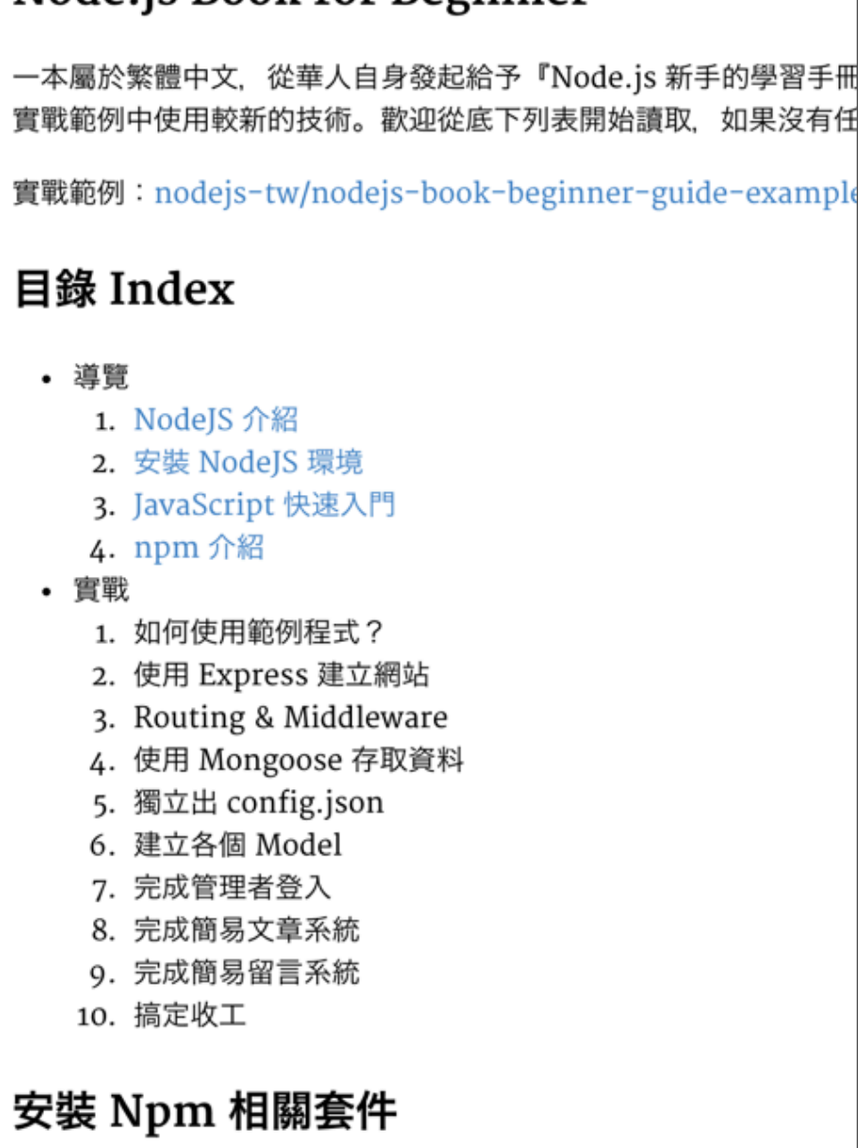
Rules

- Make events as light as posible
- Divide things to tiny
- Heavy process throw to another way
- Monitor app.js after done
- Figure out bottlenecks
- Decrease users idle / waiting time as posible

https://www.facebook.com/groups/node.js.tw/
https://www.facebook.com/NodeJS.tw
https://github.com/nodejs-tw
http://nodejs.tw/
http://2014.jsdc.tw/

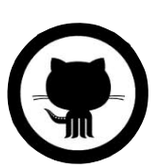
Node.js Taiwon

目錄 Index



GitBook

https://github.com/nodejs-tw/nodejs-book-beginner-guide



@clonncd