



## **Node.js and Drupal**

# Justin Randell

- Drupal.org, IRC, twitter: beejeebus

# What

---

What is this Node.js, anyway?

# Node.js

- Javascript on the server
- Based on Google's V8 engine
- Event driven
- Async I/O operations by default
- Lots of people love it, some don't:  
<http://tedddziuba.com/2011/10/node-js-is-cancer.html>



# Node.js

- Single-process, multi-threaded
- Main event loop is a single thread
- Easy to extend with C++
- Socket.IO

# Why

---

Why is this helpful?

# Apache's shortcomings

- One thread (usually one process) per connection
- Memory heavy processes (for Drupal anyway)
- Persistent connections are massive fail

# Node.js is good for...

- Concurrency!
- Asynchronous I/O
- Low overhead for maintaining open connections
- Networking applications



# How

---

How do we use Node.js with Drupal?

# Node.js module

- <http://drupal.org/project/nodejs>
- Provides a Node.js server script and a Drupal module
- All the plumbing

# Authentication

3. Node.js sends authentication token to Drupal

4. Drupal returns a list of channels for the user

nodeJS



2. Connection requested by socket.io using auth token

5. Node.js notifies client that auth is successful

1. Normal Drupal Page Load (with auth token)





# Subsequent page loads

nodeJS



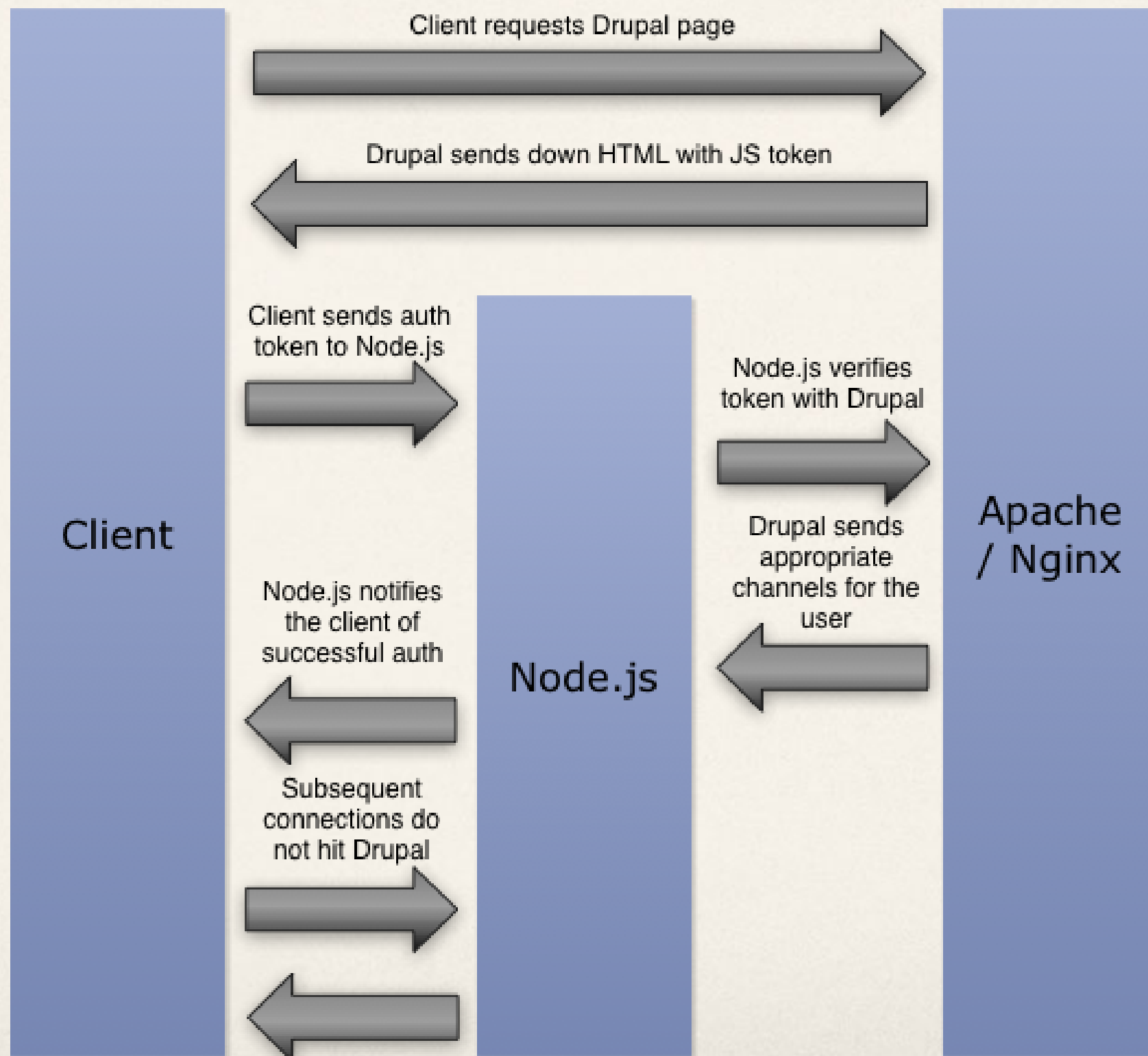
2. Node.js remembers the authenticated token, authenticating the user without a need to query Drupal again

1. Normal Drupal Page Load (with auth token)





# Put another way



# Where

---

How can my module integrate with node.js?

# PHP

```
$message = (object) array(  
    'channel' => 'example',  
    'data' => array(  
        'key' => 'value',  
        'some other key' => 'some other value',  
    ),  
);  
nodejs_enqueue_message($message);
```

# JavaScript

```
Drupal.Nodejs.callbacks.example = {  
  callback: function (message) {  
    if (message.channel = 'example') {  
      alert(message.data);  
    }  
  }  
};
```



# Add a user to a channel

```
/**  
 * Implements hook_nodejs_user_channels().  
 */  
function example_nodejs_user_channels($auth_user) {  
  $channels = array();  
  $channels[] = 'example_channel';  
  return $channels;  
}
```

# Setting and using a content channel

```
nodejs_send_content_channel_token('watchdog_dblog');

$commands[] = ajax_command_before('#admin-dblog tr:eq(1)', nodejs_watchdog_theme_row($row));
$message = (object) array(
    'channel' => 'watchdog_dblog',
    'commands' => $commands,
    'callback' => 'nodejsNodeAjax',
);
nodejs_send_content_channel_message($message);
```

# When

---

Are we getting new features?

# What's coming

- Version 1.0 is just around the corner, we're on Beta7 now
- Chatroom is being ported to rely on Node.js module
- Splitting off Node.js code to github



# New Features

- Views integration
- Generic entity updating
- Your module here...

# Demo

---

Let's see it in action!