

# EWD.js on FHIR

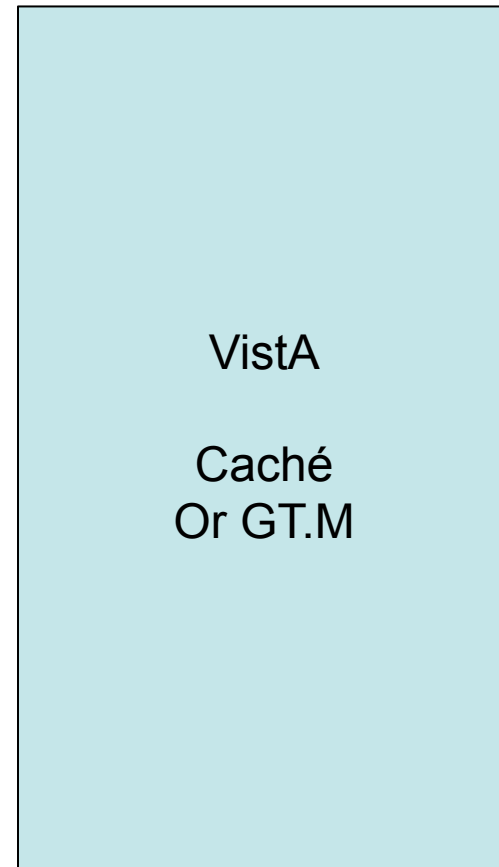
Rob Tweed  
M/Gateway Developments Ltd

Twitter: @rtweed  
Email: [rtweed@mgateway.com](mailto:rtweed@mgateway.com)

<http://www.mgateway.com>



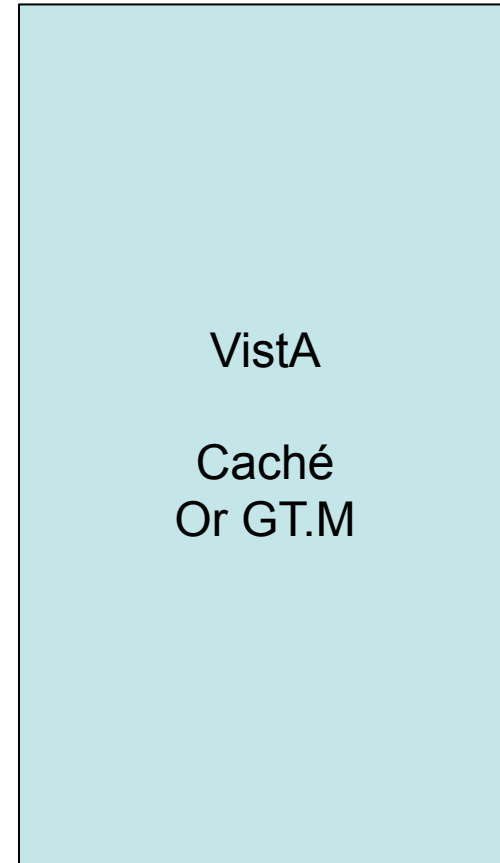
# EWD.js Architecture



# EWD.js Architecture

Mumps:

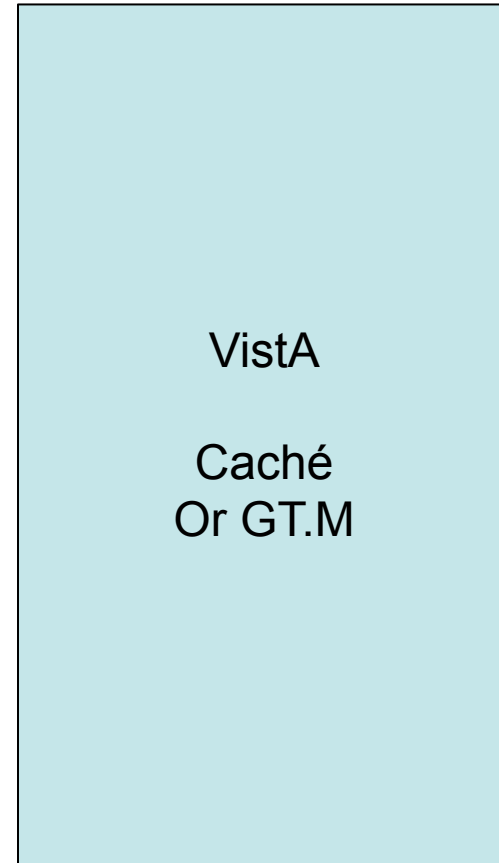
- Language
- Database



# EWD.js Architecture

Mumps:

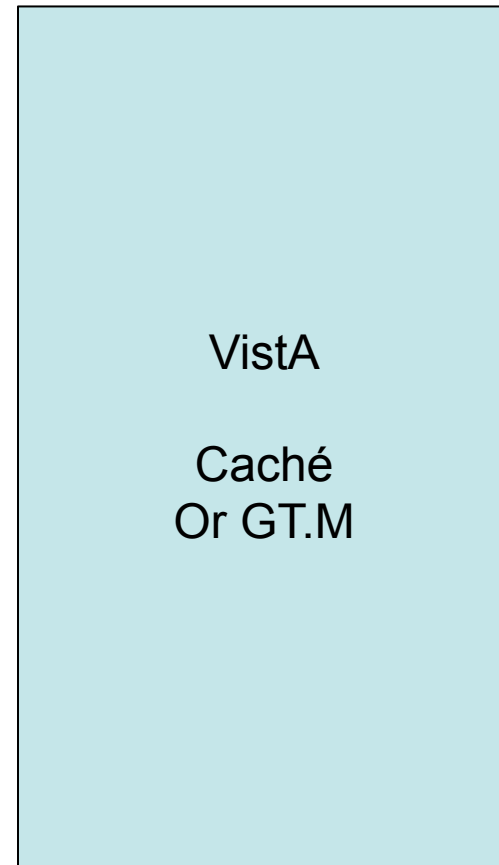
- Language
- Database



# EWD.js Architecture

Mumps:

- Language
- Database



# EWD.js Architecture

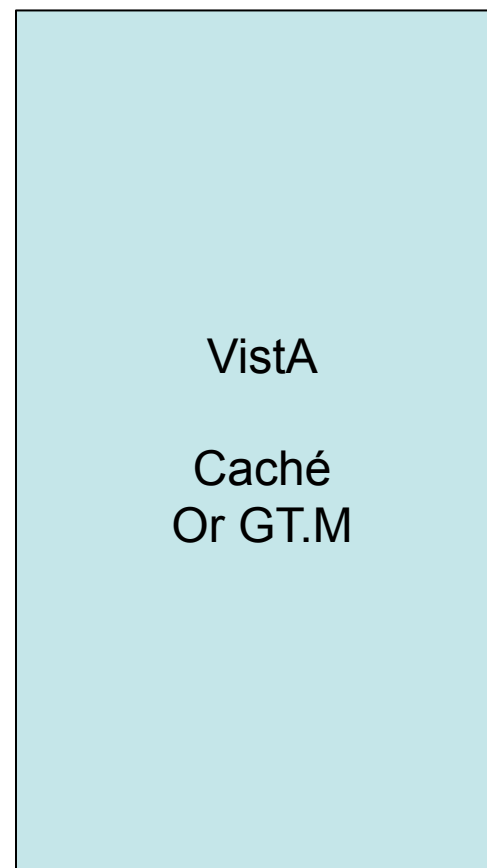
Mumps:

- Language
- Database



Exceptional NoSQL Database

- high performance
- highly scalable
- hierachical
- projectable in many ways
  - key/value
  - graph
  - object
  - relational
  - document



# EWD.js Architecture

Mumps:

~~- Language~~



- Database



VistA

Caché  
Or GT.M

# EWD.js Architecture

Mumps:

~~- Language~~



- Database



What language as a replacement?

- popular
- similar good parts to Mumps language
- quick & easy to develop in
- natural fit with database

VistA

Caché  
Or GT.M



# EWD.js Architecture

Mumps:

~~- Language~~



- Database



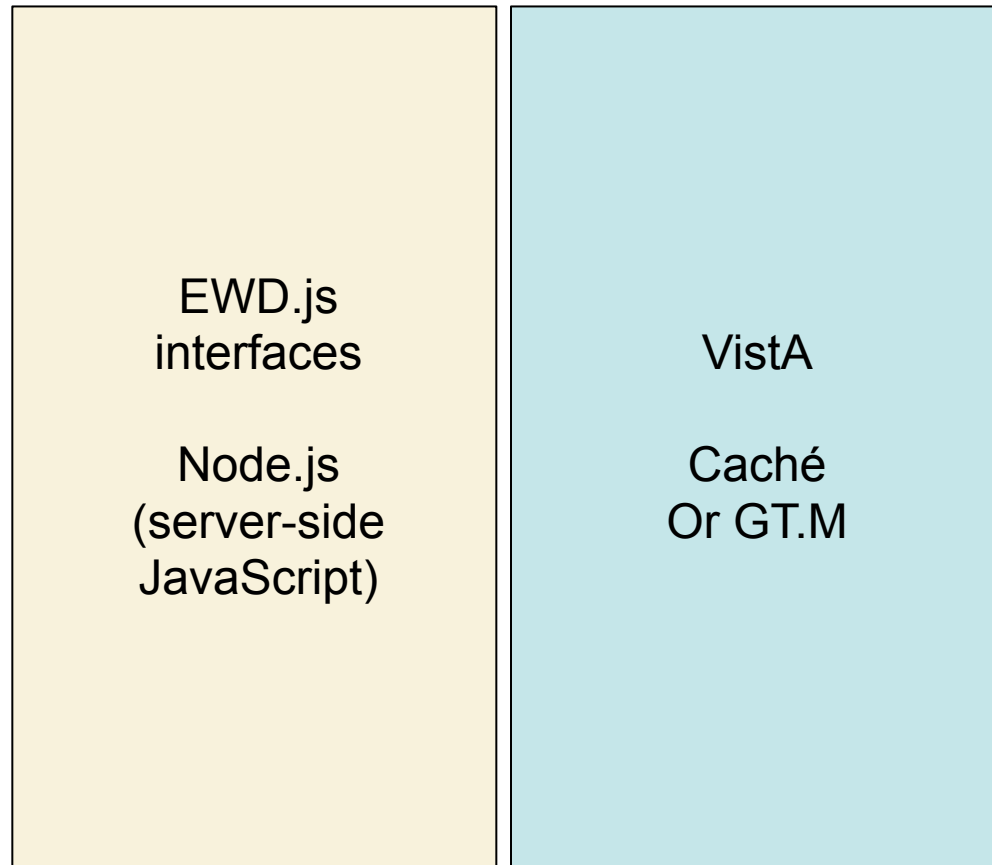
JavaScript

- hugely popular and still growing
- interpreted
- dynamic
- un-typed
- perfect fit with database
  - JSON = document database

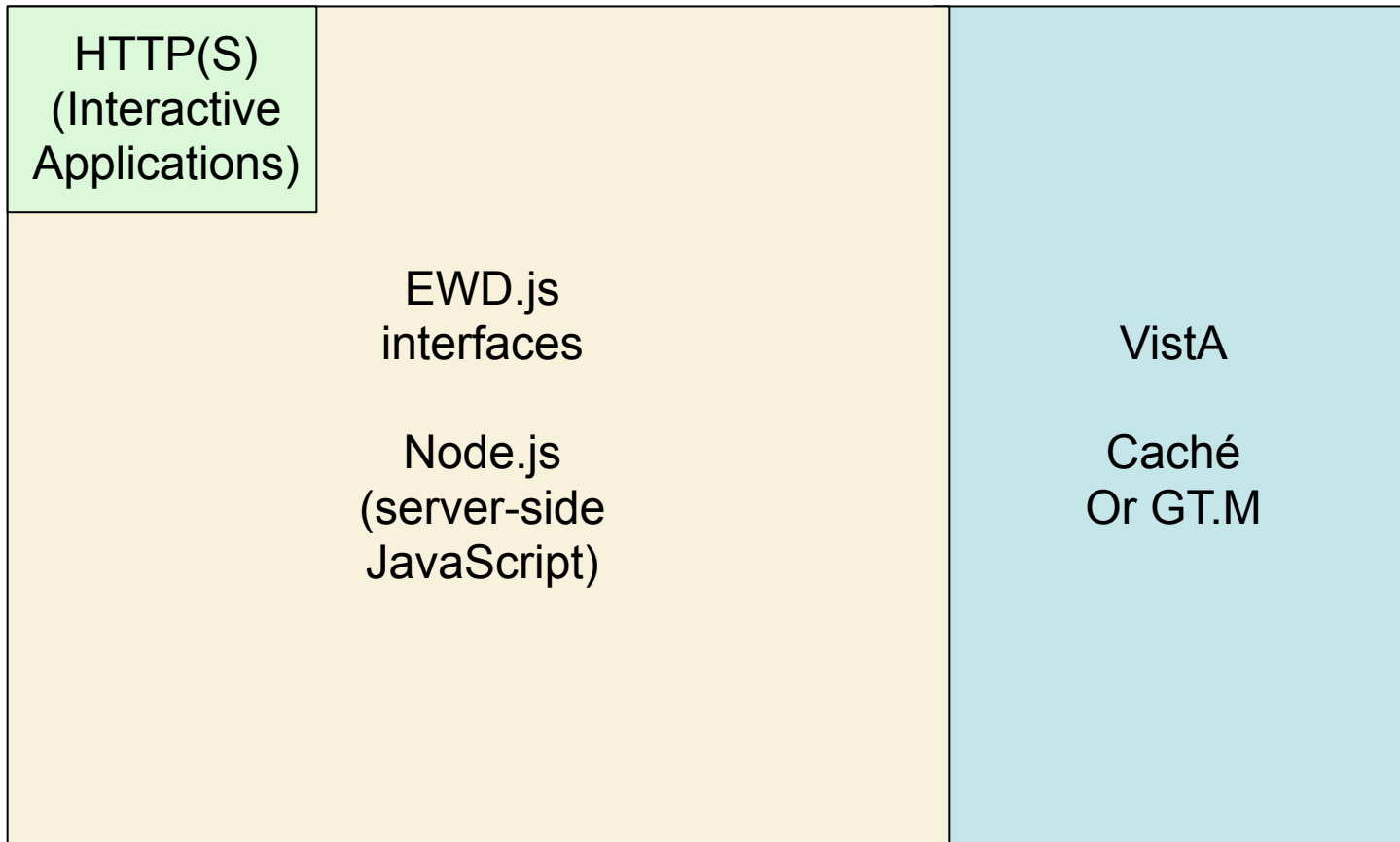
VistA

Caché  
Or GT.M

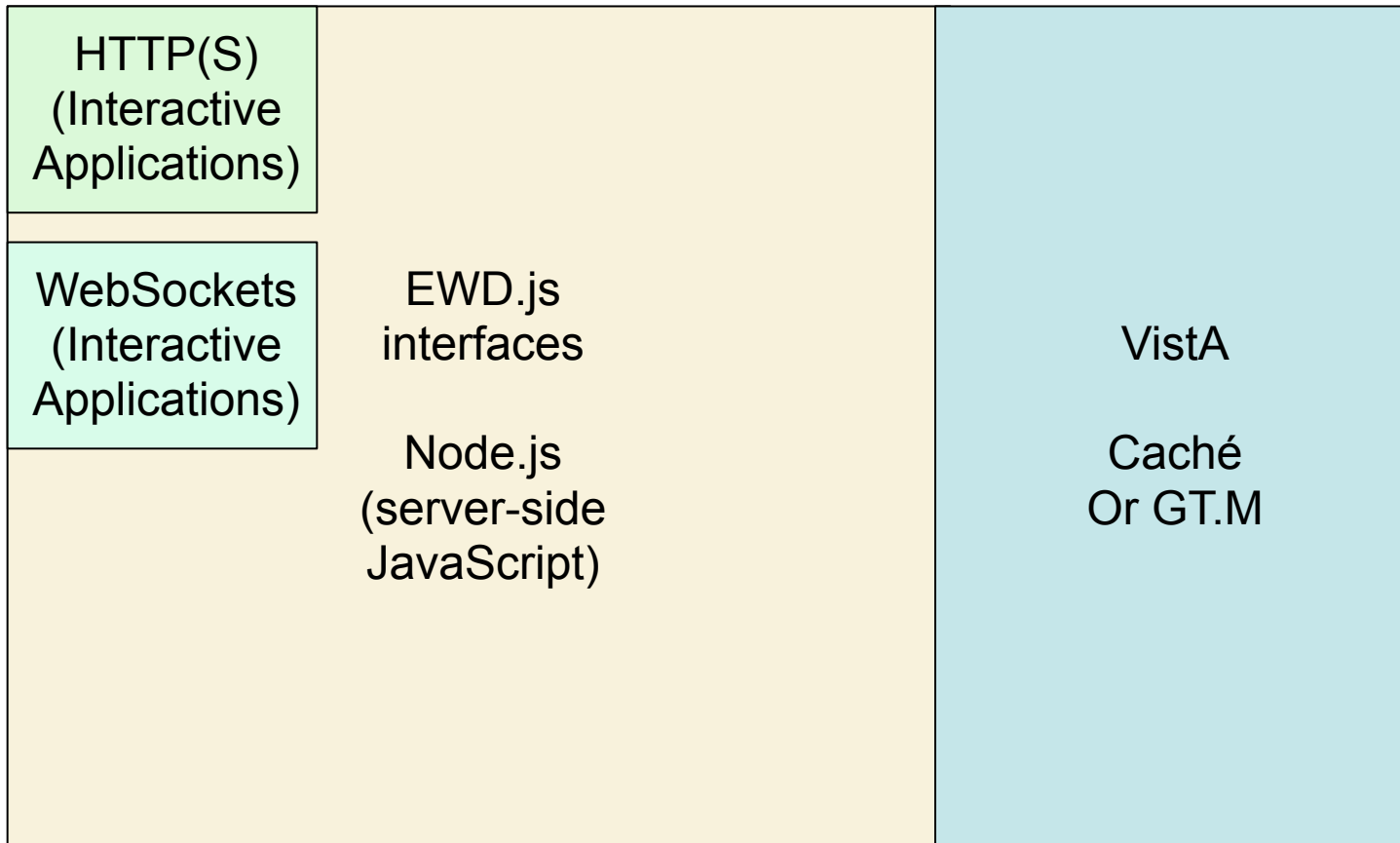
# EWD.js Architecture



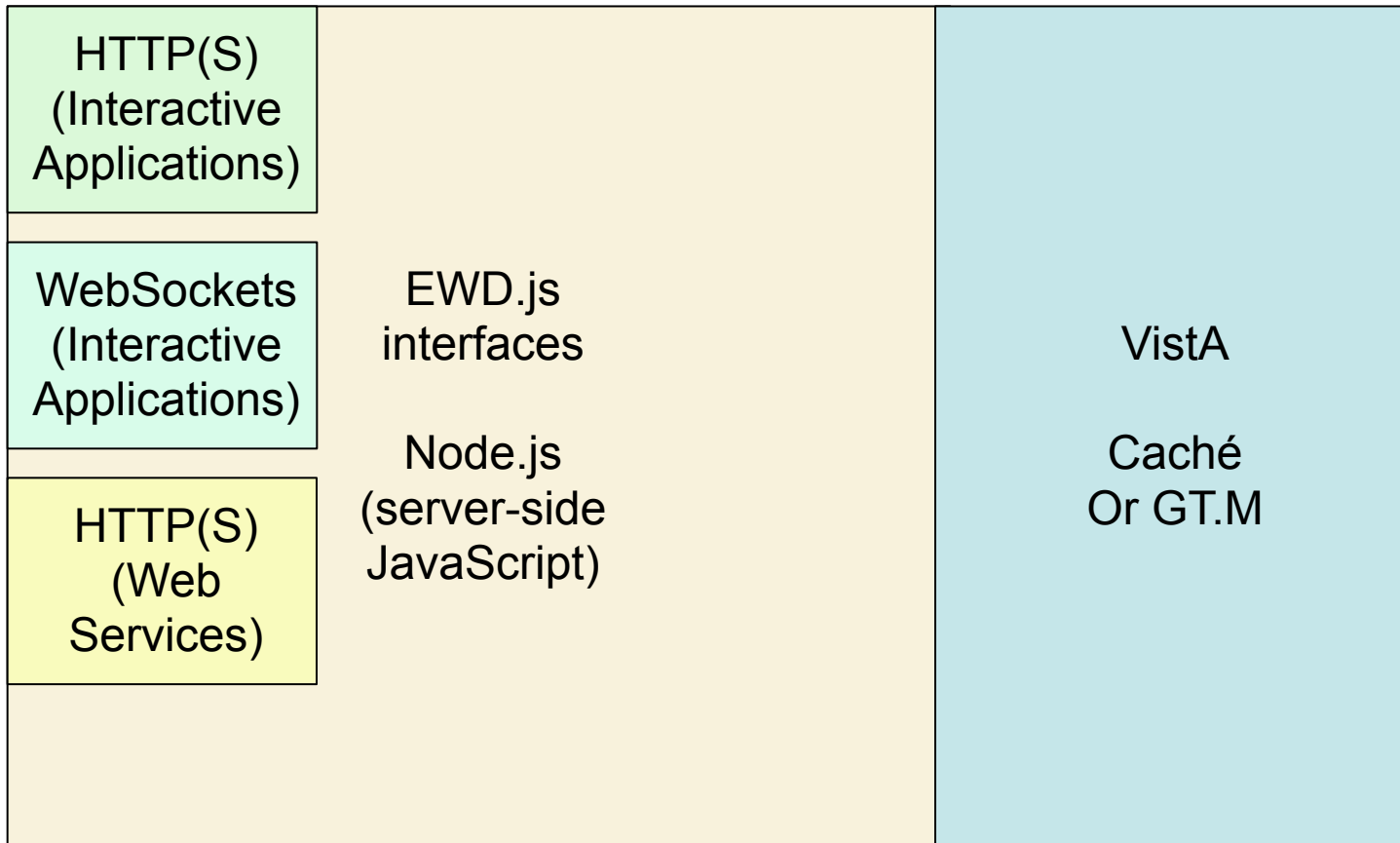
# EWD.js Architecture



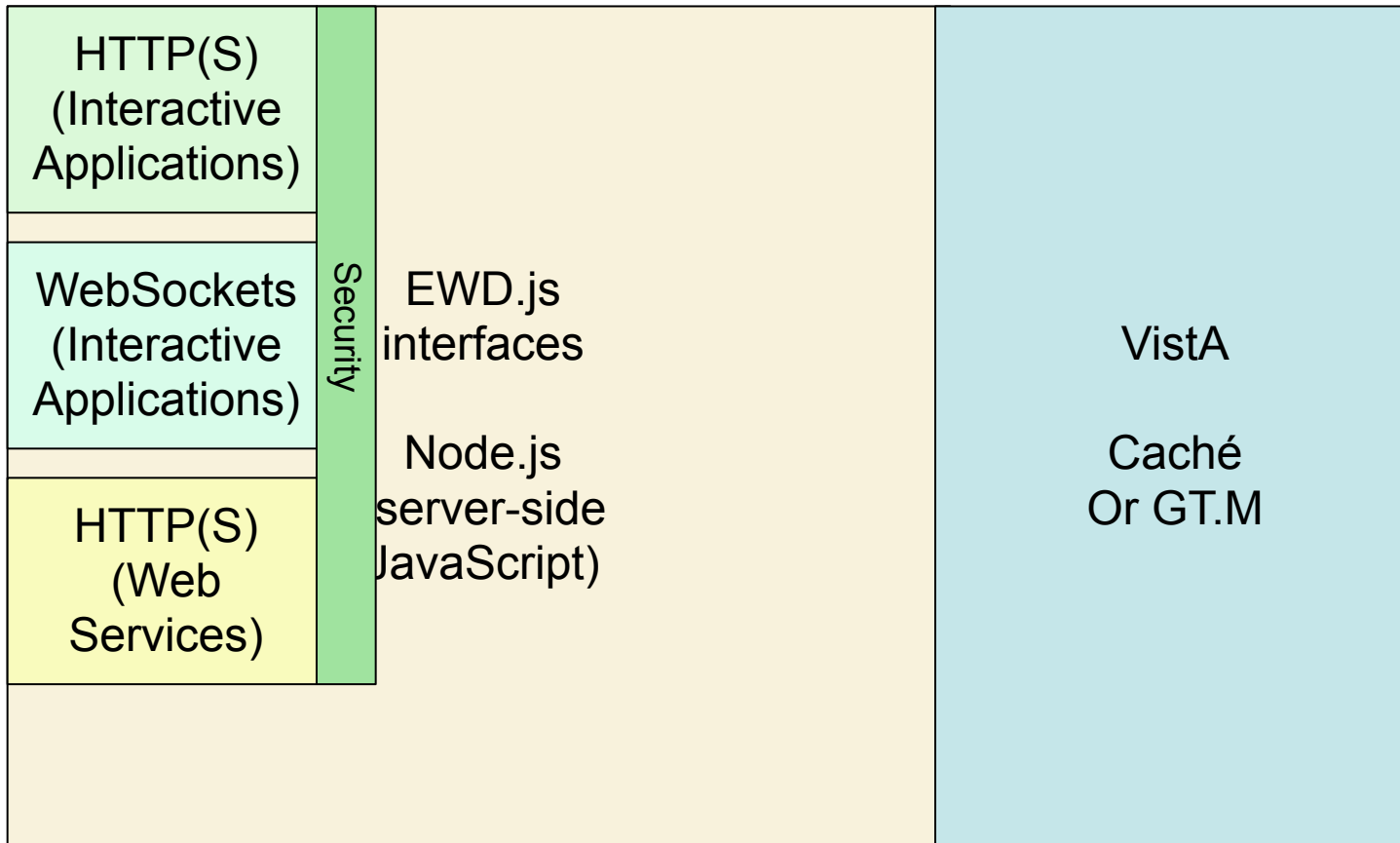
# EWD.js Architecture



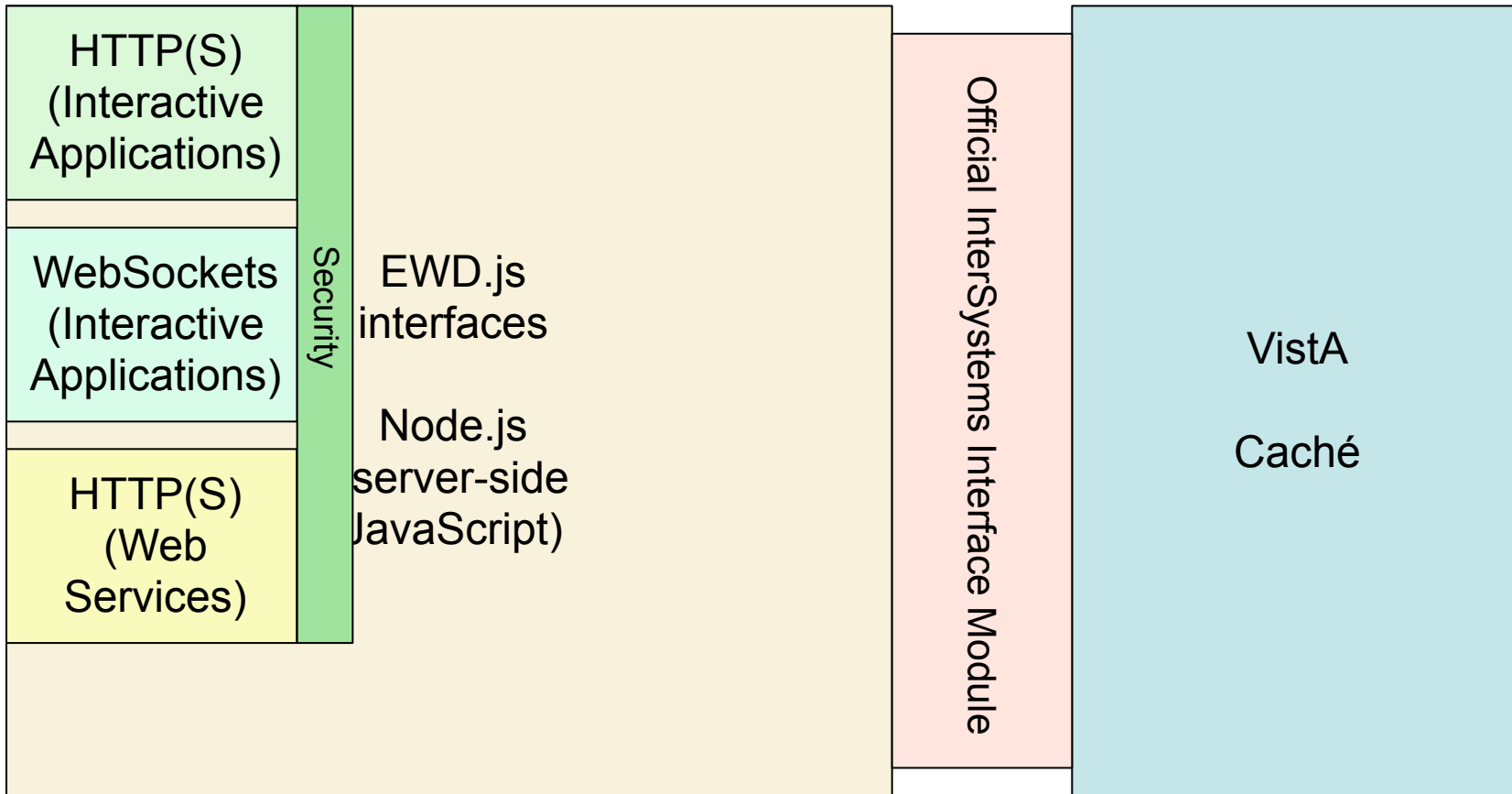
# EWD.js Architecture



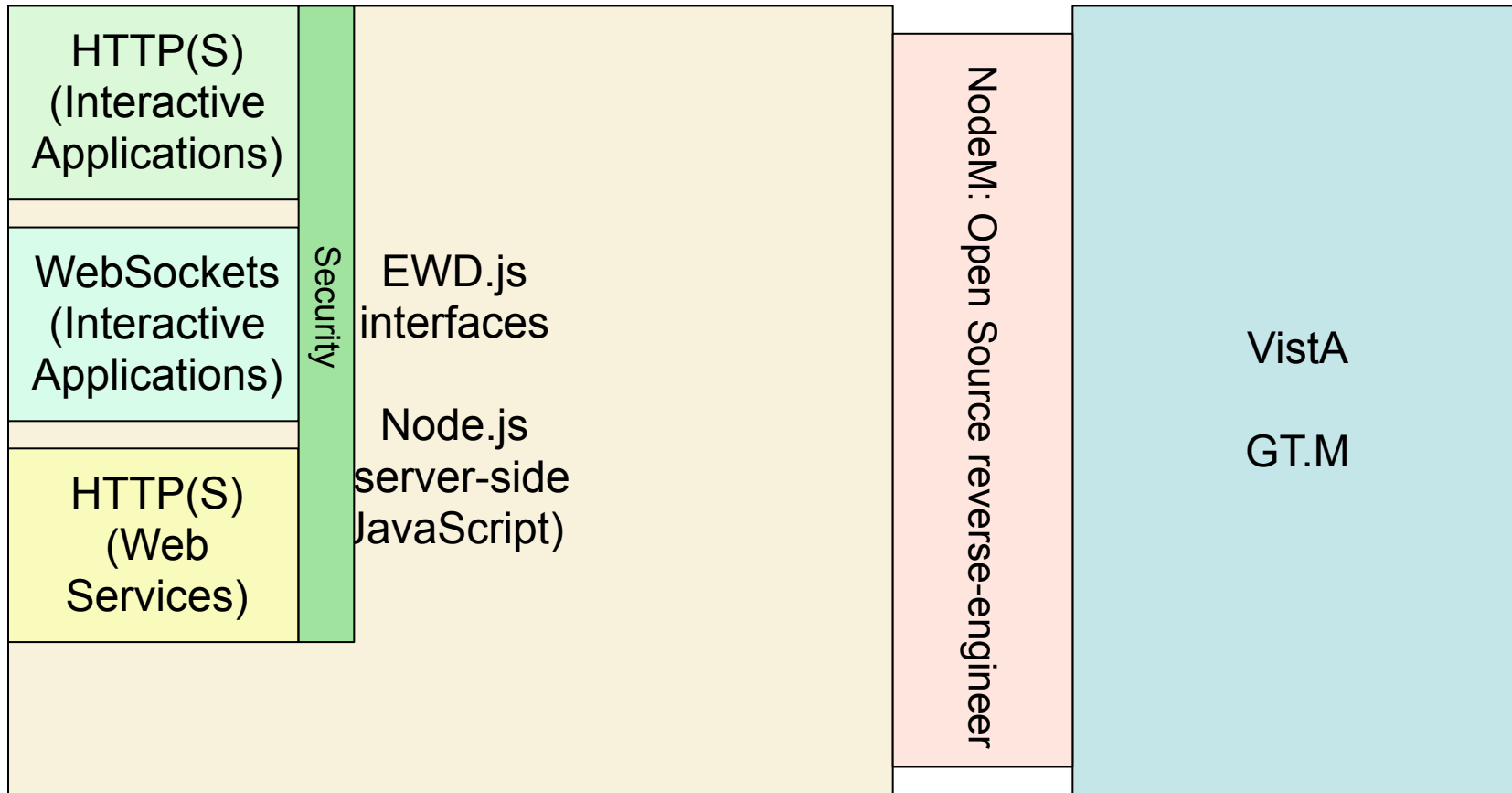
# EWD.js Architecture



# EWD.js Architecture

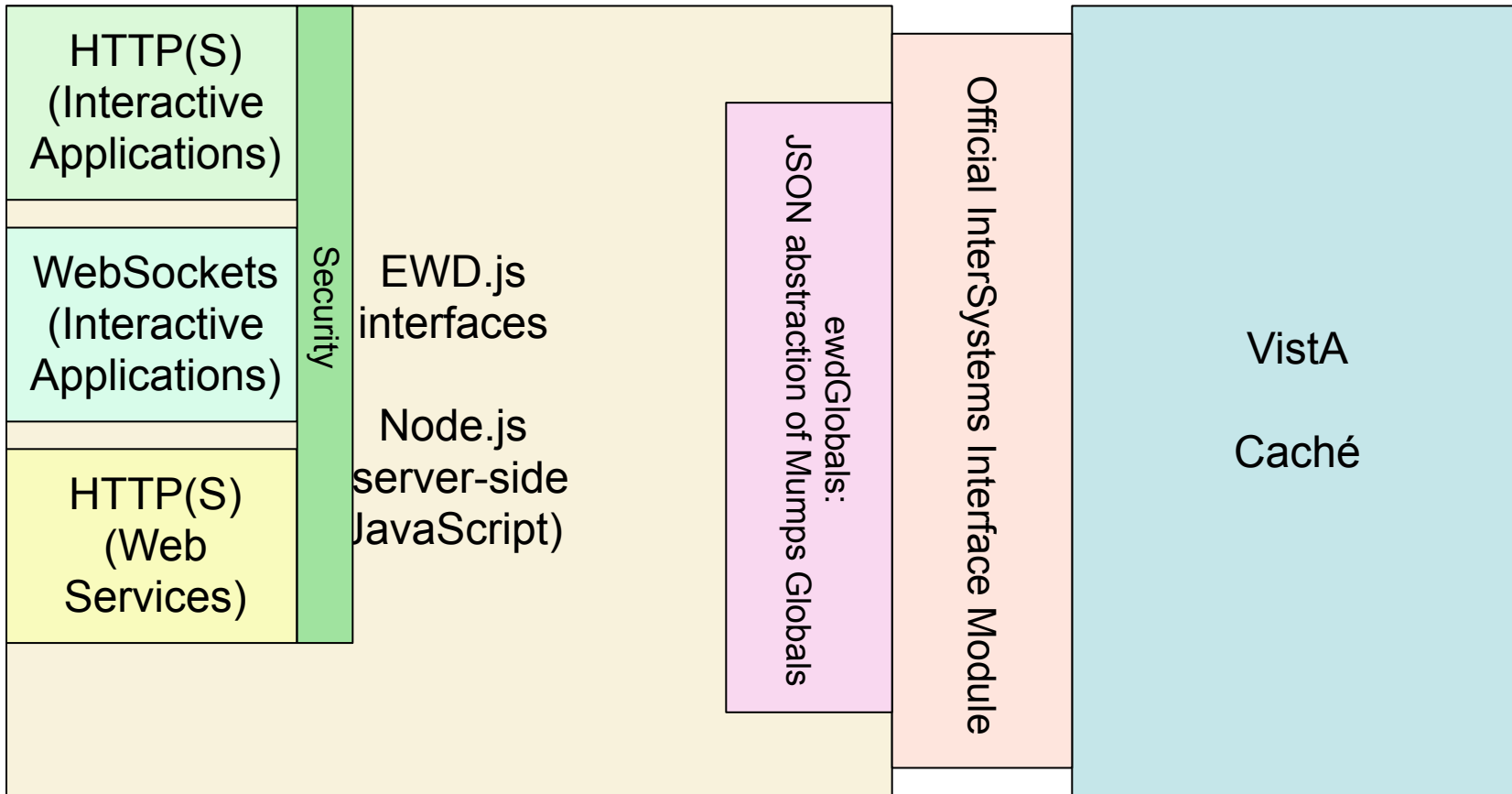


# EWD.js Architecture

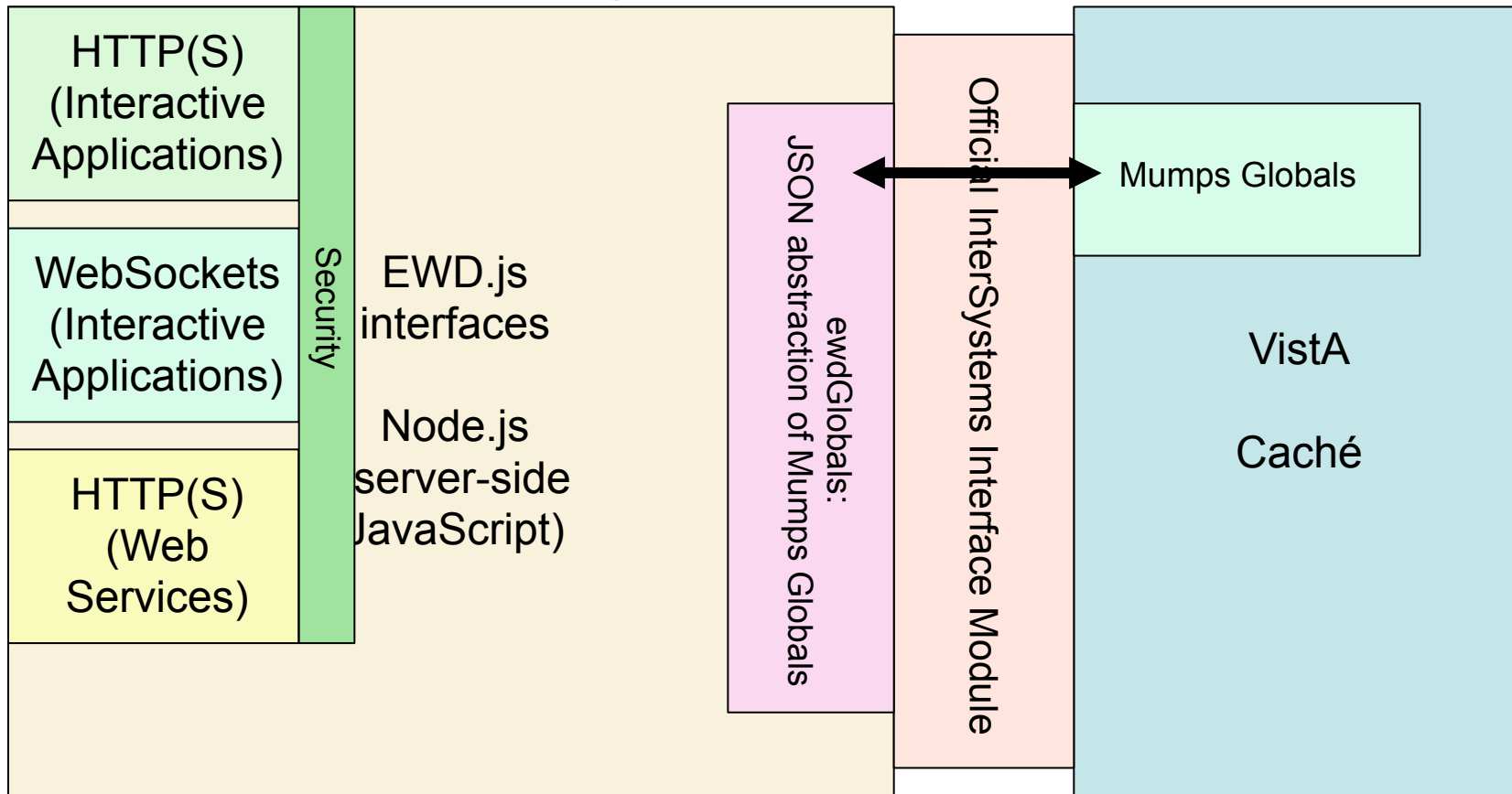




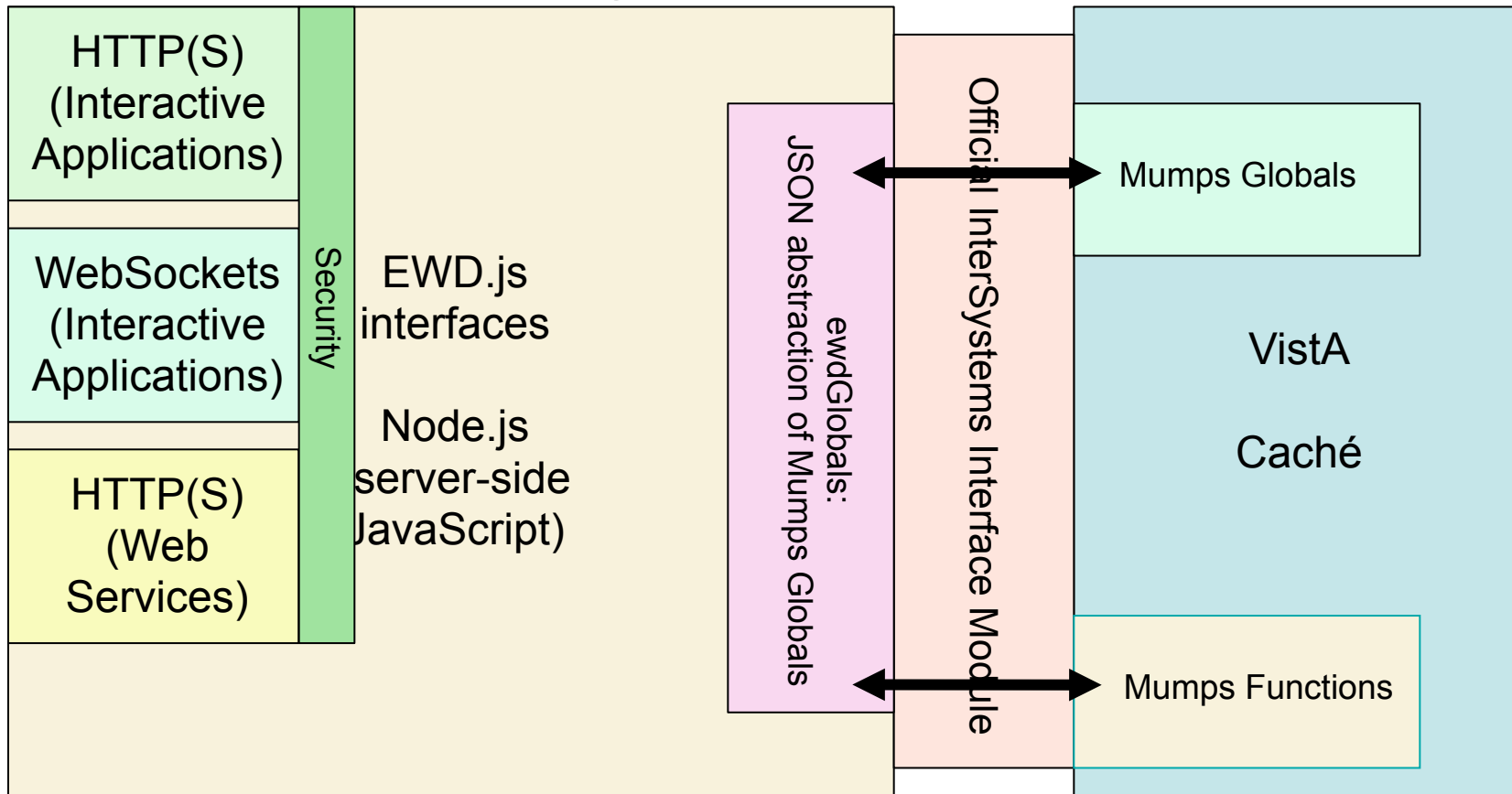
# EWD.js Architecture



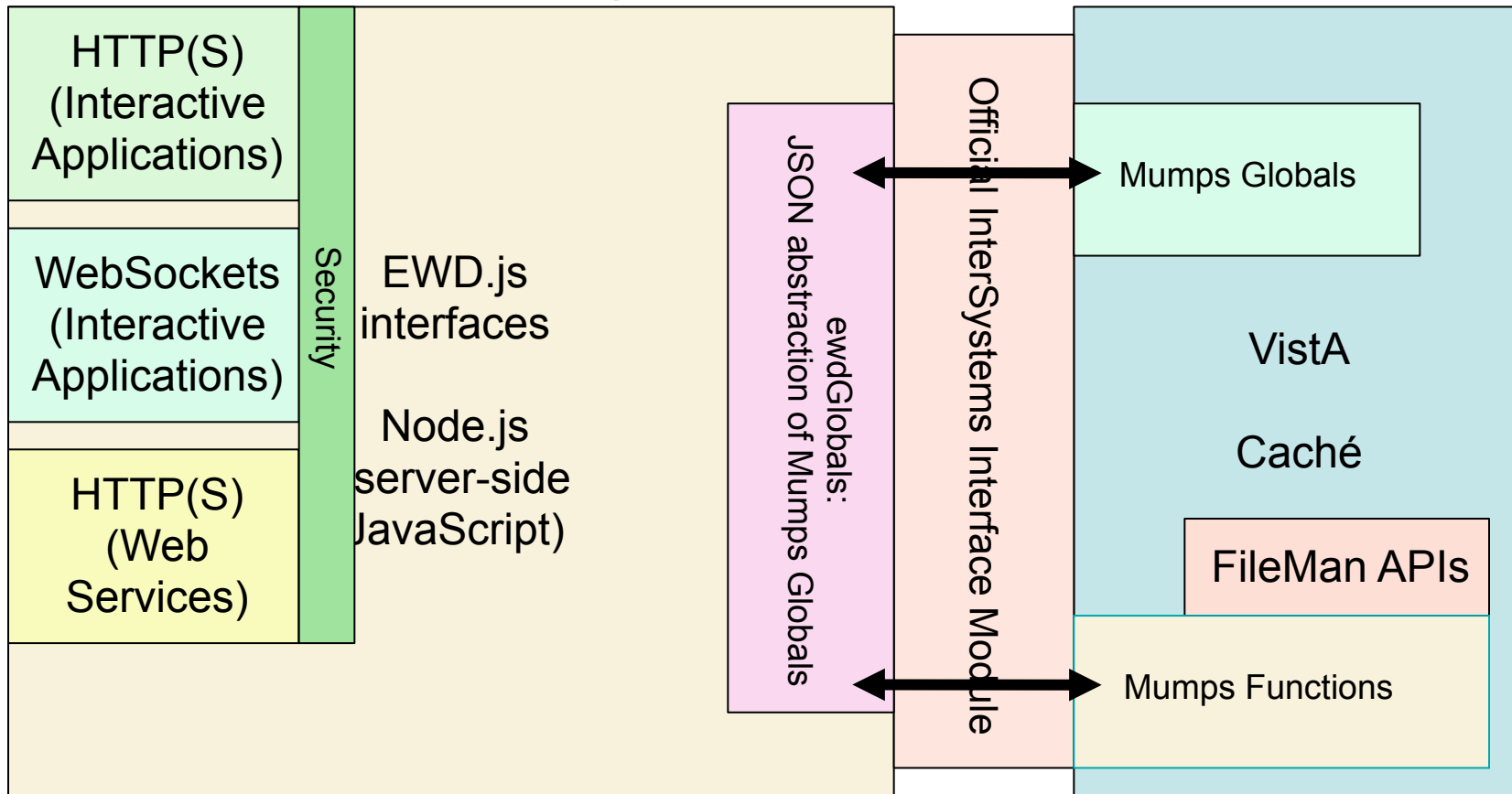
# EWD.js Architecture



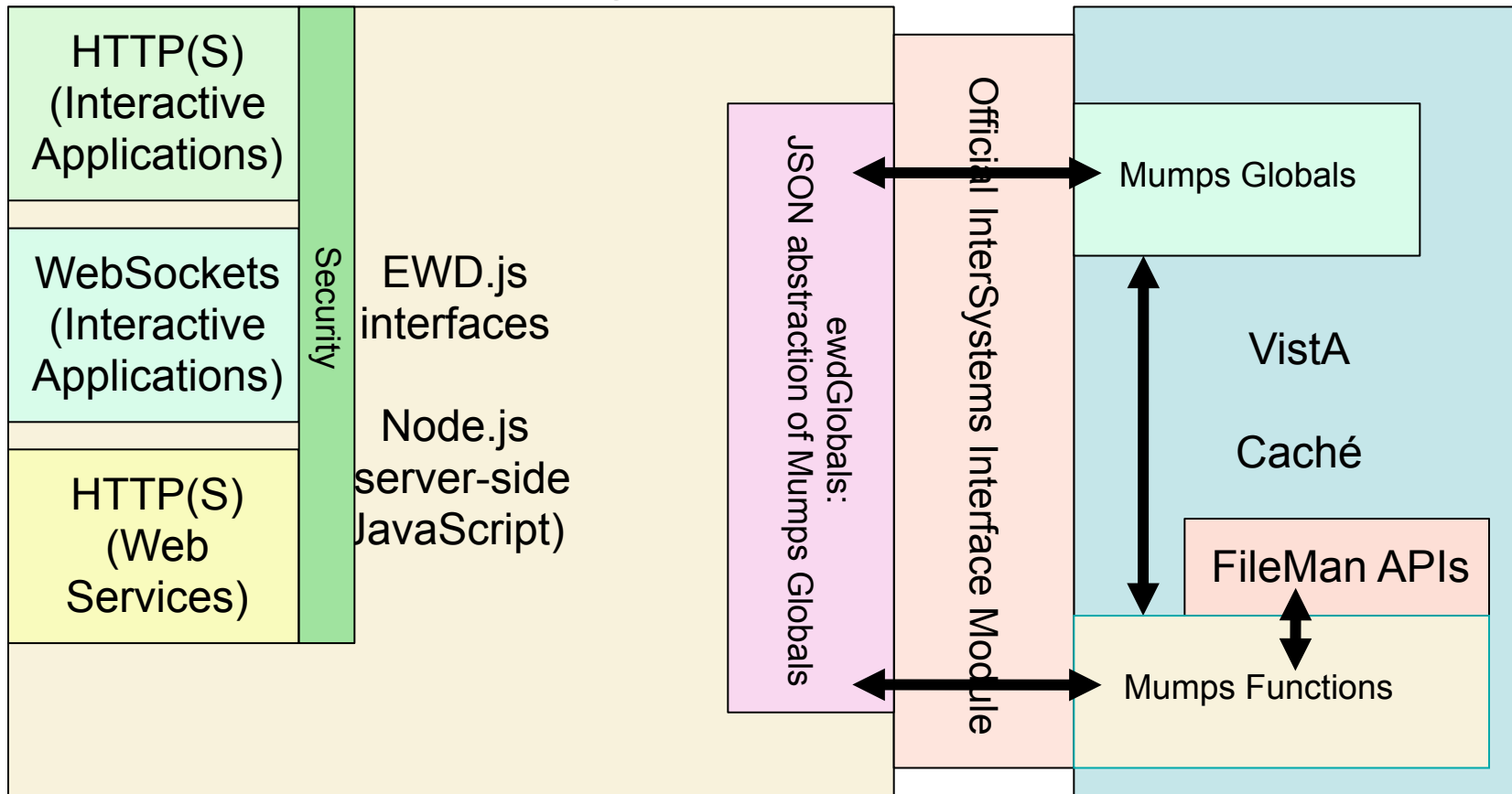
# EWD.js Architecture



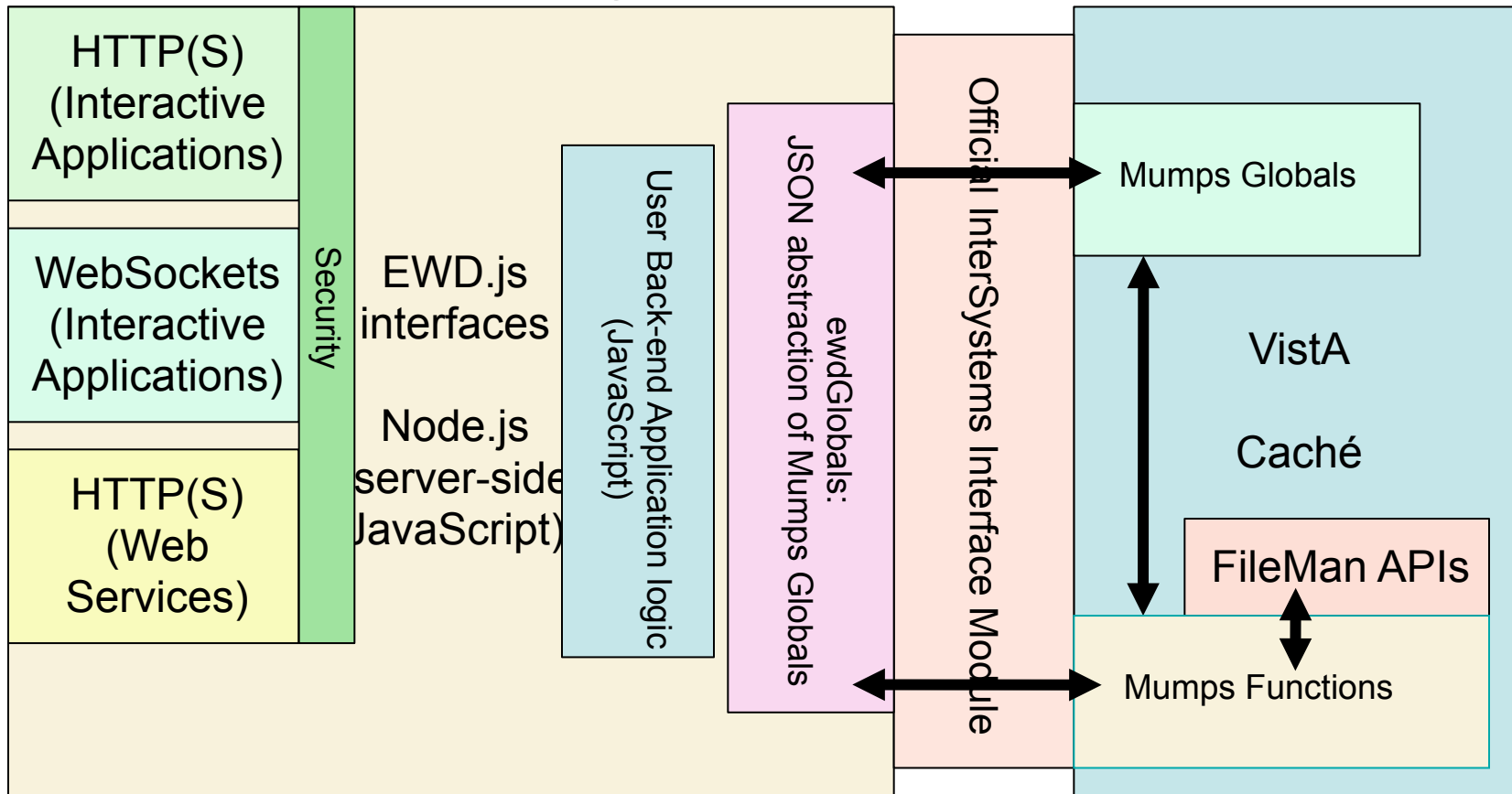
# EWD.js Architecture



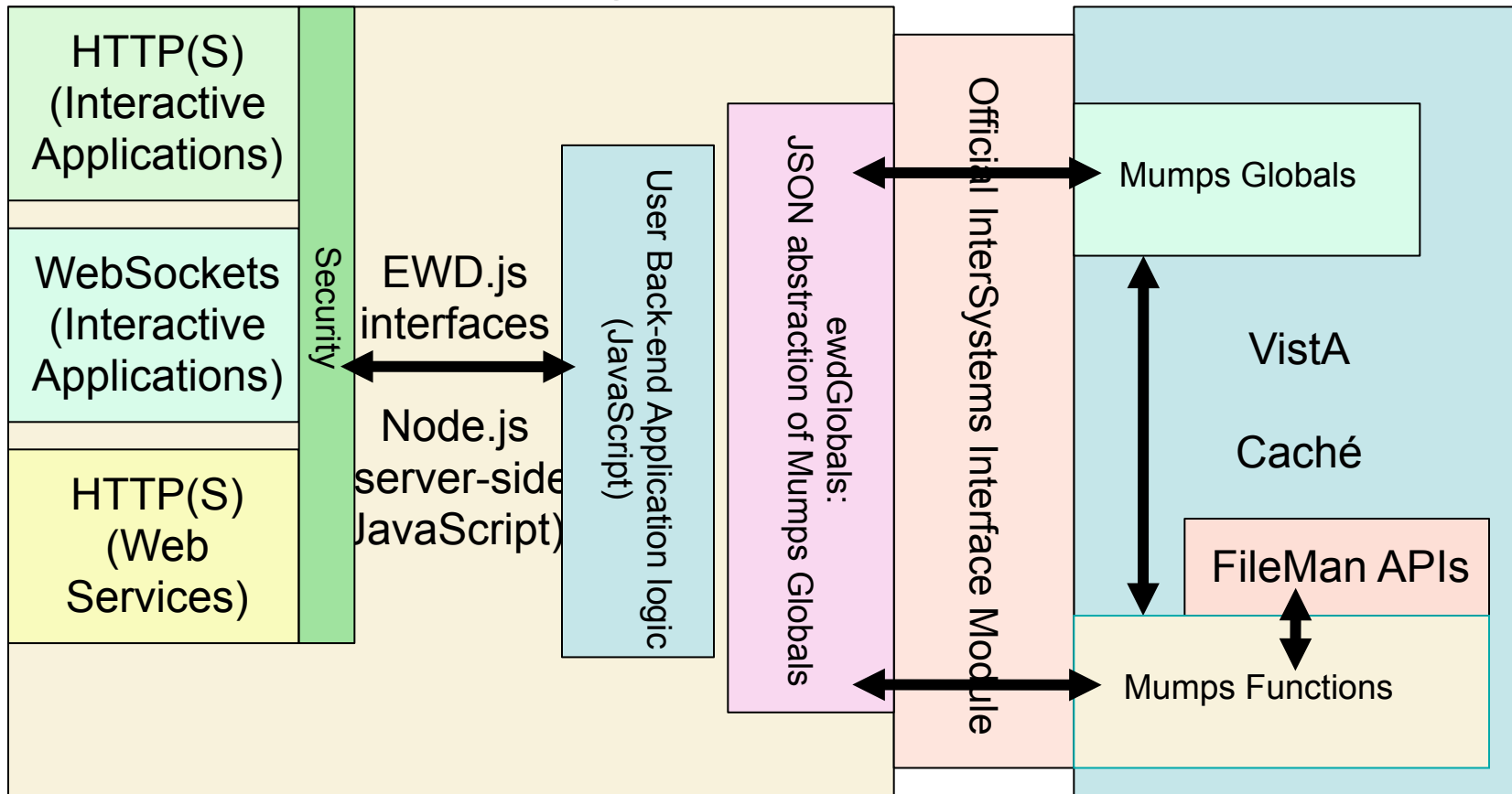
# EWD.js Architecture



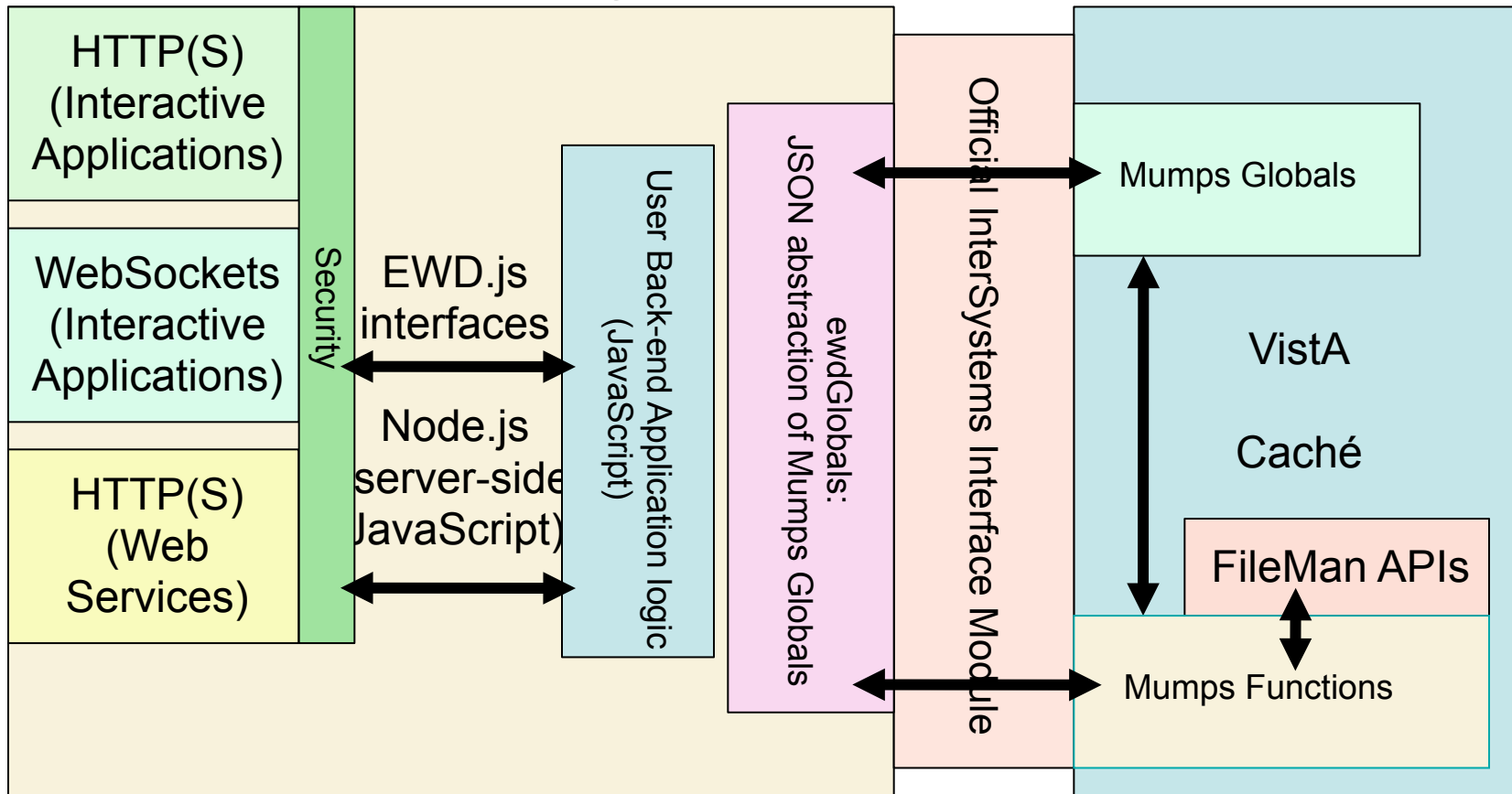
# EWD.js Architecture



# EWD.js Architecture

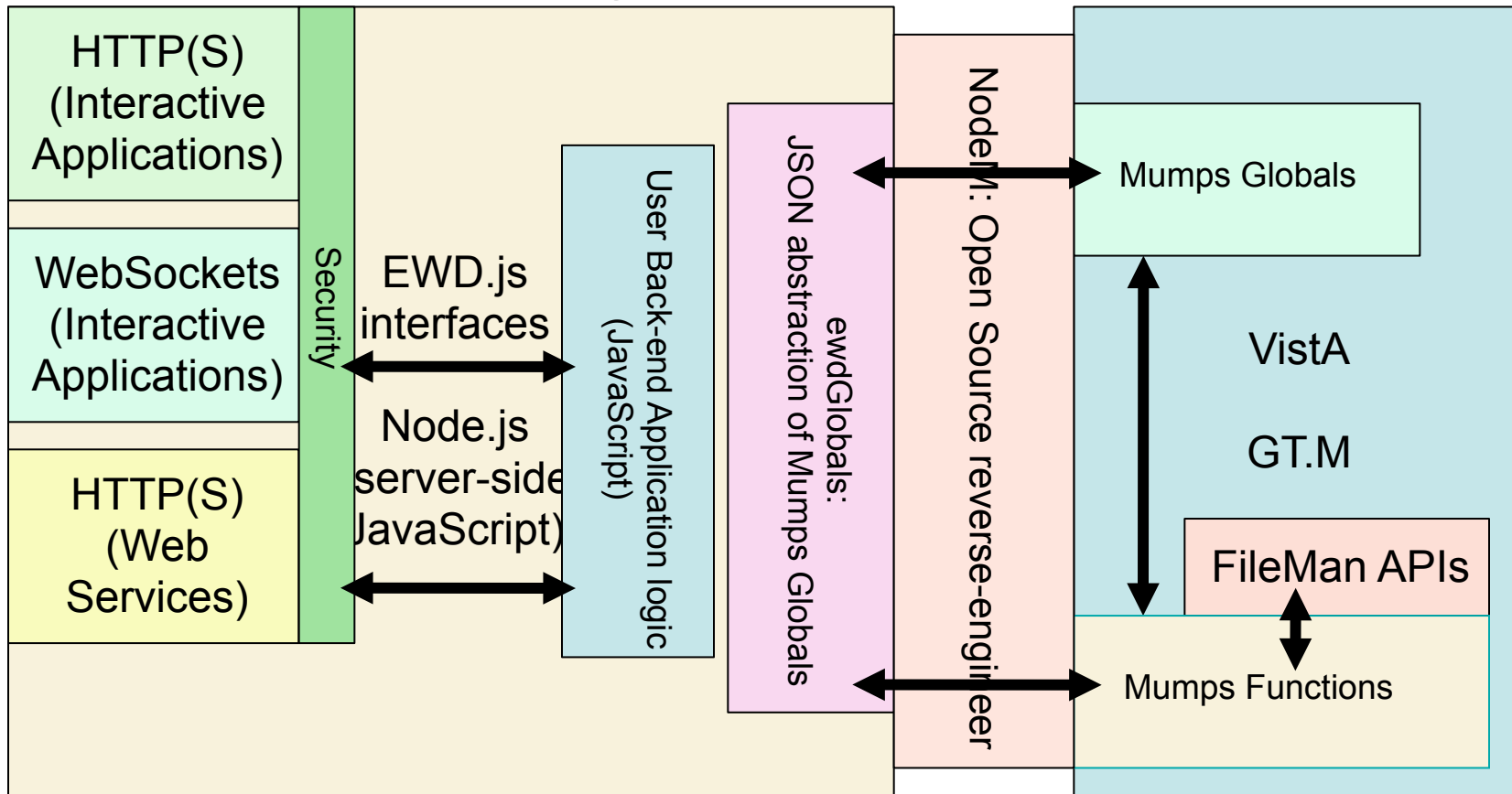


# EWD.js Architecture

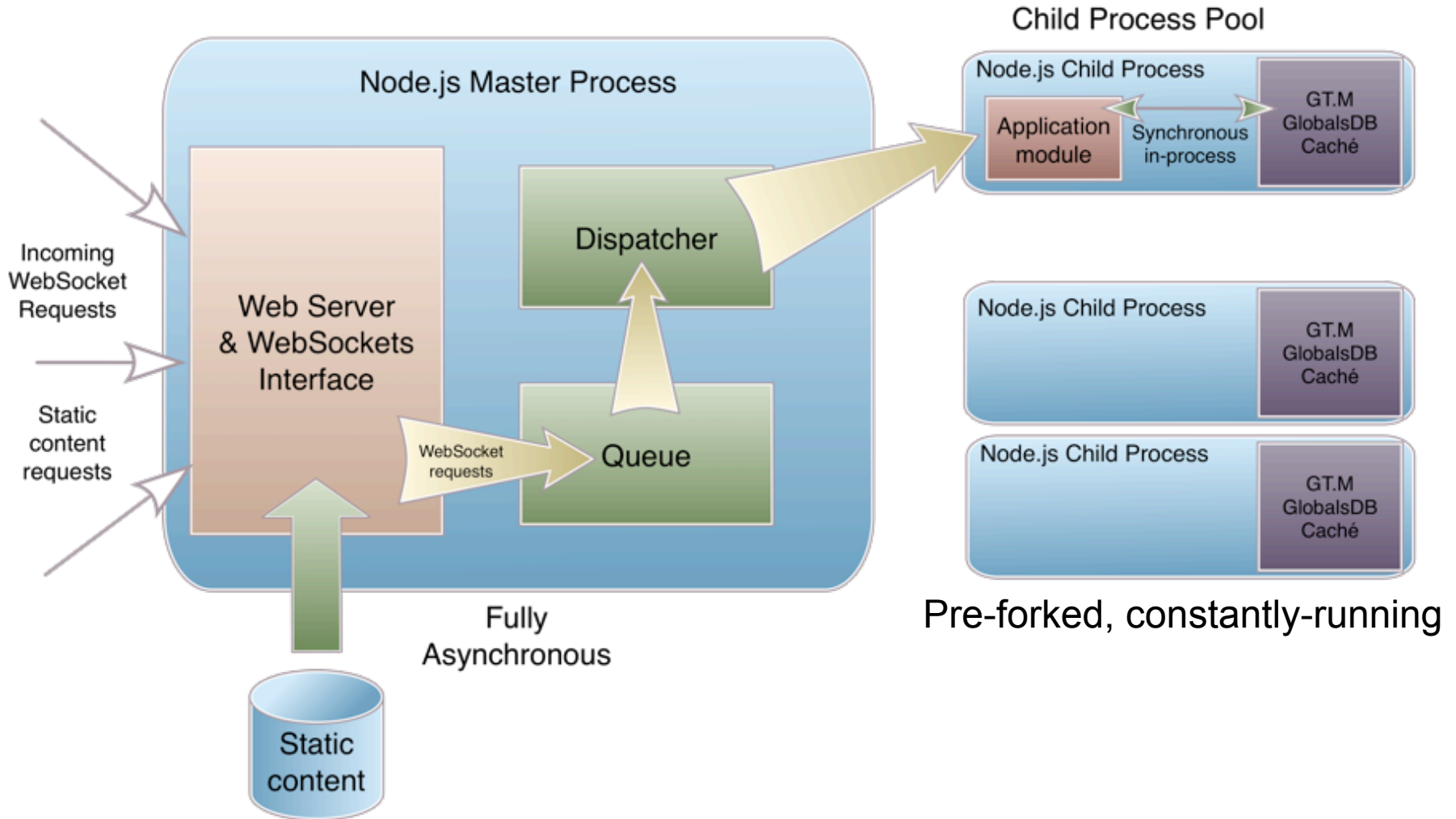




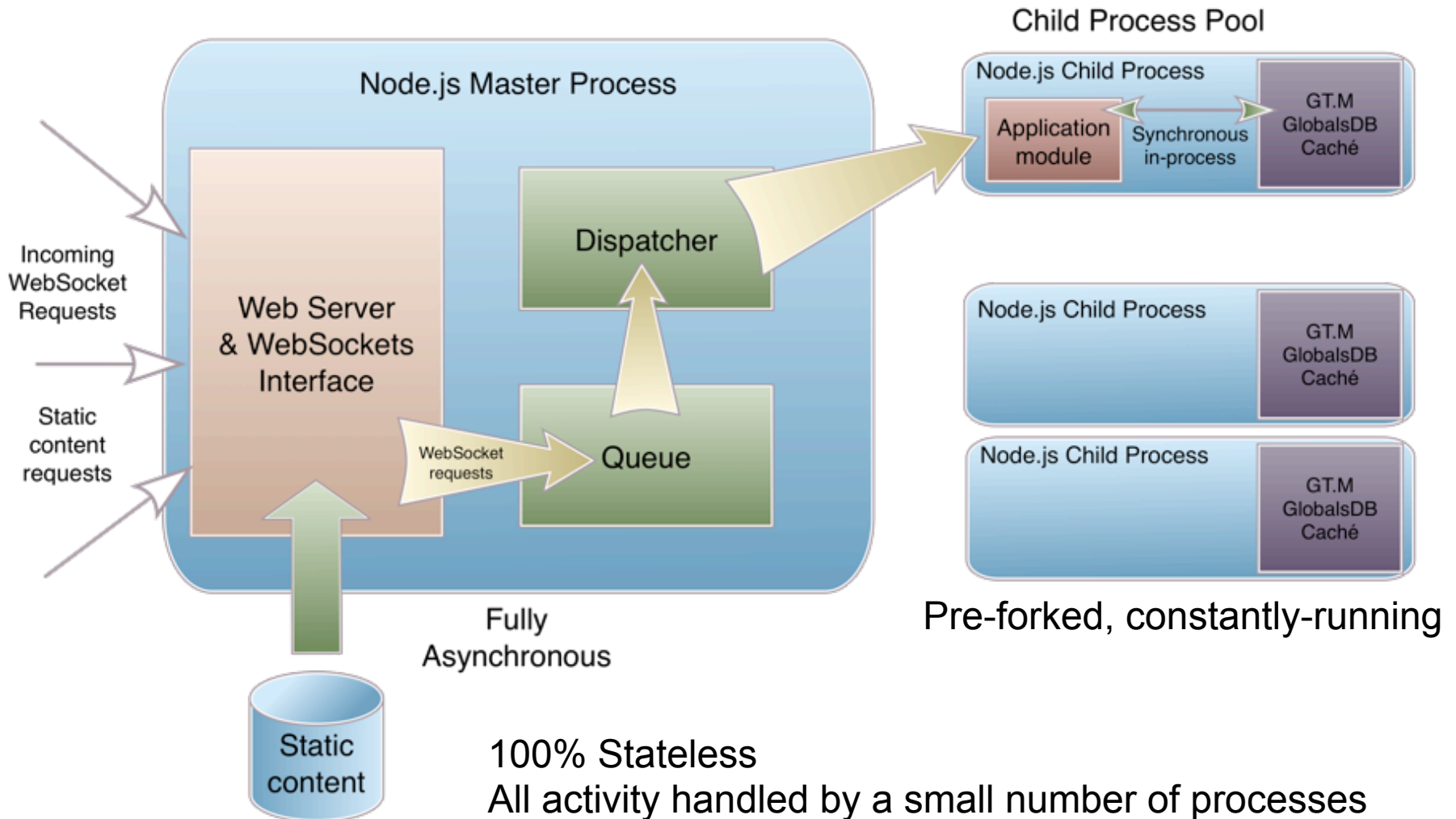
# EWD.js Architecture



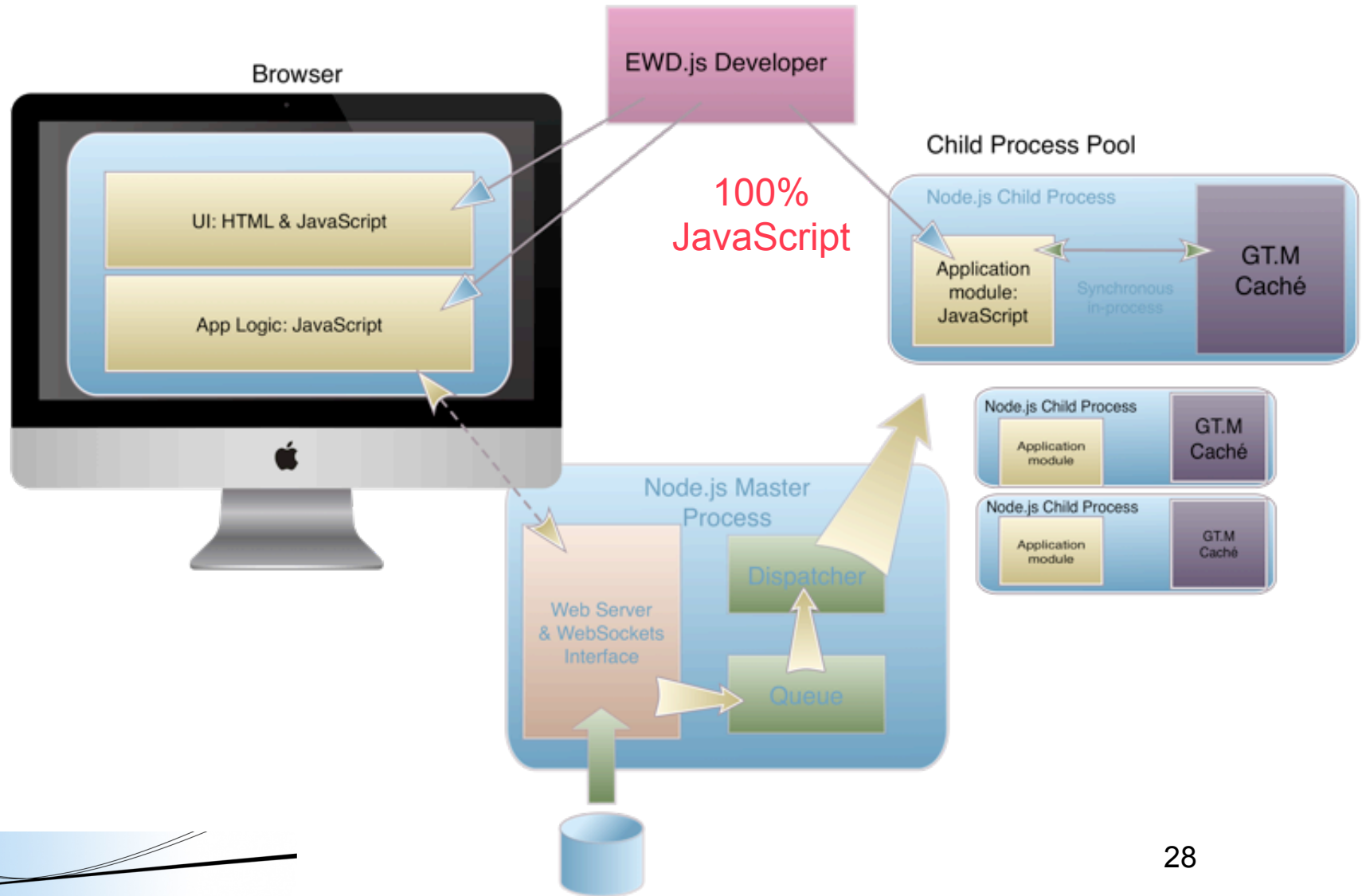
# EWD.js Architecture



# EWD.js Architecture



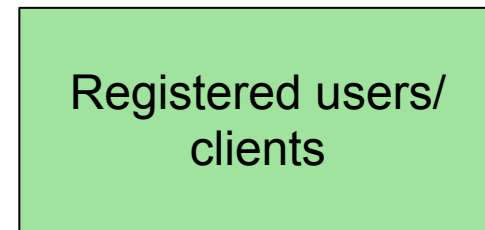
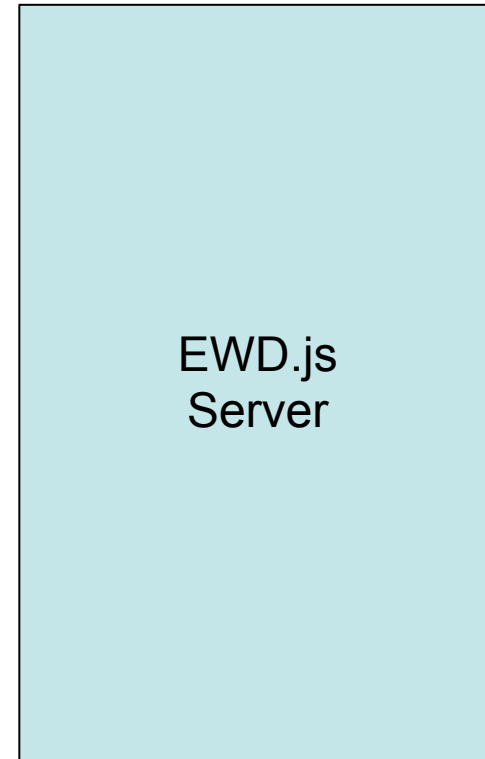
# EWD.js Development



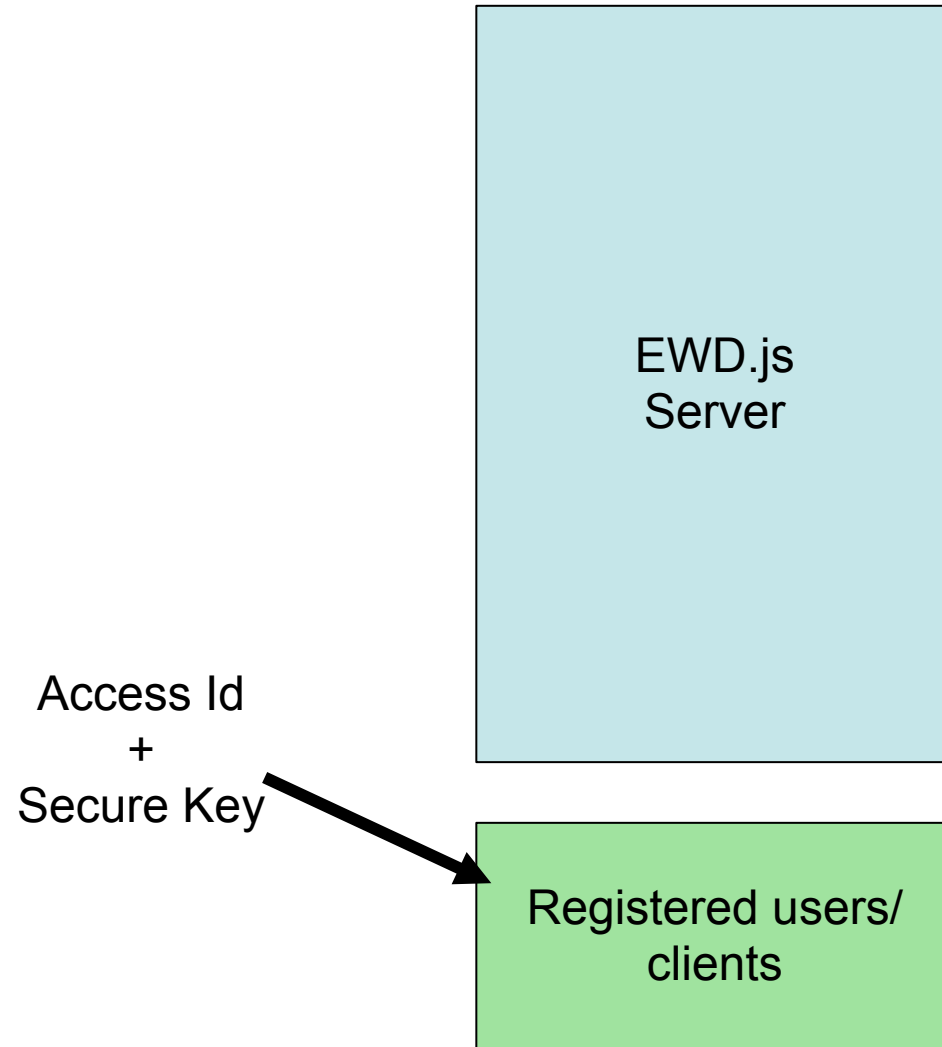
# Secured Web Services

- HTTP(S) Web Services built-in
- Need to prevent a free-for-all
- Uses same security as Amazon Web Services
  - eg for SimpleDB

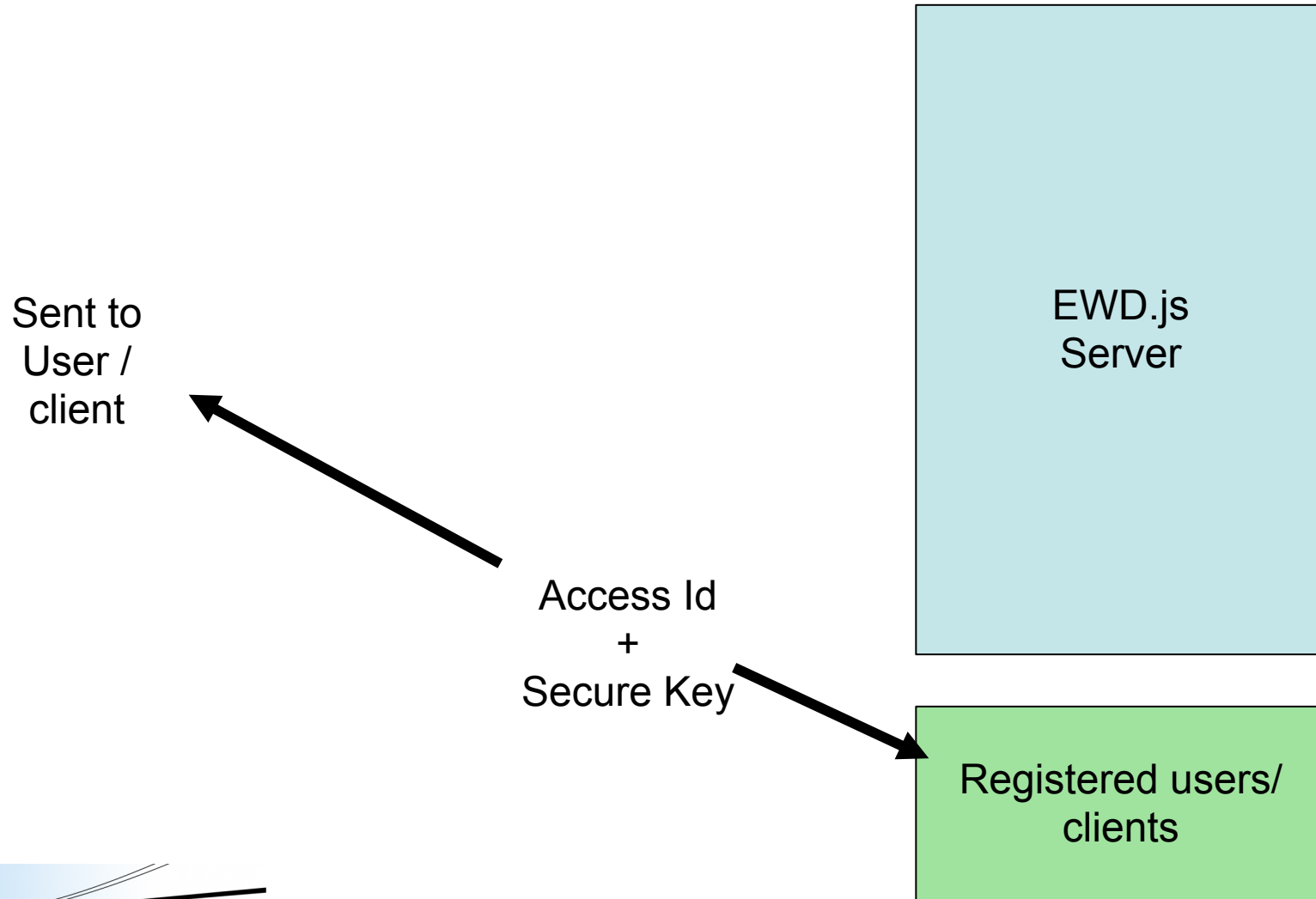
# Secured Web Services



# Secured Web Services

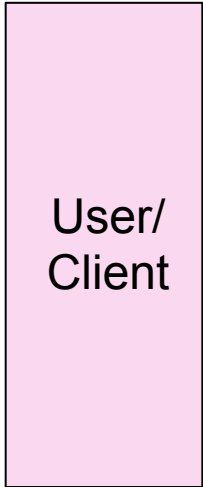


# Secured Web Services





# Secured Web Services

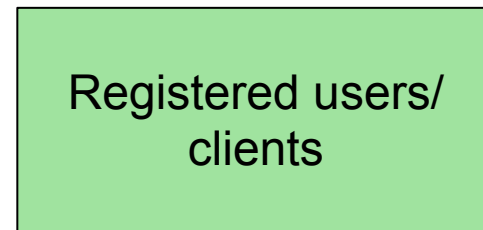


HTTP Request

&timestamp=xxxxxxx  
&accessId=rob12345



EWD.js  
Server



Registered users/  
clients

# Secured Web Services

User/  
Client

HTTP Request

Normalised

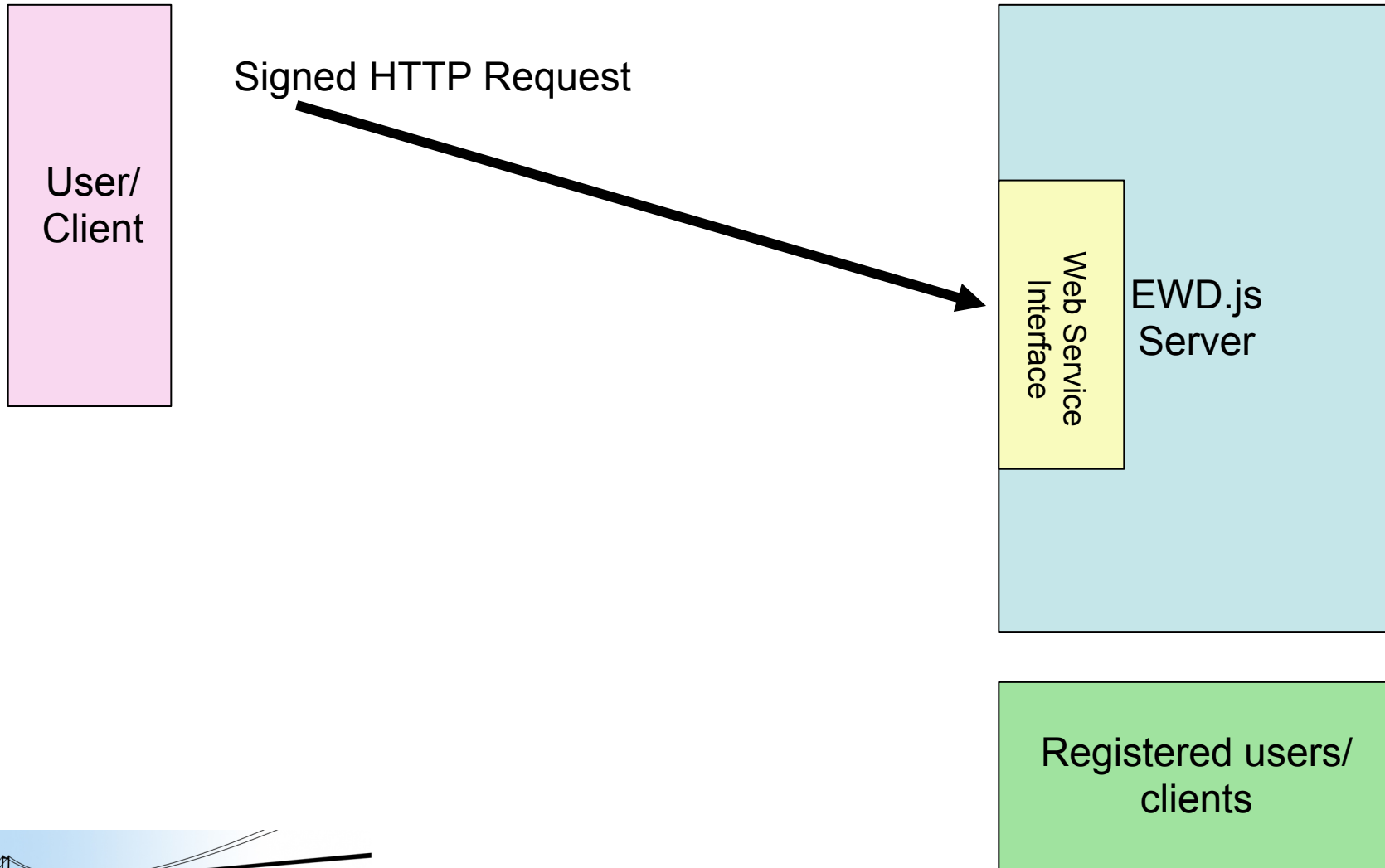
Create HMAC-SHA-256 digest  
using secret key

Add:  
&signature=xxxxxxxxxxxxxxxxxxxx

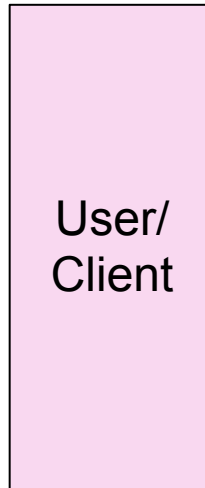
EWD.js  
Server

Registered users/  
clients

# Secured Web Services



# Secured Web Services

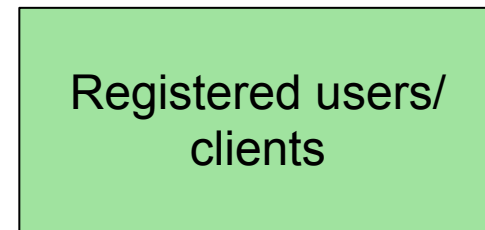
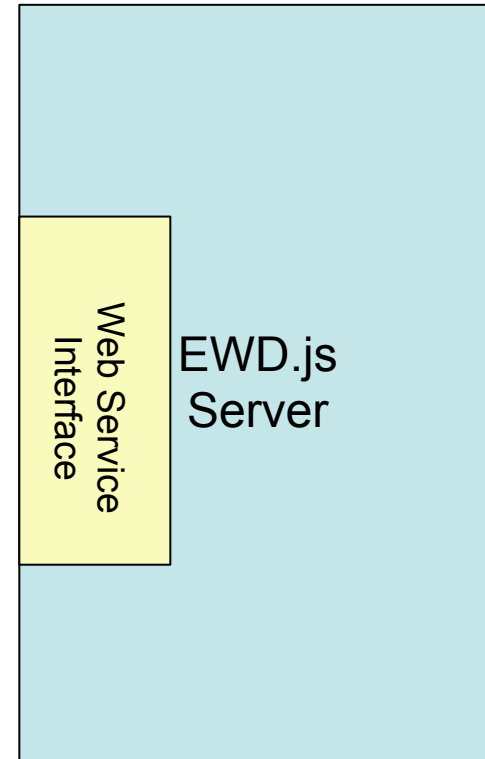


HTTP Request Processing

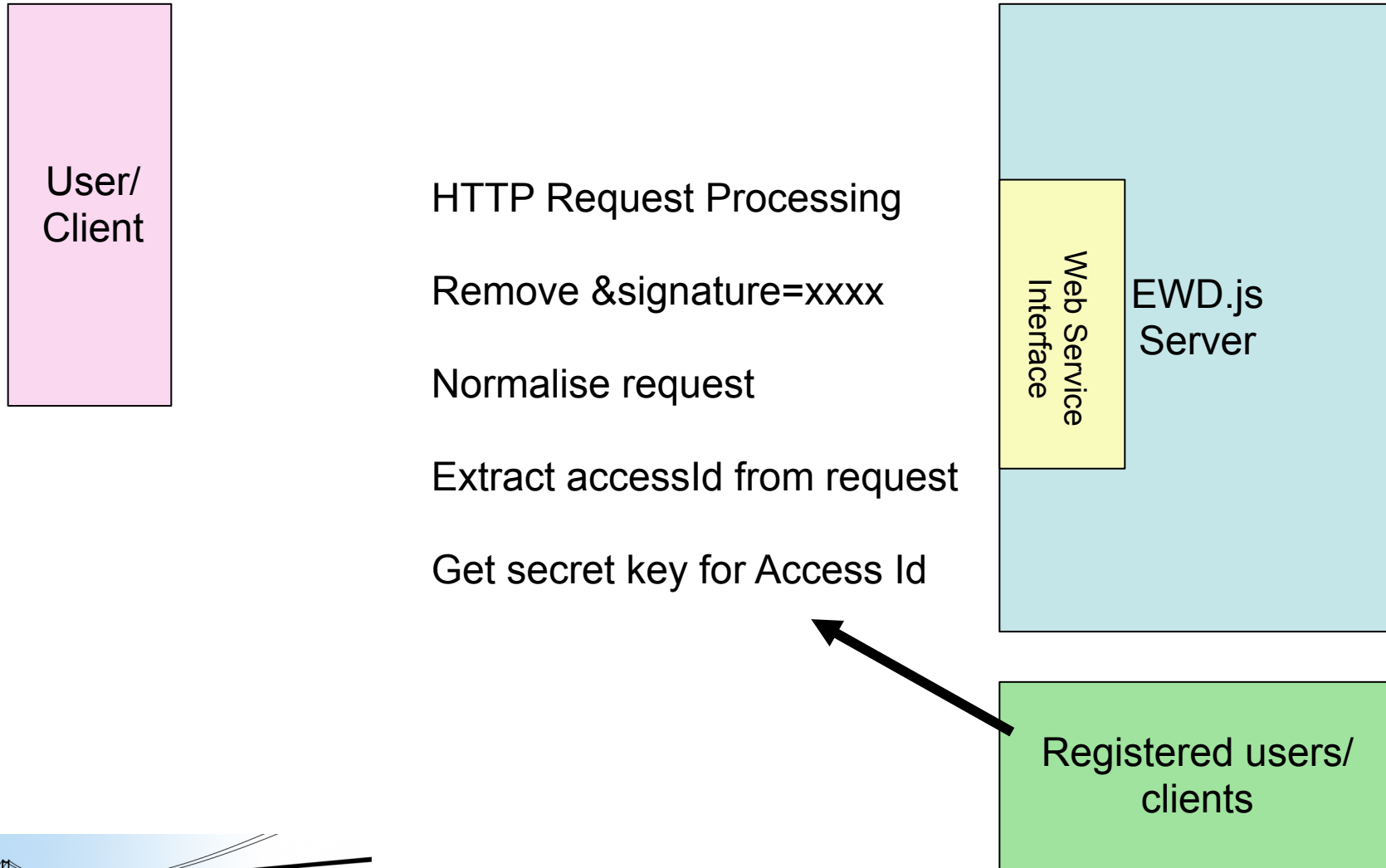
Remove &signature=xxxx

Normalise request

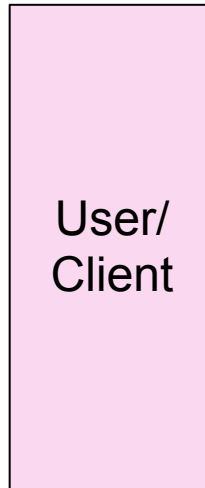
Extract accessId from request



# Secured Web Services



# Secured Web Services



HTTP Request Processing

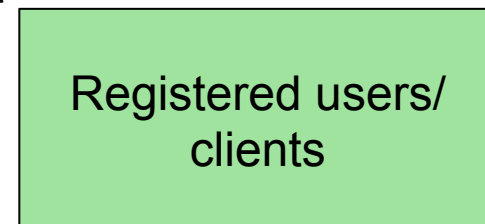
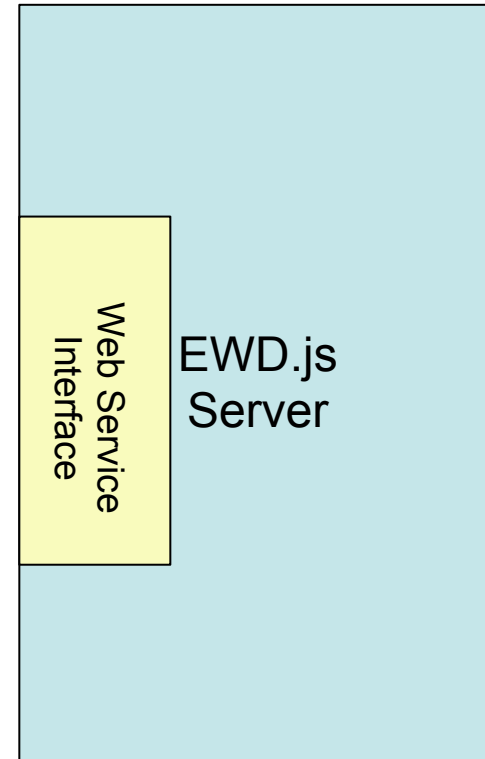
Remove &signature=xxxx

Normalise request

Extract accessId from request

Get secret key for Access Id

Generate HMAC-SHA256 digest



A green rectangular box containing the text "Registered users/clients".

# Secured Web Services

User/  
Client

HTTP Request Processing

Remove &signature=xxxx

Normalise request

Extract accessId from request

Get secret key for Access Id

Generate HMAC-SHA256 digest

- Match?



Web Service  
Interface

EWD.js  
Server

Registered users/  
clients

# Secured Web Services

User/  
Client

HTTP Request Processing

Remove &signature=xxxx

Normalise request

Extract accessId from request

Get secret key for Access Id

Generate HMAC-SHA256 digest

- Match?

- Don't Match?



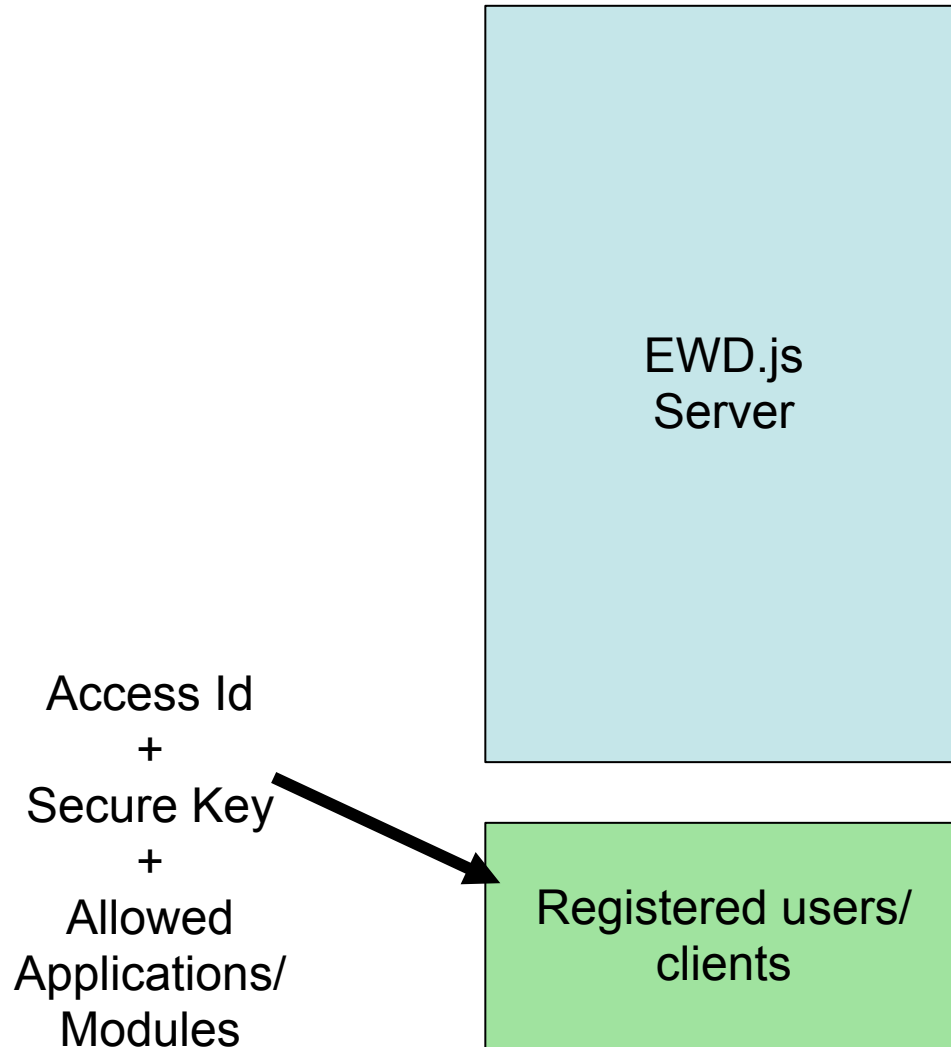
Web Service  
Interface

EWD.js  
Server

Registered users/  
clients



# Secured Web Services



# REST

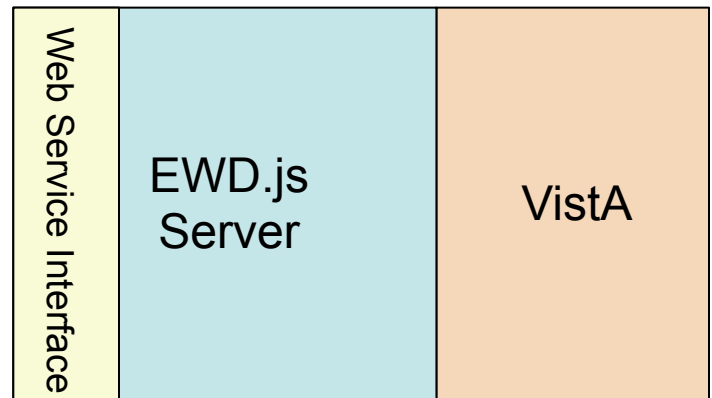
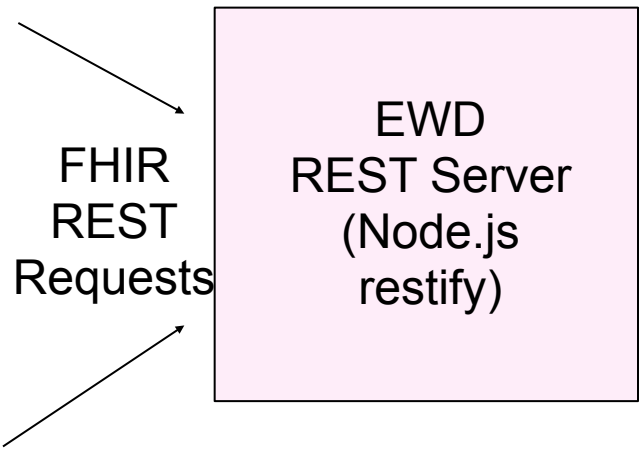
- Growing requirement in healthcare
  - many projects/initiatives in VA based on REST-ful interfaces to VistA
  - HL7 FHIR
    - REST is a key part of the standard

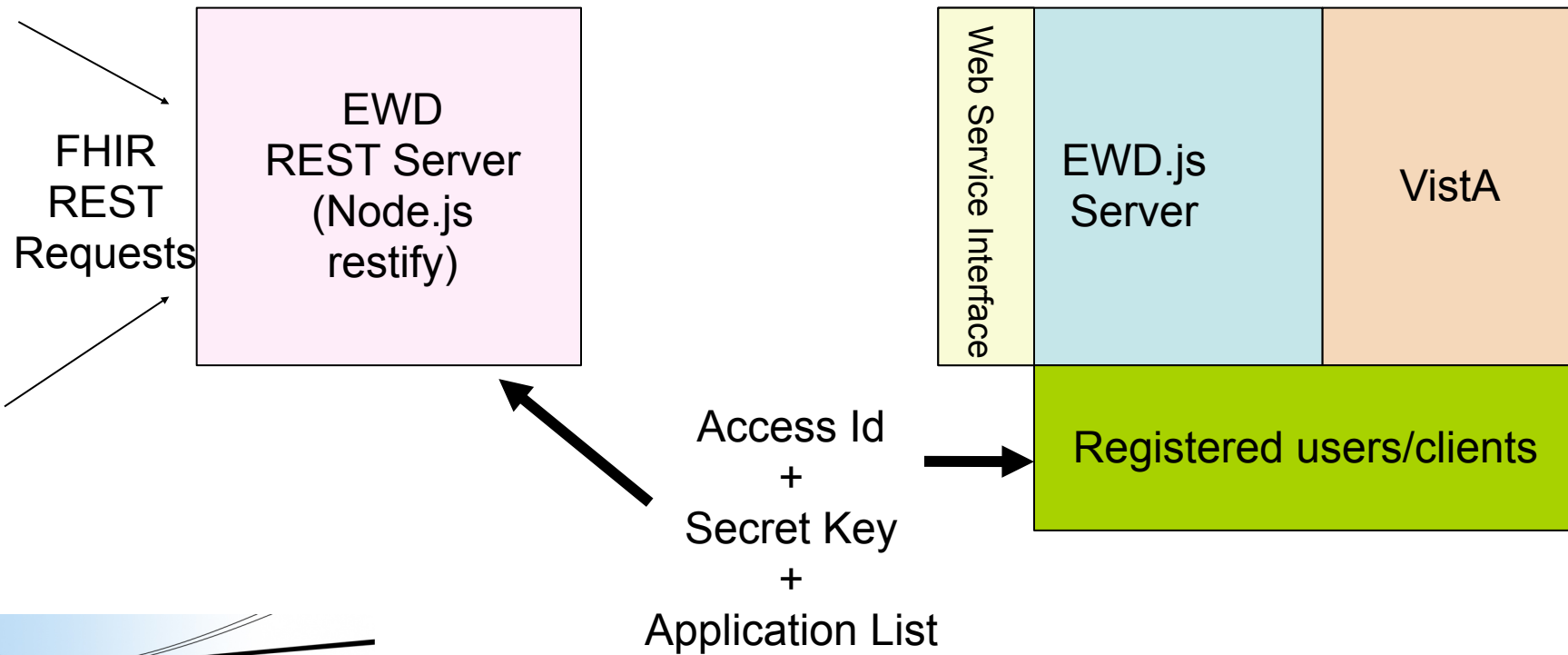
# REST

- Growing requirement in healthcare
  - many projects/initiatives in VA based on REST-ful interfaces to VistA
  - HL7 FHIR
    - REST is a key part of the standard
  - EWD.js doesn't support REST
    - implement REST interface?

# REST

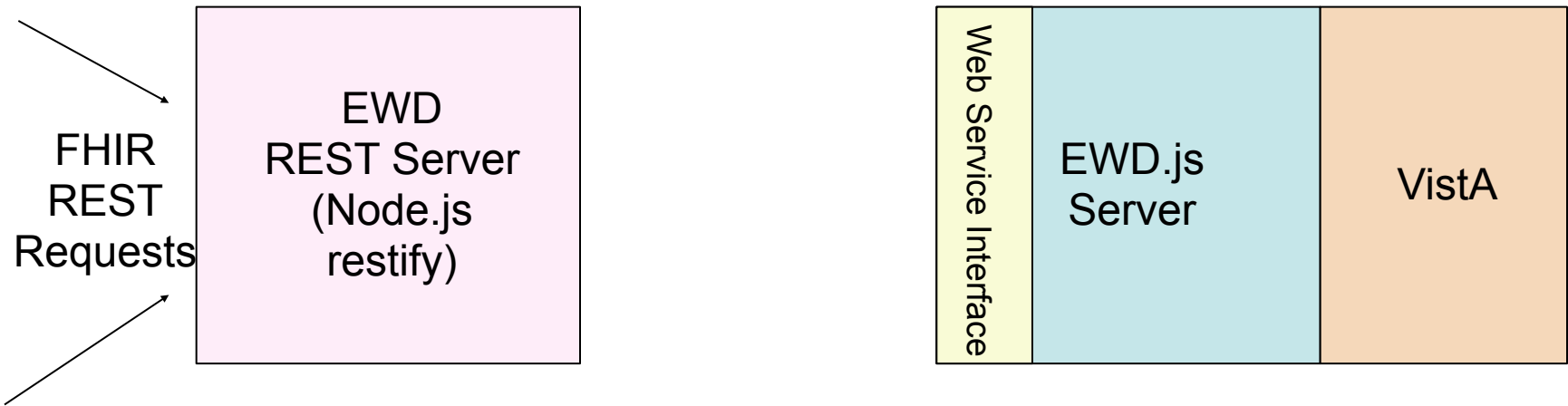
- Growing requirement in healthcare
  - many projects/initiatives in VA based on REST-ful interfaces to VistA
  - HL7 FHIR
    - REST is a key part of the standard
  - EWD.js doesn't support REST
    - implement REST interface?
    - instead, separate EWD REST Server
      - Node.js module: Restify
        - » de facto standard REST implementation for Node.js
        - » tried, tested, stable, fast
        - » customisable!





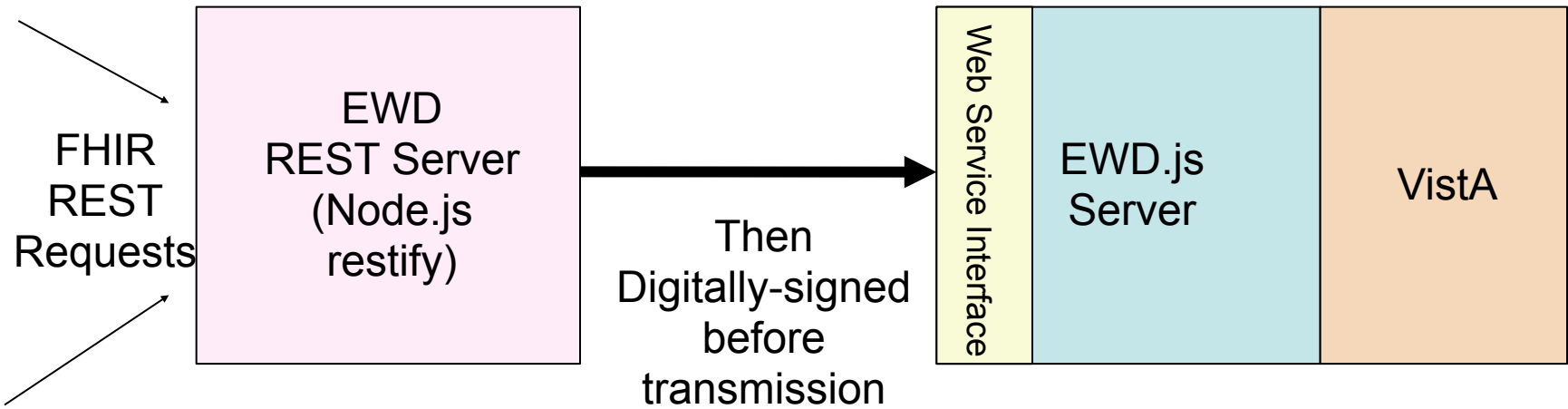
## URL Re-writing:

REST requests rewritten  
As EWD.js HTTP Web  
Service Requests

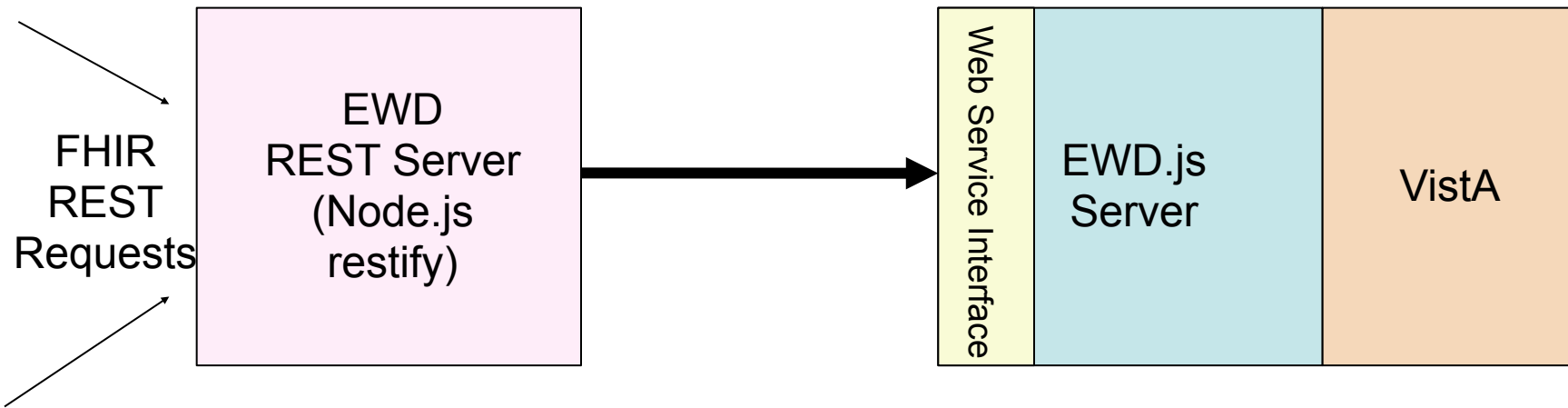


## URL Re-writing:

REST requests rewritten  
As EWD.js HTTP Web  
Service Requests



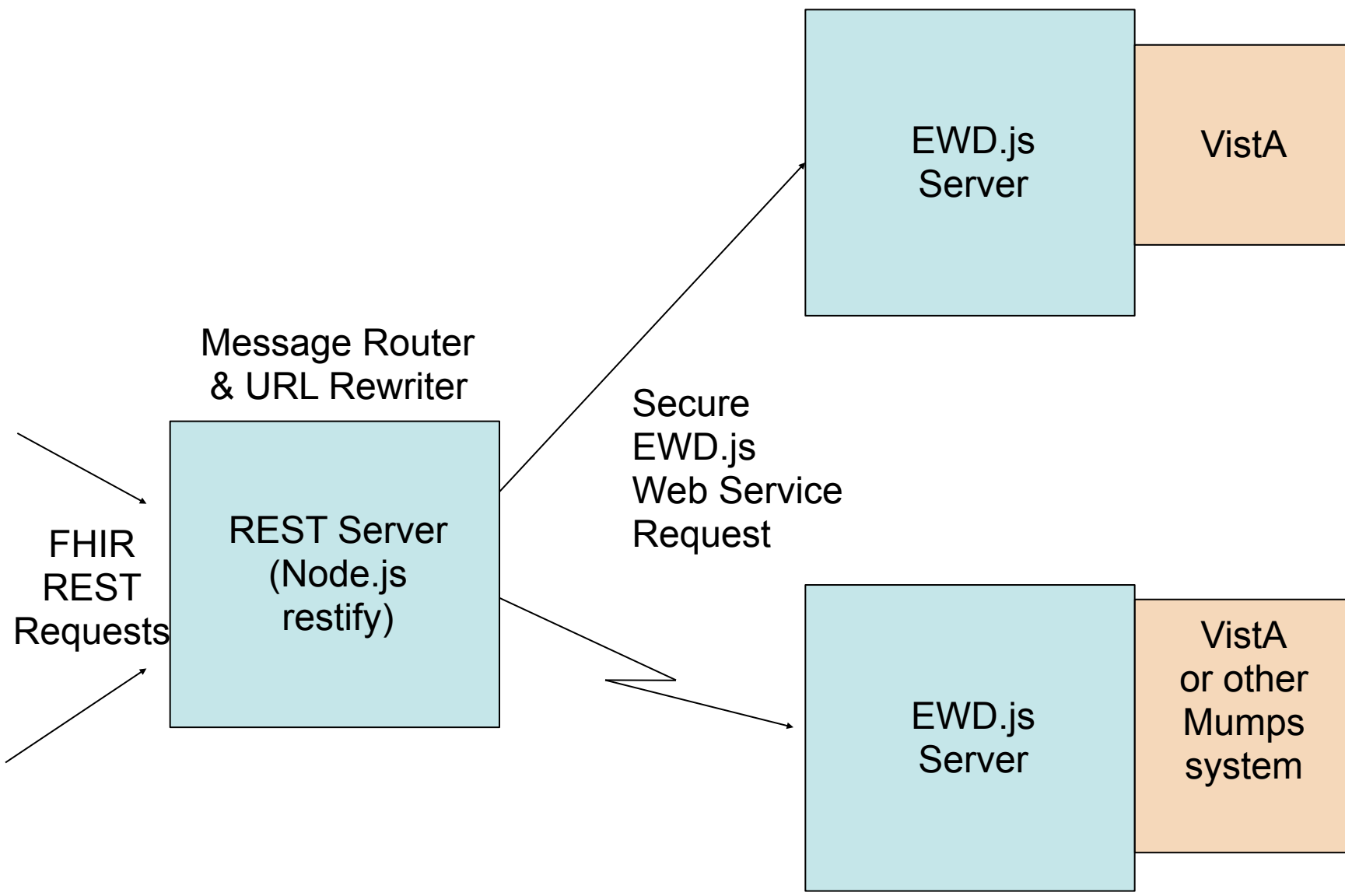


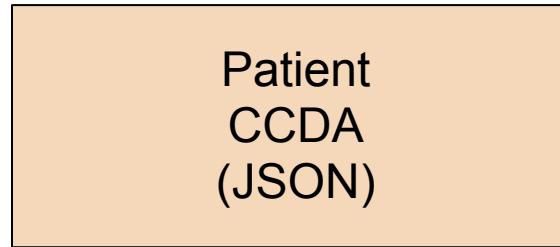


No Change required  
To EWD.js

# URL Re-Router

- Get Observations for patient 1234 from EC2 server
- <http://192.168.1.2:8081/fhir/ec2/patient/@1234/observation>
- Get Observations for patient 1234 from Oroville server
- <http://192.168.1.2:8081/fhir/oroville/patient/@1234/observation>

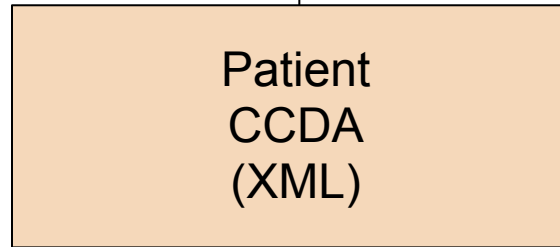




`_setDocument()`



Stored in Mumps DB



File

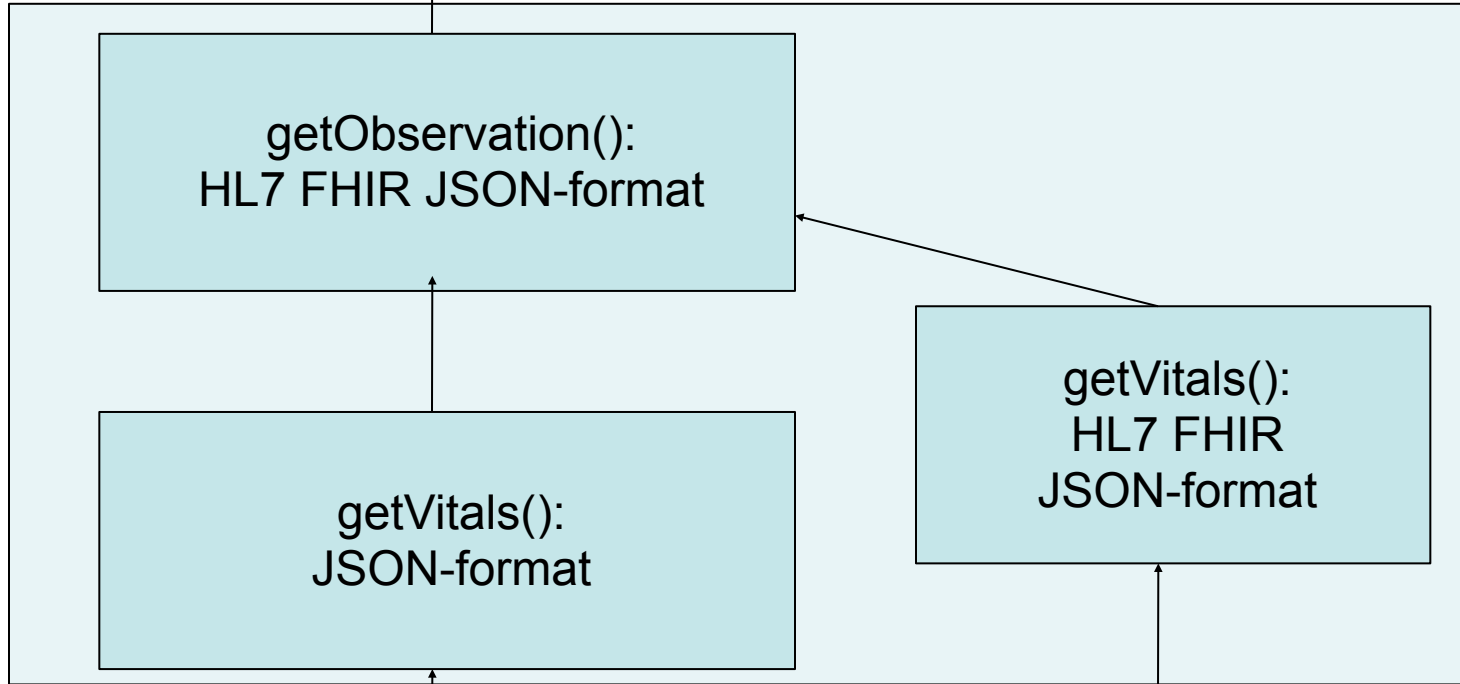
George Lilly's  
work



Forwarded to FHIR REST Server

EWD.js

EWD.js  
Back-end  
FHIR  
Module



getObservation():  
HL7 FHIR JSON-format

getVitals():  
JSON-format

getVitals():  
HL7 FHIR  
JSON-format

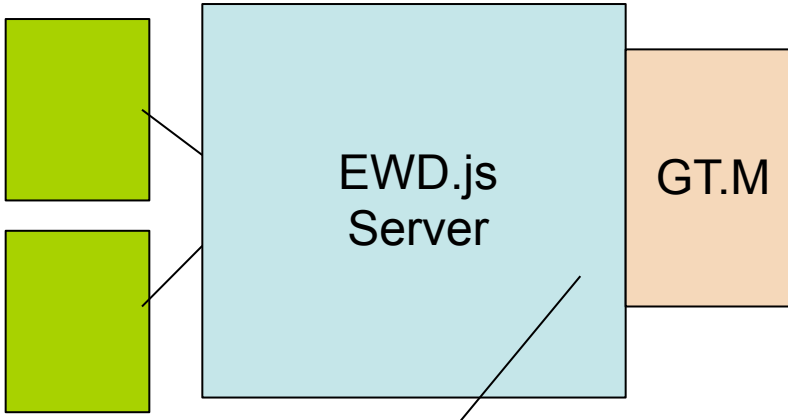
Mumps function  
API wrapper

JSON-formatted CCDA  
Stored in Mumps  
Or MongoDB

VistA

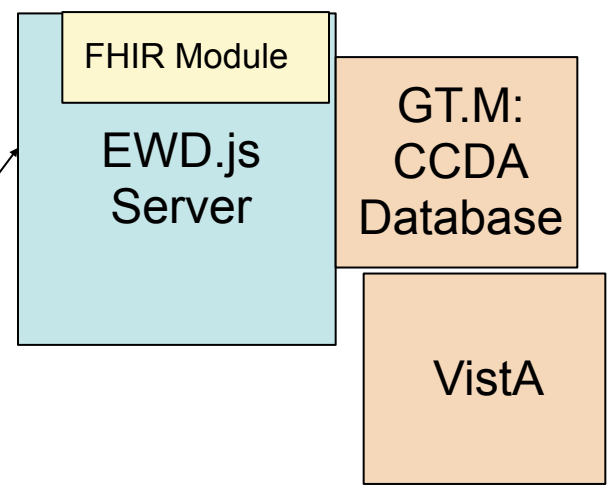
Josh Mandel's  
Blue-button logic  
Adapted for EWD.js



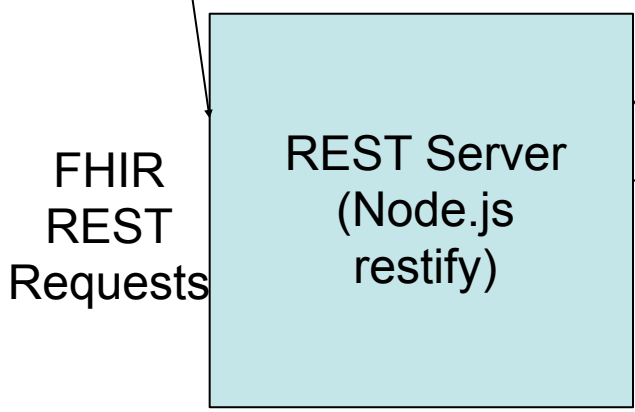


Blood-pressure Application

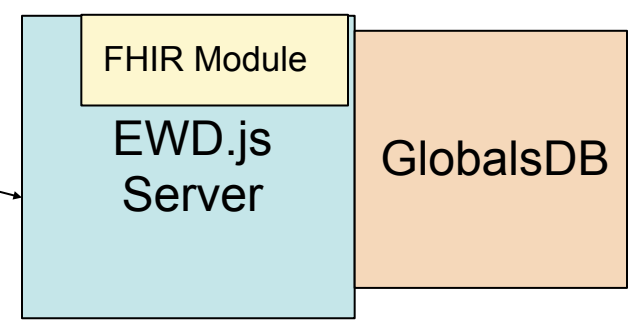
### George Lilly's Rackspace Server



Secure EWD.js Web Service Requests

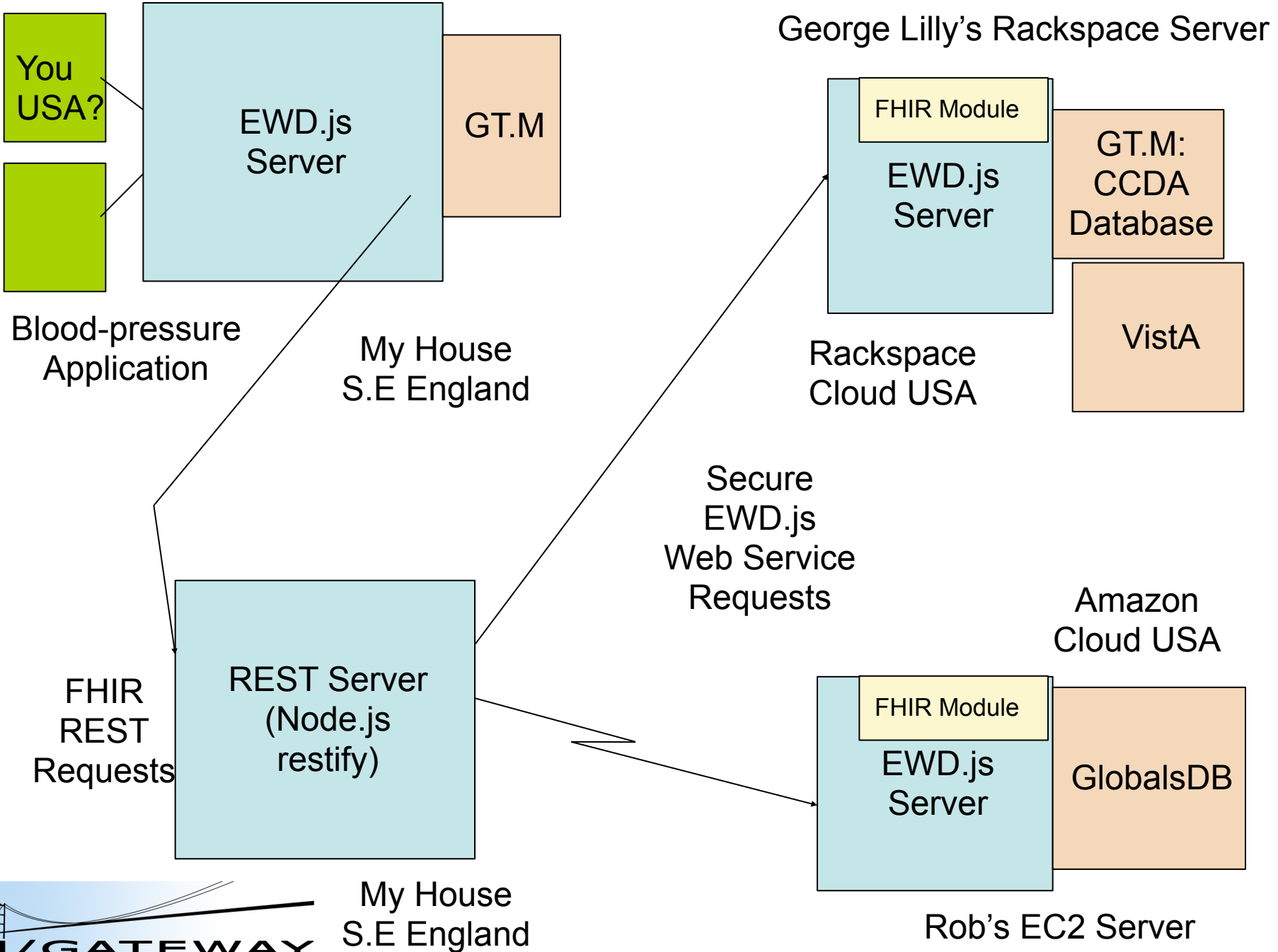


FHIR REST Requests



### Rob's EC2 Server



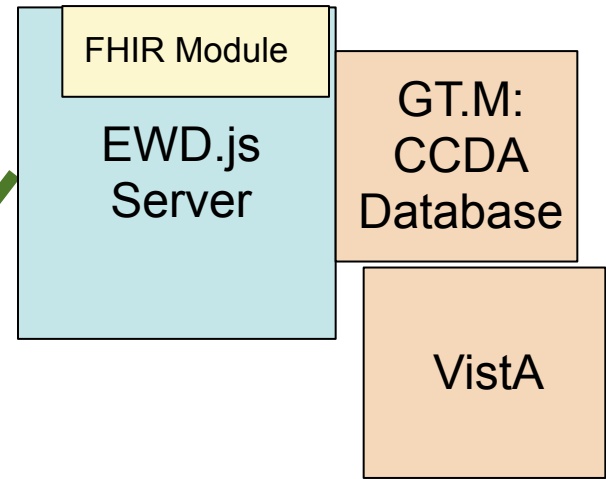


# Try it

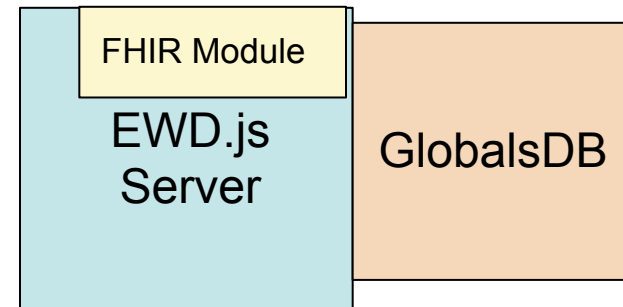
- <https://www.mgateway.com:38080/ewd/FHIRDemo/index.html>
- Username: rob
- Password: secret
- Note: not designed for use with IE!
  - use Chrome, Firefox or Safari



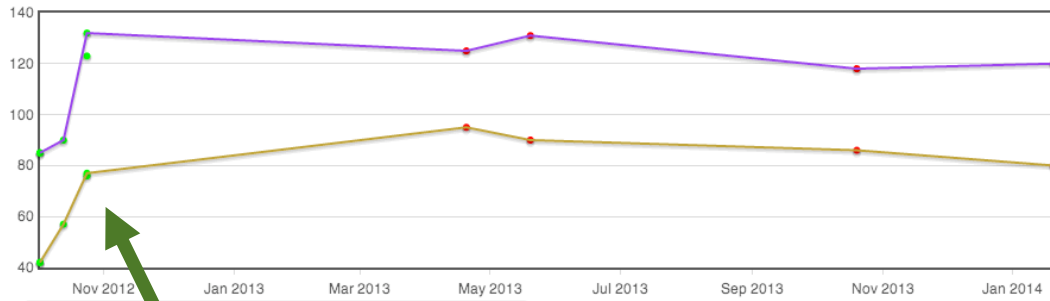
# George Lilly's Rackspace Server



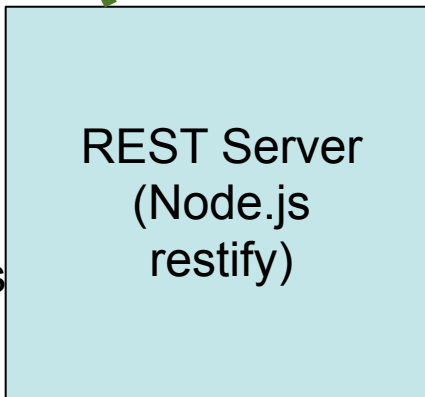
HL7 FHIR  
Observation



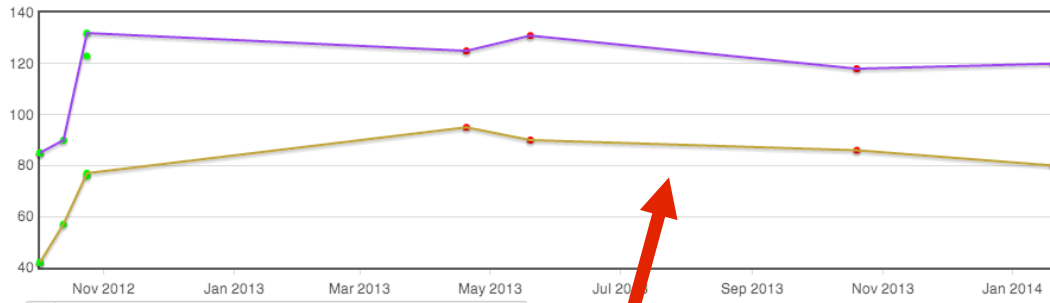
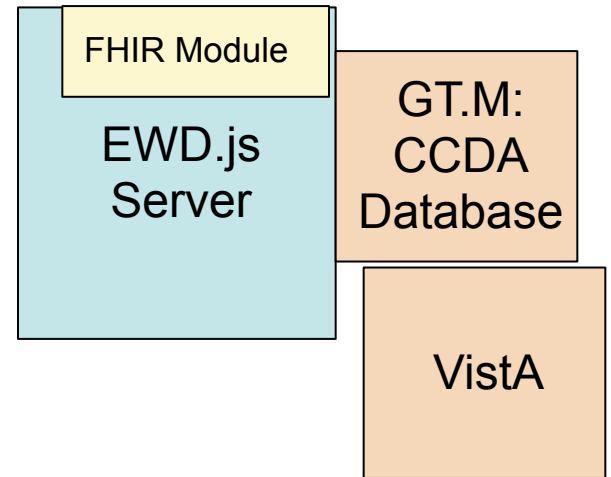
# Rob's EC2 Server



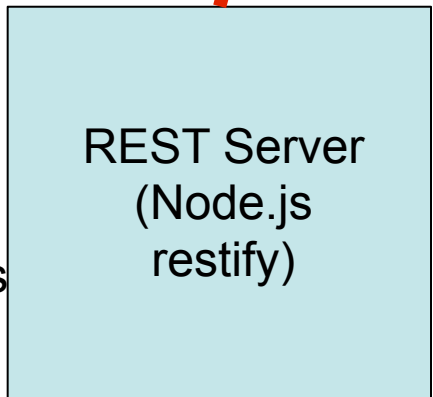
FHIR  
REST  
Requests



# George Lilly's Rackspace Server

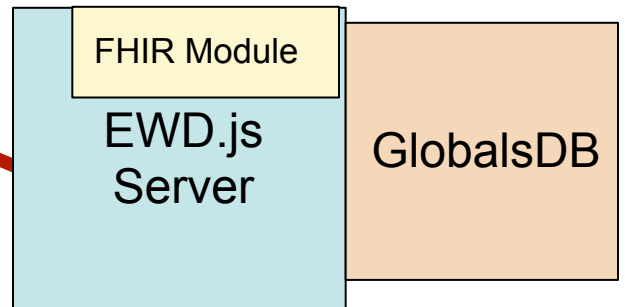


FHIR  
REST  
Requests



REST Server  
(Node.js  
restify)

HL7 FHIR  
Observation



EWD.js  
Server

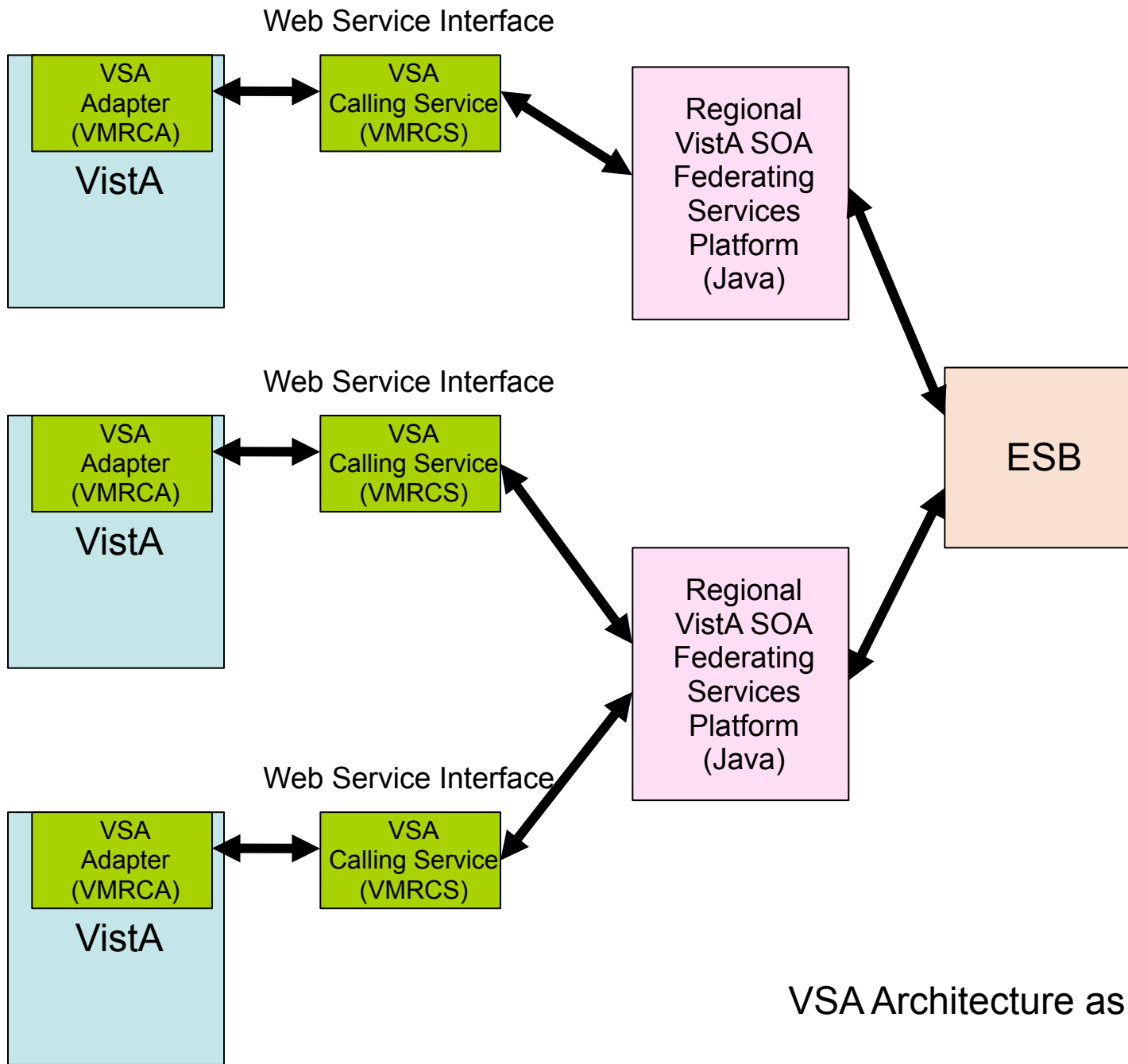
GlobalsDB

Rob's EC2 Server

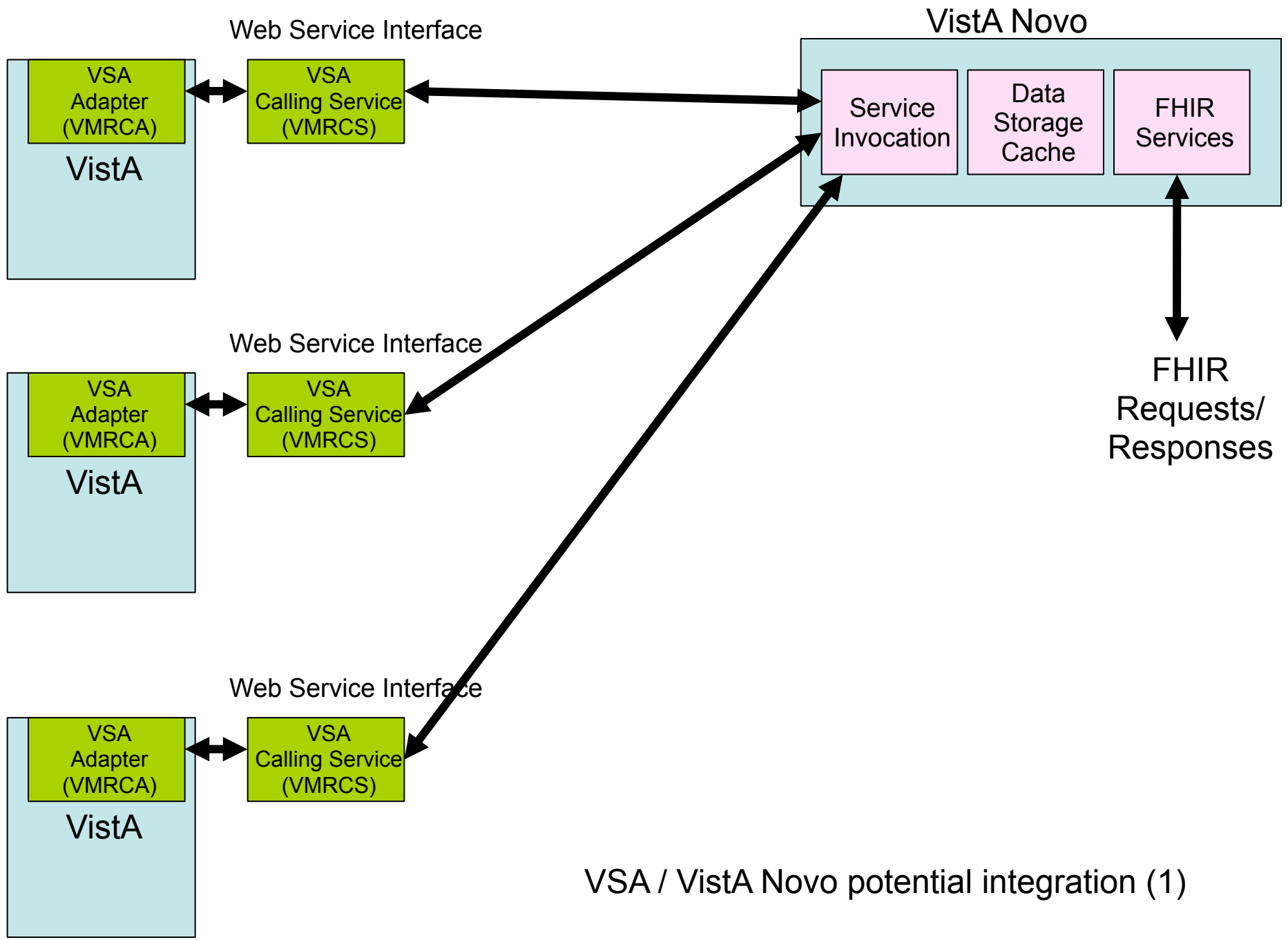


# EWD.js in relation to:

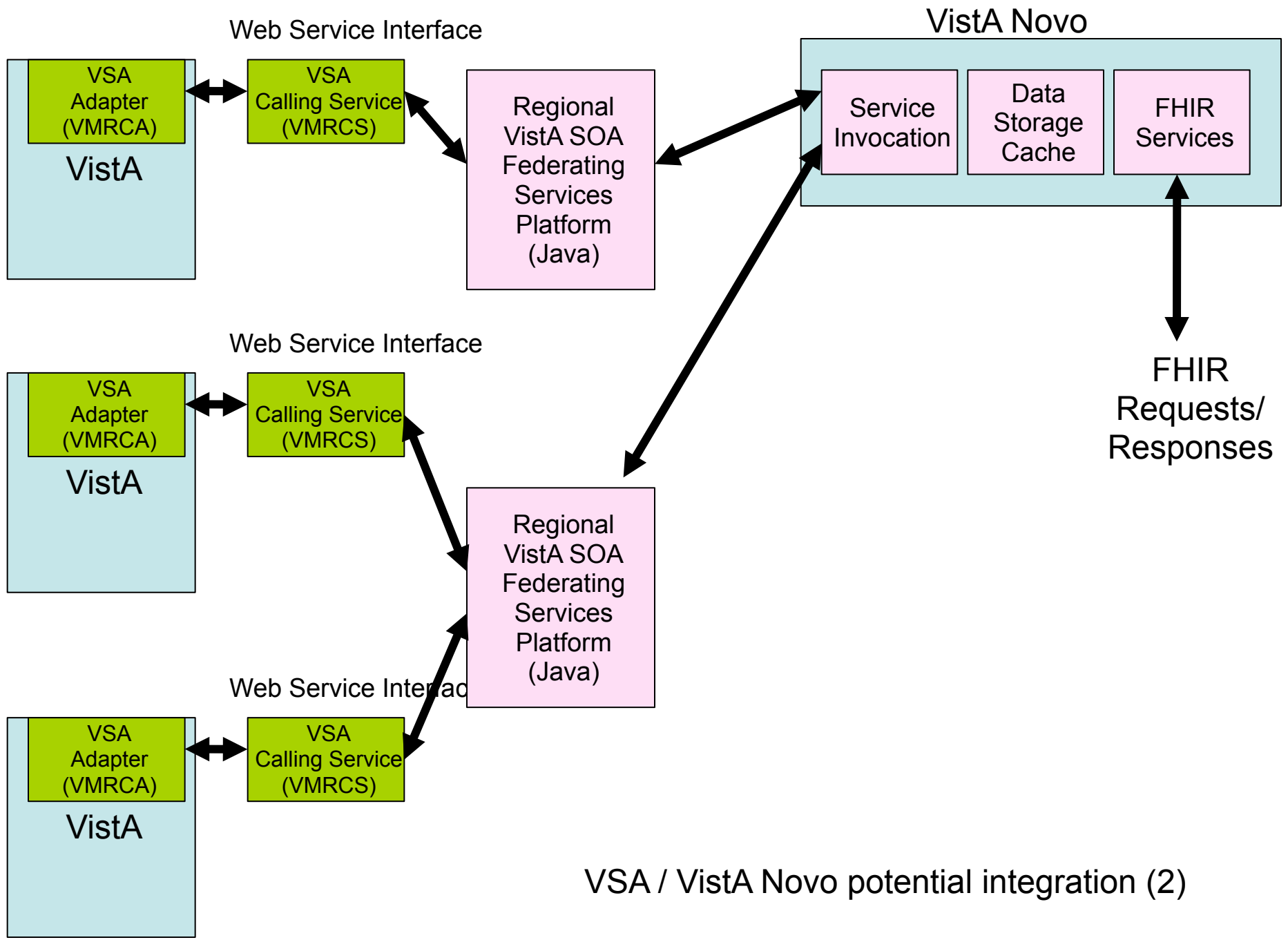
- VistA Novo
- VSA



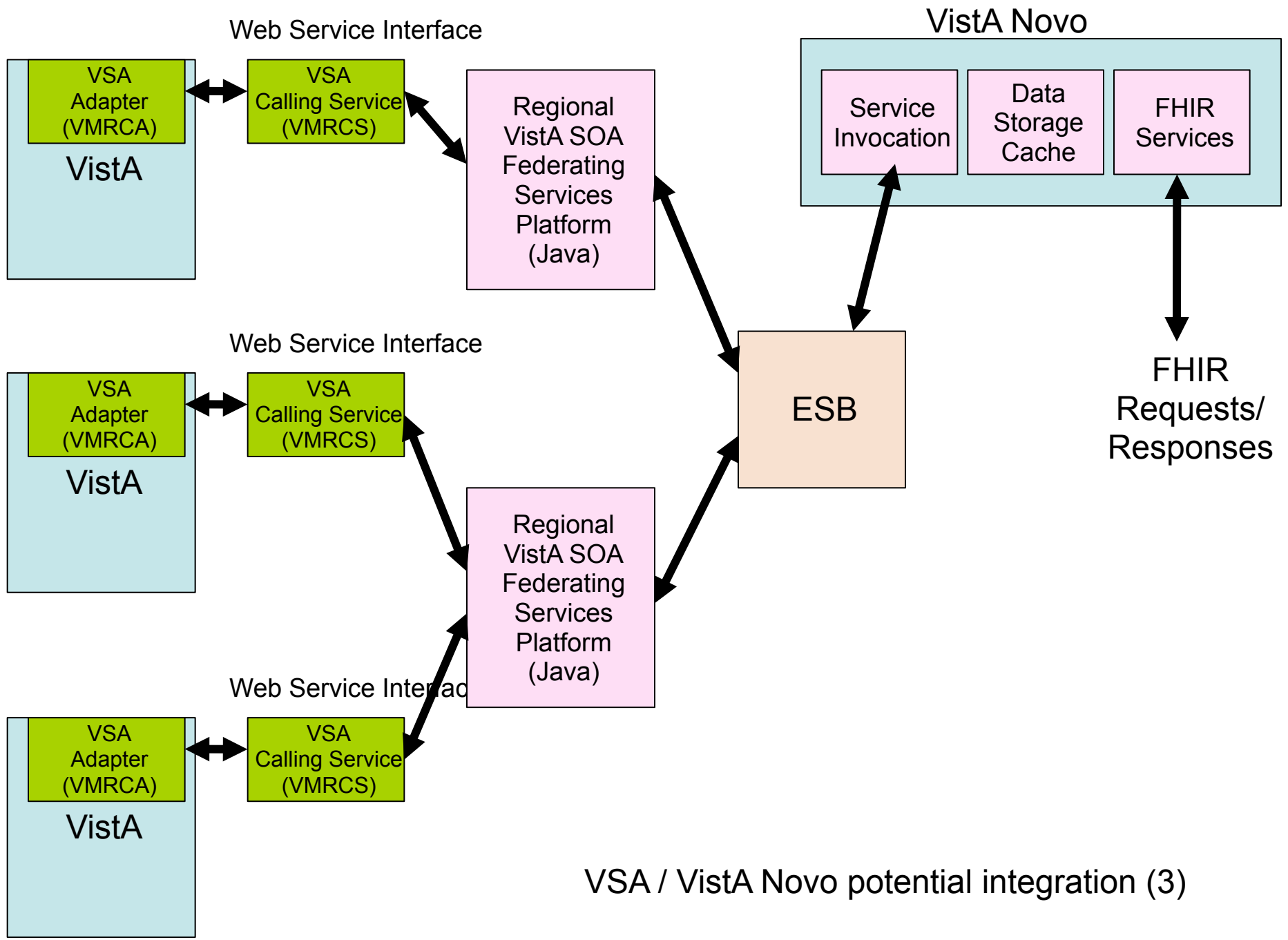
VSA Architecture as currently proposed



VSA / VistA Novo potential integration (1)

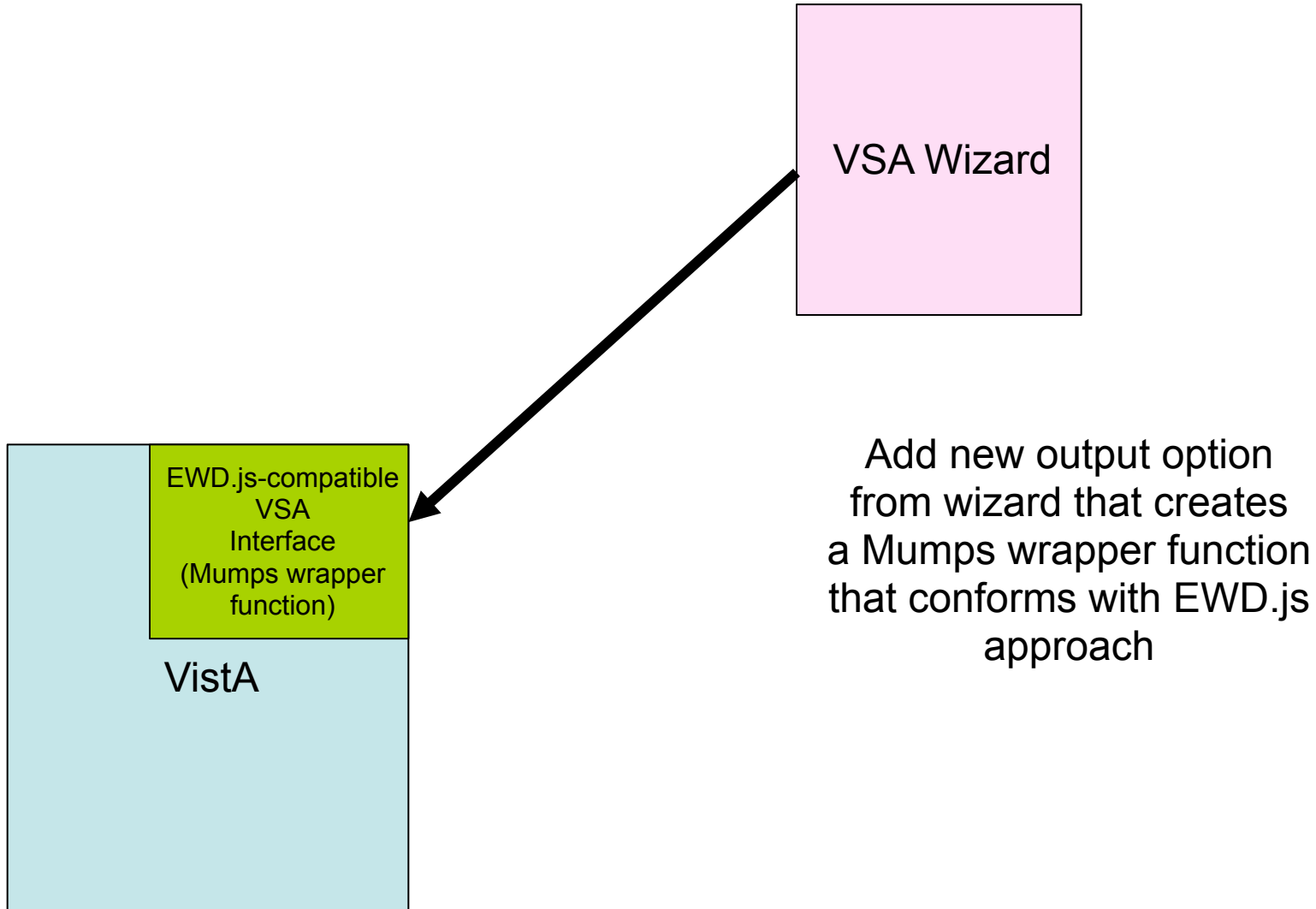


VSA / VistA Novo potential integration (2)

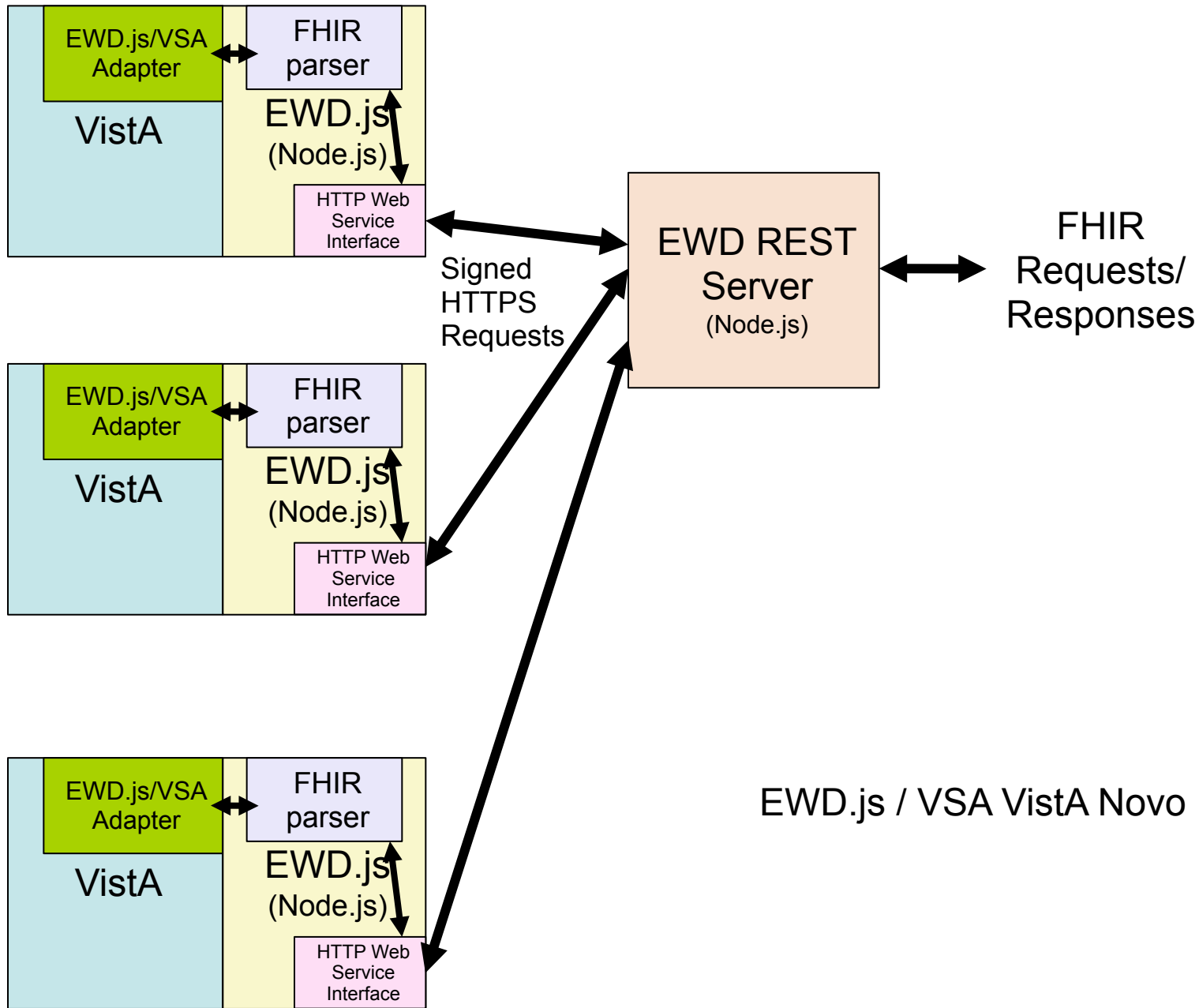


VSA / VistA Novo potential integration (3)

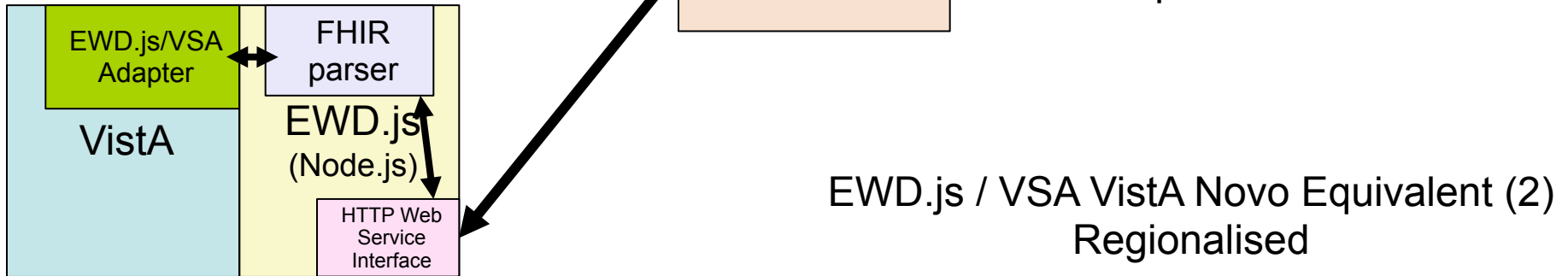
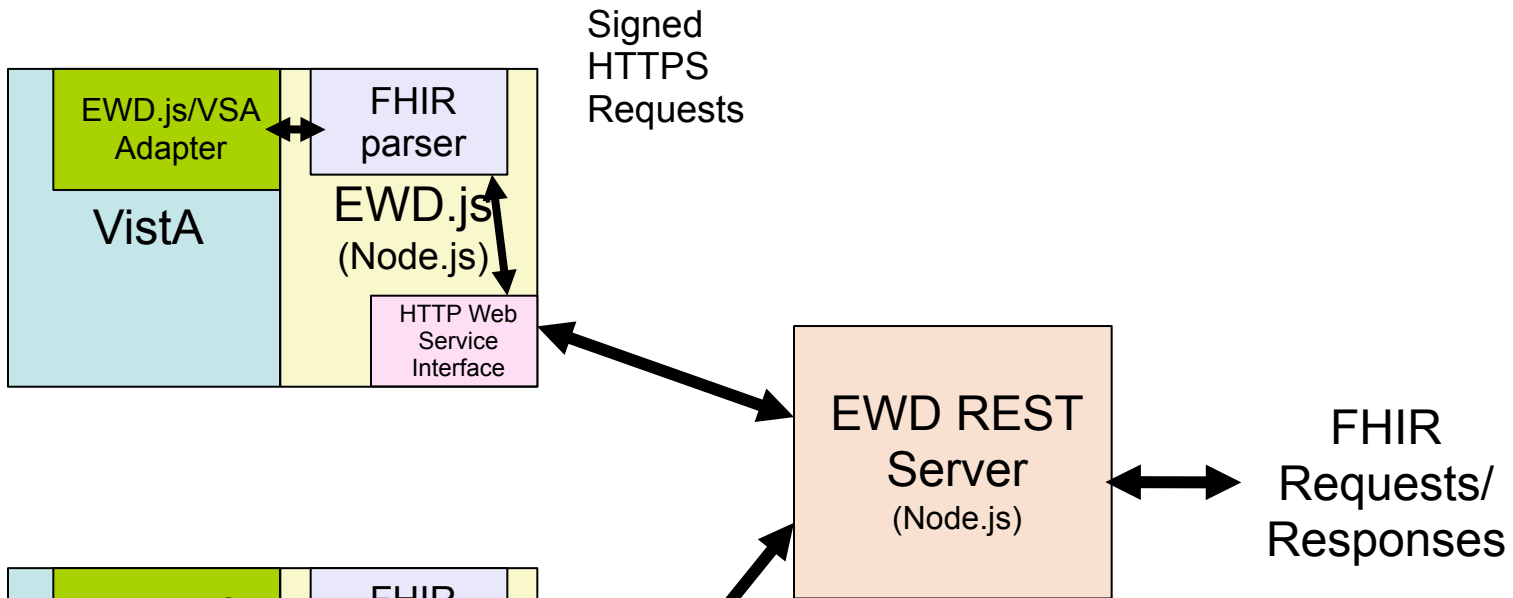
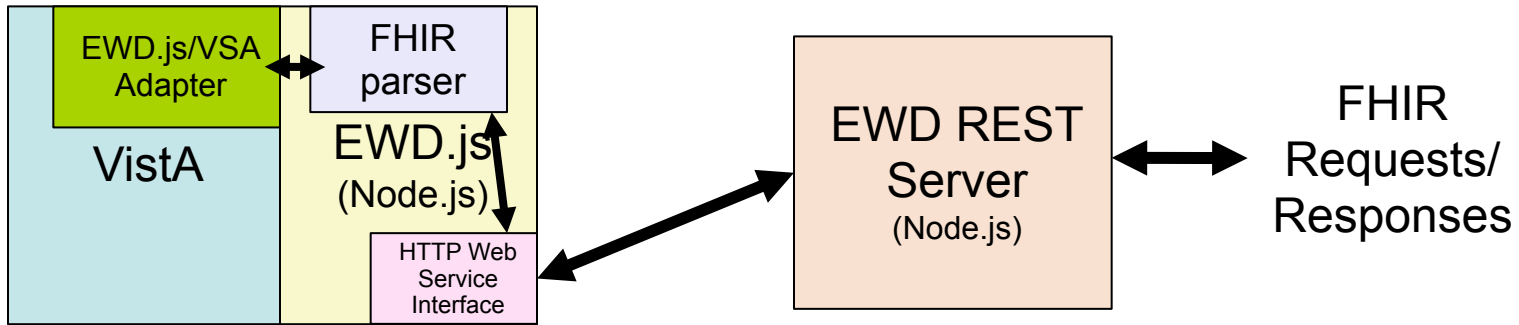
# EWD.js / VSA Integration – step 1



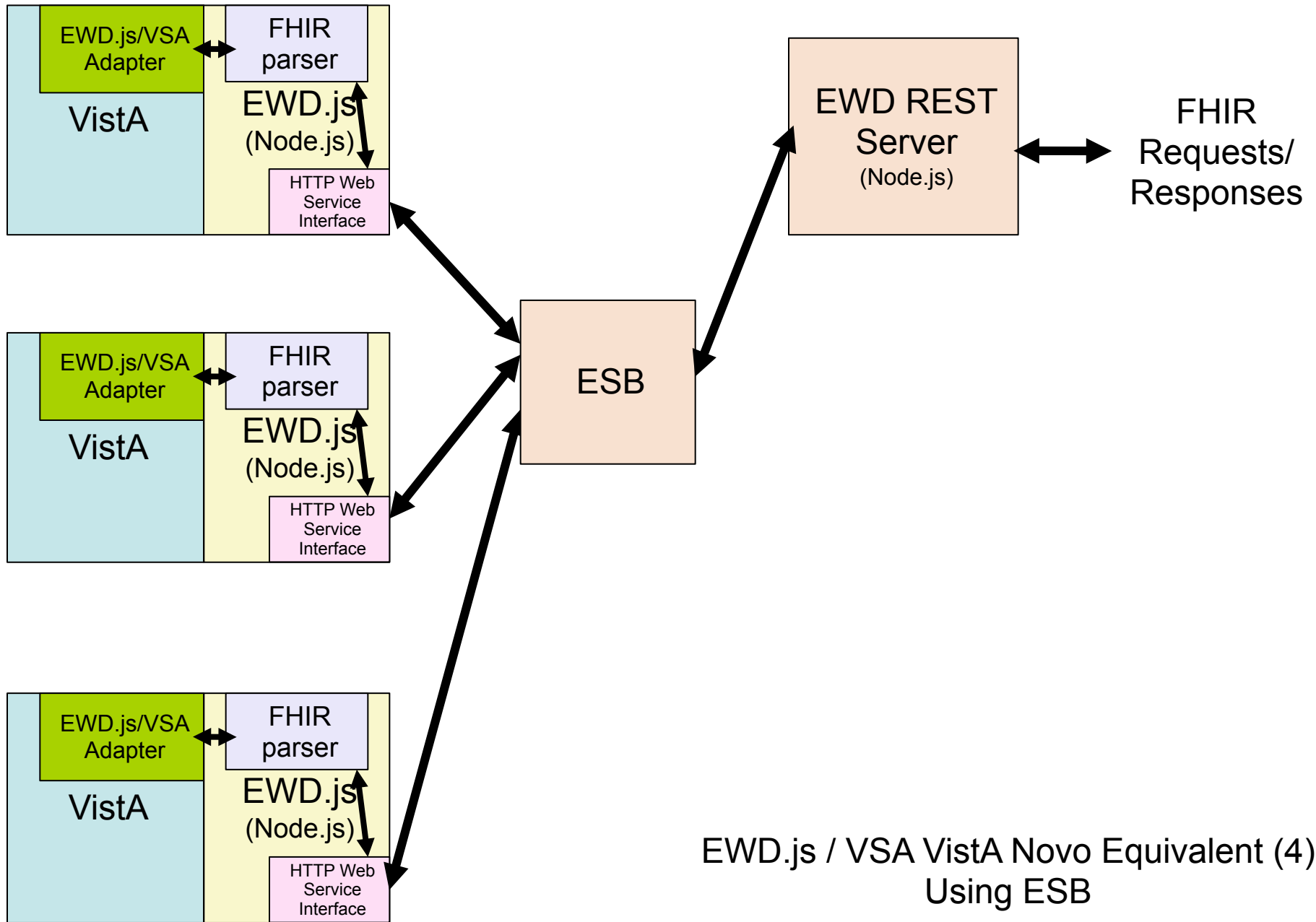




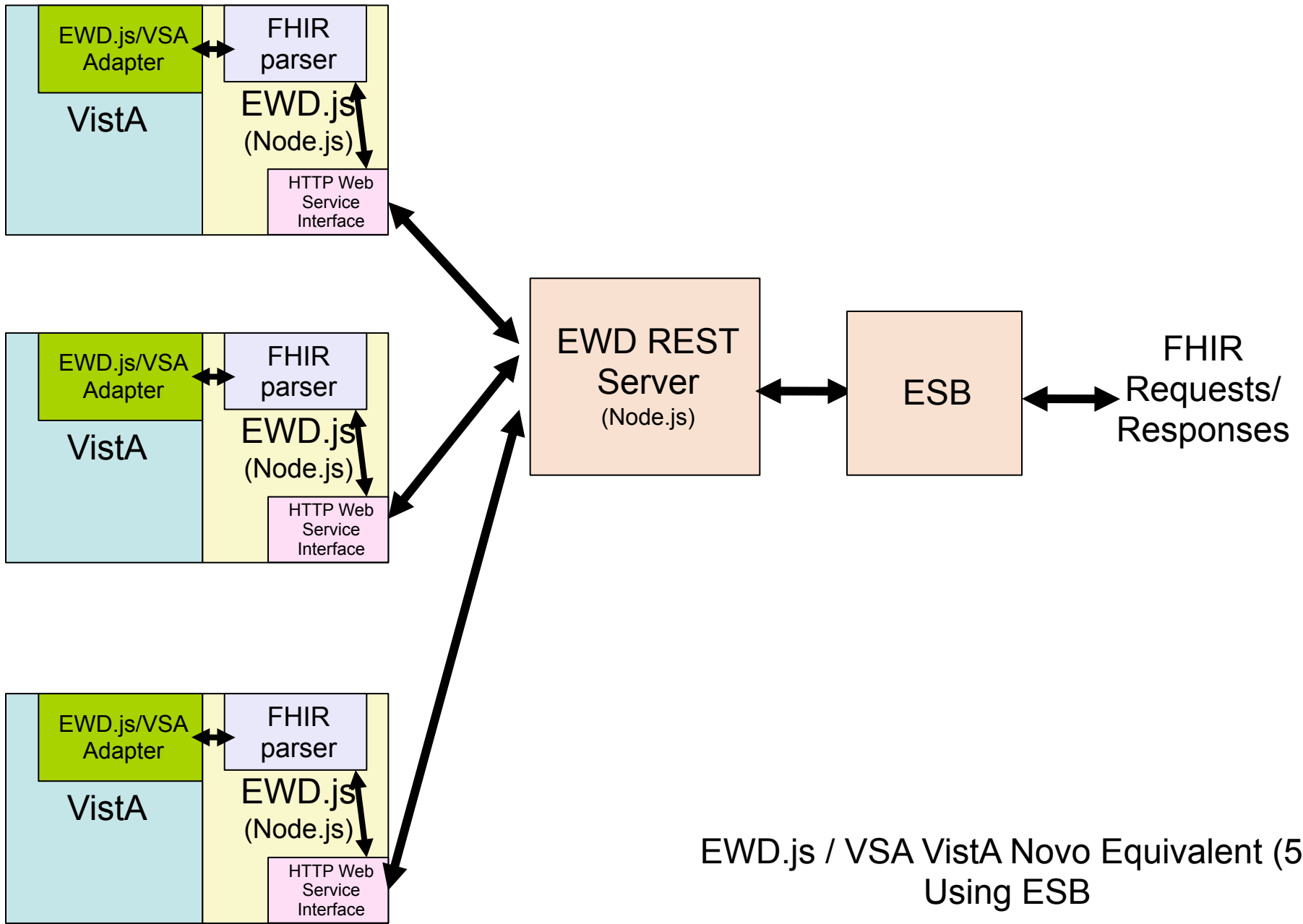
EWD.js / VSA VistA Novo Equivalent (1)







EWD.js / VSA VistA Novo Equivalent (4)  
Using ESB



EWD.js / VSA VistA Novo Equivalent (5)  
Using ESB

# Try it all out

- Christopher Edwards (OSEHRA)
  - VistA Installer
    - VistA up and running in 30 minutes
    - includes EWD.js and EWD REST Server
      - fully installed, configured and running
      - example applications
      - example REST services
  - Full details:
  - <http://www.osehra.org/blog/ewdjs-and-vista-ready-use-30-minutes>

# More security details

- Detailed summary of EWD.js security for:
  - interactive (WebSocket) applications;
  - HTTP-based Web Services
  - REST
- Read:
  - <http://robtweed.wordpress.com/2014/03/10/ewd-js-application-web-service-and-rest-security/>

# EWD.js on FHIR

## Copy of slides:

<http://gradvs1.mgateway.com/download/EWDjs-VistANovo.pdf>

Rob Tweed

M/Gateway Developments Ltd

Twitter: @rtweed

Email: [rtweed@mgateway.com](mailto:rtweed@mgateway.com)

