

# Oh Node(.js)!

Drupal & Node.js

# Oh Node(.js)!

Drupal & Node.js



# Hi, I'm tizzo

- @tizzo on twitter, Drupal.org, IRC, & anywhere I can get it
- VP of Engineering at Zivtech
- Drupaler since 4.7

# Node.js

What is it?

# Node.js



# Node.js



- Google's V8 javascript engine developed for chrome

# Node.js



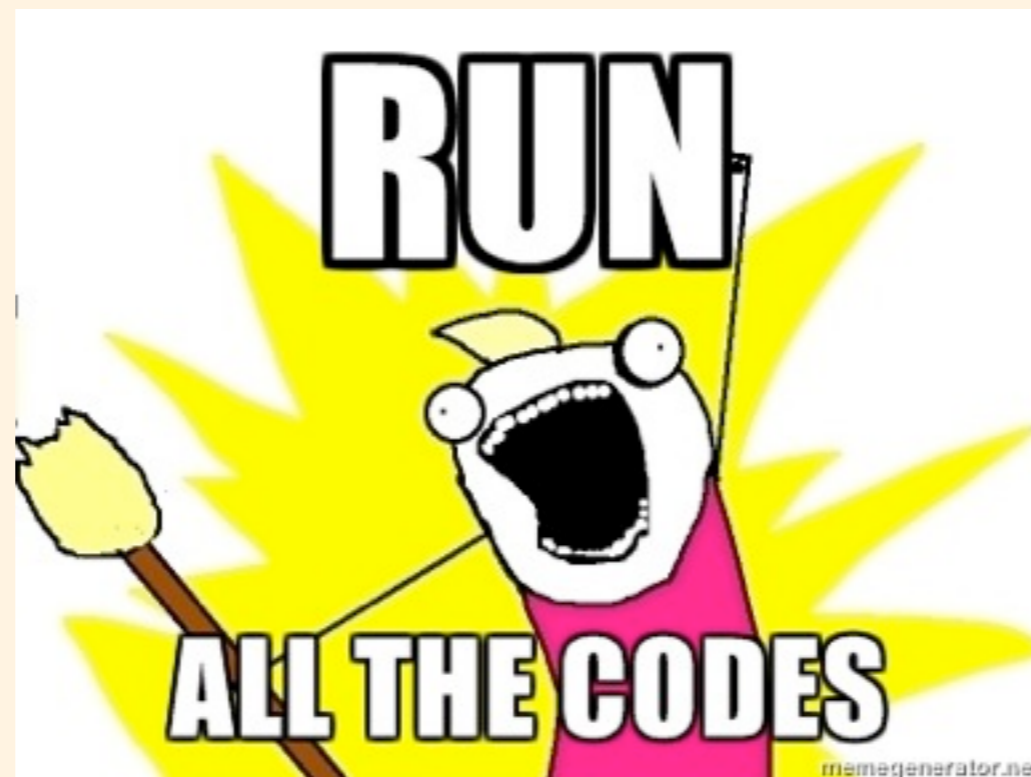
- Google's V8 javascript engine developed for chrome
- Packaged I/O libraries making it easy to use as a web server

# Drupal vs. Node

FIGHT!



# Drupal



Suppose you want to record a date with a form and save it to a database. You'd need an HTML form, a timestamp (integer) field in your schema, and a few lines of code. Throw in a stable jQuery date popup widget and you have more code but not much more complexity. Or you could imagine every possible date permutation, all theoretically accessible to non-developers, and you end up with the **14,673 lines in Drupal's [Date](#) module.**

**Ben Buckman**

<http://benbuckman.net/drupal-excessive-complexity>

# Node.js



# Different tools for different Jobs

# Different tools for different Jobs

- Node.js is *AWESOME* at

# Different tools for different Jobs

- Node.js is *AWESOME* at
  - I/O bound processes

# Different tools for different Jobs

- Node.js is AWESOME at
  - I/O bound processes
  - building only what you need

# Different tools for different Jobs

- Node.js is *AWESOME* at
  - I/O bound processes
  - building only what you need
- Drupal is *AWESOME* at



# Different tools for different Jobs

- Node.js is *AWESOME* at
  - I/O bound processes
  - building only what you need
- Drupal is *AWESOME* at
  - web-app time to market

# Different tools for different Jobs

- Node.js is *AWESOME* at
  - I/O bound processes
  - building only what you need
- Drupal is *AWESOME* at
  - web-app time to market
  - part A & part B fitting together

# Drupal & Node

So happy together!

# Node.js Module

# Node.js Module

- <http://drupal.org/project/nodejs>

# Node.js Module

- <http://drupal.org/project/nodejs>
- Give Drupal push capabilities

# Node.js Module

- <http://drupal.org/project/nodejs>
- Give Drupal push capabilities
- Notifications

# Node.js Module

- <http://drupal.org/project/nodejs>
- Give Drupal push capabilities
- Notifications
  - Channels



# Node.js Module

- <http://drupal.org/project/nodejs>
- Give Drupal push capabilities
- Notifications
  - Channels
    - General (whole site)

# Node.js Module

- <http://drupal.org/project/nodejs>
- Give Drupal push capabilities
- Notifications
  - Channels
    - General (whole site)
    - Content (what I'm looking at)

# Node.js Module

- <http://drupal.org/project/nodejs>
- Give Drupal push capabilities
- Notifications
  - Channels
    - General (whole site)
    - Content (what I'm looking at)
  - Broadcast (everyone, everywhere)

```
1 <?php
2 $message = new stdClass;
3 $message->broadcast = TRUE;
4 $message->channel = 'badcamp';
5 $message->data = array(
6     'foo' => 'bar',
7 );
8 nodejs_enqueue_message($message);
9
```

```
1 Drupal.Nodejs.callbacks.example = {  
2   callback: function(message) {  
3     if (message.channel === 'badcamp') {  
4       alert(message.data.foo);  
5     }  
6   }  
7 }
```

**BUT HOW DOES IT  
WORK?!?!**

# Authentication

nodeJS

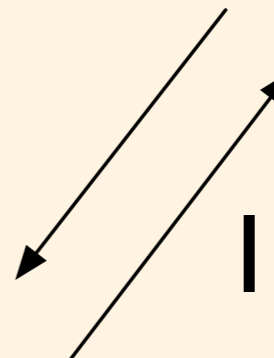


# Authentication

nodeJS



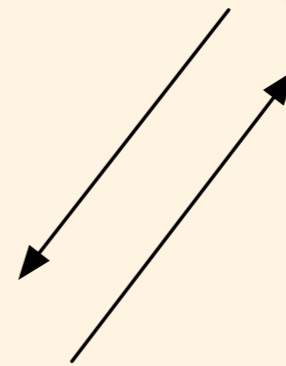
I. Normal Drupal  
Page Load  
(with auth token)





# Authentication

nodeJS

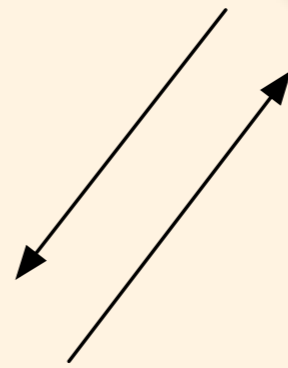
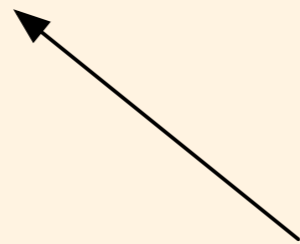


# Authentication

nodeJS

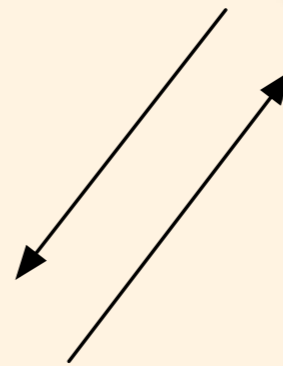
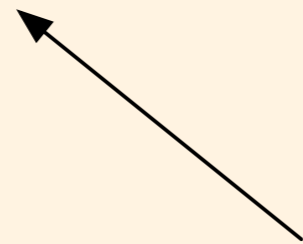


2. Connection requested by socket.io using auth token



# Authentication

nodeJS

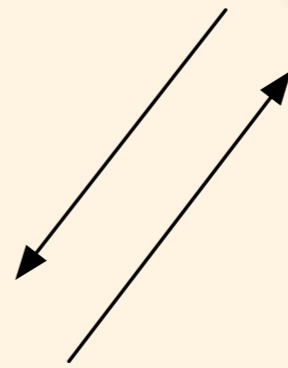
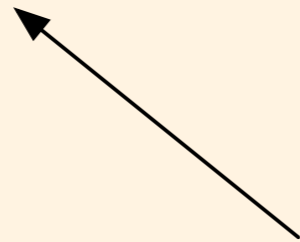


# Authentication

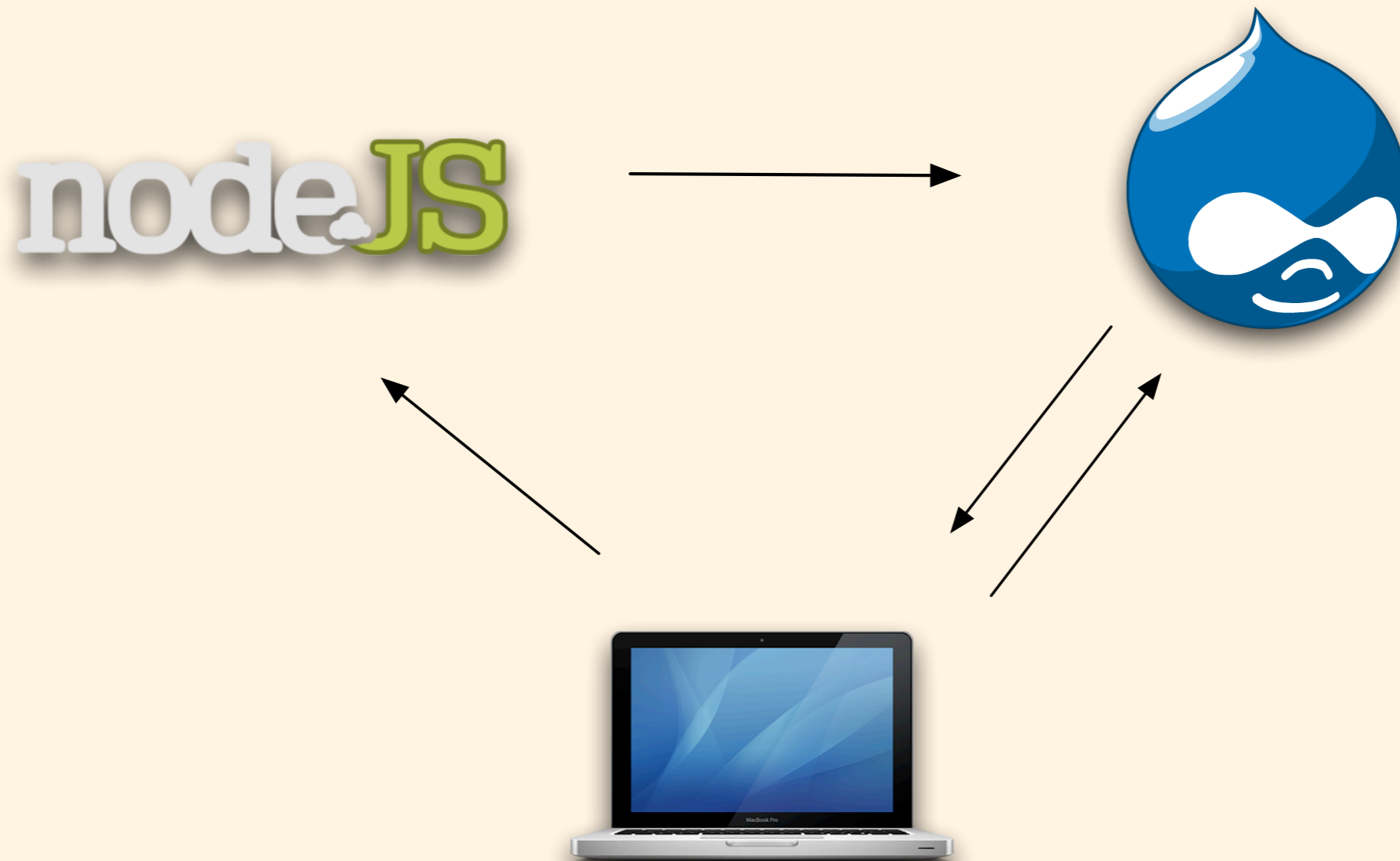
3. Node.js sends  
auth token to

Drupal →

nodeJS



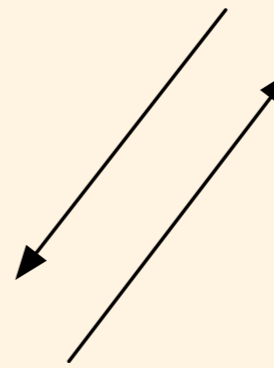
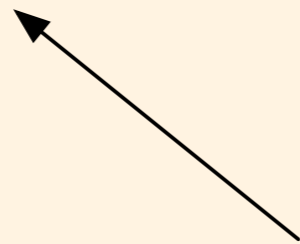
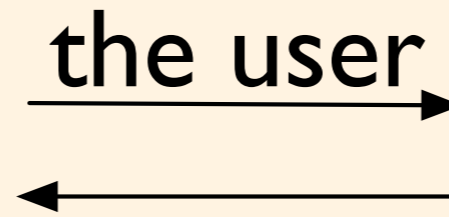
# Authentication



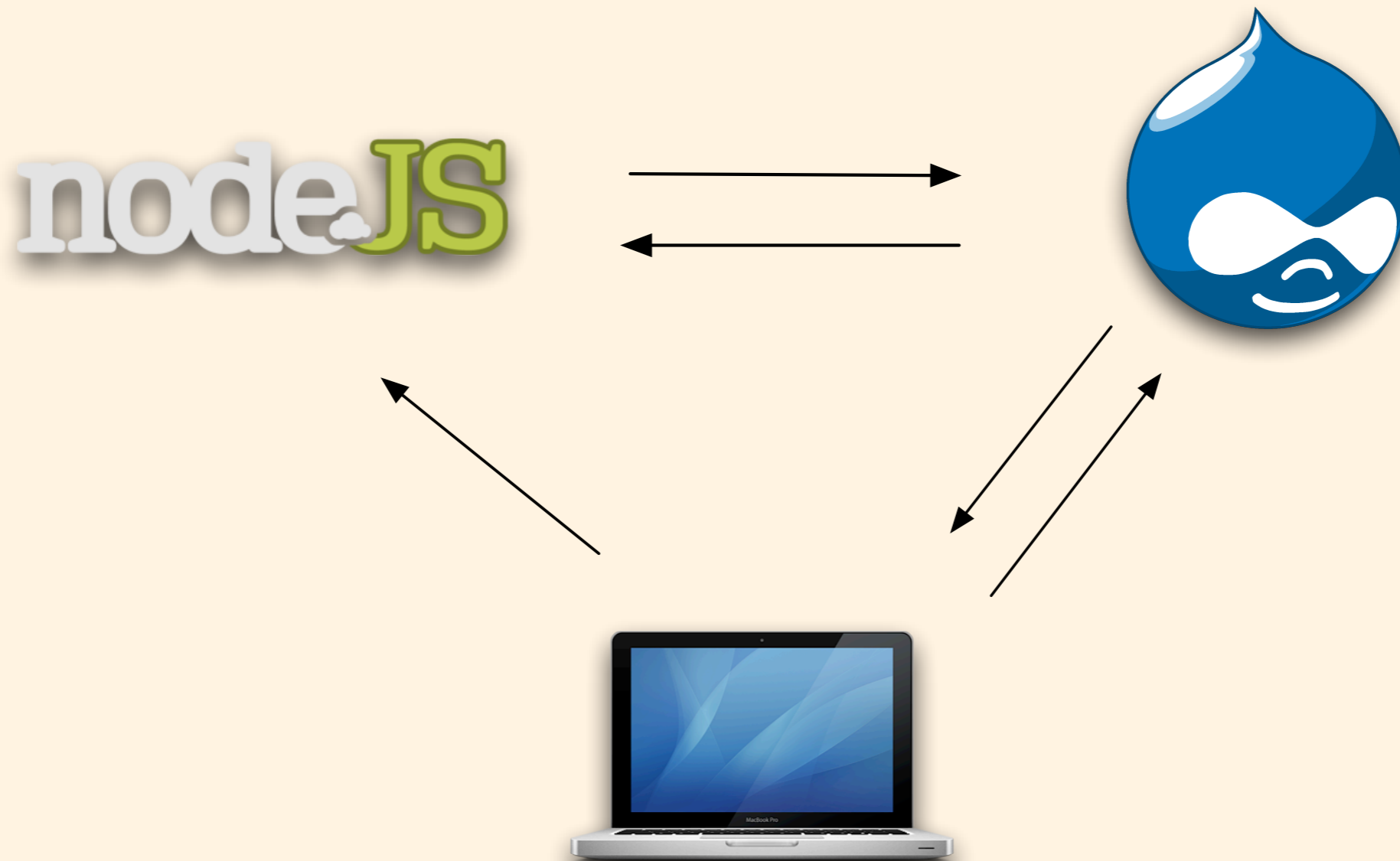
# Authentication

4. Drupal returns a list of channels for the user

nodeJS

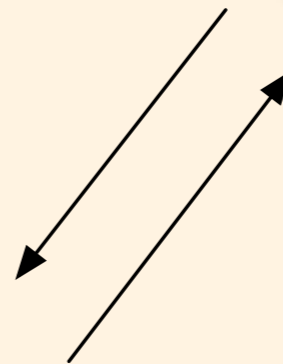
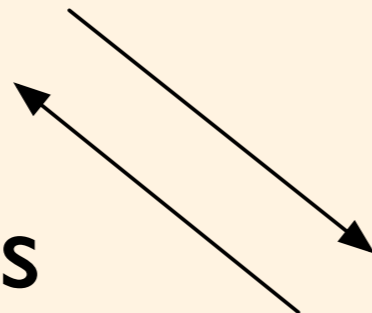
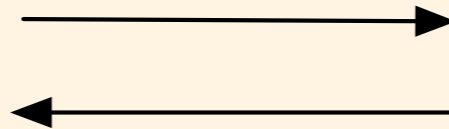


# Authentication



# Authentication

node.JS

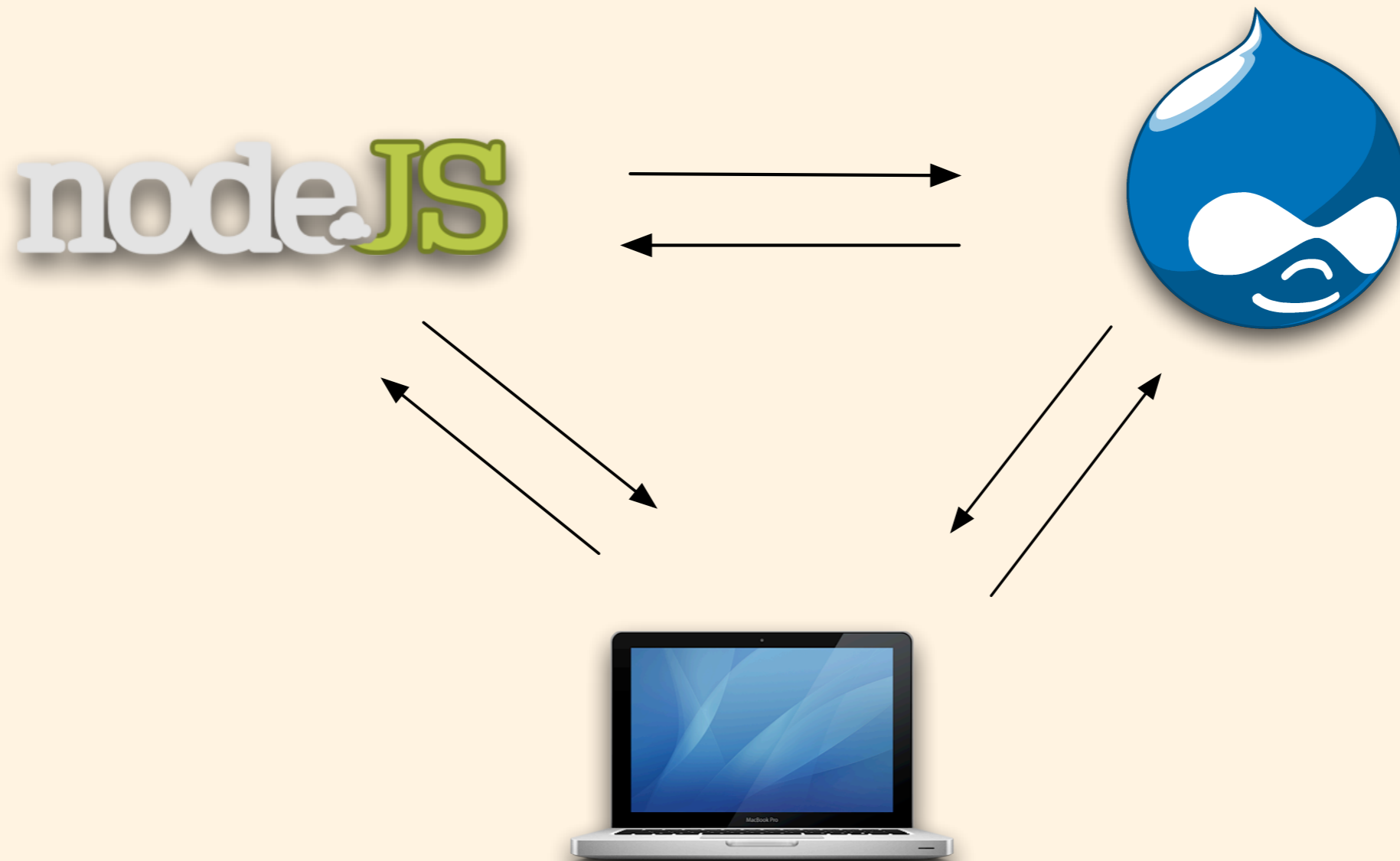


5. Node.js notifies client that auth is successful





# Authentication



# Subsequent page loads

nodeJS

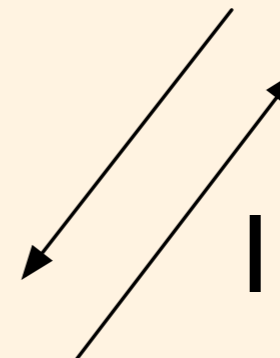


# Subsequent page loads

nodeJS



I. Normal Drupal  
Page Load  
(with auth token)



# Subsequent page loads

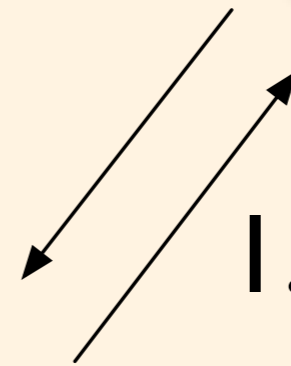
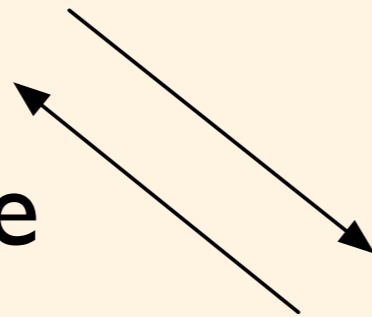
nodeJS



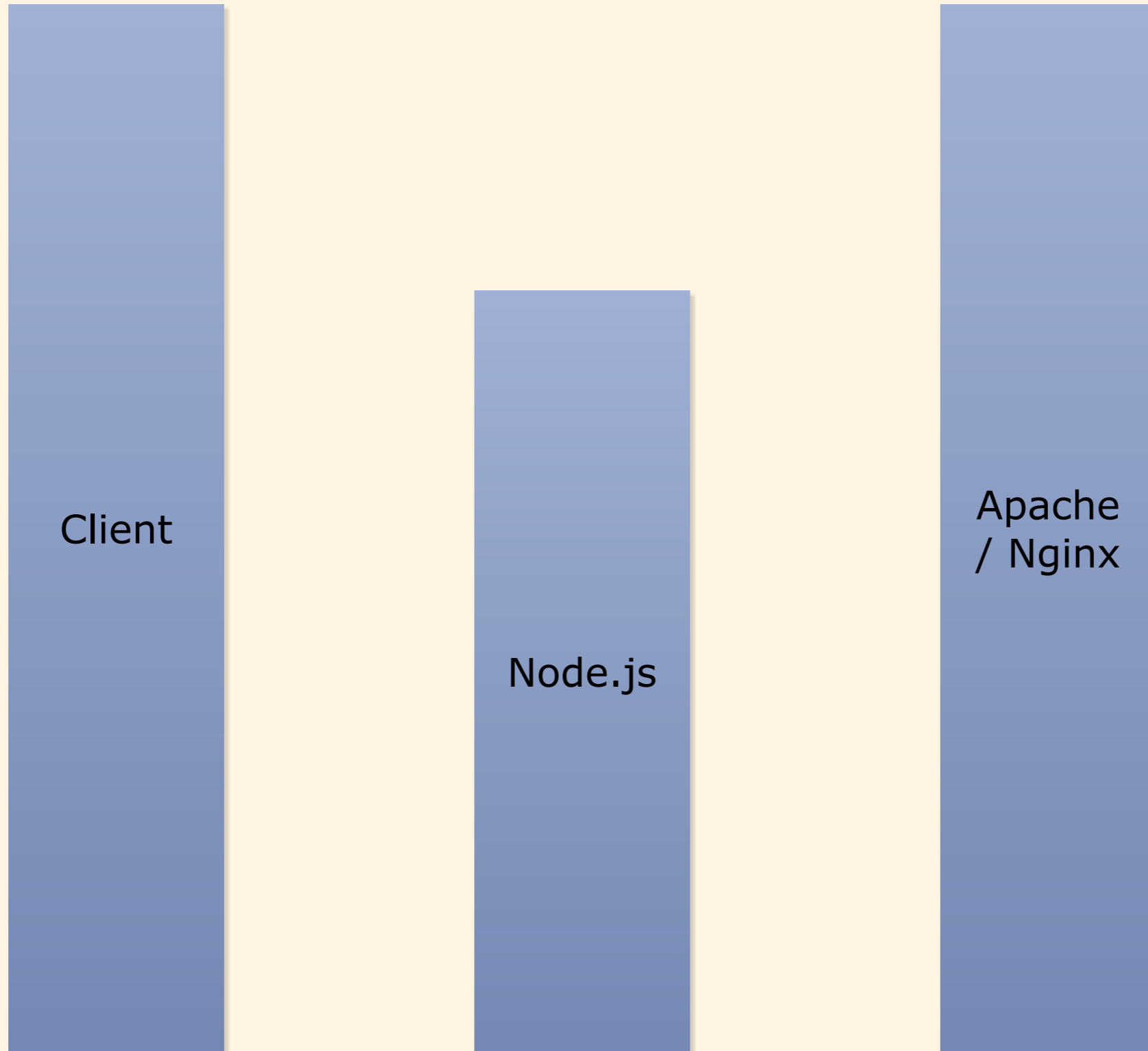
2. Node.js remembers the authenticated token, authenticating the user without a need to query Drupal again



1. Normal Drupal Page Load (with auth token)



# Put another way

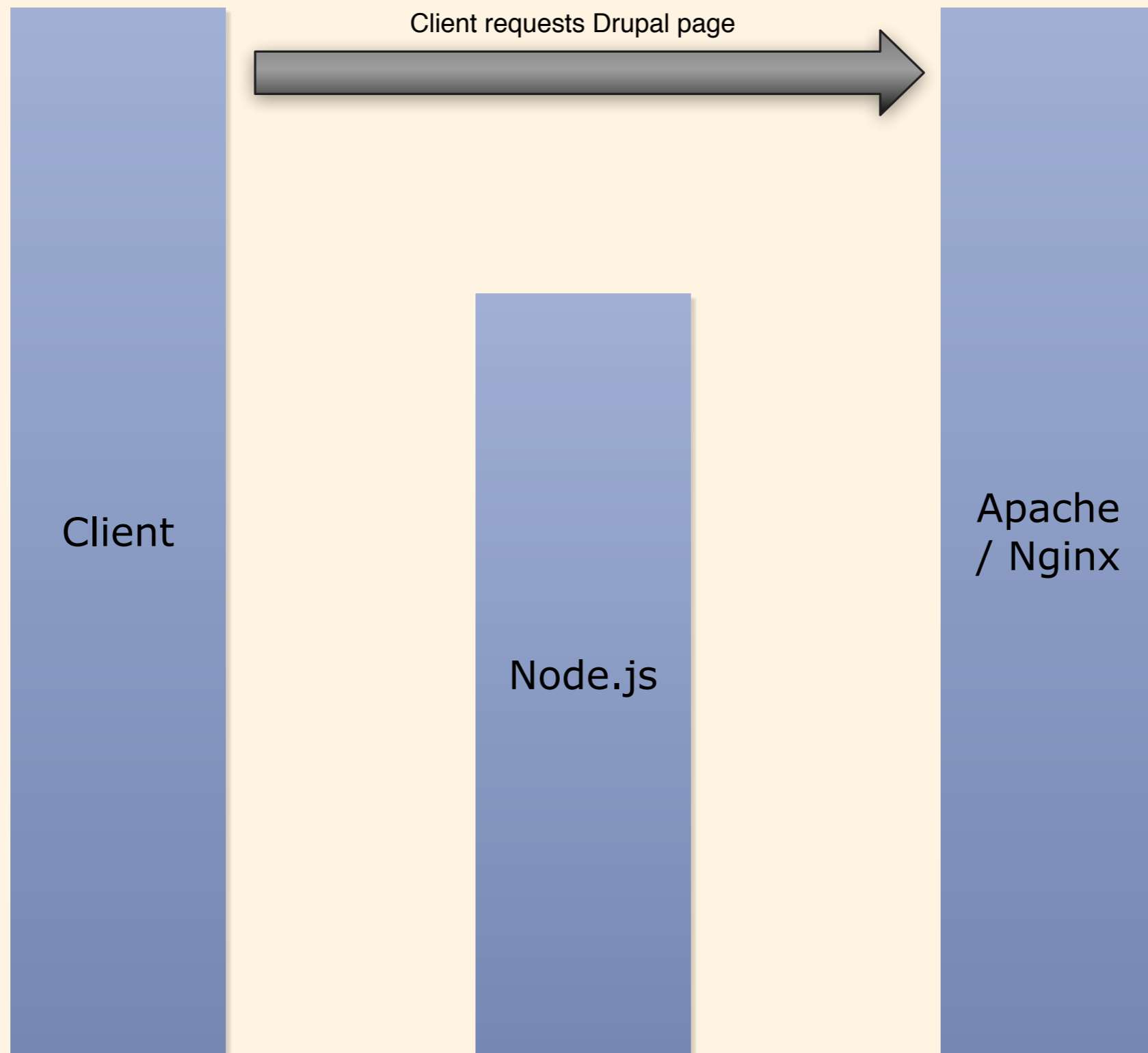


Client

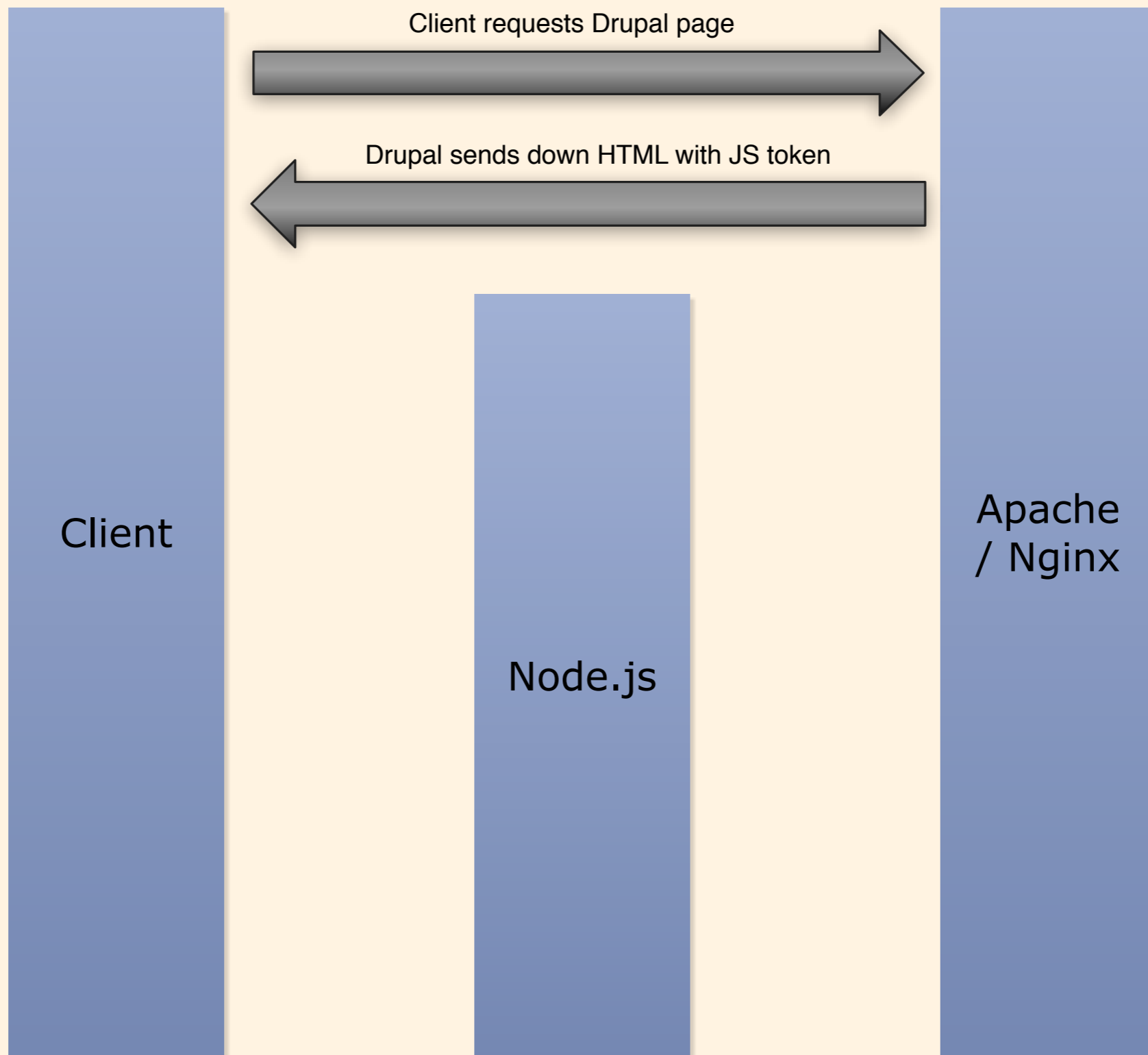
Node.js

Apache  
/ Nginx

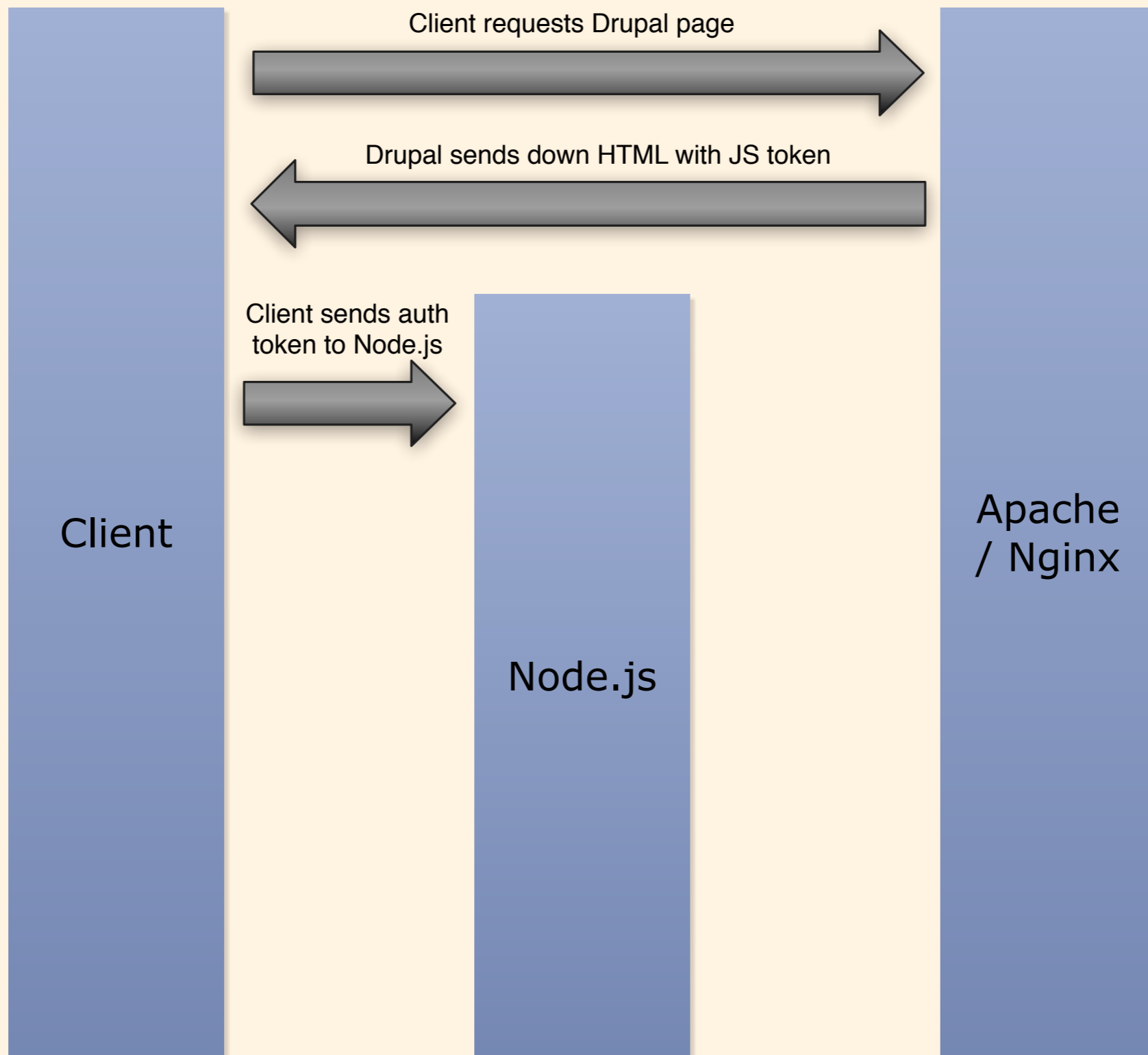
# Put another way



# Put another way

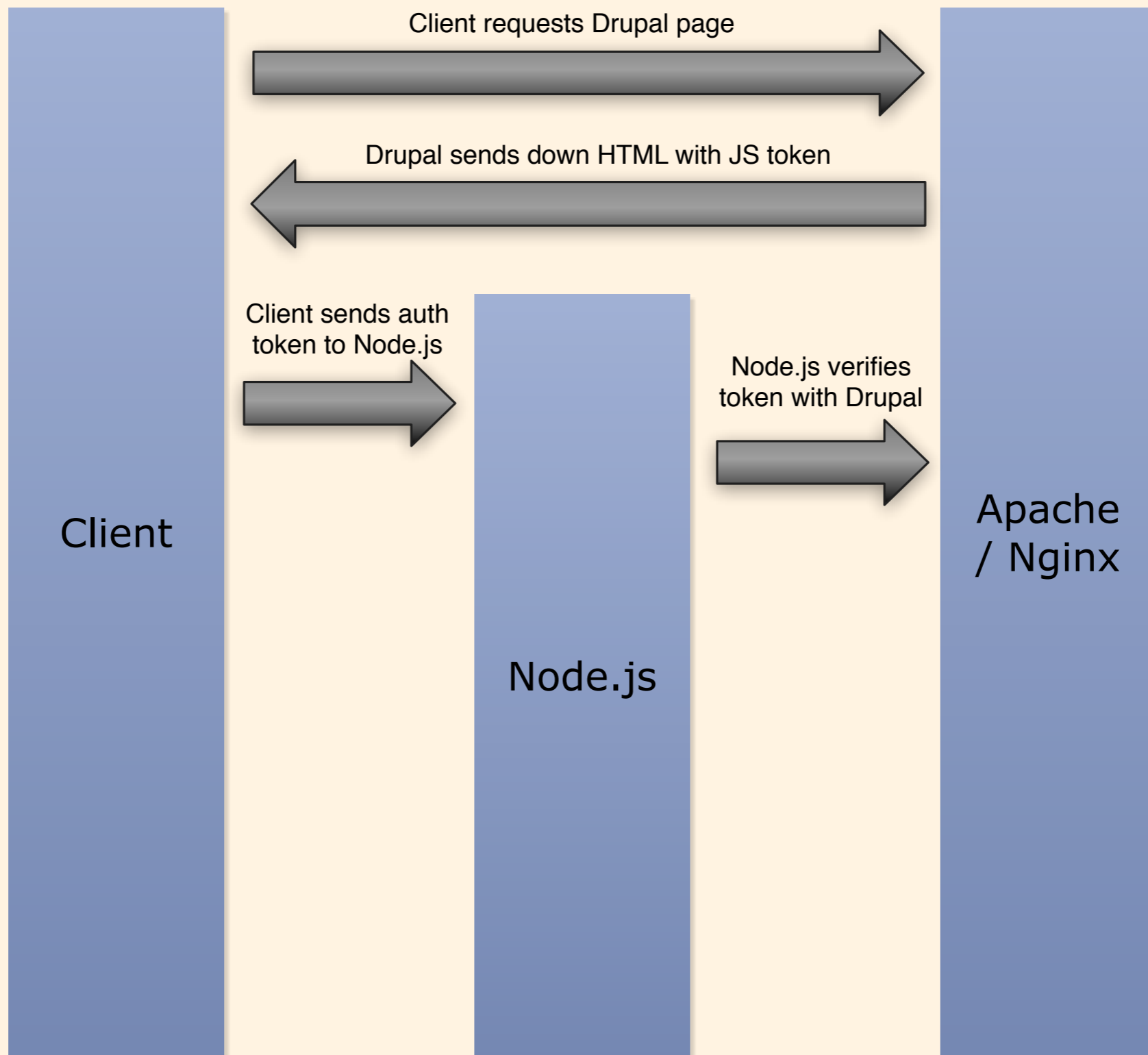


# Put another way

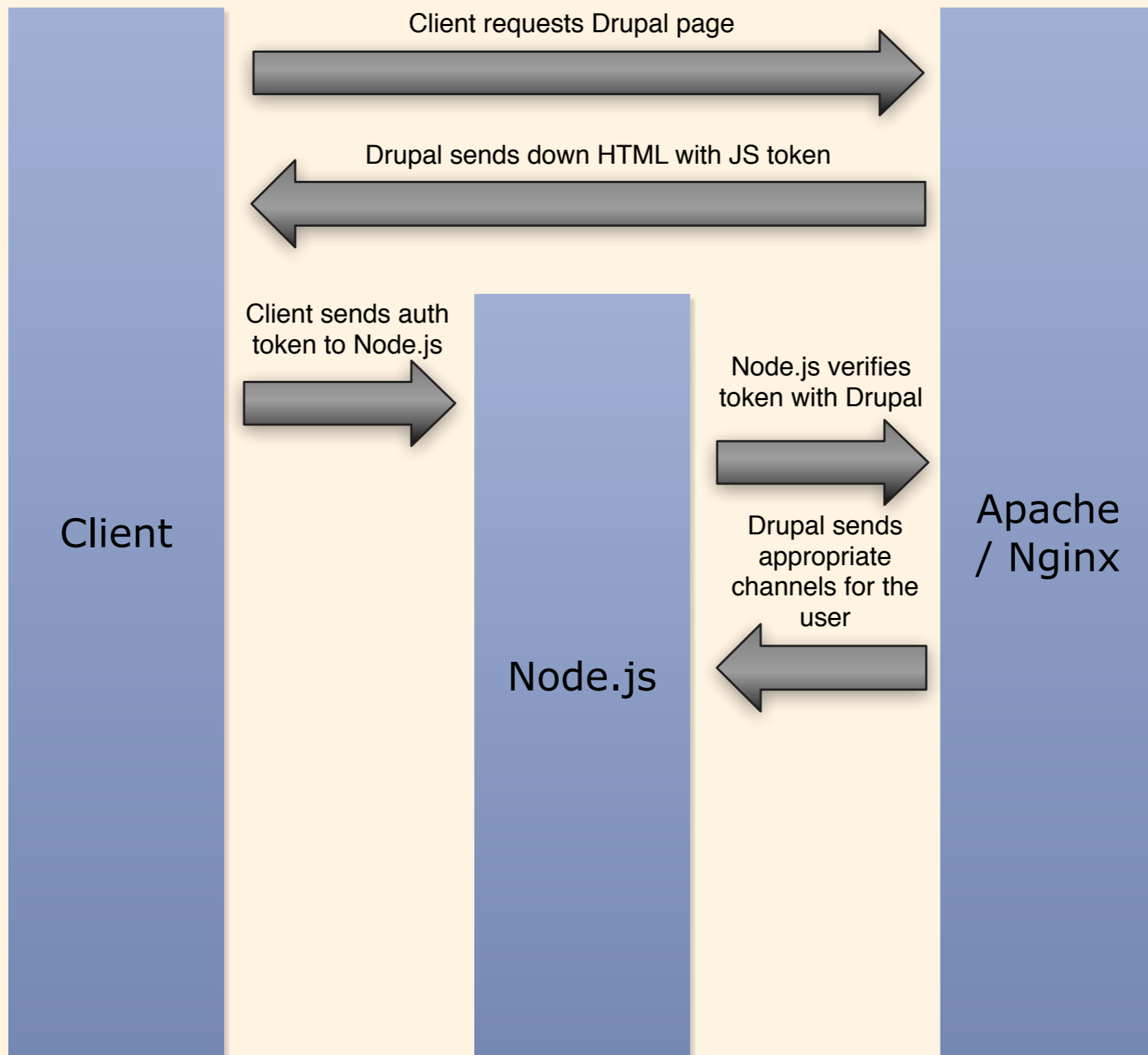




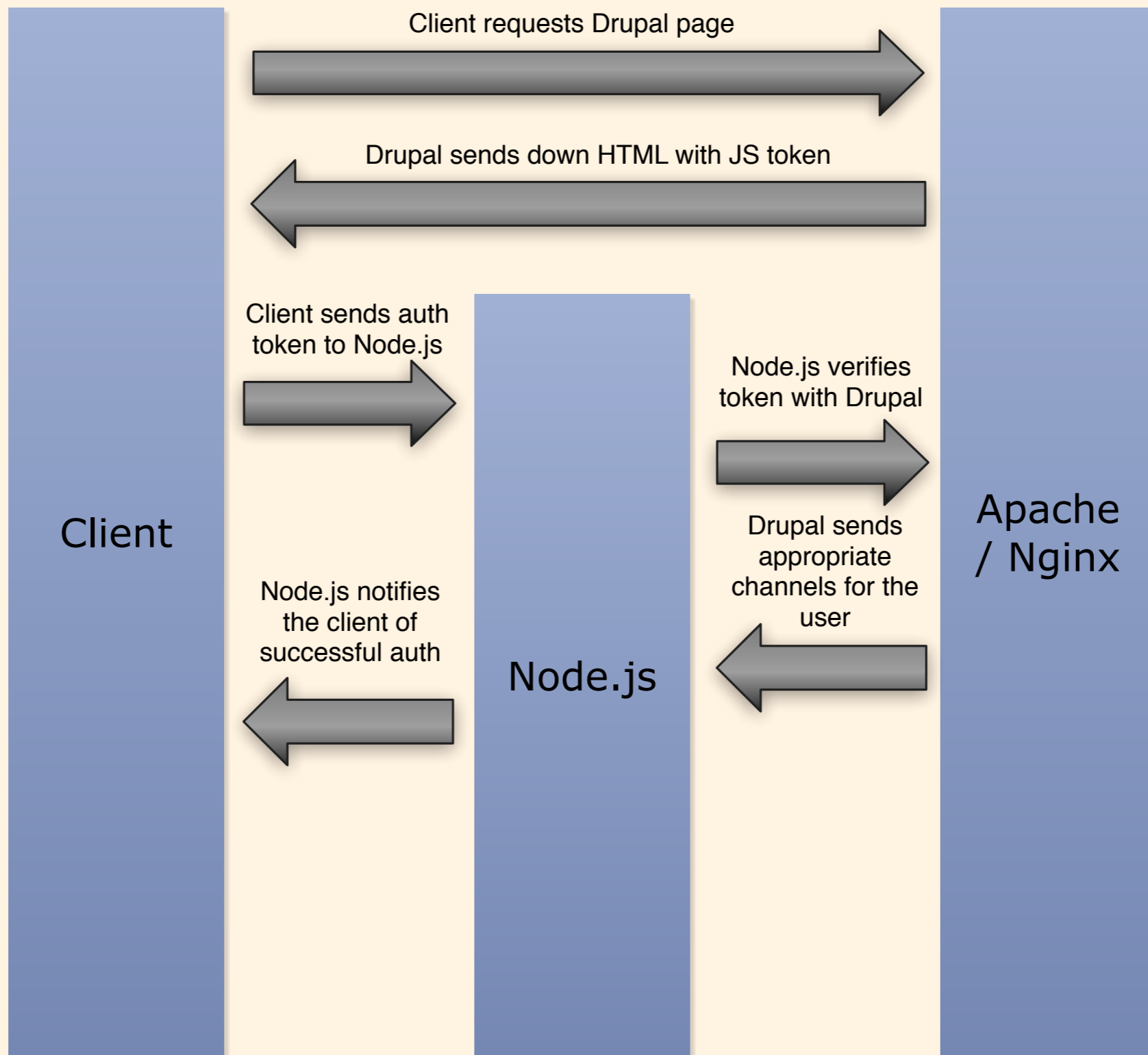
# Put another way



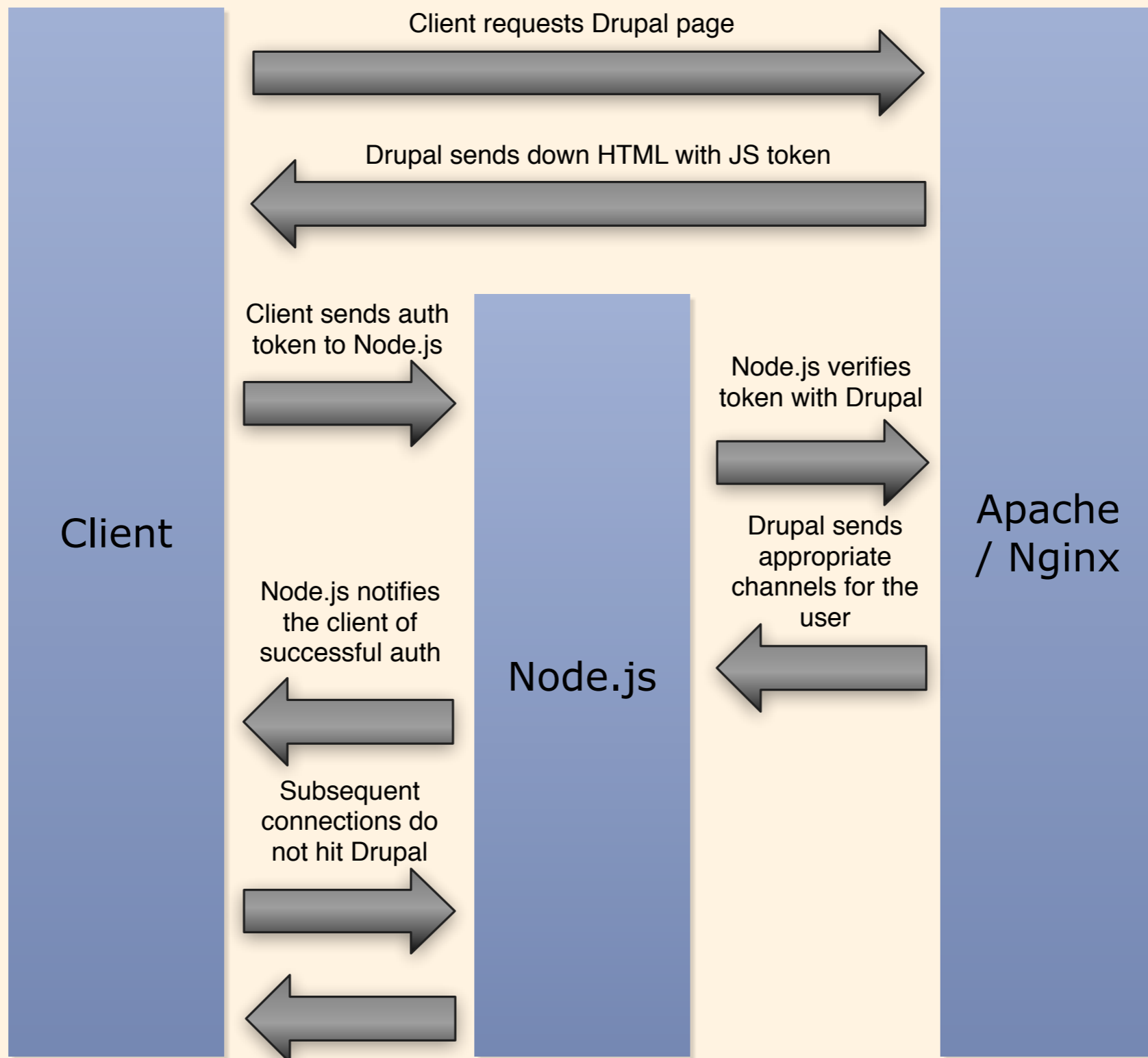
# Put another way



# Put another way



# Put another way



# Deploying

So you actually want to run this thing?

# Supervisor

# Supervisord

- A python thing

# Supervisord

- A python thing
- Not node-supervisor



# Supervisord

- A python thing
- Not node-supervisor
- Daemonize anything

# Supervisord

- A python thing
- Not node-supervisor
- Daemonize anything
- Stable, easy to configure, easy to display

```
1 [program:nodejs]
2 command=/usr/bin/node server.js
3 user=nodejs
4 directory=/var/lib/nodejs
5 autostart=true
6 autorestart=true
7
8 stdout_logfile=/var/log/nodejs/nodejs.log
9 stderr_logfile=/var/log/nodejs/nodejs-errors.log
10 stdout_logfile_maxbytes=104857600
11 stderr_logfile_maxbytes=104857600
```

# Node.js Awesomeness

Drupal's not the only one with nice modules

# NPM

- Node Package Manager
- NPM has over 17,000 modules
- Drupal has 18,922
- <http://npmjs.org/>

# Socket.io



# Socket.io

- <http://socket.io/>



# Socket.io

- <http://socket.io/>
- OMG, socket.io.





# Socket.io

- <http://socket.io/>
- OMG, socket.io.
- Powers the Drupal node.js module



# Server

```
var io = require('socket.io').listen(80);

io.sockets.on('connection', function (socket) {
  socket.emit('news', { hello: 'world' });
  socket.on('my other event', function (data) {
    console.log(data);
  });
});
```

# Client

```
<script src="/socket.io/socket.io.js"></script>
<script>
  var socket = io.connect('http://localhost');
  socket.on('news', function (data) {
    console.log(data);
    socket.emit('my other event', { my: 'data' });
  });
</script>
```

# async

# async

- Not actually node.js, just javascript

# async

- Not actually node.js, just javascript
- <https://github.com/caolan/async>

# async

- Not actually node.js, just javascript
- <https://github.com/caolan/async>
- Insanely useful for things like API requests

```
1  async = require('async');
2  loadedRecords = [];
3  asyncGetRecords = function(id, callback) {
4      loadById(id, function(error, record) {
5          loadedRecords.push(record);
6          // Error should be null on success.
7          callback(error);
8      });
9  }
10 async.forEachLimit([0, 1, 2, 3], 2, asyncGetRecords, function(error) {
11     console.log(loadedRecords);
12 });
13
```

# Coffeescript



# Coffeescript

- <http://coffeescript.org/>

# Coffeescript

- <http://coffeescript.org/>
- It's like javascript, python and ruby had a beautiful baby

# Coffeescript

- <http://coffeescript.org/>
- It's like javascript, python and ruby had a beautiful baby
- That make you kind of hate debugging

```
1 someClosure = (someArg, someCallback) ->
2   someCallback "#{someArg} is awesome."
3
4 callingCode = (project) ->
5   someClosure project, (someString) ->
6     console.log someString
7
8 callingCode('drupal')
9 callingCode('node.js')
10
```

```
// Generated by CoffeeScript 1.3.3
(function() {
  var callingCode, someClosure;

  someClosure = function(someArg, someCallback) {
    return someCallback("" + someArg + " is awesome.");
  };

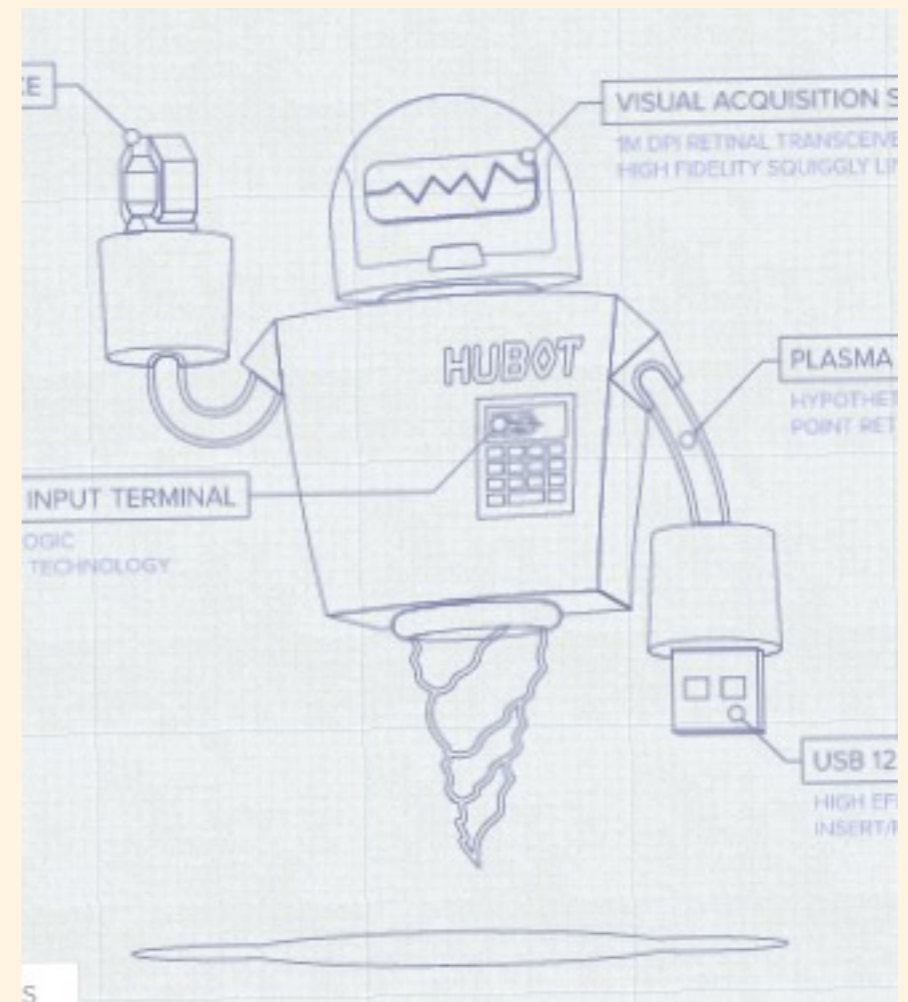
  callingCode = function(project) {
    return someClosure(project, function(someString) {
      return console.log(someString);
    });
  };

  callingCode('drupal');

  callingCode('node.js');

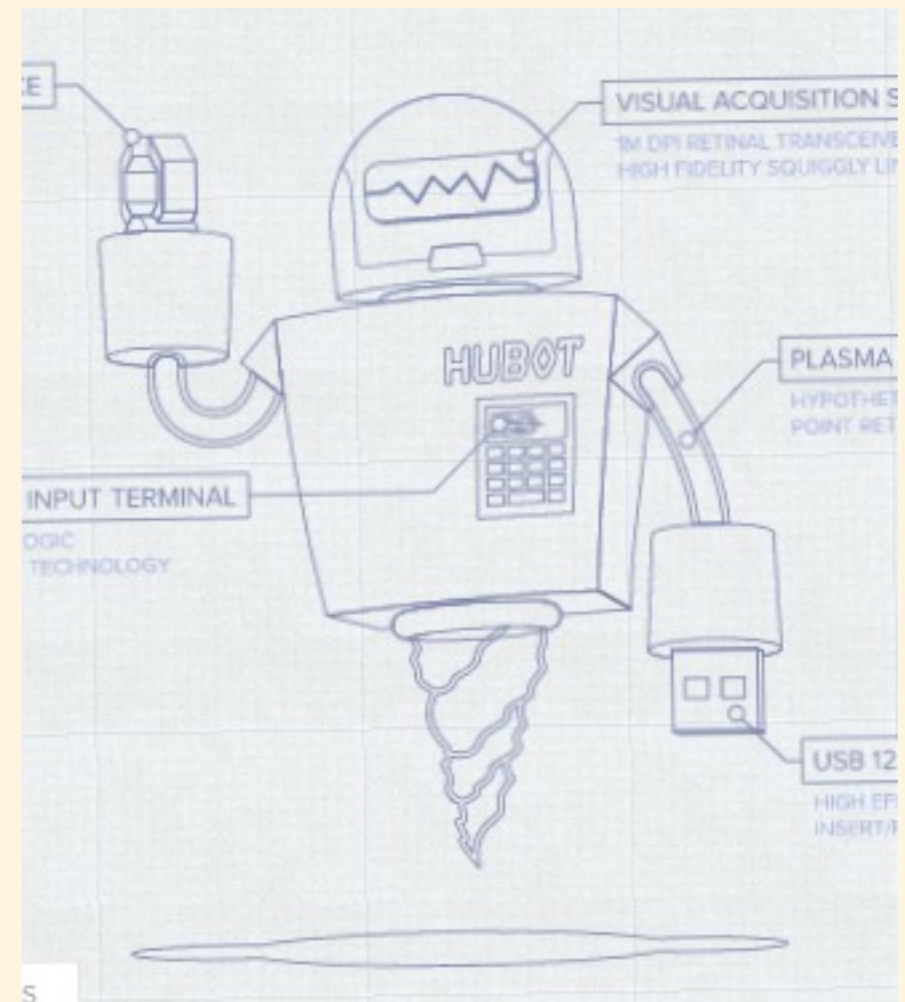
}).call(this);
```

# Hubot



# Hubot

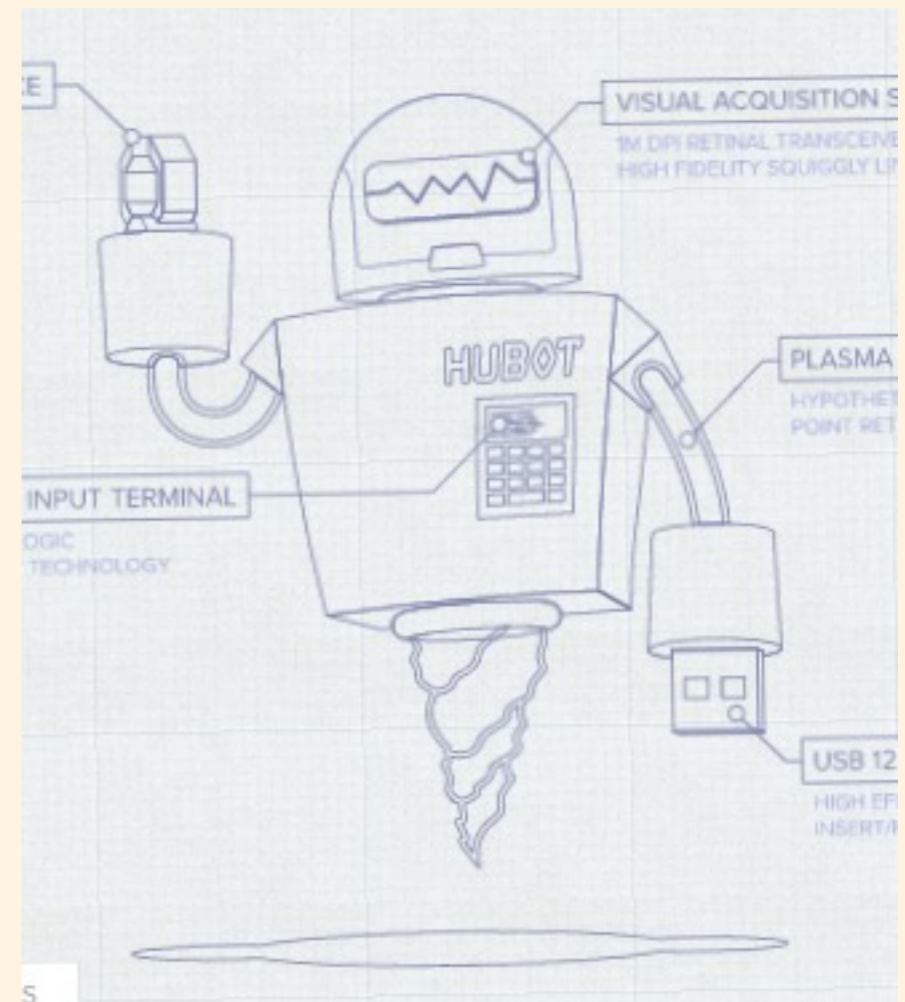
- Protocol agnostic chat bot





# Hubot

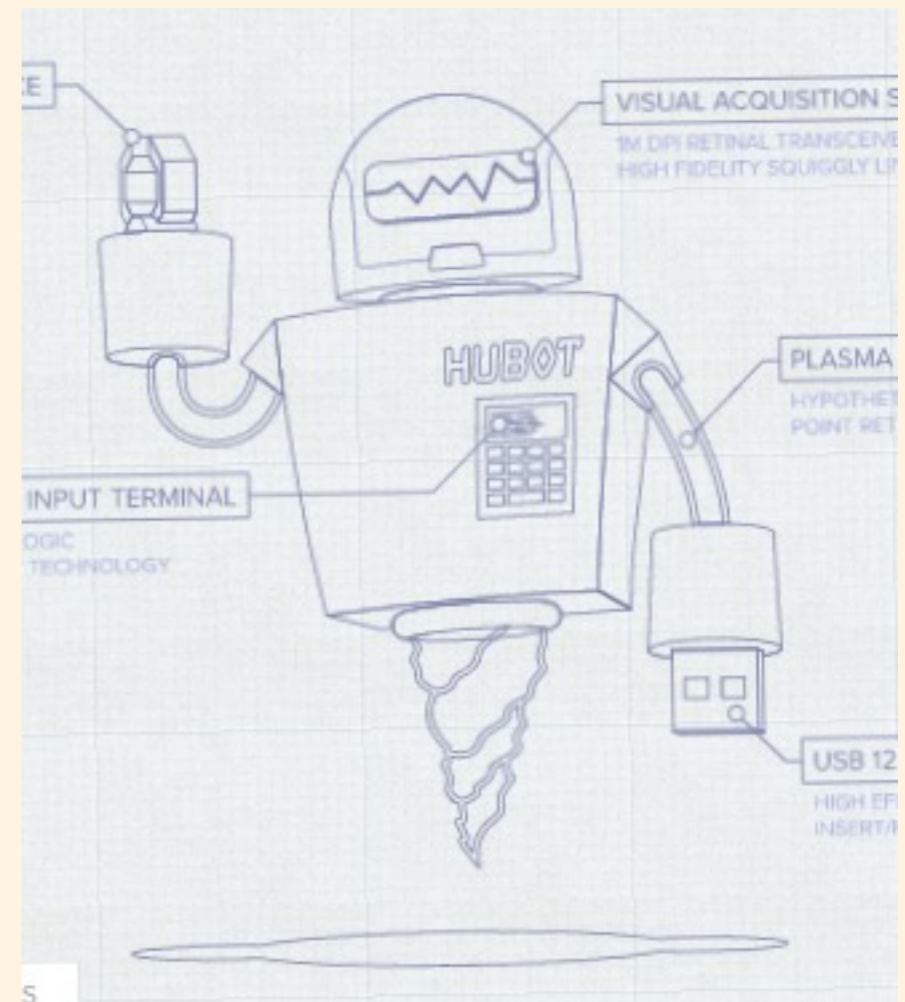
- Protocol agnostic chat bot
- Easy to extend!





# Hubot

- Protocol agnostic chat bot
- Easy to extend!
- Out of the box integration with Github, Jenkins, Jira, Redmine, etc.





```
3 # dude me - Grab a random quote from the most recent page of the http://thebiglebowski.t
4 module.exports = (robot) ->
5
6   robot.respond /(dude me)/i, (msg) ->
7     request = require('request')
8     FeedParser = require('feedparser')
9     parser = new FeedParser()
10
11    request.get 'http://thebiglebowski.tumblr.com/rss', (error, response, body) ->
12      quotes = []
13      parser.parseString body, (error, meta, articles) ->
14        article = msg.random articles
15        msg.send article.description
```

# Web Frameworks

# Web Frameworks

- Express (<http://expressjs.com/>)

# Web Frameworks

- Express (<http://expressjs.com/>)
- Flatiron (<http://flatironjs.com/>)

# Web Frameworks

- Express (<http://expressjs.com/>)
- Flatiron (<http://flatironjs.com/>)
- Getty (<http://geddyjs.org/>)

# Web Frameworks

- Express (<http://expressjs.com/>)
- Flatiron (<http://flatironjs.com/>)
- Getty (<http://geddyjs.org/>)
- Meteor (<http://meteor.com/>)

# Web Frameworks

- Express (<http://expressjs.com/>)
- Flatiron (<http://flatironjs.com/>)
- Getty (<http://geddyjs.org/>)
- Meteor (<http://meteor.com/>)
- So many more...



# Supervisor

# Supervisor

- <https://github.com/isaacs/node-supervisor>

# Supervisor

- <https://github.com/isaacs/node-supervisor>
- Hot code reloading (save a file, restart the server)