

EWD.js

The Future Starts Here

Rob Tweed

Twitter: @rtweed

rob.tweed@gmail.com

<http://www.mgateway.com>



Copyright © 2015 M/Gateway Developments Ltd

How EWD.js tends to be viewed



- Just one of many possible technical options
- Tomorrow will bring another shiny technology
- Rob's hobby horse

What if...



- EWD.js was adopted at the VA?
- What's the view like from up there?

EWD.js: the bigger picture



- What does it make possible?
- A glimpse of the vision behind EWD.js

What is EWD.js

- Development framework
- Run-time platform

<http://ewdjs.com>



Copyright © 2015 M/Gateway Developments Ltd

What is EWD.js

- Development framework
 - Client-server applications in the browser
 - Web Services
 - Legacy Mumps application modernisation
- Run-time platform

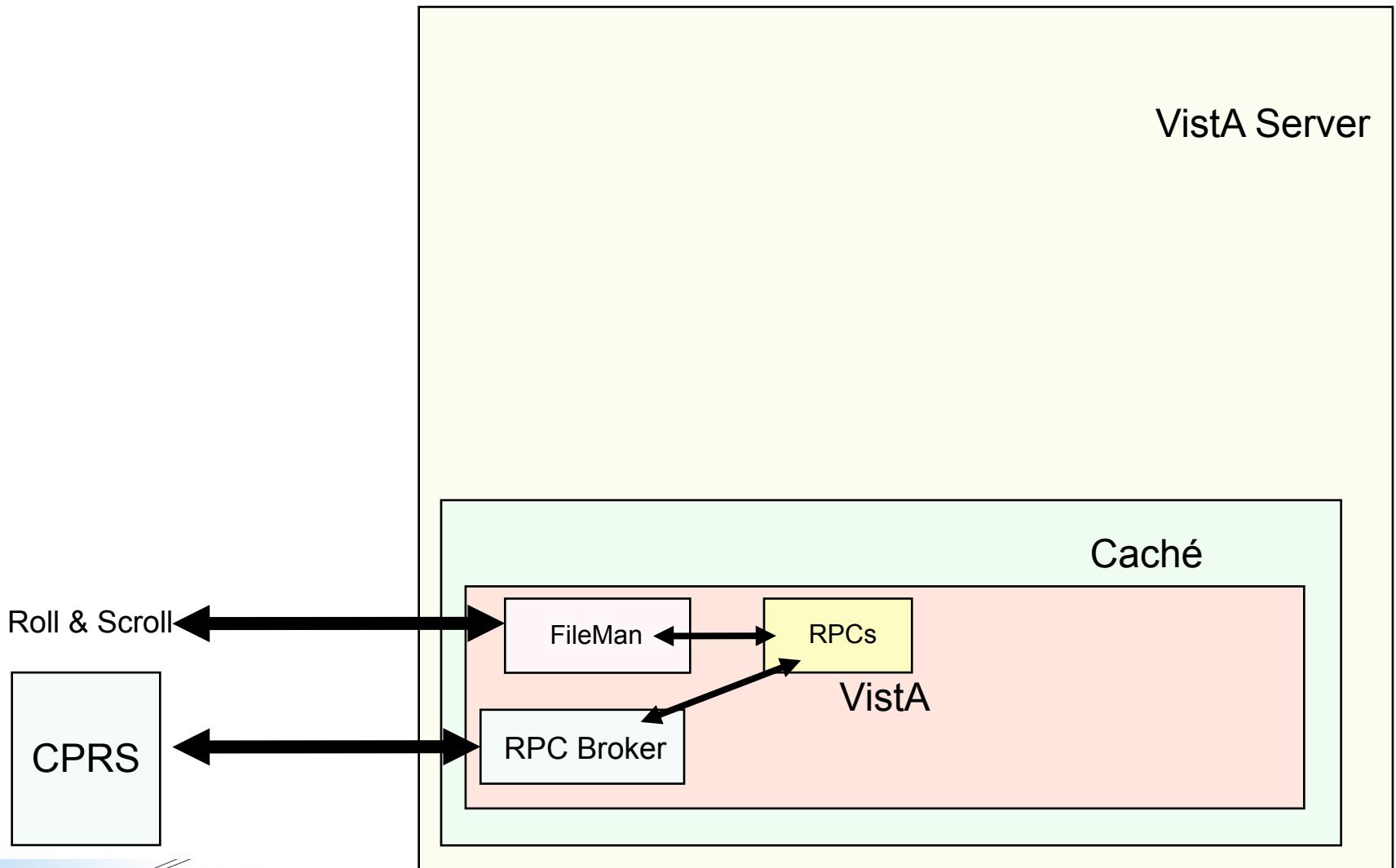
What is EWD.js

- Development framework
- Run-time platform
 - Node.js-based application server
 - Web Server
 - Integrated JSON database
 - Caché, GT.M, GlobalsDB, MongoDB

EWD.js key features

- 100% JavaScript
 - Browser
 - Back-end (business logic): Node.js
- 100% JSON-based data
 - Browser
 - Back-end
 - Database
- Legacy Mumps code can be accessed from Node.js
- High performance
 - Node.js
 - In-process interface to database
- Highly scalable
 - stateless

Current VistA



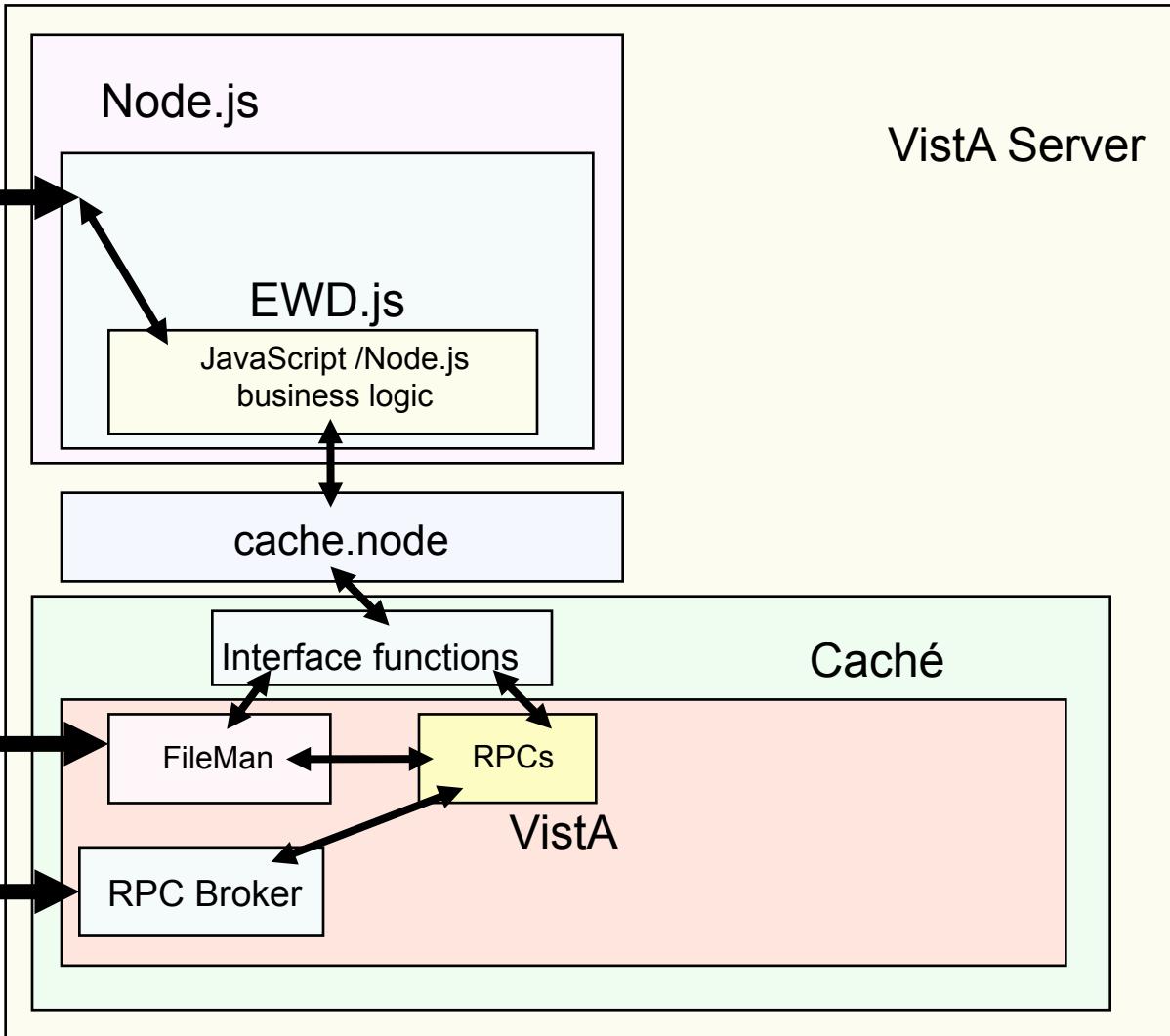
Stateful

EWD.js-enabled VistA

Stateless

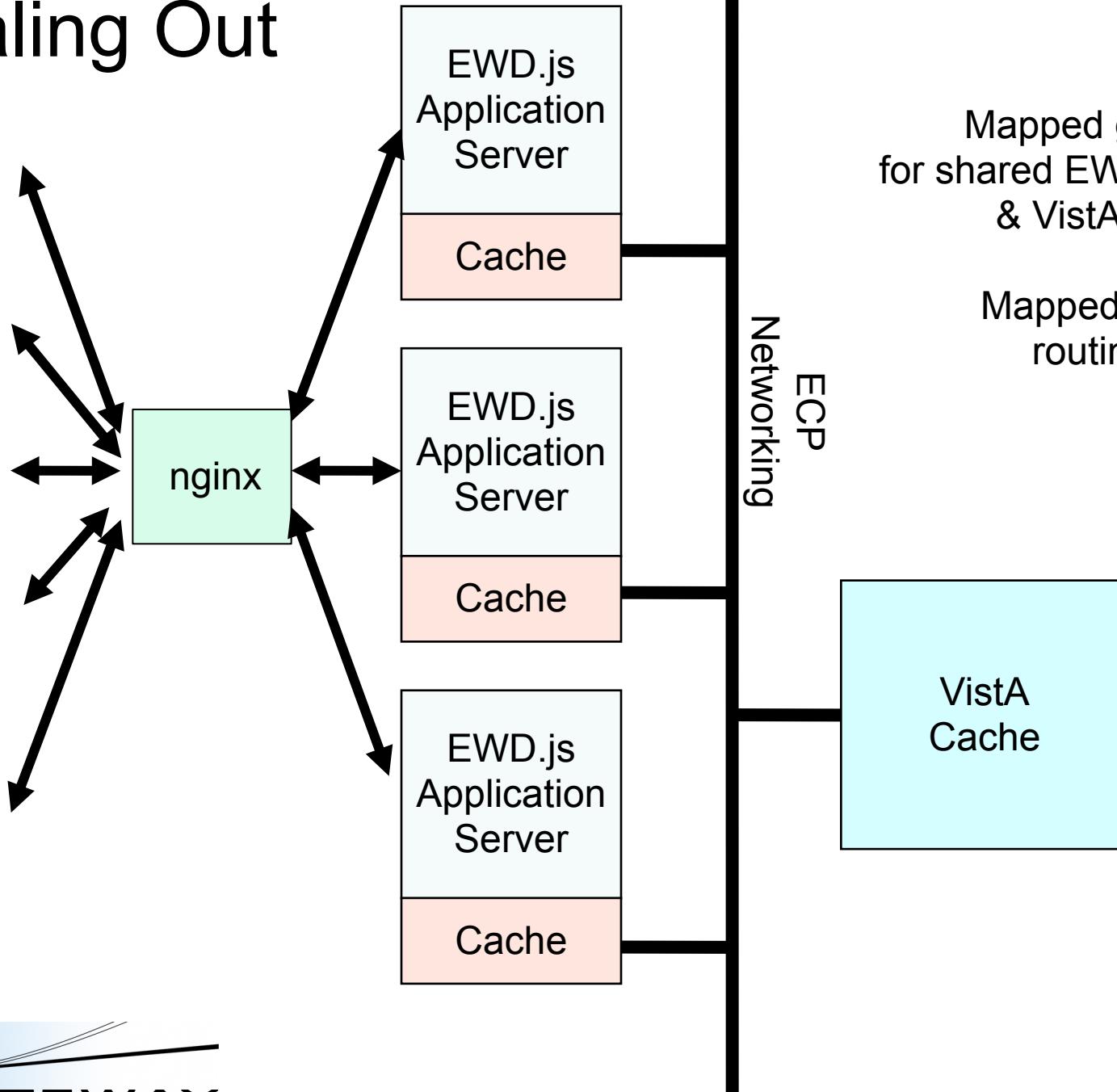
Web
Service
Clients

HTTPS +
HMAC SHA256



Scaling Out

Incoming
Web Service
Traffic



Mapped globals
for shared EWD Sessions
& VistA data

Mapped VistA
routines

Benefits to the VA

- Single language skill
- Controlled migration:
 - from rare Mumps development resources
 - To readily-available JavaScript developers



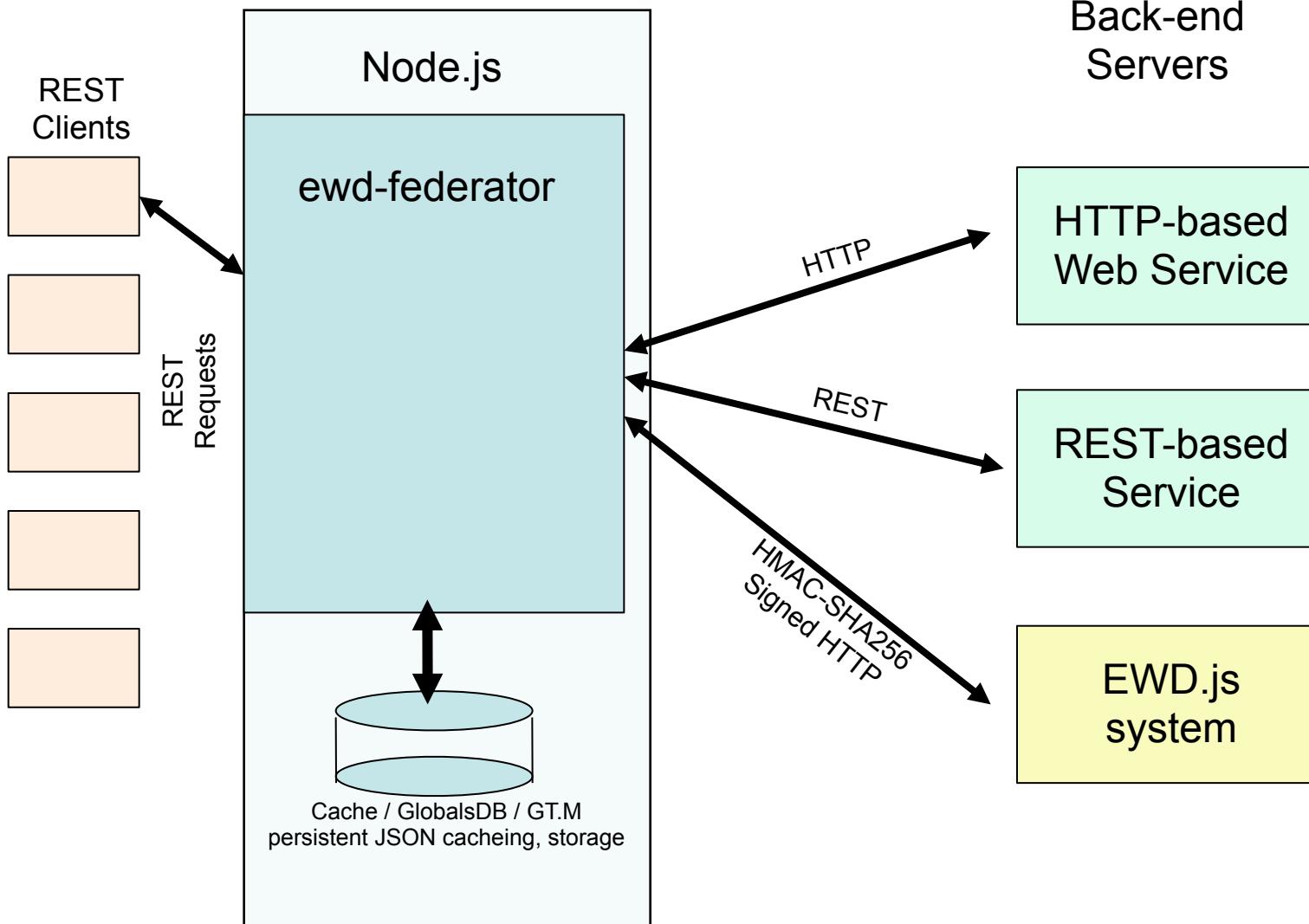
Benefits to the VA

- Single language skill
- Controlled migration:
 - from rare Mumps development resources
 - To readily-available JavaScript developers
 - One-off interfacing of core VistA functionality via JavaScript functions

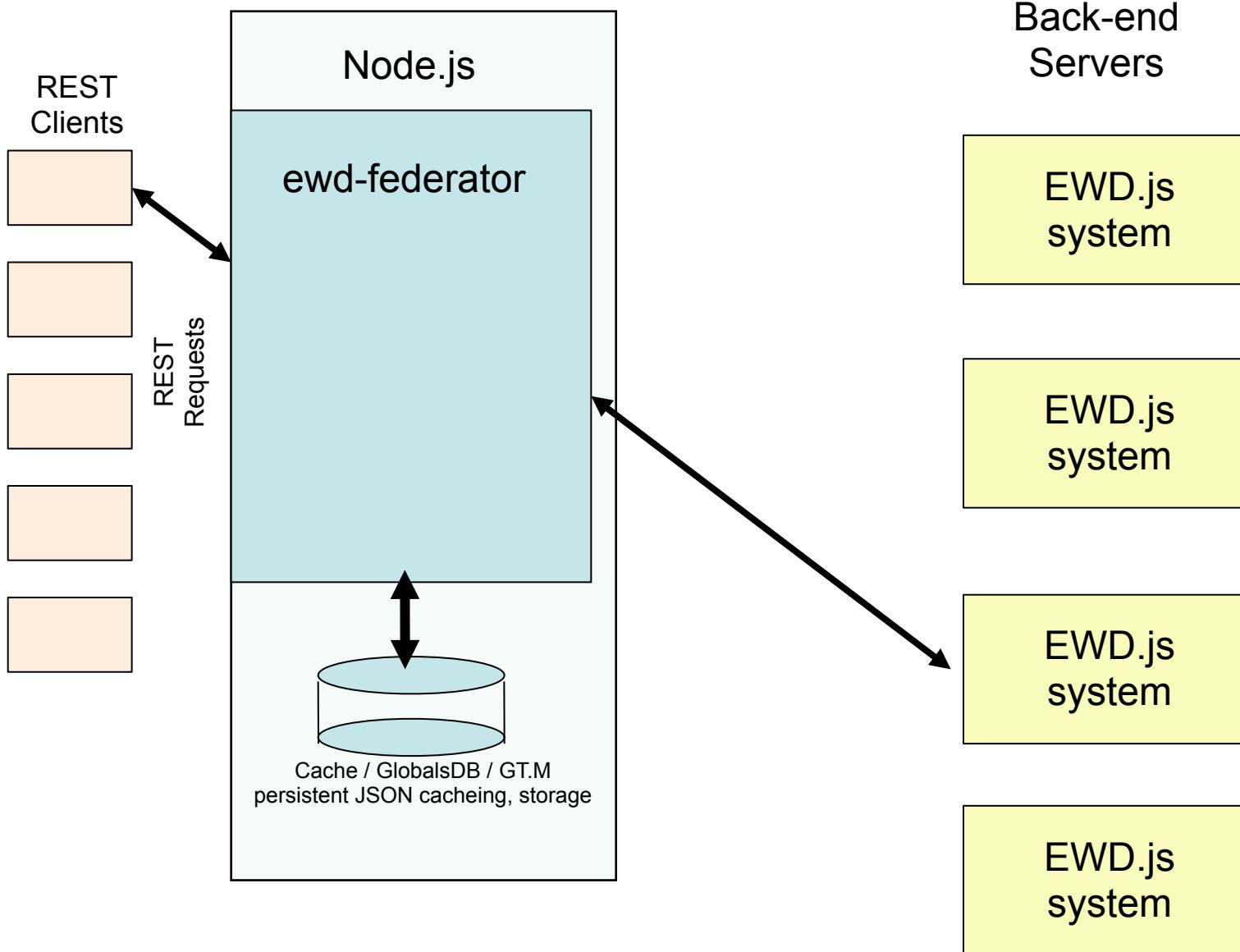
Federated access across the VA?

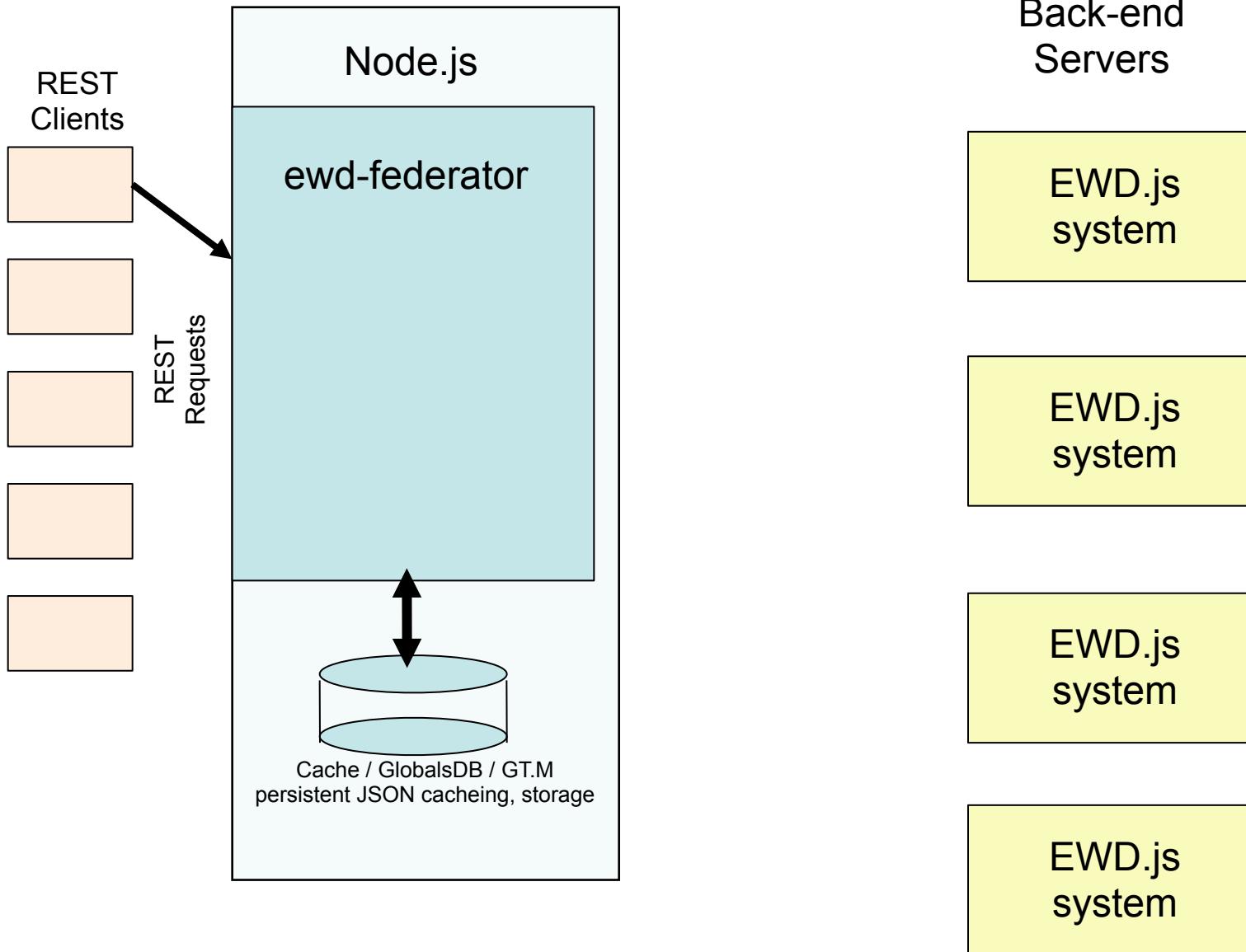


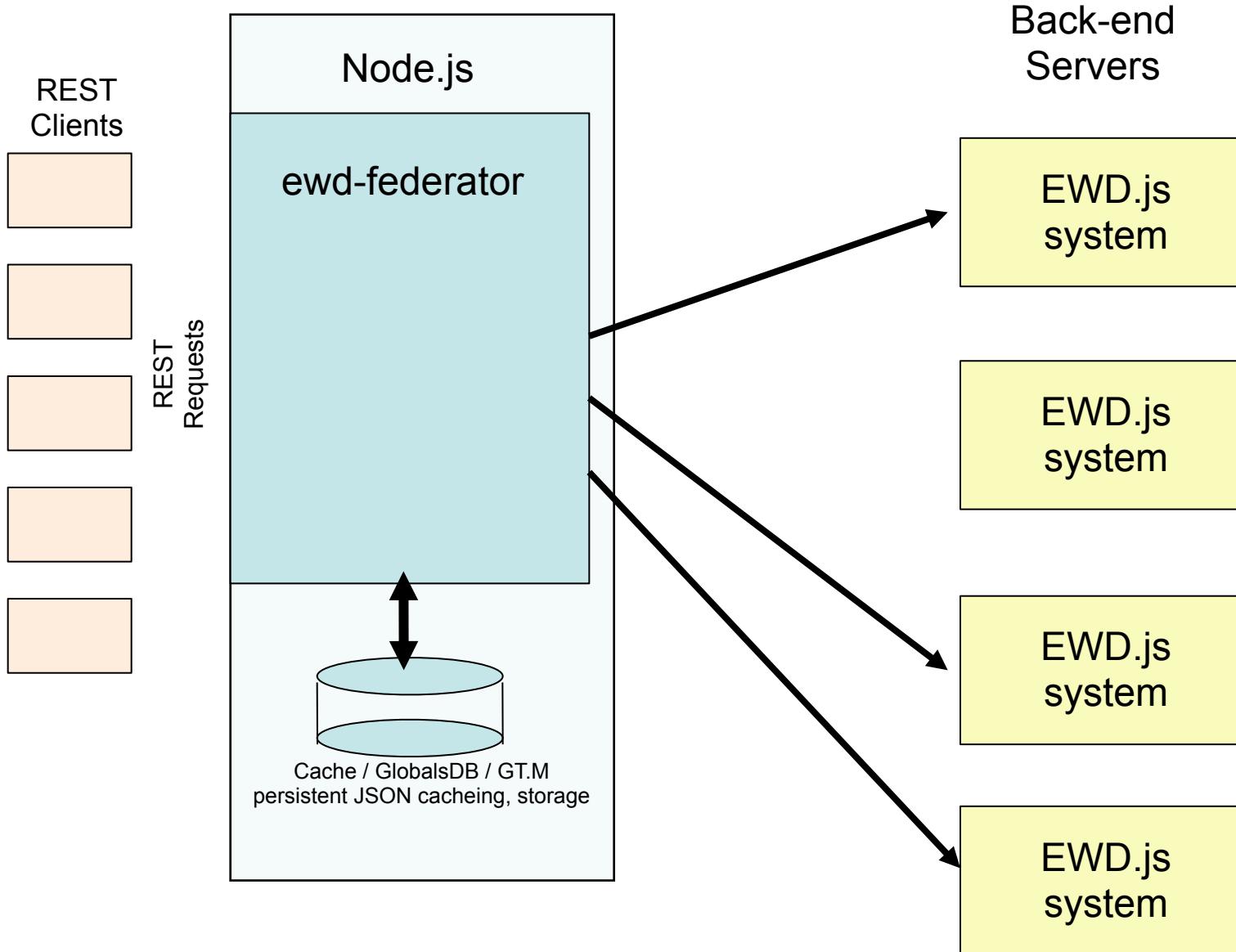
Copyright © 2015 M/Gateway Developments Ltd

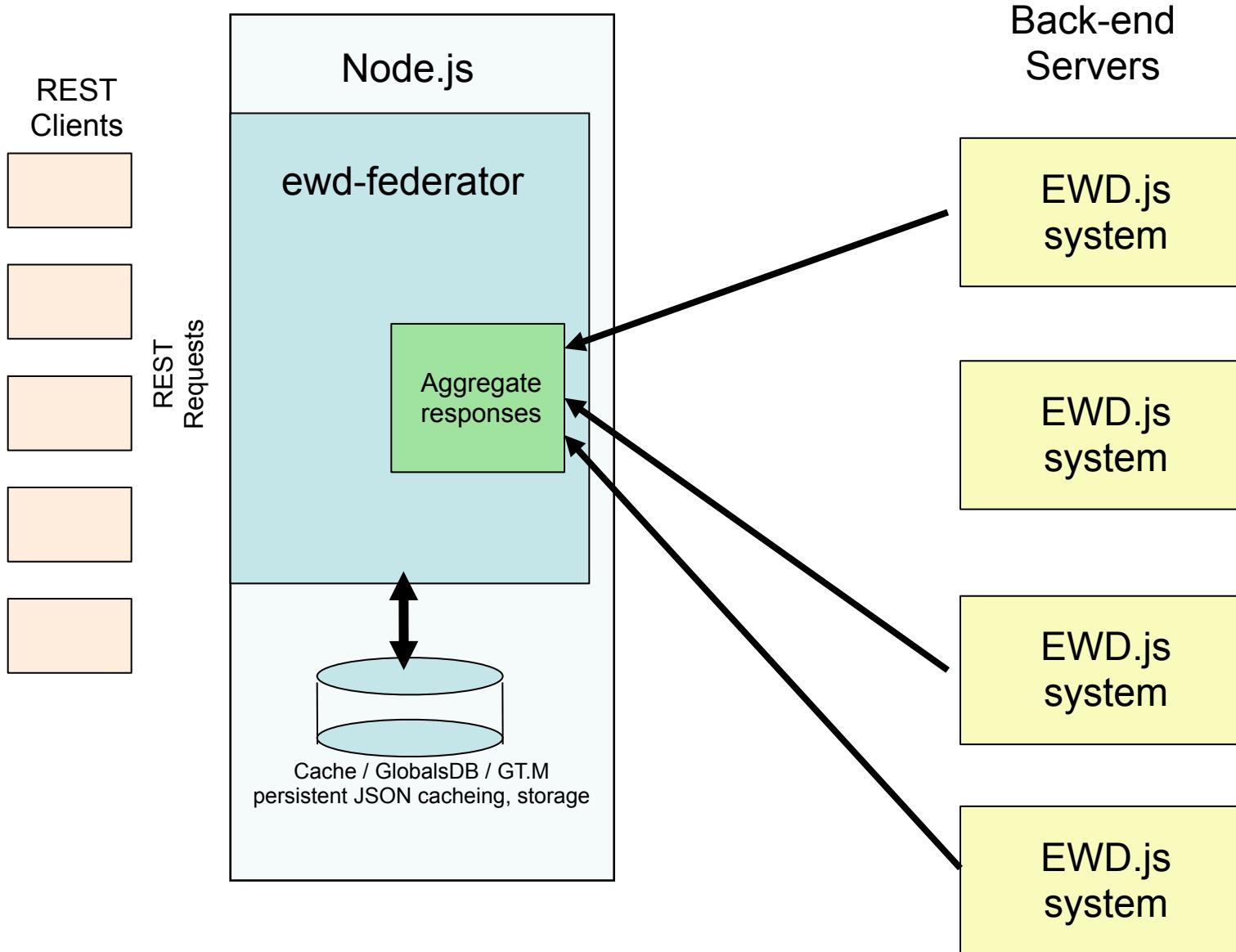


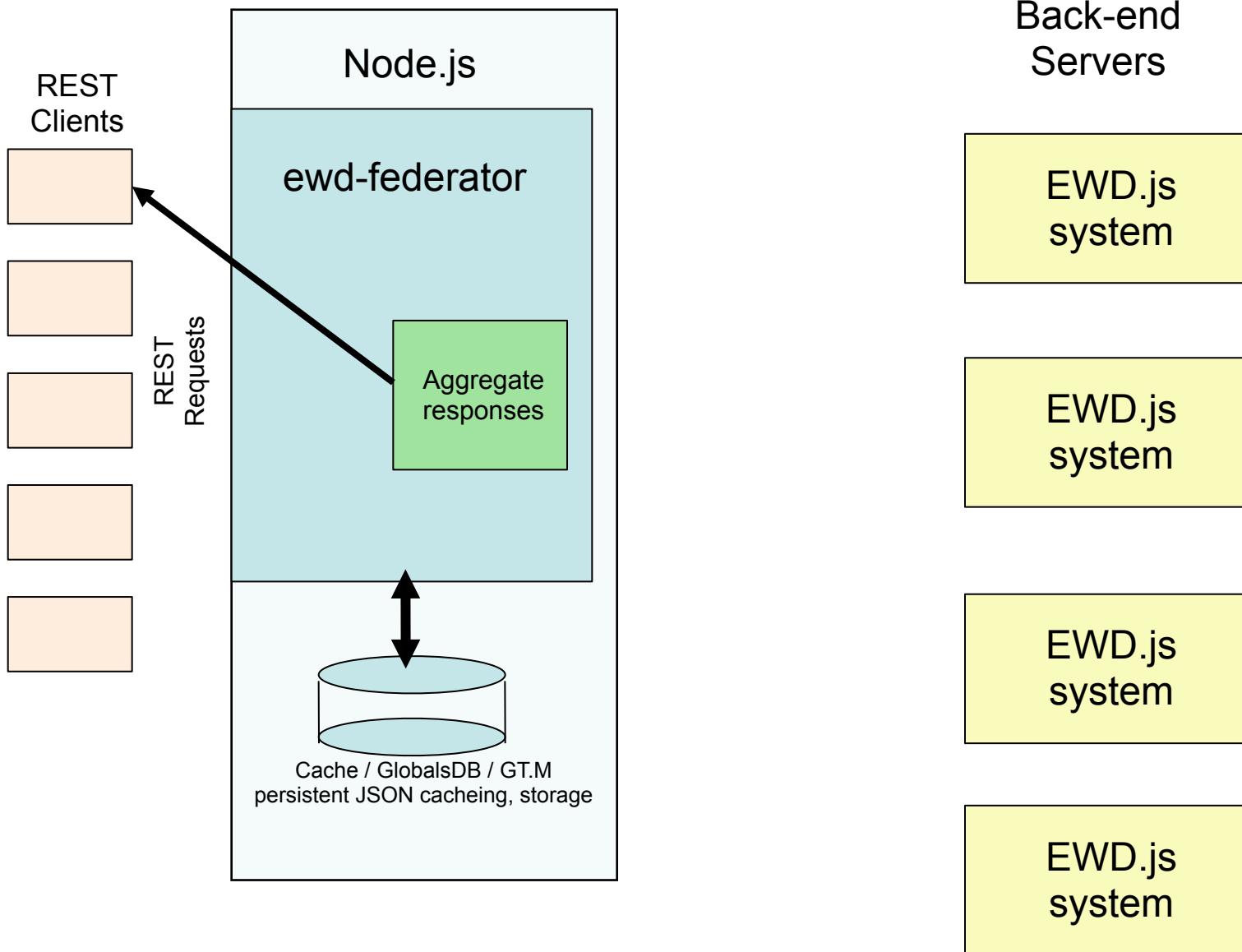
<https://github.com/robtweed/ewd-federator>

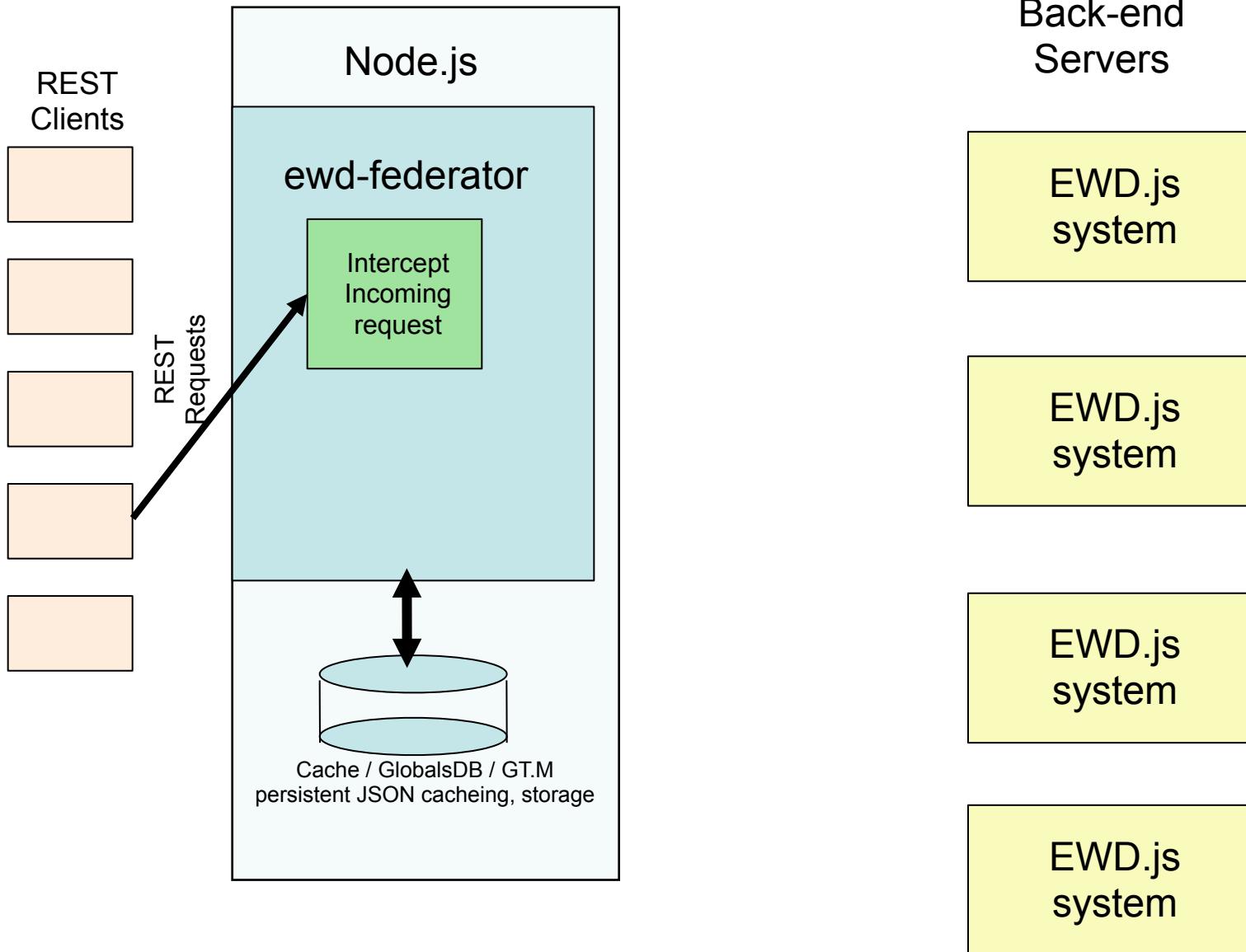


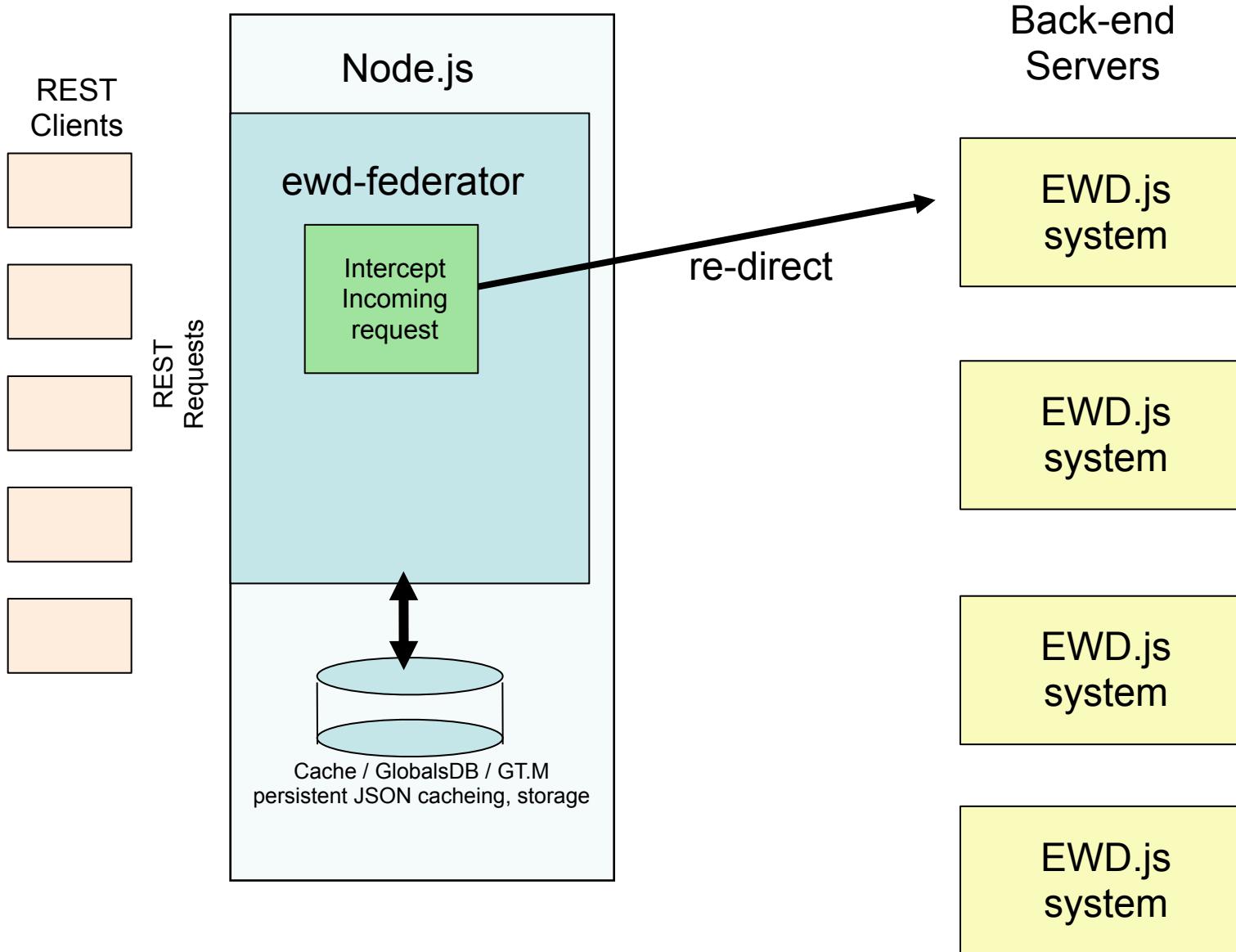


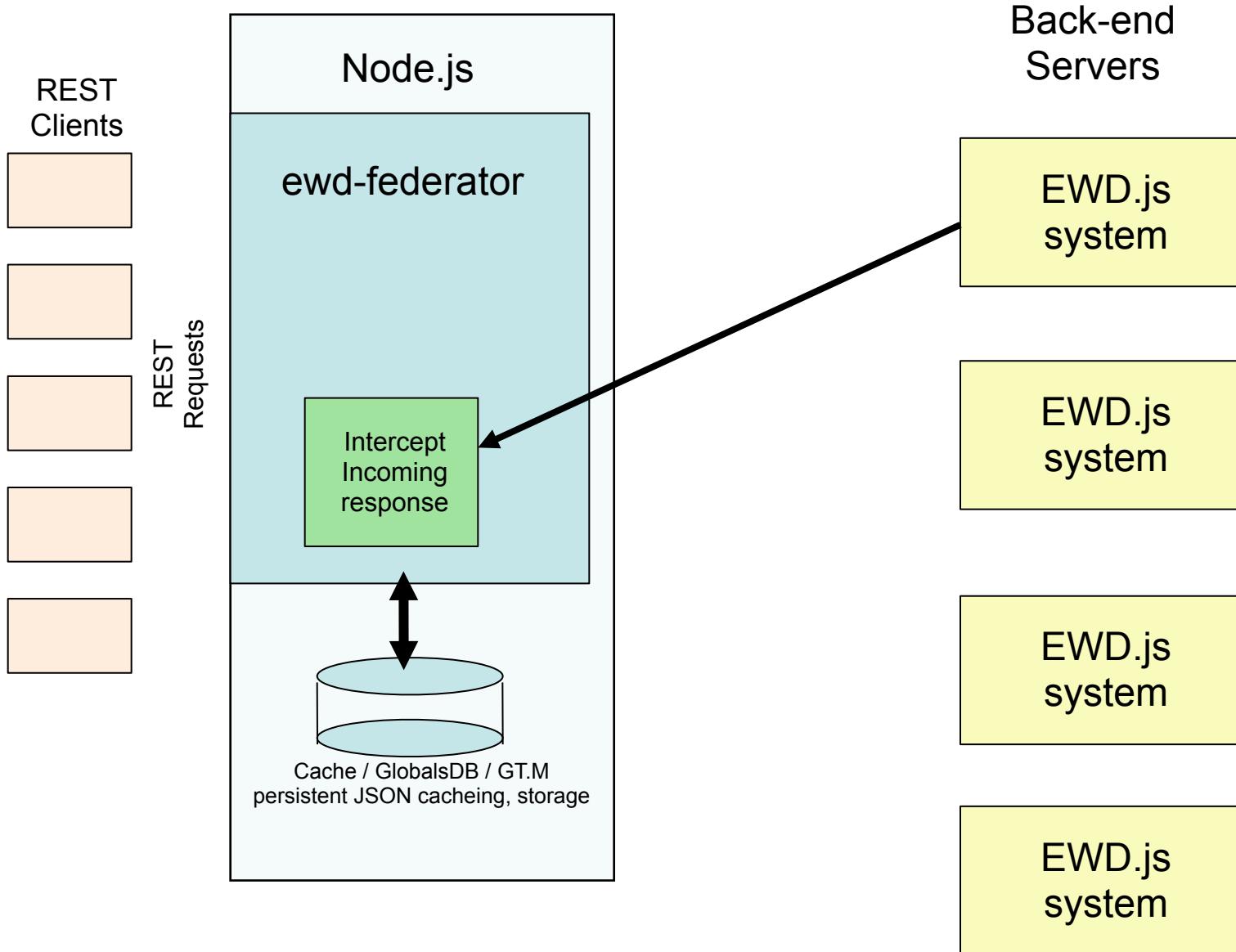


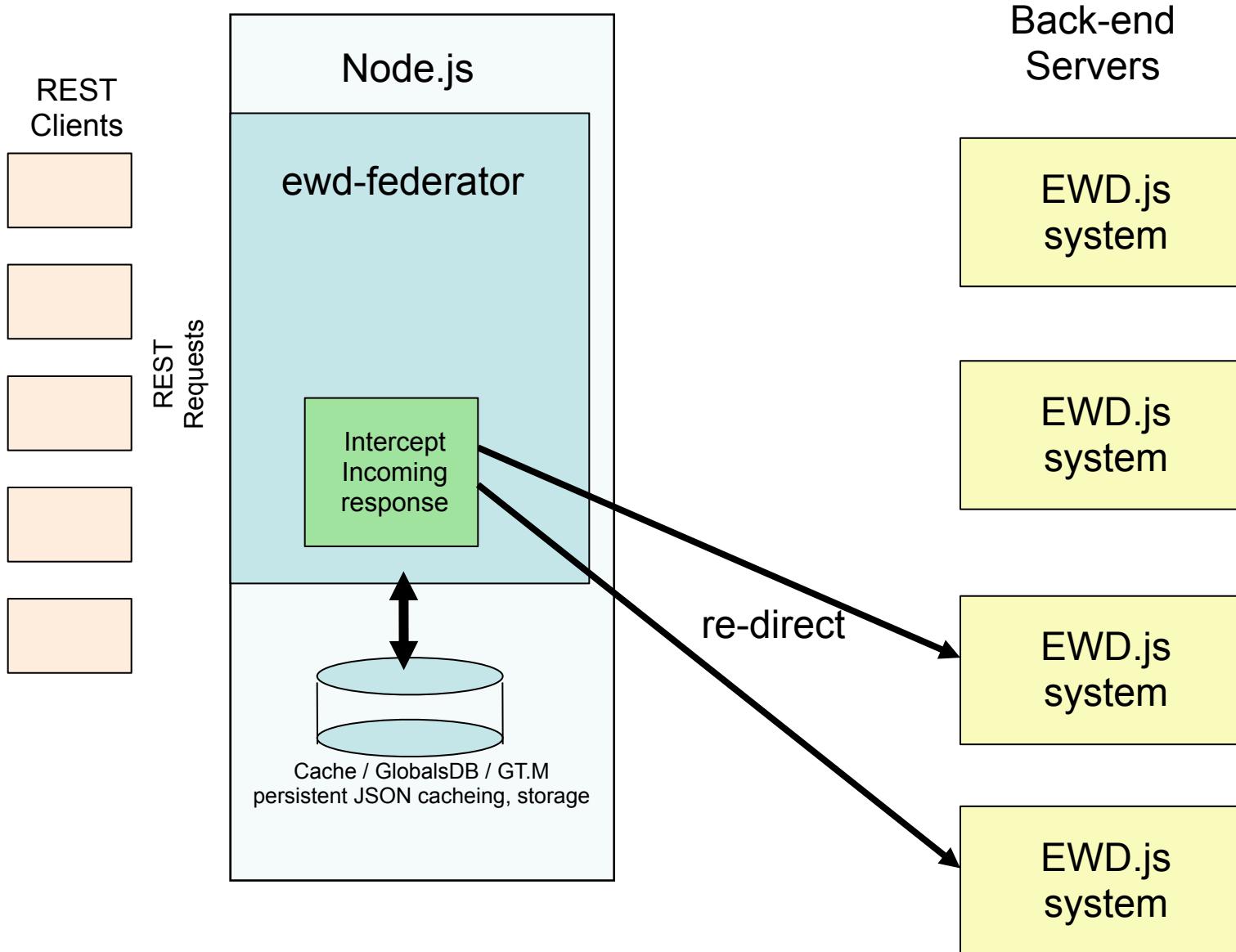


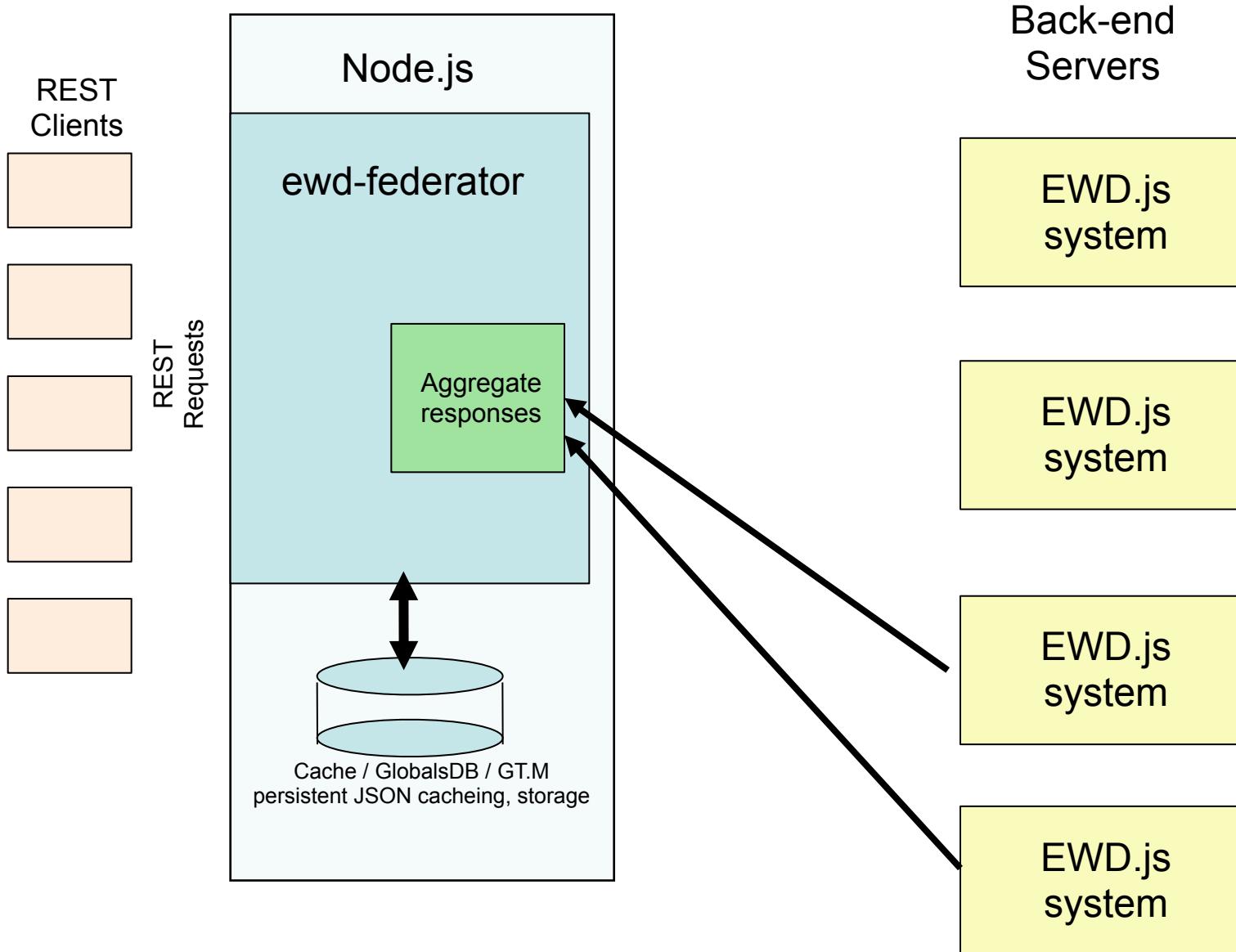


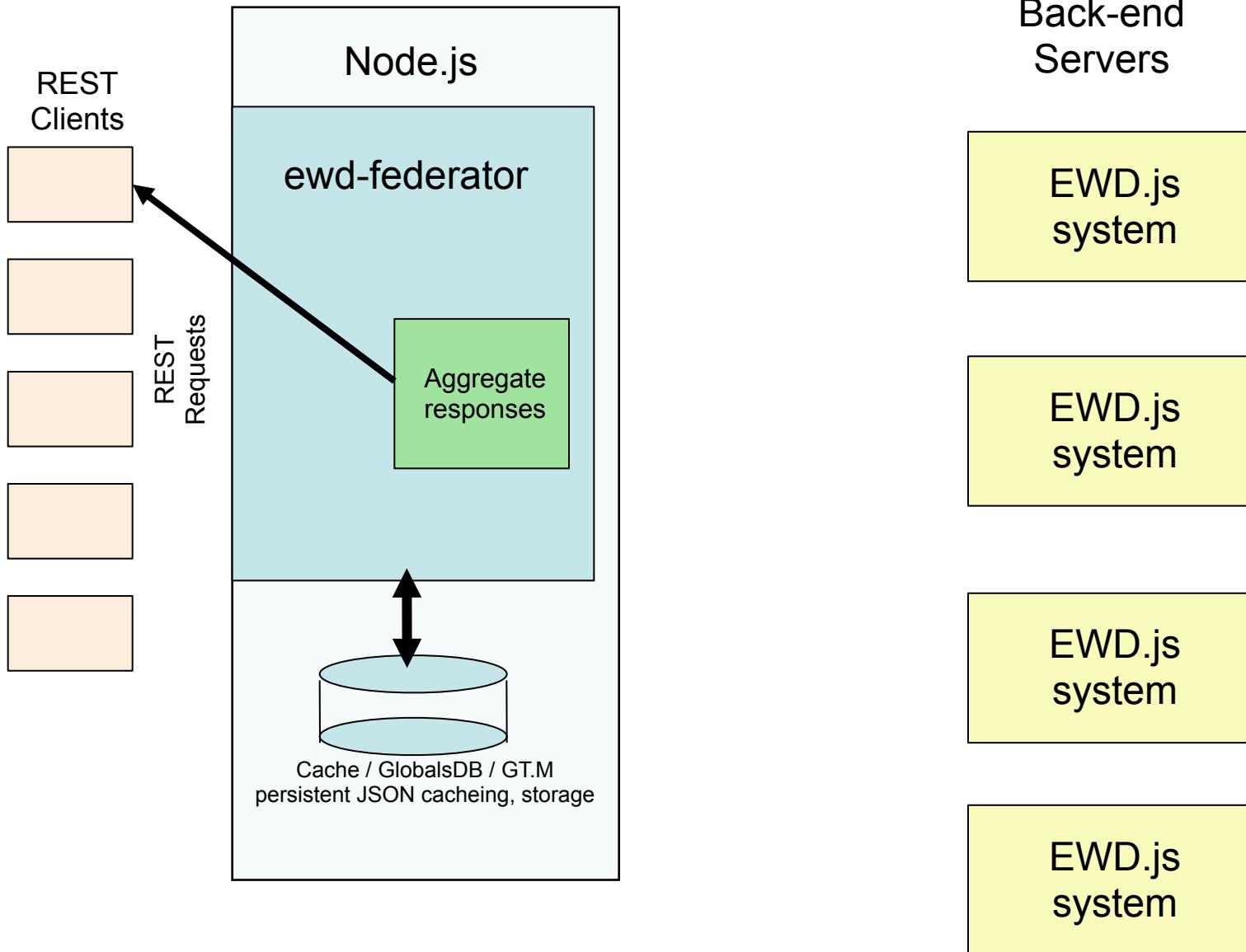












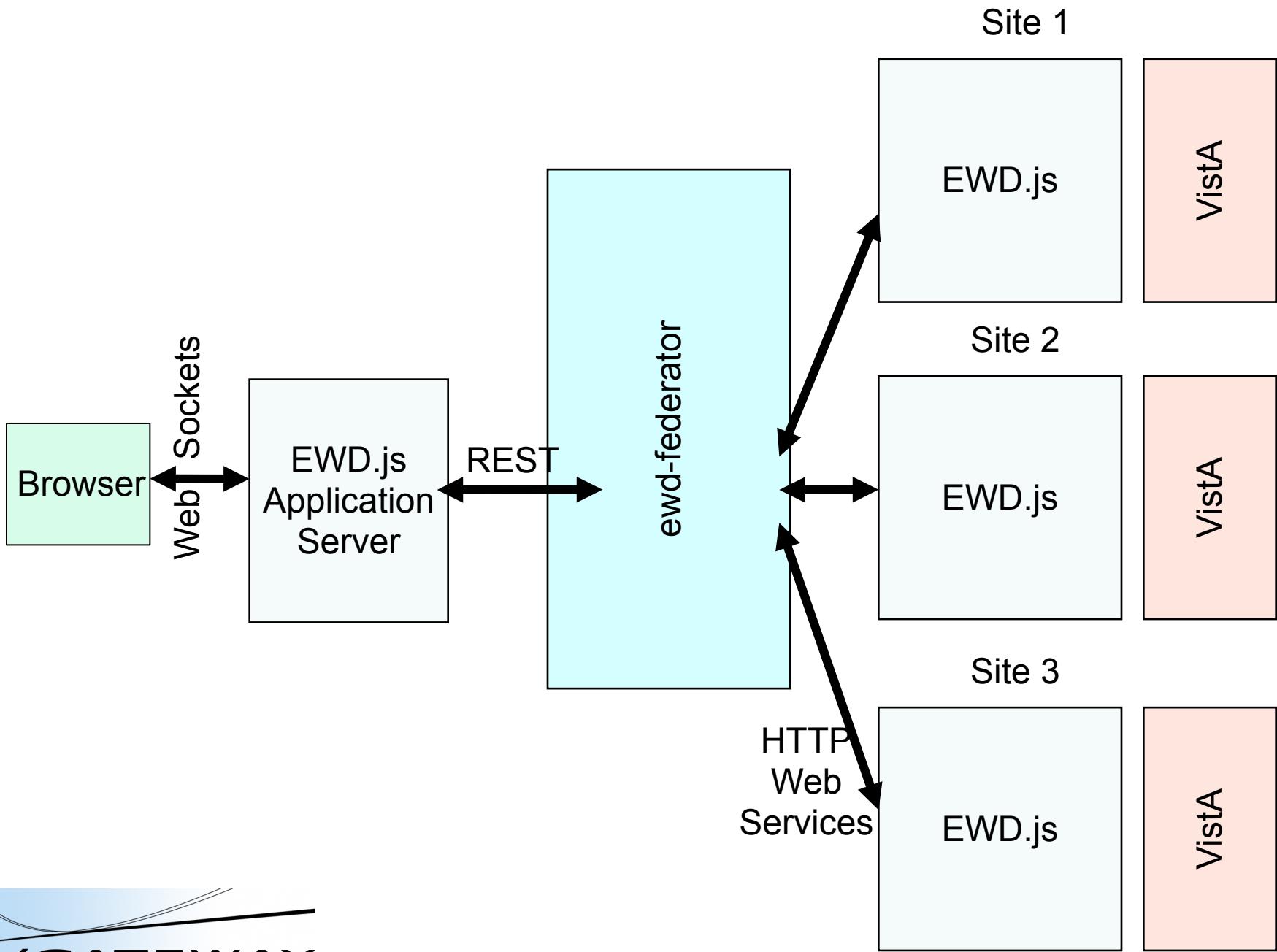
ewd-federator

- Ability to create complex "dances" between EWD.js and other Web/REST servers
 - Single Sign On
 - BSE rules
- Ability to save information on ewd-federator
 - Caching to avoid repeated "dances"
 - Storing authorisation tokens

Potential benefits to the VA

- Open source, lightweight, JavaScript-based solution to federation
- Very fast, scalable
 - Same ECP-based scale-out could be used
- Secure:
 - EWD.js web service traffic encrypted using HMAC-SHA256 (cf Amazon Web Services)
 - EWD.js web services inaccessible without authorisation at each EWD.js server

So that's REST...
What about browser-based apps?



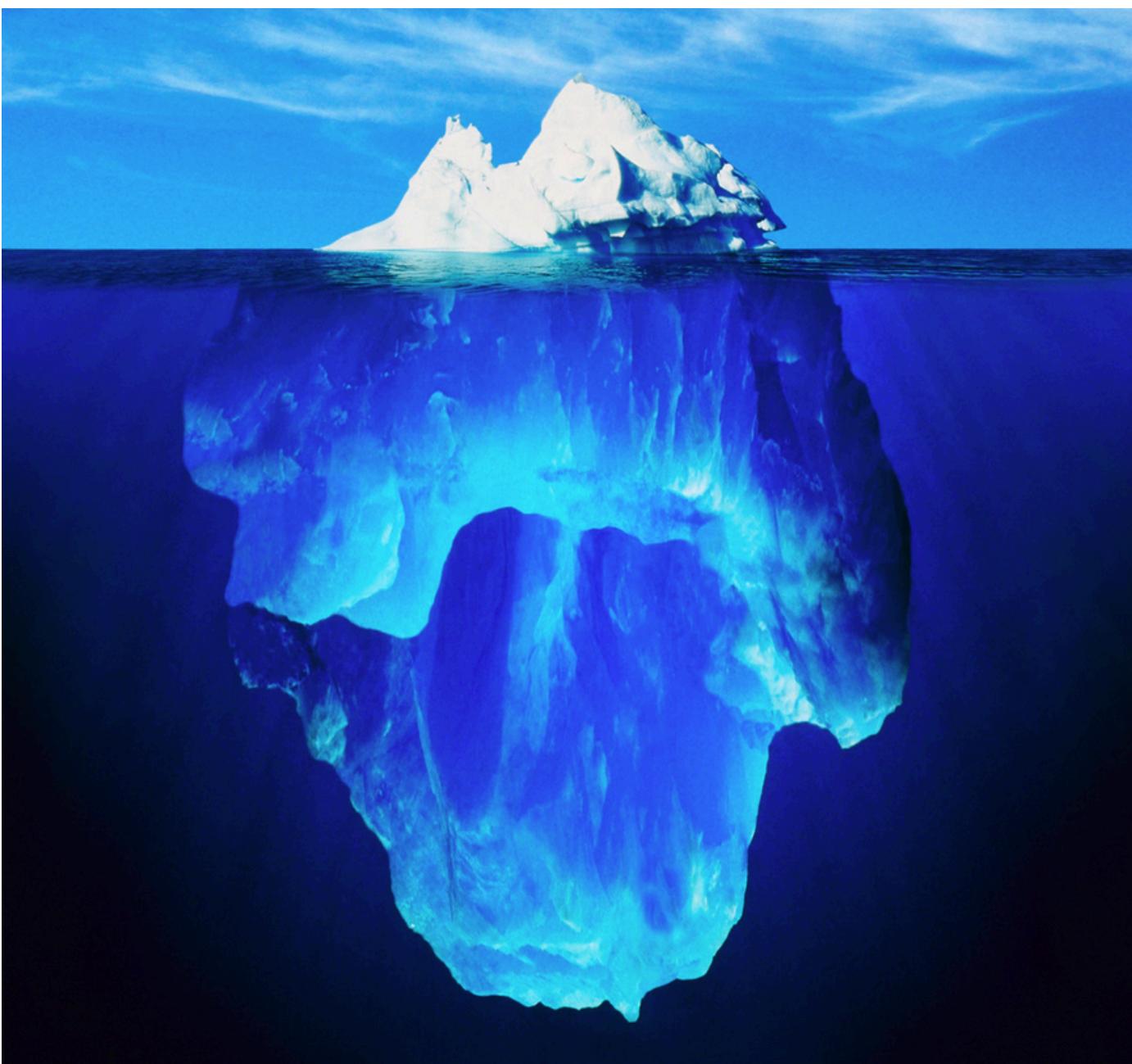
EWD.js Browser Applications

- Single Page Applications
- Use Web Sockets instead of Ajax / HTTP
 - Bi-directional persistent connection
 - Server or browser can send at any time
 - No more polling
 - Many studies suggest faster than HTTP
- Client-server in a browser
- Works with any JavaScript framework

All good stuff, but



It's the narrow perspective on why EWD.js





Copyright © 2015 M/Gateway Developments Ltd

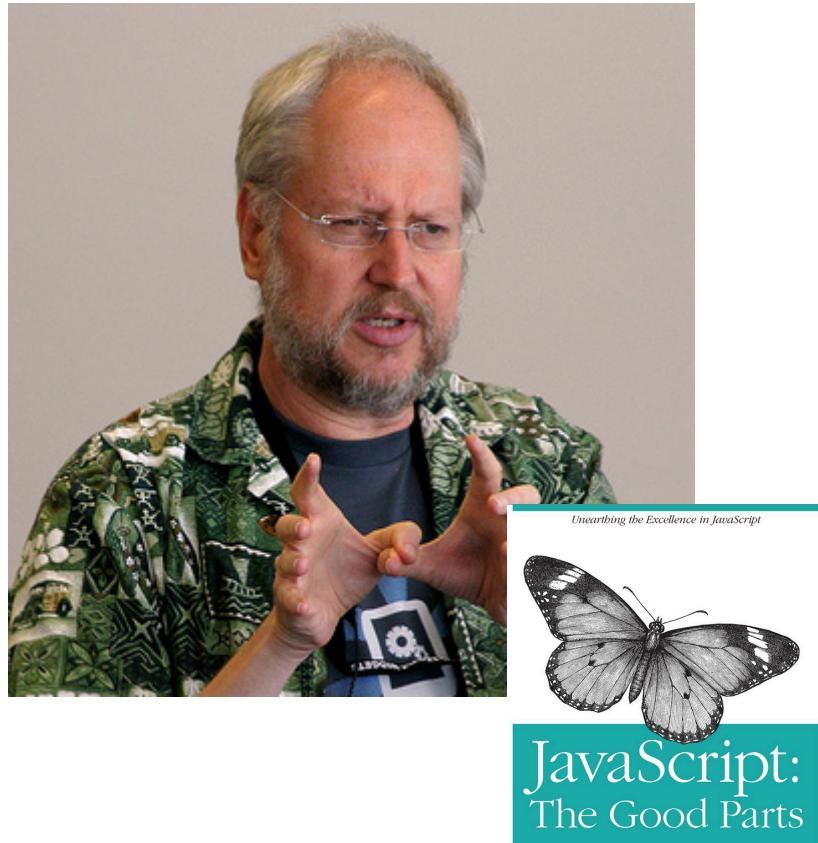
JavaScript



Brendan Eich

Netscape 1995

JavaScript credibility raised



O'REILLY® | YAHOO! PRESS

Douglas Crockford



Browsers

JavaScript

Node.js



Browsers

JavaScript

Node.js

There's never been a more exciting time to be in IT

And all the big innovations and developments are focused
In these 3 areas



Node.js:

There's a module for everything!

Tap into a huge and growing pool
of innovations

So, browser-based VistA Apps

The right way to go?



- Myth #1: Browser UIs just aren't good enough for clinical/medical apps
- Myth #2: Browsers and web technologies aren't fast enough / too much latency
- Myth #3: Browsers and web technologies aren't secure enough



- Myth #1: Browser UIs just aren't good enough for clinical/medical apps

It seems they're good enough for everyone else

There's nothing a browser can't do these days
The days of being limited to simple pages were decades ago



- Myth #1: Browser UIs just aren't good enough for clinical/medical apps

Do you use:

- Google Docs
- Microsoft Office 365
- Do you bank, shop, watch videos, etc on your browser at home?

MYTHBUSTERS

- Myth #1: Browser UIs just aren't good enough for clinical/medical apps

The screenshot shows the Cloud9 IDE interface. The top menu bar includes File, Edit, View, Windows, Help, a file icon, debug, and Preview. On the left, there's a sidebar with icons for Project Files (selected), Open Files, Run, and Preferences. The main area has tabs for skin.xml and statusbar.js. The code editor displays the following JavaScript code:

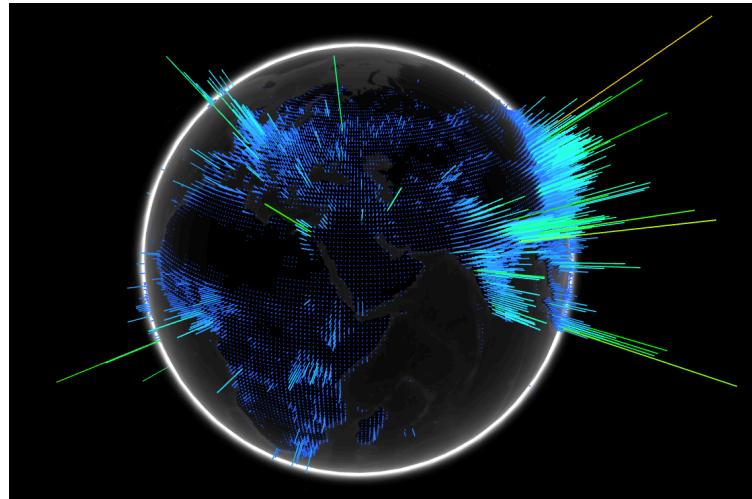
```
module.exports = ext.register("ext/statusbar/statusbar", {
  name : "Status bar",
  dev : "Cloud9 IDE, Inc.",
  alone : true,
  type : ext.GENERAL,
  markup : markup,
  skin : {
    id : "statusbar",
    data : skin,
    "media-path" : ide.staticPrefix + "/style/images/",
    "icon-path" : ide.staticPrefix + "/style/icons/"
  },
  expanded: false,
  nodes : [],
  toolItems: [],
  prefsItems: [],
  horScrollbarAutoHide : "false",
  edgeDistance : 3,
  hook : function(){
    var _self = this;
    ide.addEventListener("openfile", function(){
      setTimeout(function(){
        ext.initExtension(_self);
      }, 1000);
    });
    ide.addEventListener("loadsettings", function(e){
      var strSettings = e.model.queryValue("auto/statusbar");
      if (strSettings === "true") {

```



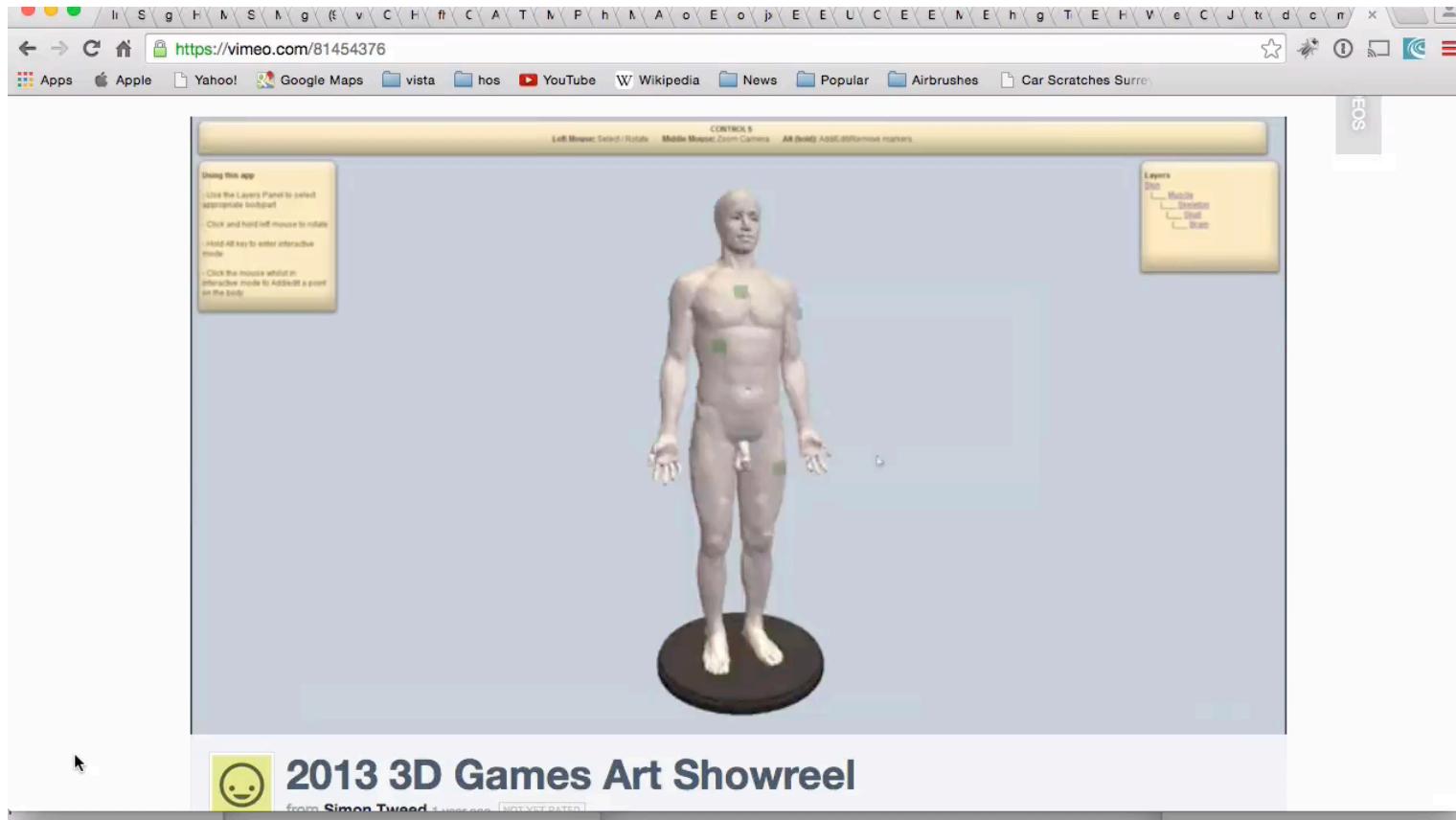
MYTH BOOSTERS

- Myth #1: Browser UIs just aren't good enough for clinical/medical apps
 - Sophisticated graphics:
 - 2-d
 - 3-d eg <http://data-arts.appspot.com/globe/>



MYTH BOOSTERS

- Myth #1: Browser UIs just aren't good enough for clinical/medical apps



<http://gradvs1.mgateway.com/main/simon3d.html>

Copyright © 2015 M/Gateway Developments Ltd



- Myth #1: Browser UIs just aren't good enough for clinical/medical apps

Fileman GUI

Finally, see the work of Astute Semantics
VistA modernisation using EWD.js

<http://astutesemantics.com/>

Copyright © 2015 M/Gateway Developments Ltd



- Myth #1: Browser UIs just aren't good enough for clinical/medical apps



MYTH BOOSTERS

- Myth #2: Browsers and web technologies aren't fast enough / too much latency

Are the following hopelessly slow for you?

- Amazon
- EBay
- Google
- Do you watch Netflix or YouTube in your browser?

MYTH BOOSTERS

- Myth #2: Browsers and web technologies aren't fast enough / too much latency

Demonstrations showing federated access to multiple VistA systems using ewd-federator and EWD.js:

- Sub-second response

Asynchronous pre-emptive data fetching based on usage statistics

MYTH BOOSTERS

- Myth #2: Browsers and web technologies aren't fast enough / too much latency



MYTH BOOSTERS

- Myth #3: Browsers and web technologies aren't secure enough

It seems they're good enough for online banking....

MYTHBUSTERS

- Myth #3: Browsers and web technologies aren't secure enough





So what's the view like from up there?

The future is browser-shaped

- Web-sockets
 - Server can send messages to the browser at any time:
 - Alerts, message of the day
 - Real-time graphing of clinical results on tablet browser by the bed-side
 - Real-time bed state display
 - Cross-browser inter-communication between provider and patient
 - Legacy green screen terminal in browser
 - <https://robtweed.wordpress.com/2014/03/04/vista-the-hackable-open-source-ehr-hackable-platform/>

The future is browser-shaped

- WebRTC
 - Real time communications
 - Skype-like behaviour in a browser, without plug-ins
 - Video
 - Audio
 - File transfer
 - Key to browser-based telemedicine

Asm.js

- Subset of JavaScript that can be compiled and run at near native C++ speed in asm.js-enabled browsers
- Emscripten compiler
- Being driven by the games industry
 - Unreal Engine ported to run in browser, without plug-ins
 - Try this:
 - http://www.flohofwoe.net/demos/dsomapviewer_asmjs.html

Asm.js

- Application of gaming thinking to healthcare problems
- High performance UIs with extremely rapid user interaction
- Gaming is all about high-speed reaction to highly optimised visualisations of large amounts of data
 - Sound familiar?

The future is browser-shaped

- ServiceWorkers:
 - New upcoming feature for the web platform
 - Designed particularly for offline browser problem
 - HTML5 AppCache is declarative and has many known limitations
 - ServiceWorkers:
 - Allow a script to persistently cache resources
 - Can handle all resource requests
 - Even when network unavailable
 - See: <https://github.com/slightlyoff/ServiceWorker/blob/master/explainer.md>

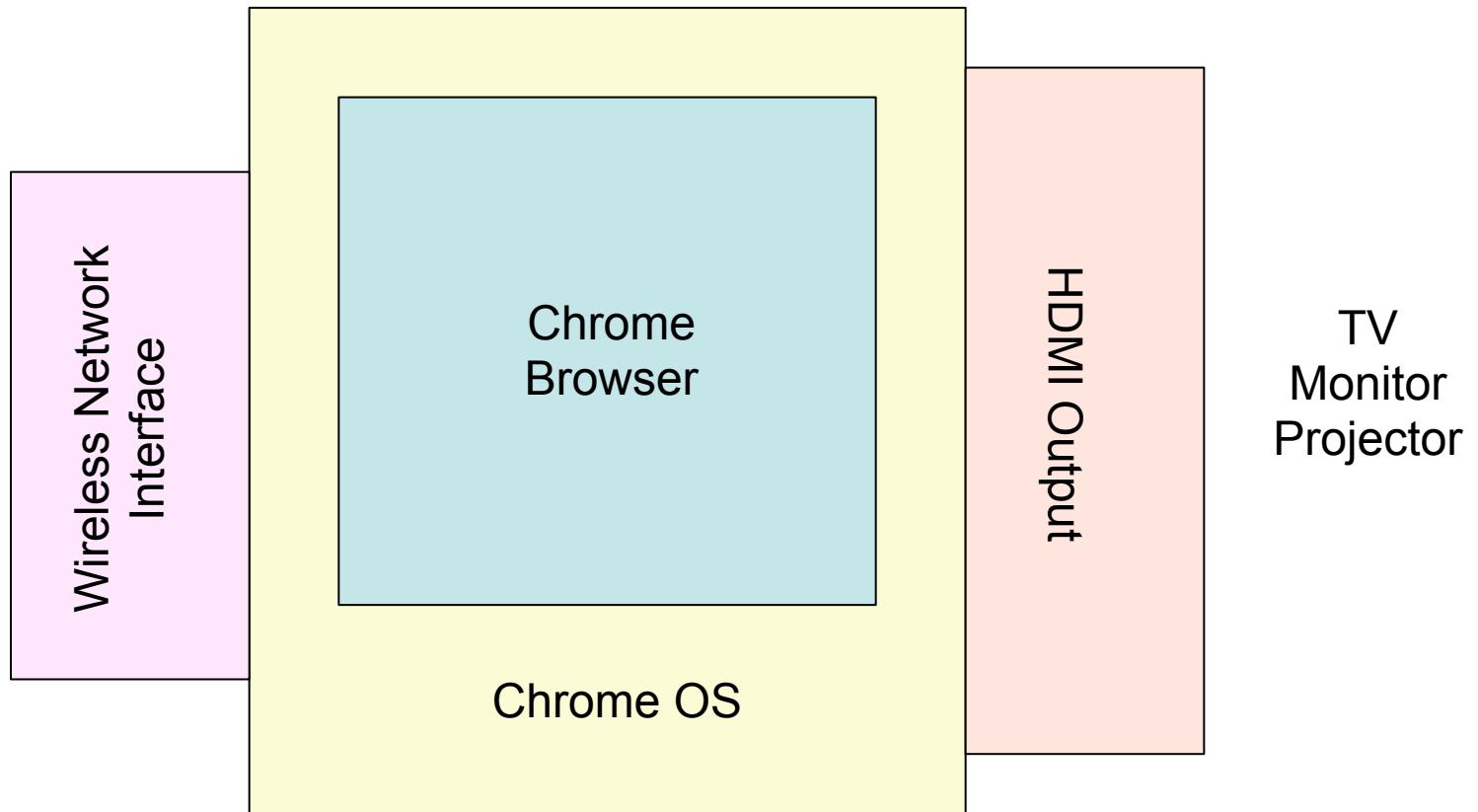
The future is browser-shaped



Chromecast

- \$35 Device from Google
- Marketed as a video streaming device for your TV
- It actually has a lot more potential, particularly in healthcare

Chromecast Schematic



The future is browser-shaped

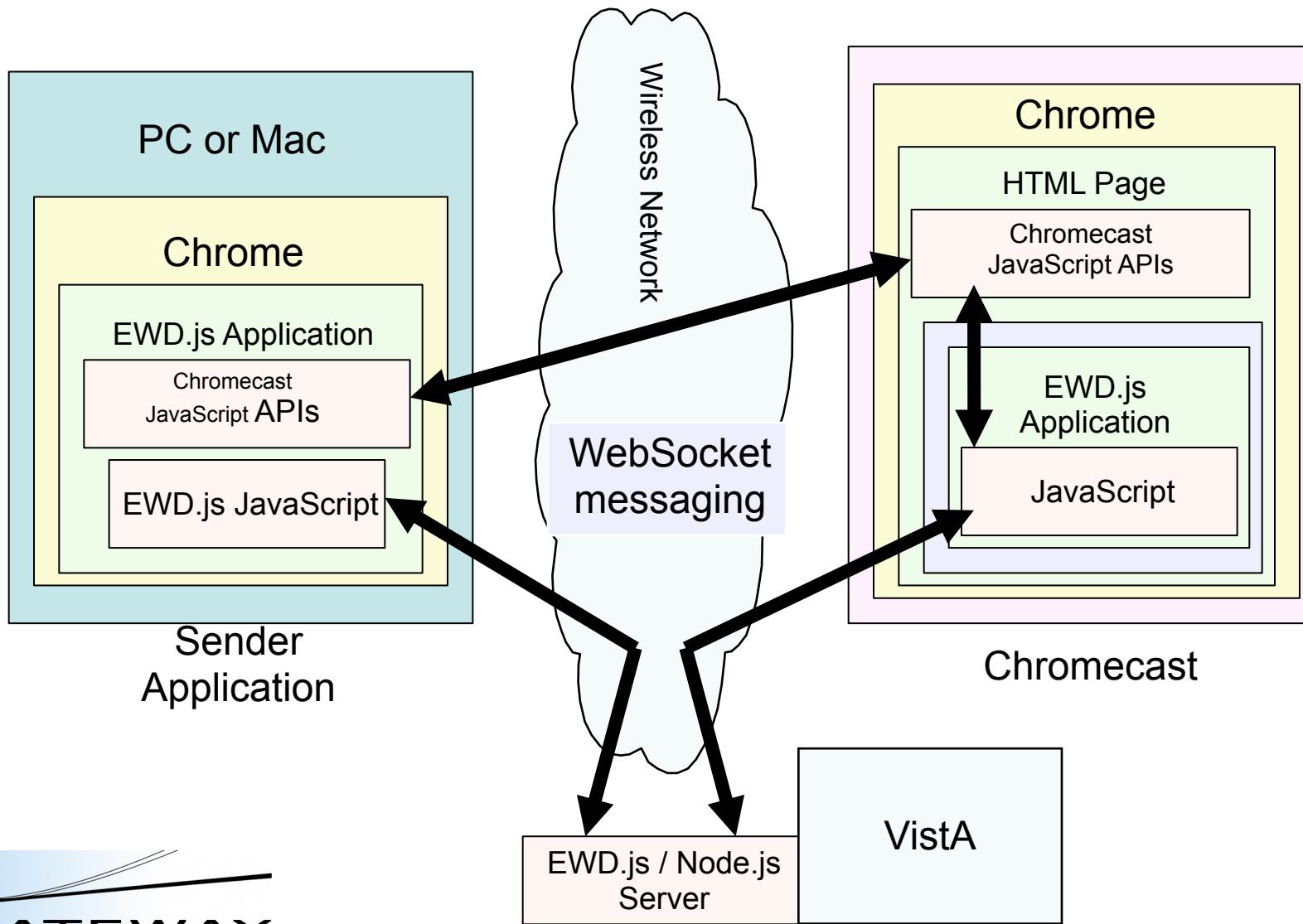


Chromecast

- Turns an HDMI-interfaced monitor or TV into a sophisticated, controllable display

<https://robtweed.wordpress.com/2014/04/05/chromecast-understanding-its-potential/>

VistA Chromecast Applications



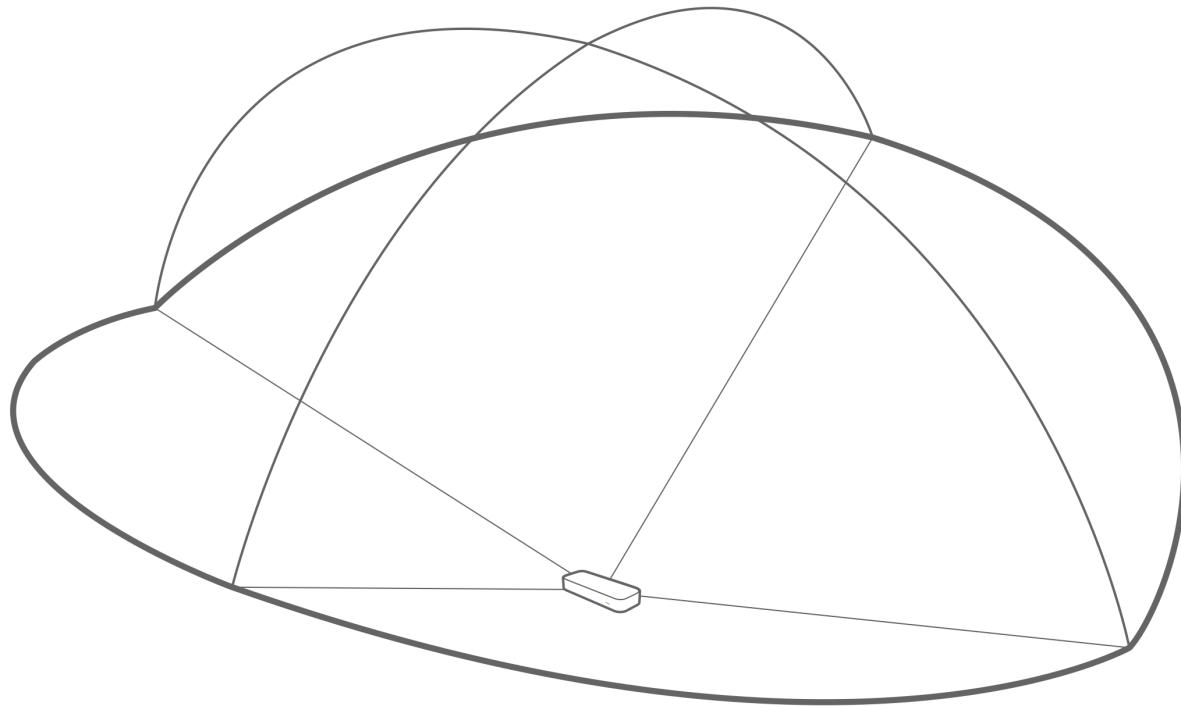
The future is JavaScript-based

LEAP



Costs \$80

What does it do?

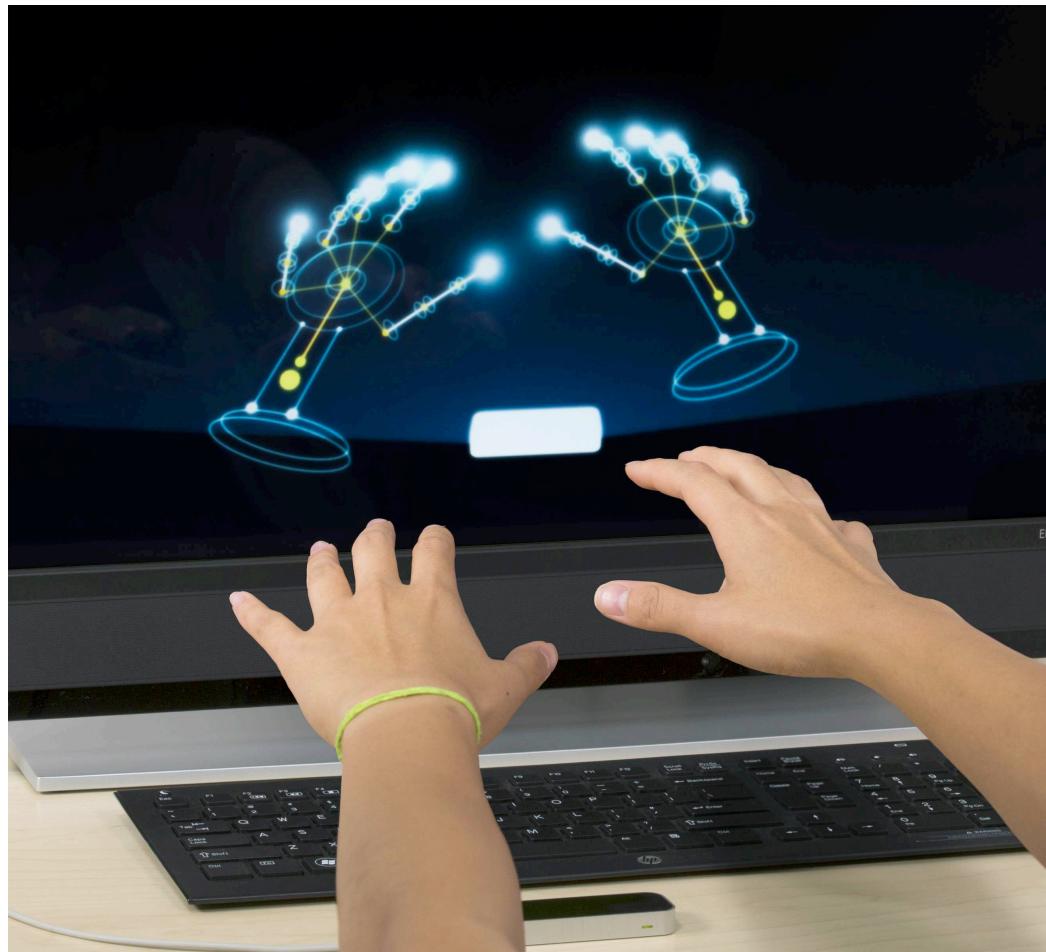


Interaction Area

2 feet above the controller, by 2 feet wide on each side
(150° angle), by 2 feet deep on each side (120° angle)

Detects hand movements
and gestures in 3D area
above and around the device

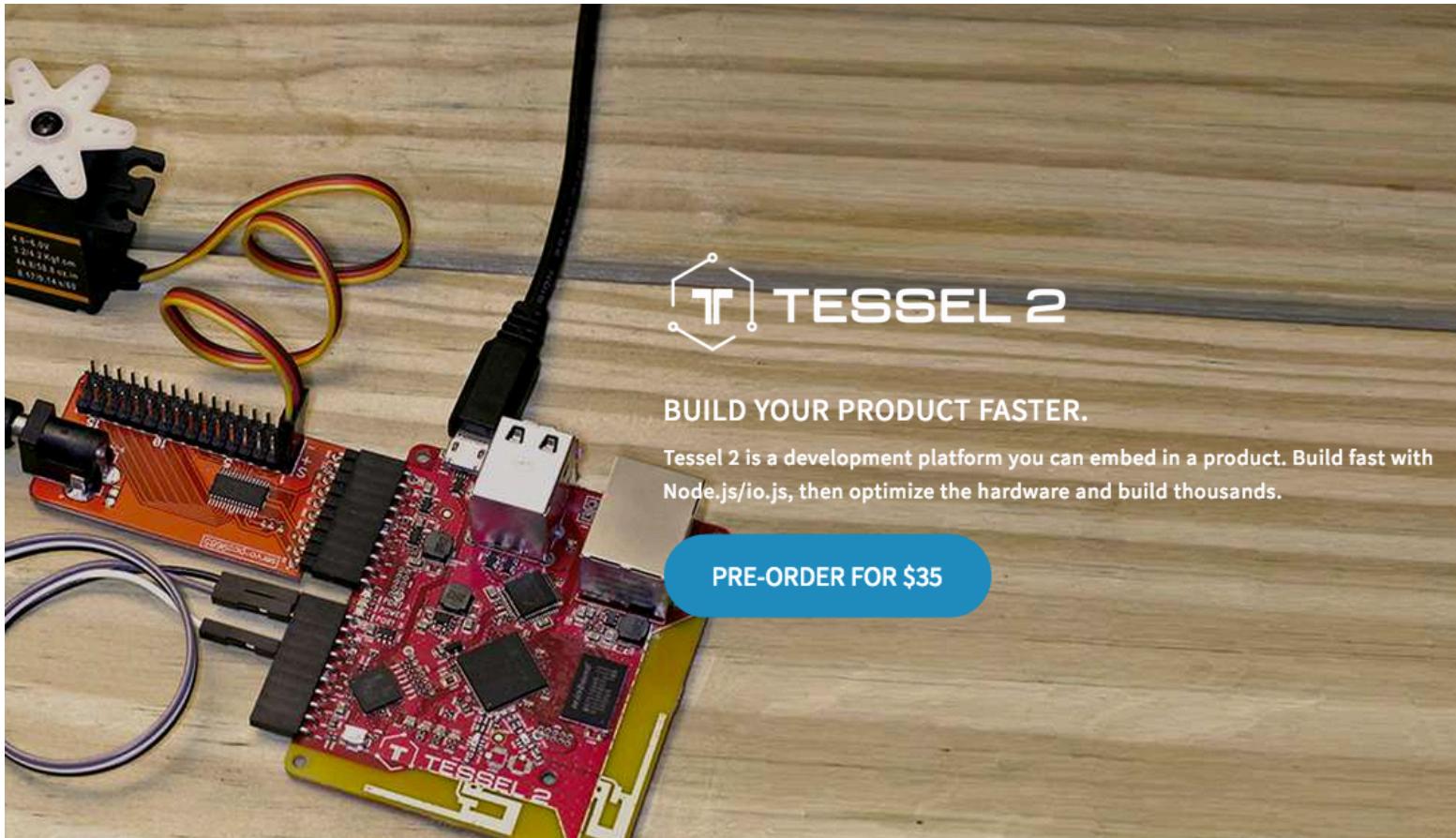
How do you use it?



- USB Device
- Move one or more hands above it
- Software can detect:
 - Palms
 - Fingers & thumbs
 - Including joints
 - Position
 - Rotation in 3D
- Samples at whatever frequency you wish
- Sub-millimeter accuracy
- Program in JavaScript

<http://osehra.org/sites/default/files/QuantitativeHandTremorTweed.pdf>

The future is JavaScript-shaped



TESSEL 2

BUILD YOUR PRODUCT FASTER.

Tessel 2 is a development platform you can embed in a product. Build fast with Node.js/io.js, then optimize the hardware and build thousands.

PRE-ORDER FOR \$35

The future is JavaScript-based



Google Authenticator for 2-factor authentication

EWD.js provides support "out of the box"

More info:
<http://bit.ly/1ERPXRf>

Video:
<http://gradvs1.mgateway.com/main/authenticator.html>

Consequences for the VA

- Applications move to browser using stateless EWD.js architecture
- Resource-hungry stateful applications diminish
- Resources freed up on VistA servers

Consequences for the VA

- Convergence of mobile & desktop development:
 - Responsive design
 - Build once, run on all platforms
 - One language skill: JavaScript
- All but the most demanding mobile applications can be browser-based
- ServiceWorkers seen as solution for offline

Consequences for the VA

- VistA access moves to browser
- Why are PCs needed any more?
- Chromebooks would support all applications
 - Including roll & scroll
- Cheap: \$200 - 300
- Highly secure
- No risk if stolen
- No need for such costly and complex security procedures
- <https://robweed.wordpress.com/2013/12/19/chromebooks-the-ideal-healthcare-desktop/>

Consequences for the VA

- **ewd-federator:**
 - Code distribution
 - Software update management
 - Synchronising configurations
 - Federated control over user privileges
 - Centralised performance monitoring / dashboards

I could go on....

So imagine if...



EWD.js was adopted at the VA..

You really should see the view up here!



Client-server browser-based applications
able to exploit all the IT industry's major advances

Browsers, JavaScript & Node.js:
the powerhouses of innovation



Copyright © 2015 M/Gateway Developments Ltd

EWD.js

The Future Starts Here

Rob Tweed

Twitter: @rtweed

rob.tweed@gmail.com

<http://www.mgateway.com>



Copyright © 2015 M/Gateway Developments Ltd