

What's new in Rails 5

Turbolinks 3.0

ActionCable

# Turbolinks 3.0

navigating web application faster

# How Turbolinks works

1. intercepts all clicks to links to the same domain
2. changes the browser's URL using the History API
3. requests the new page using XHR
4. replaces the current `<body>` element and merges the contents of the `<head>` element

# Turbolinks 3.0

Partial Replacement

# Turbolinks Example

# ActionCable

full-stack, real-time, publish-subscribe framework for  
Rails 5

# Usage

- Chat
- Multiplayer Games
- Updates
- Feeds
- Collaborative Editing and Coding
- etc.



# WebSocket

- TCP-based protocol
- Persistent (long-lived)
- Bi-directional (full-duplex)
- Designed to be implemented in web browsers and web servers
- Supports encrypted communication
- Introduces URI schemes: ws, wss

# WebSocket: Standards

Standardised in 2011

IETF: RFC 6455

**The WebSocket Protocol**

W3C Web IDL

**The WebSocket API**

IE	Edge <sup>*</sup>	Firefox	Chrome	Safari	Opera	iOS Safari <sup>*</sup>	Opera Mini <sup>*</sup>	Android Browser <sup>*</sup>	Chrome for Android
			29						
			45					4.3	
8			48			8.4		4.4	
9		45	49	9	36	9.2		4.4.4	
11	13	46	50	9.1	37	9.3	8	50	50
	14	47	51	TP	38				
		48	52		39				
		49	53						

can-i-use-it

Yes!



# WebSocket: Handshake

protocol switch from HTTP to WebSocket

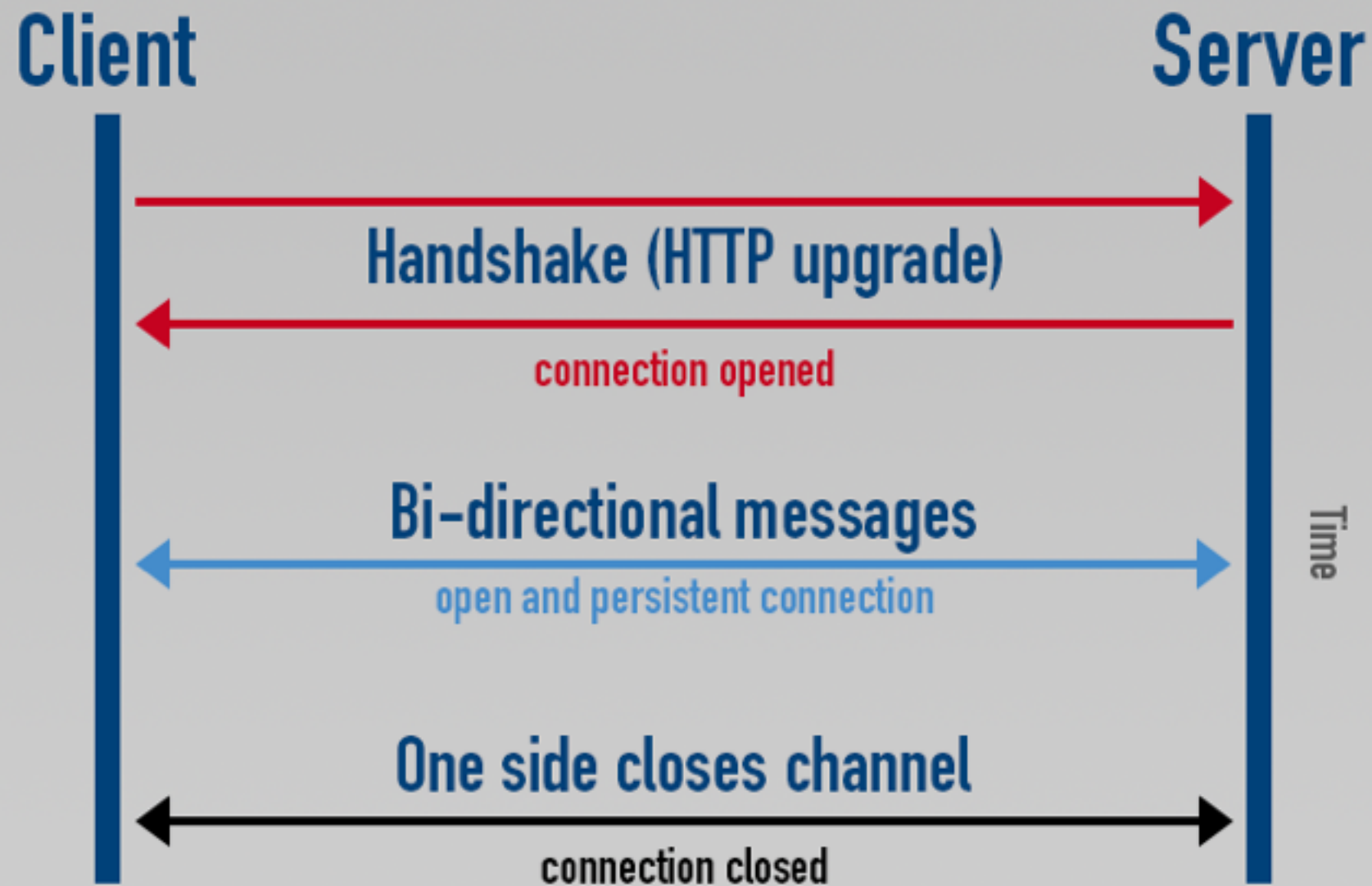
# WebSocket: Handshake Example

## CLIENT REQUEST

GET /cable HTTP/1.1  
Host: zeroeleven.rs:3000  
Cookie: session=N3Yyc0Z0cVdm  
Upgrade: websocket  
Connection: Upgrade  
Sec-WebSocket-Key:  
dGh1IHNhbXBsZSBub25jZQ==  
Sec-WebSocket-Version: 13

## SERVER RESPONSE

HTTP/1.1 101 Switching Protocols  
Upgrade: websocket  
Connection: Upgrade  
Sec-WebSocket-Accept:  
s3pPLMBiTxaQ9kYGzzhZRbK+x0o=



# WebSocket

Visual Representation

# WebSocket: API

```
var ws = new WebSocket("ws://zeroeleven.rs");

ws.onopen = function(evt) { alert("Connection open..."); };
ws.onmessage = function(evt) { alert("Received Message: " +
    evt.data); };
ws.onclose = function(evt) { alert("Connection closed."); };

ws.send("Hello from Start-it!");
ws.close();
```

# Terminology

Connection

Consumer

Channel

Subscriber

Subscription

Pub/Sub

Broadcasting



# Pub/Sub Adapters

async

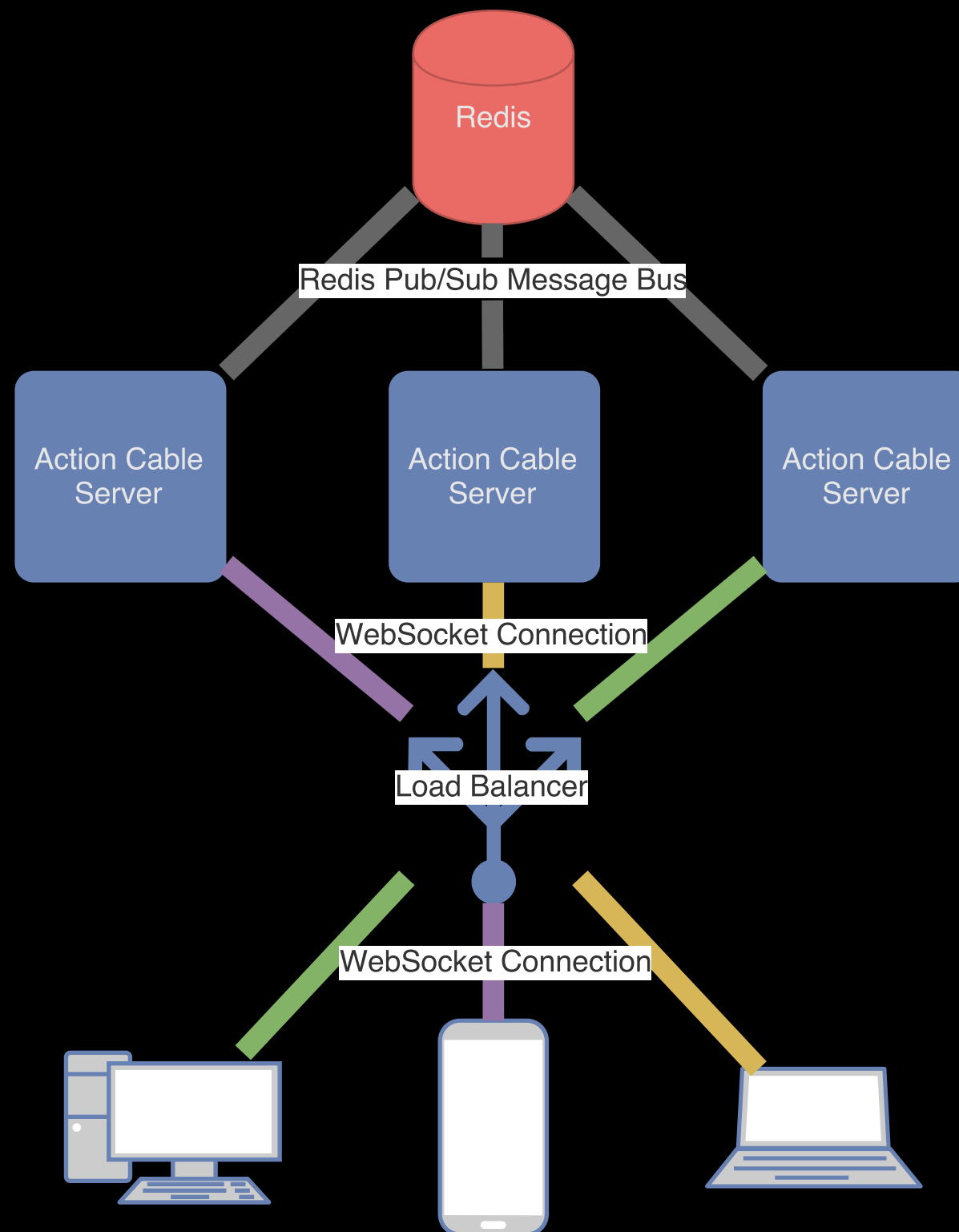
Redis

PostgreSQL

# Running the ActionCable Server

In-App

Standalone



Visualisation of ActionCable in Production

```
$ rails new start-it
```

```
app/  
  channels/  
    application_cable/  
      channel.rb  
      connection.rb
```

```
app/  
  assets/  
    javascripts/  
      cable.js
```

```
# app/channels/application_cable/connection.rb
```

```
module ApplicationCable
  class Connection < ActionCable::Connection::Base
    identified_by :current_user

    def connect
      self.current_user = find_verified_user
    end

    def find_verified_user
      if verified_user = User.find_by(id: cookies.signed[:user_id])
        verified_user
      else
        reject_unauthorized_connection
      end
    end
  end
end
```

```
// app/assets/javascripts/cable.js
```

```
//= require action_cable
```

```
//= require_self
```

```
//= require_tree ./channels
```

```
(function() {
```

```
  this.App || (this.App = {});
```

```
  App.cable = ActionCable.createConsumer();
```

```
}).call(this);
```

\$ rails generate channel room speak



```
app/  
  channels/  
    room_channel.rb
```

```
app/  
  assets/  
    javascripts/  
      channels/  
        room.coffee
```

```
# app/channels/room_channel.rb
```

```
class RoomChannel < ApplicationCable::Channel
  def subscribed
    stream_from "room_channel"
  end

  def unsubscribed
  end

  def speak(data)
    ActionCable.server.broadcast 'room_channel', message:
data['message']
  end
end
```

```
# app/assets/javascripts/channels/room_channel.js
```

```
App.room = App.cable.subscriptions.create "RoomChannel",  
  connected: ->
```

```
    disconnected: ->
```

```
    received: (data) ->  
      $("#messages").append data["message"];
```

```
    speak: (message) ->  
      @perform "speak", message: message
```

```
App.room.speak "SIR, YES SIR!"
```



# ActionCable Example