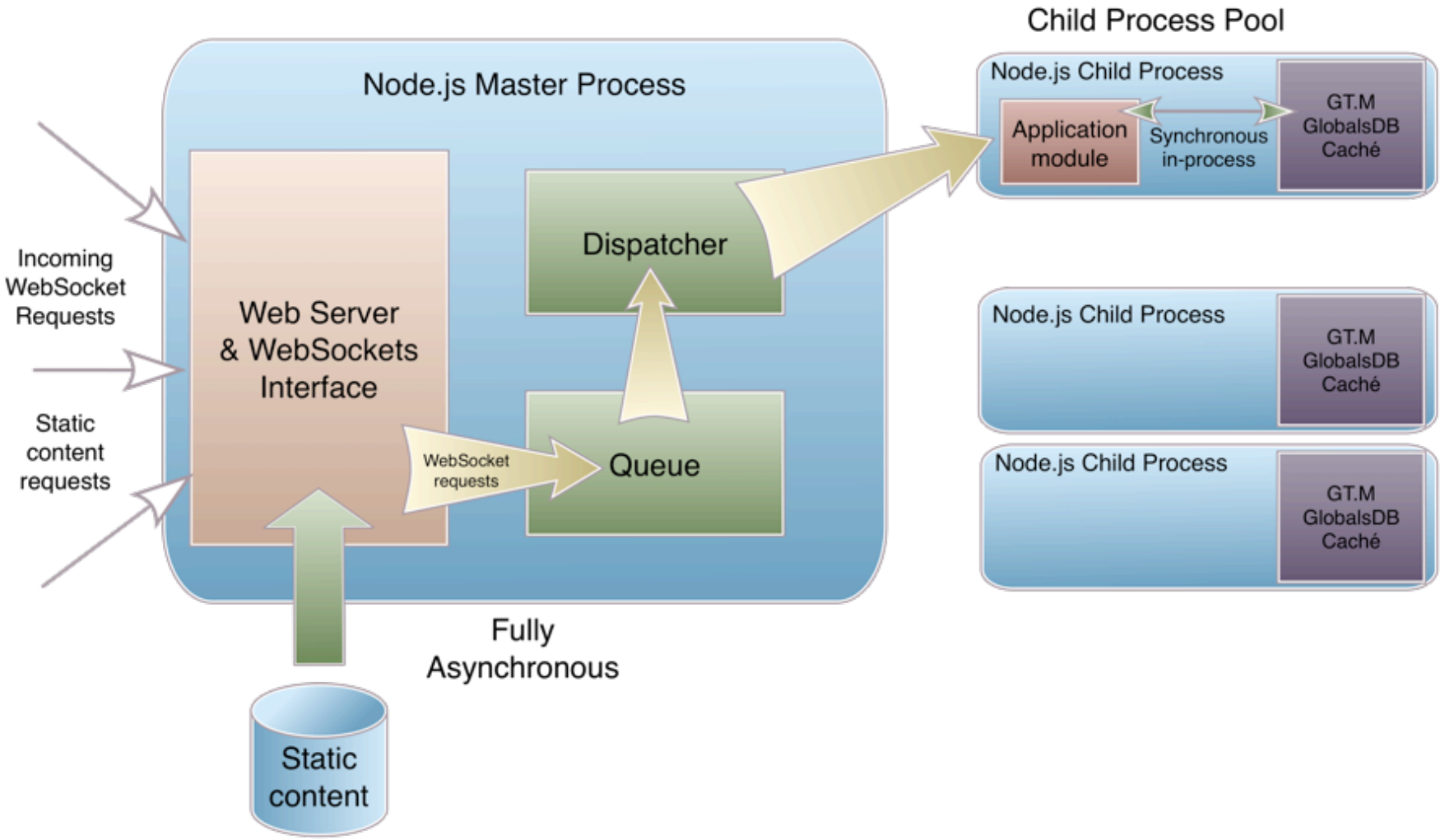
EWD.js Architecture

Rob Tweed
M/Gateway Developments Ltd

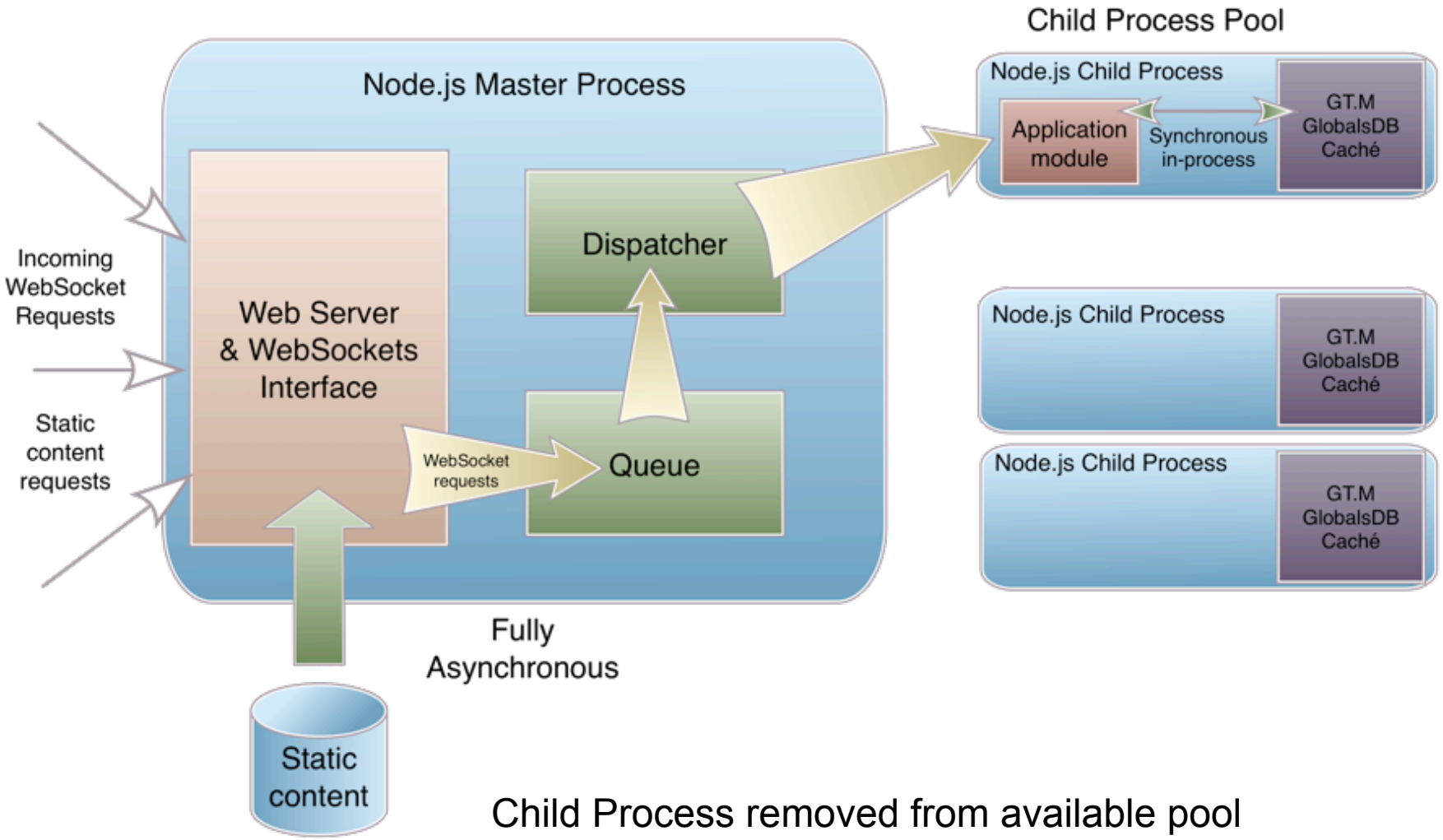
Twitter: @rtweed
Email: rtweed@mgateway.com

EWD.js Architecture



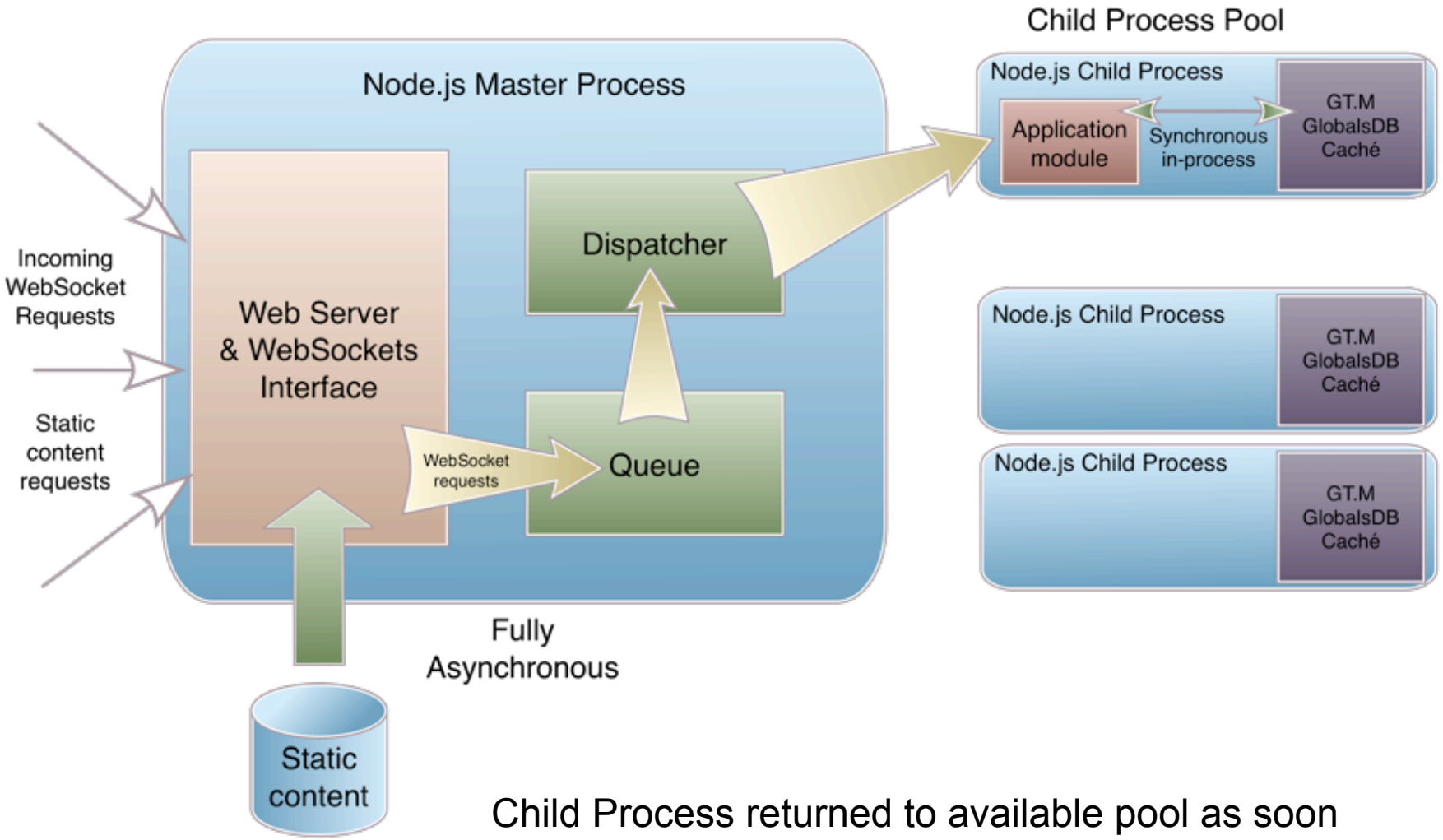
M/GATEWAY DEVELOPMENTS

EWD.js Architecture



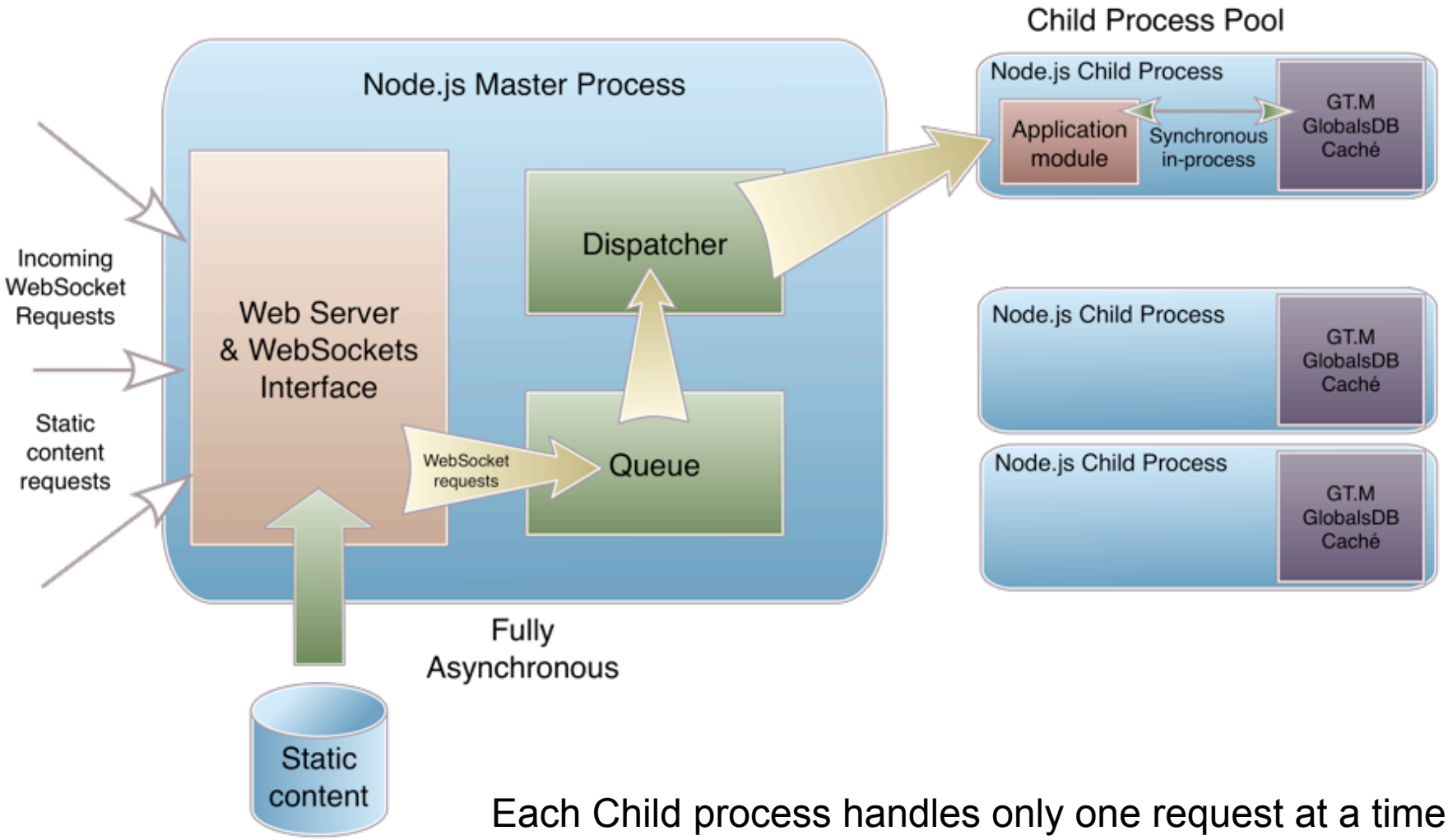
Child Process removed from available pool as soon as a request is sent to it

EWD.js Architecture



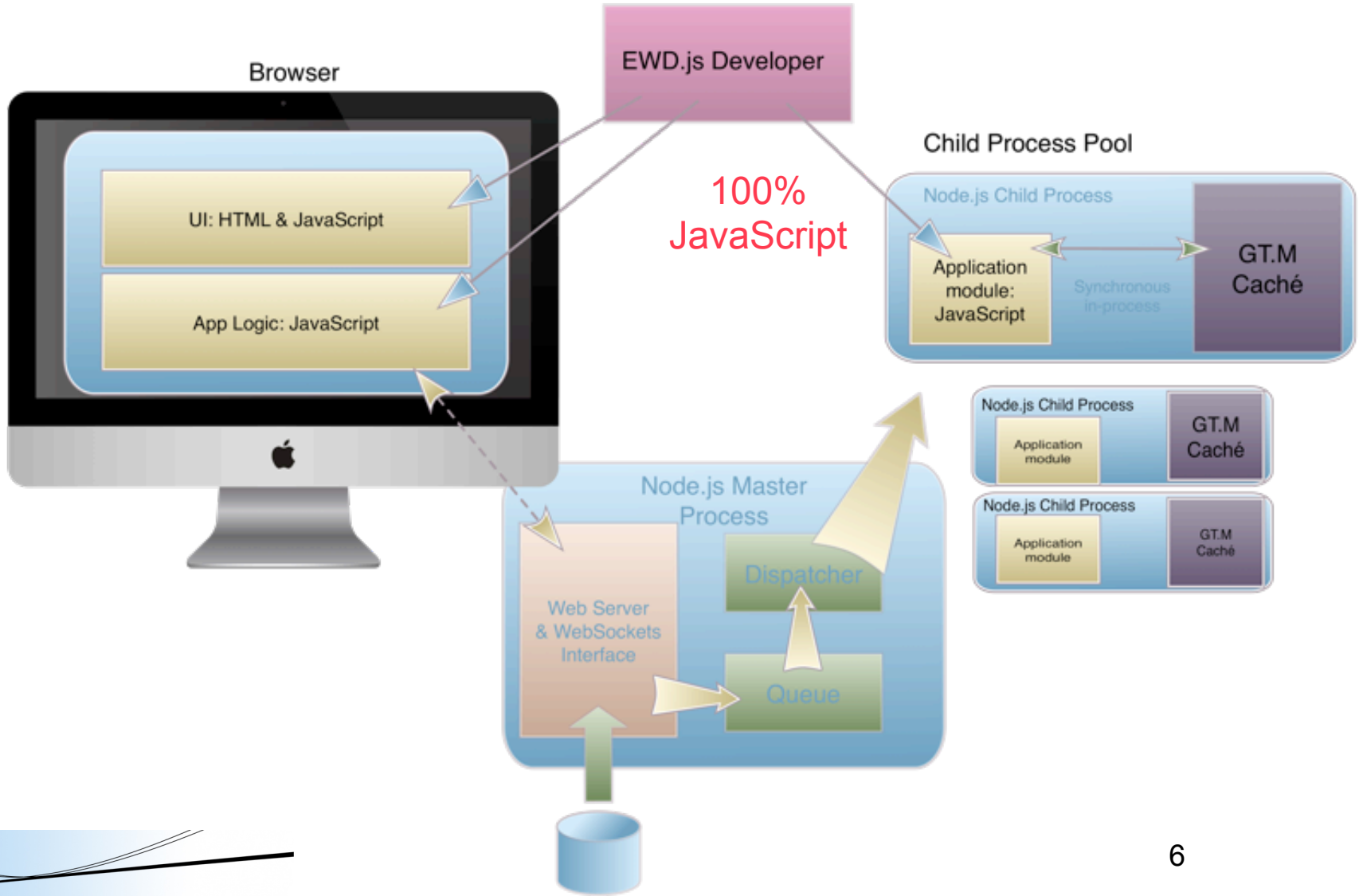
Child Process returned to available pool as soon as processing is completed

EWD.js Architecture



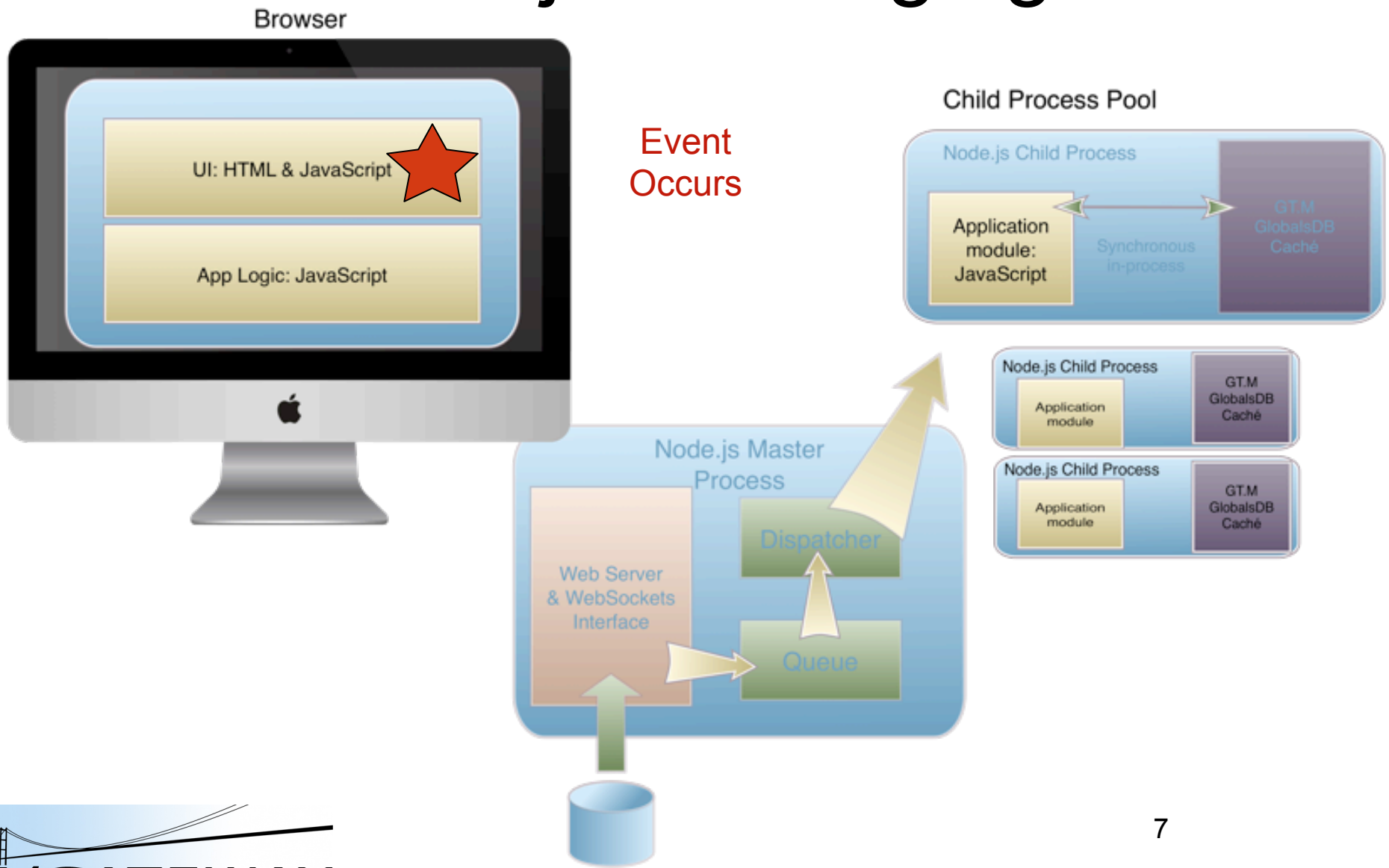
Each Child process handles only one request at a time

EWD.js Development



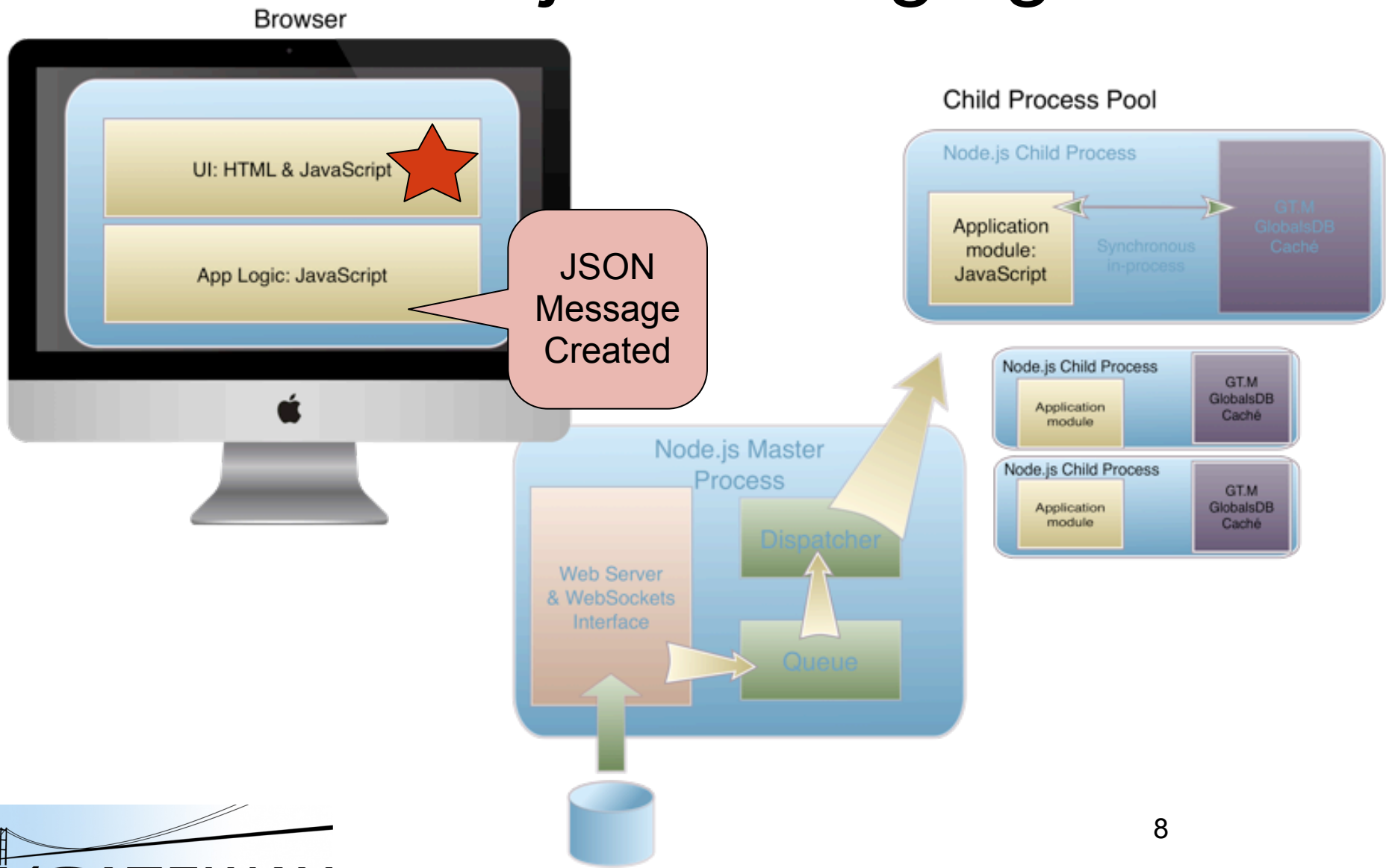
M/GATEWAY DEVELOPMENTS

EWD.js Messaging



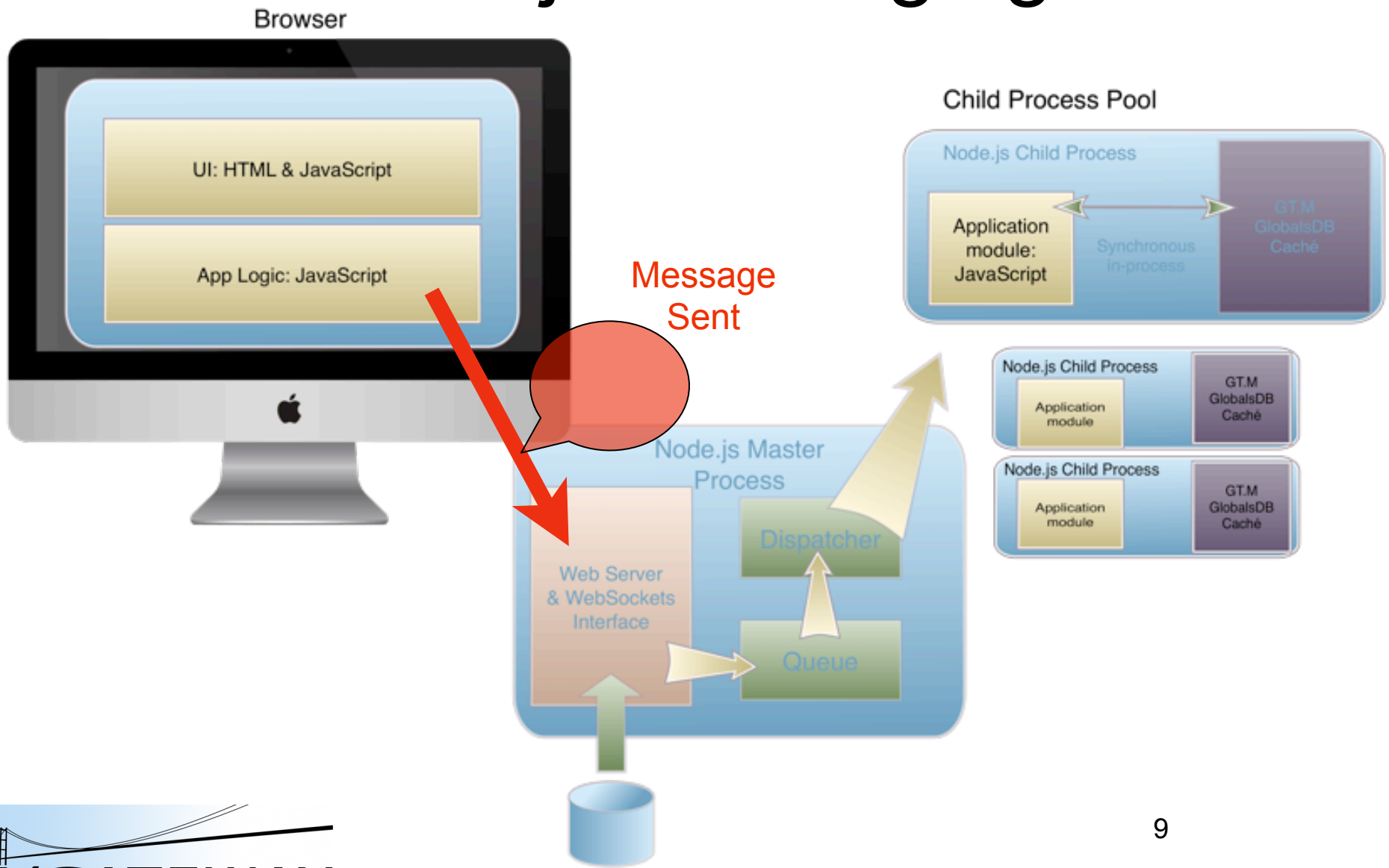
M/GATEWAY DEVELOPMENTS

EWD.js Messaging

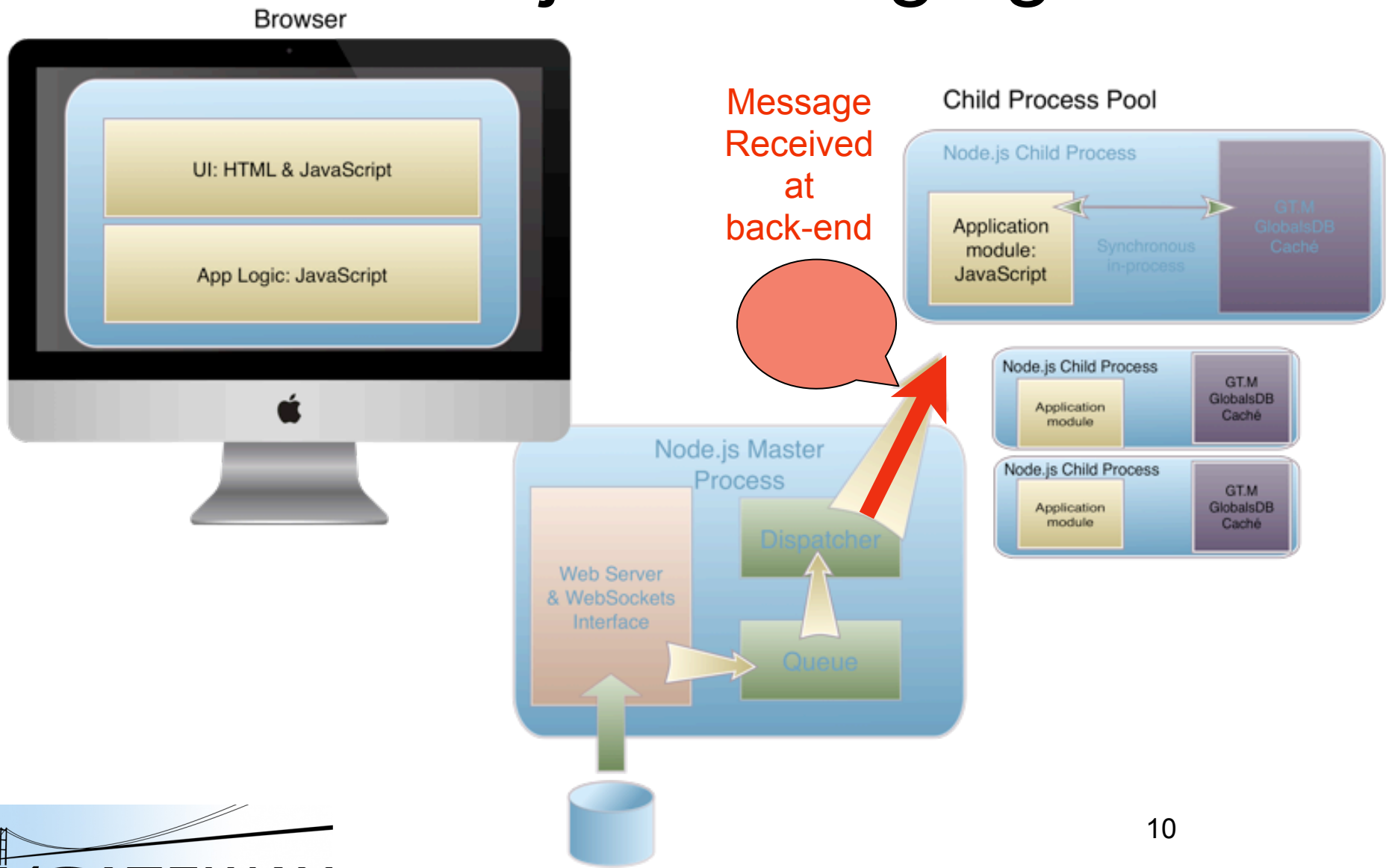


M/GATEWAY DEVELOPMENTS

EWD.js Messaging

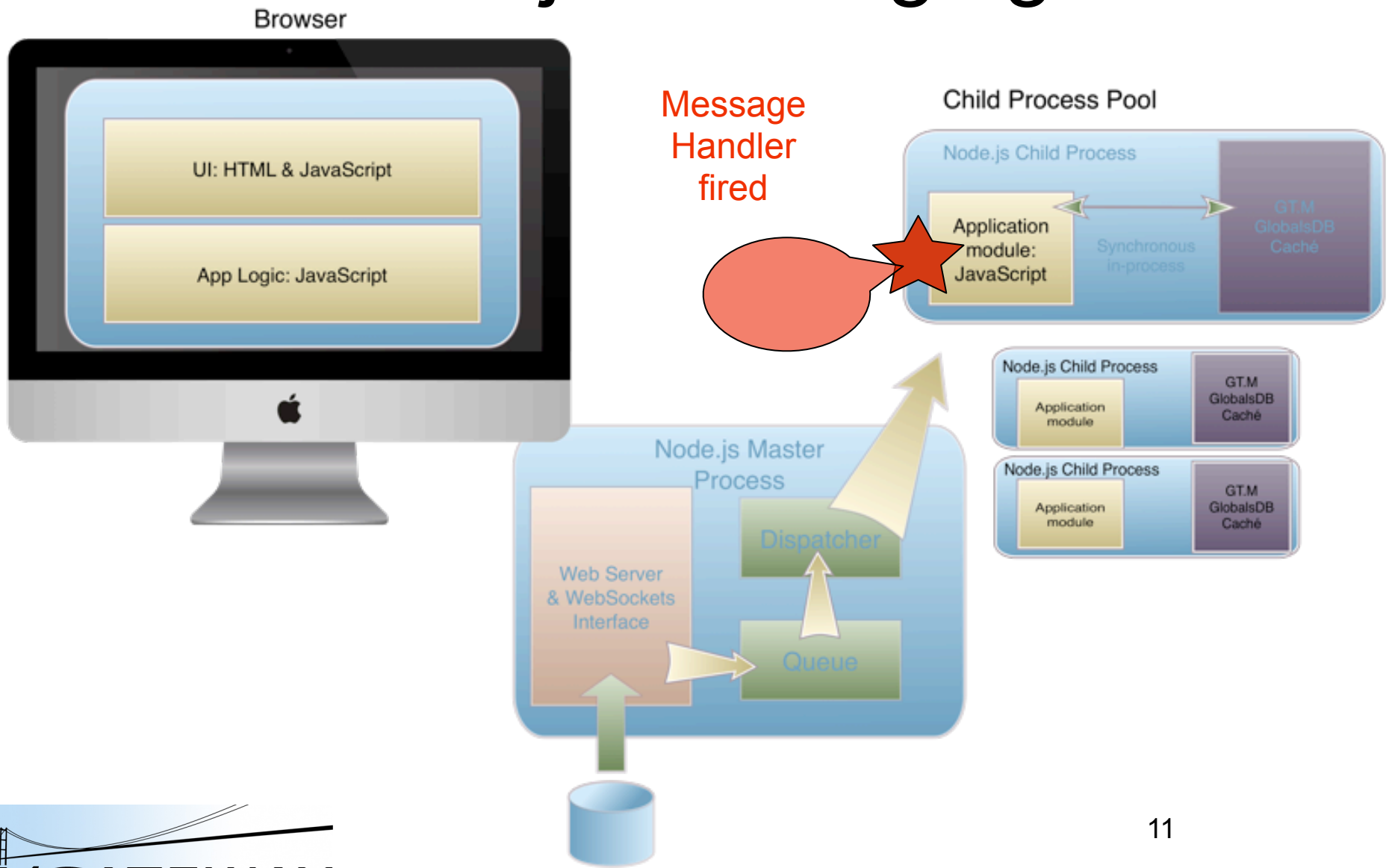


EWD.js Messaging



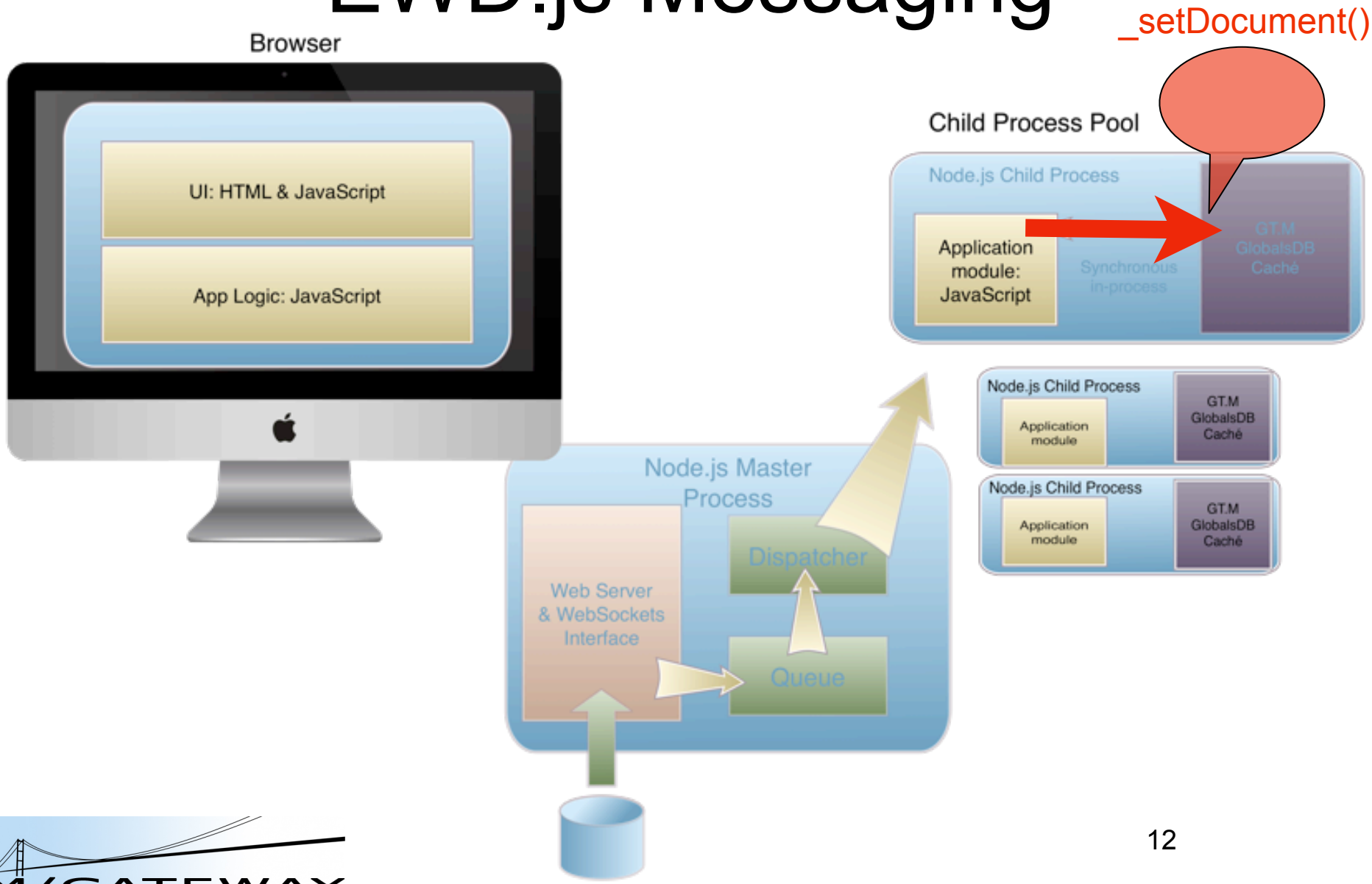
M/GATEWAY DEVELOPMENTS

EWD.js Messaging



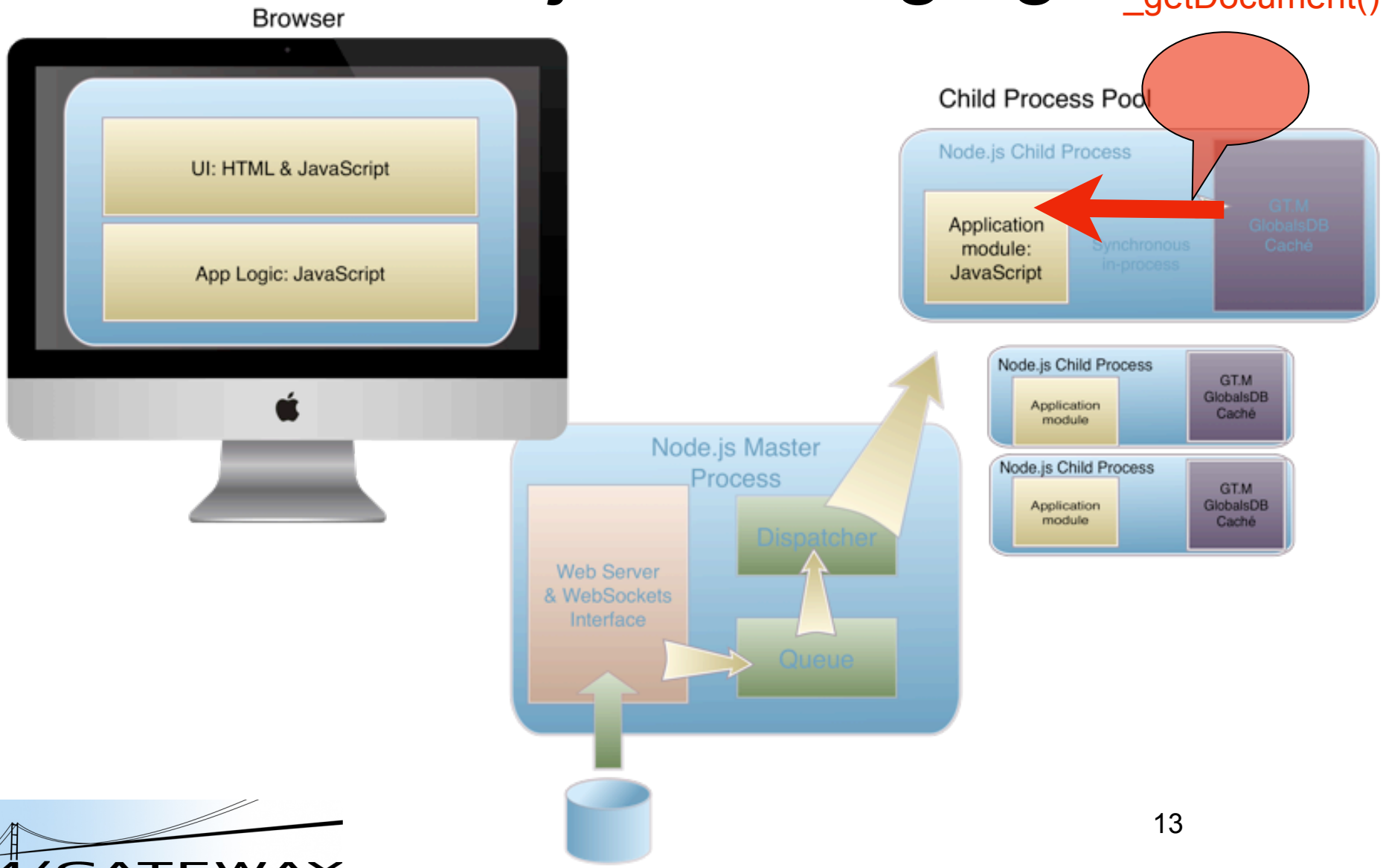
M/GATEWAY DEVELOPMENTS

EWD.js Messaging



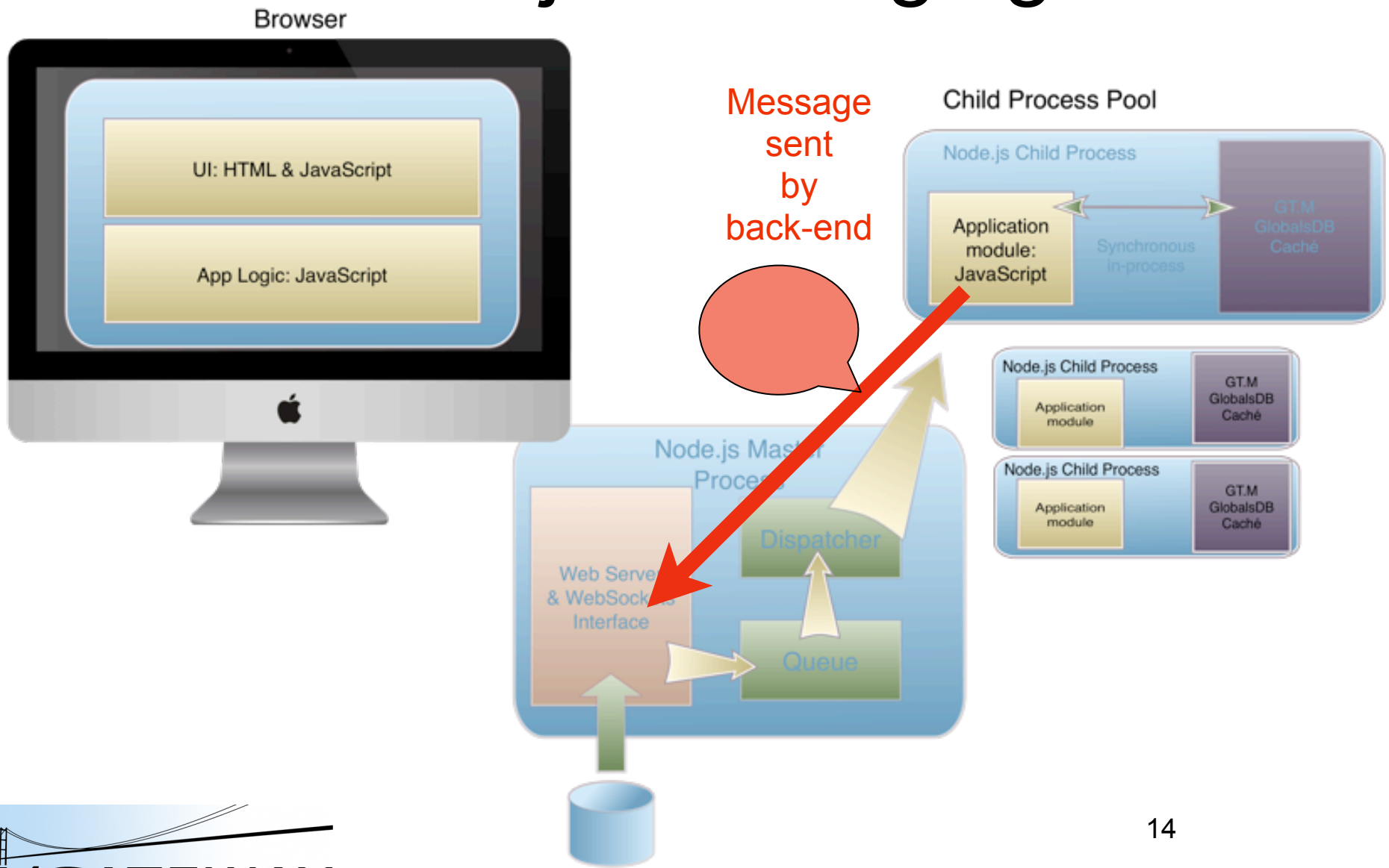
M/GATEWAY DEVELOPMENTS

EWD.js Messaging



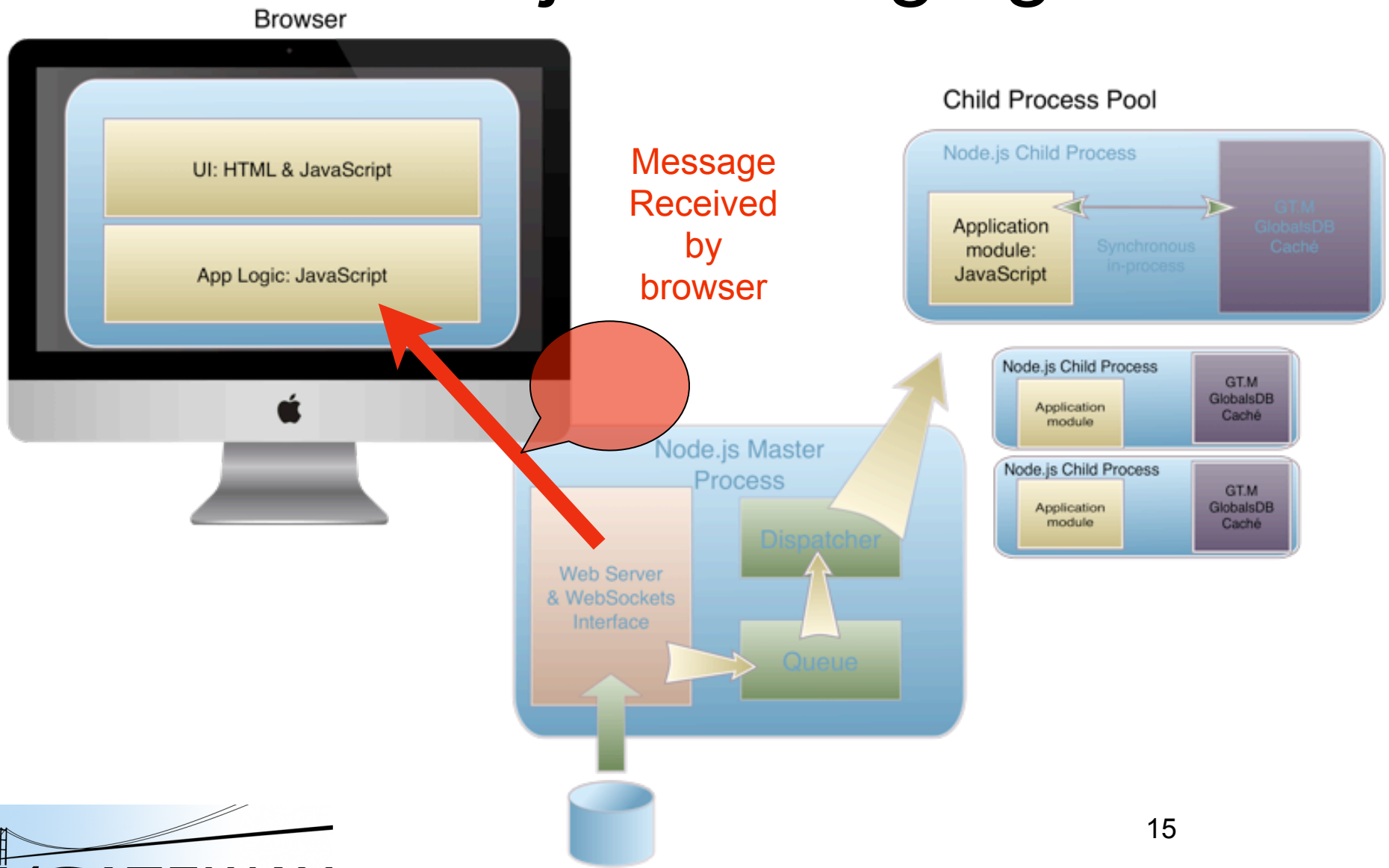
M/GATEWAY DEVELOPMENTS

EWD.js Messaging



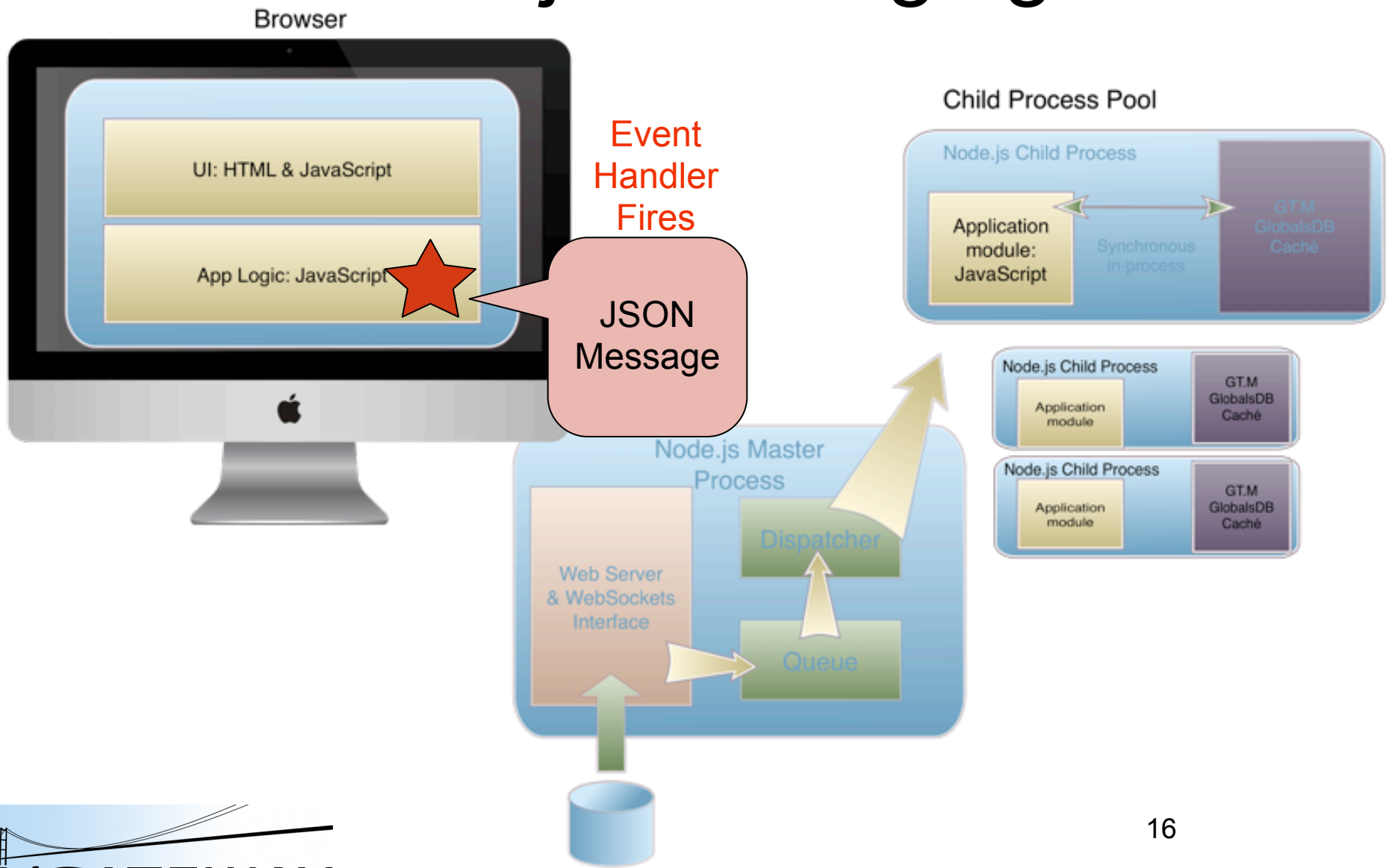
M/GATEWAY DEVELOPMENTS

EWD.js Messaging



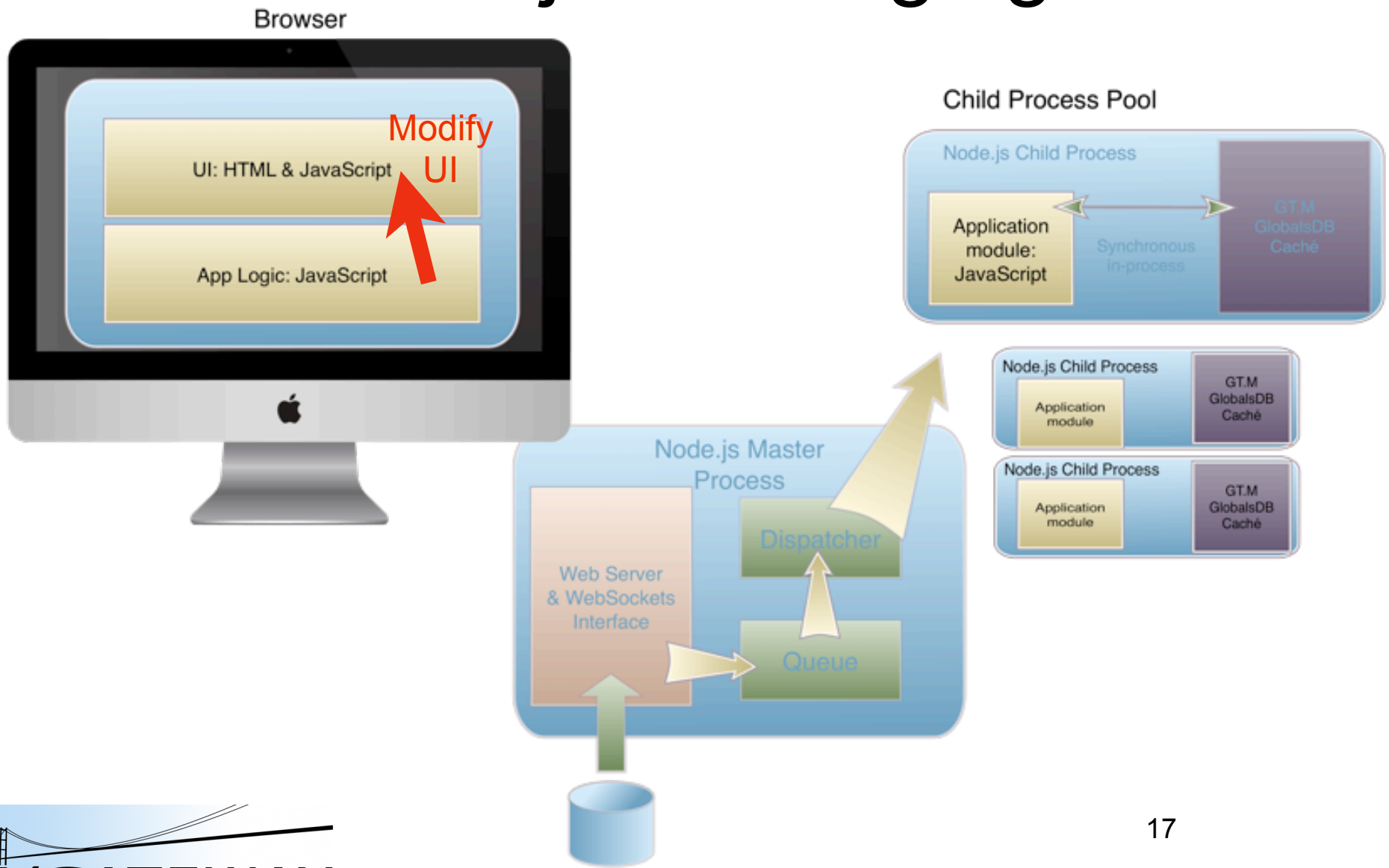
M/GATEWAY DEVELOPMENTS

EWD.js Messaging



M/GATEWAY DEVELOPMENTS

EWD.js Messaging



M/GATEWAY DEVELOPMENTS

Send message from browser

```
EWD.sockets.sendMessage({
  type: "sendHelloWorld",
  params: {
    text: 'Hello World!',
    sender: 'Rob',
    date: new Date().toUTCString()
  }
});
```

Send message from browser

```
EWD.sockets.sendMessage({
  type: "sendHelloWorld",
  params: {
    text: 'Hello World!',
    sender: 'Rob',
    date: new Date().toUTCString()
  }
});
```

User-defined type

Send message from browser

```
EWD.sockets.sendMessage({
  type: "sendHelloWorld",
  params: {
    text: 'Hello World!',
    sender: 'Rob',
    date: new Date().toUTCString()
  }
});
```

JSON payload

Back-end Module

```
module.exports = {
  onSocketMessage: function(ewd) {
    var wsMsg = ewd.webSocketMessage;
    var type = wsMsg.type;
    var params = wsMsg.params;
    var sessid = ewd.session.$('ewd_sessid')._value;
  }
};
```

Back-end Module

```
module.exports = {
  onSocketMessage: function(ewd) {
    var wsMsg = ewd.webSocketMessage;
    var type = wsMsg.type;
    var params = wsMsg.params;
    var sessid = ewd.session.$('ewd_sessid')._value;
  }
};
```

Message Handler: fires whenever a message is received

Back-end Module

```
module.exports = {
  onSocketMessage: function(ewd) {
    var wsMsg = ewd.webSocketMessage;
    var type = wsMsg.type;
    var payload = wsMsg.params;
    var sessid = ewd.session.$('ewd_sessid')._value;
    if (type === 'sendHelloWorld') {
      // do whatever is required with payload
      return {received: true};
    }
  }
};
```

Handle the message we sent from browser

Back-end Module

```
module.exports = {
  onSocketMessage: function(ewd) {
    var wsMsg = ewd.webSocketMessage;
    var type = wsMsg.type;
    if (type === 'sendHelloWorld') {
      var savedMsg = new ewd.mumps.GlobalNode('myMessage', []);
      savedMsg._setDocument(wsMsg);
      return {savedInto: '^myMessage'};
    }
  }
};
```

Saves the entire message into ^myMessage

Back-end Module

```
module.exports = {
  onSocketMessage: function(ewd) {
    var wsMsg = ewd.webSocketMessage;
    var type = wsMsg.type;
    if (type === 'sendHelloWorld') {
      var savedMsg = new ewd.mumps.GlobalNode('myMessage', []);
      savedMsg._setDocument(wsMsg);
      return {savedInto: '^myMessage'};
    }
  }
};
```

Returns a *sendHelloWorld* message back to browser

Browser-side Handler

```
EWD.onSocketMessage = function(messageObj) {
  if (messageObj.type === 'sendHelloWorld') {
    var text = 'Your message was successfully saved into ' +
       messageObj.message.savedInto;
    document.getElementById('response').innerHTML = text;
    setTimeout(function() {
      document.getElementById('response').innerHTML = '';
    },2000);
  }
};
```

Built-in Event Handler function

Browser-side Handler

```
EWD.onSocketMessage = function(messageObj) {
  if (messageObj.type === 'sendHelloWorld') {
    var text = 'Your message was successfully saved into ' +
       messageObj.message.savedInto;
    document.getElementById('response').innerHTML = text;
    setTimeout(function() {
      document.getElementById('response').innerHTML = '';
    },2000);
  }
};
```

Just like at the back-end!

Browser-side Handler

```
EWD.onSocketMessage = function(messageObj) {
  if (messageObj.type === 'sendHelloWorld') {
    var text = 'Your message was successfully saved into ' +
       messageObj.message.savedInto;
    document.getElementById('response').innerHTML = text;
    setTimeout(function() {
      document.getElementById('response').innerHTML = '';
    },2000);
  }
};
```

Modify the UI

No Polling!

- With WebSockets, the back-end can send a message *at any time* to:
  - a specific browser
  - all browsers running a specific EWD.js application
  - all currently-connected browsers

Back-end sending a message

```
var savedMsg = new ewd.mumps.GlobalNode('myMessage', []);

ewd.sendWebSocketMsg({
  type: 'savedMessage',
  message: savedMsg._getDocument()
});
```

Built-in secured Web Services

- Any back-end JavaScript method can be exposed as a JSON Web Service
- Access is automatically secured
  - HMAC-SHA256 digital signatures required for every HTTP request
  - The same security used by Amazon Web Services
- Lightweight peer-to-peer access between EWD.js systems

Example Web Service

```
webServiceExample: function(ewd) {
  var patient = new ewd.mumps.GlobalNode('CLPPats', [ewd.query.id]);
  if (!patient._exists) return {error: 'Patient ' + ewd.query.id + ' does not exist'};
  return patient._getDocument();
}
```

```
https://192.168.1.89:8080/json/demo/webServiceExample?
  id=1233&
  accessId=rob12kjh1i23&
  timestamp=Wed, 19 Jun 2013 14:14:35 GMT&
  signature=P0bIakNehj2TkuadxbKRsIgJCGIhY1EvntJdSce5XvQ=
```

Node.js EWD.js Web Service client:

npm install ewdliteclient

Example Web Service

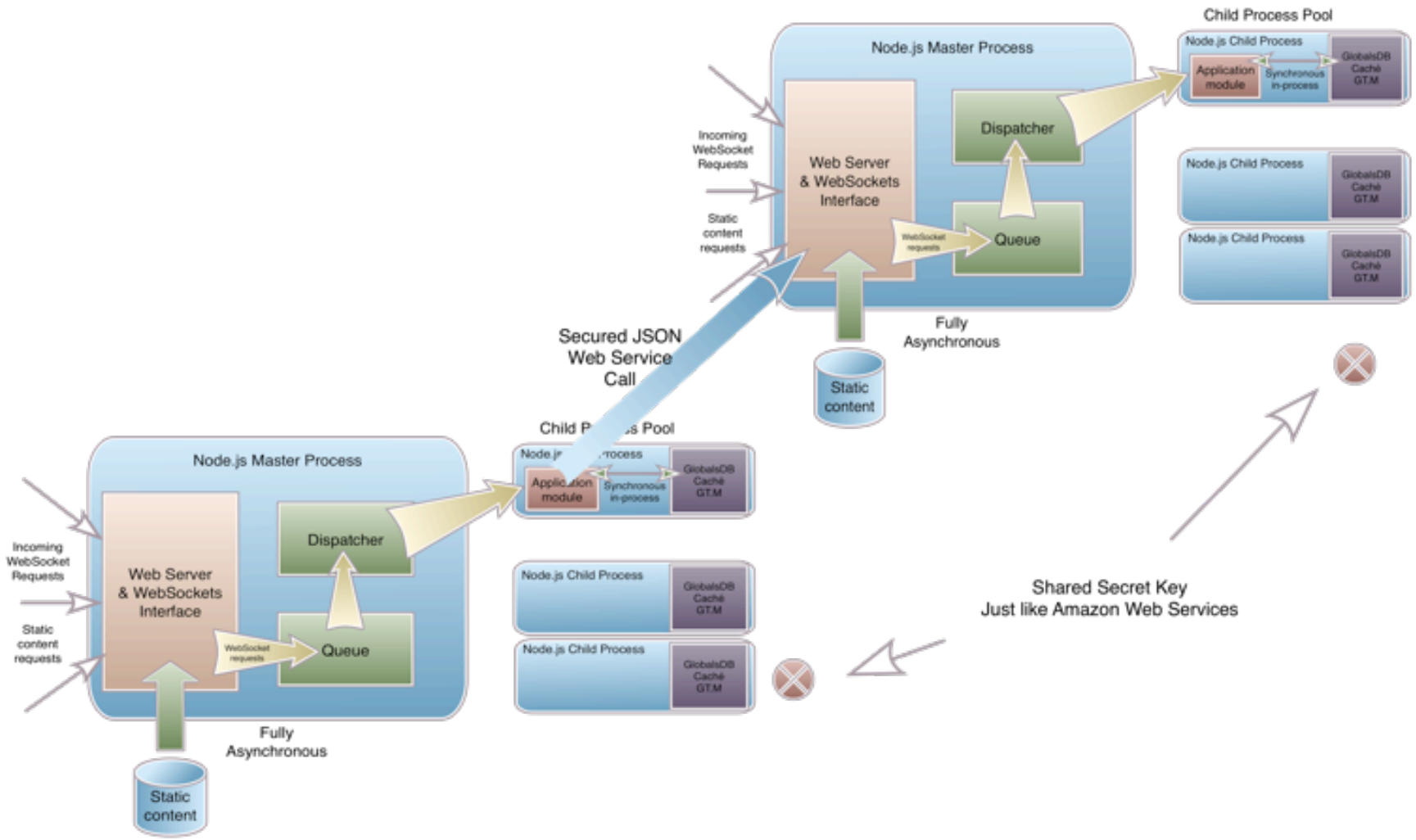
```
webServiceExample: function(ewd) {
  var patient = new ewd.mumps.GlobalNode('CLPPats', [ewd.query.id]);
  if (!patient._exists) return {error: 'Patient ' + ewd.query.id + ' does not exist'};
  return patient._getDocument();
}
```

```
https://192.168.1.89:8080/json/demo/webServiceExample?
  id=1233&
  accessId=rob12kjh1i23&
  timestamp=Wed, 19 Jun 2013 14:14:35 GMT&
  signature=P0blakNehj2TkuadxbKRsIgJCGIhY1EvntJdSce5XvQ=
```

Node.js EWD.js Web Service client:
npm install ewdliteclient

The perfect architecture to support VSA

Secured Linked Systems



M/GATEWAY DEVELOPMENTS

Node.js Custom Events

addMedication({params})

Node.js Custom Events

addMedication({params})

```
addMedication = function(params) {
... code for adding medication

ewd.emit('audit', {auditParams};
ewd.emit('stockControl', {stockParams});
};
```

Node.js Custom Events

```
addMedication({params})
```

```
addMedication = function(params) {
... code for adding medication

ewd.emit('audit', {auditParams};
ewd.emit('stockControl', {stockParams});
};
```

```
ewd.on('audit', function(params) {
  .... code for adding audit record

  possibly via WebService to remote
  system
};
```

Node.js Custom Events

```
addMedication({params})
```

```
addMedication = function(params) {
... code for adding medication

ewd.emit('audit', {auditParams};
ewd.emit('stockControl', {stockParams});
};
```

```
ewd.on('audit', function(params) {
  .... code for adding audit record
};
```

```
ewd.on('stockControl', function(params {
  .... code for changing stock record
};
```

EWD.js requirements

- Ideally browsers that support HTML5 WebSockets
  - however, EWD.js uses Node.js socket.io library
    - emulates websockets using other techniques if not available
    - even works with old versions of Internet Explorer!

Licensing & Availability

- Apache 2
- https://github.com/robtweed/ewdGateway2
- Installing on Node.js:
  - *npm install ewdgateway2*

Getting Started

- http://gradvs1.mgateway.com/download/EWDjs.pdf
- dEWDrop VM (http://www.fourthwatchsoftware.com)
- Mike Clayton's Ubuntu Installer
- Raspberry Pi!
- Training Courses:
  - Watch for announcments at:
    - http://robtweed.wordpress.com

EWD.js

Rob Tweed
M/Gateway Developments Ltd

Twitter: @rtweed
Email: rtweed@mgateway.com