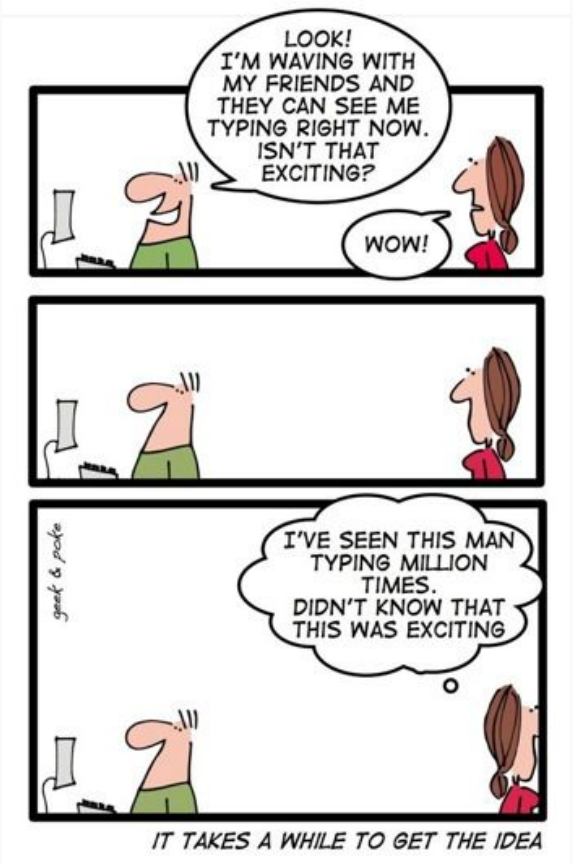
WebSockets

Parce que le temps réel ça poutre quand même...



Ok… Peut-être pas pour tous mais quand même !

Définition

WebSocket est un standard du Web désignant un protocole réseau de la couche application et une interface de programmation du World Wide Web visant à créer des canaux de communication full-duplex par dessus une connexion TCP. Le protocole a été normalisé par l'IETF dans la RFC 6455 en 2011 et l'interface de programmation est en cours de standardisation par le W3C.

Communication WEB depuis les dinosaures.

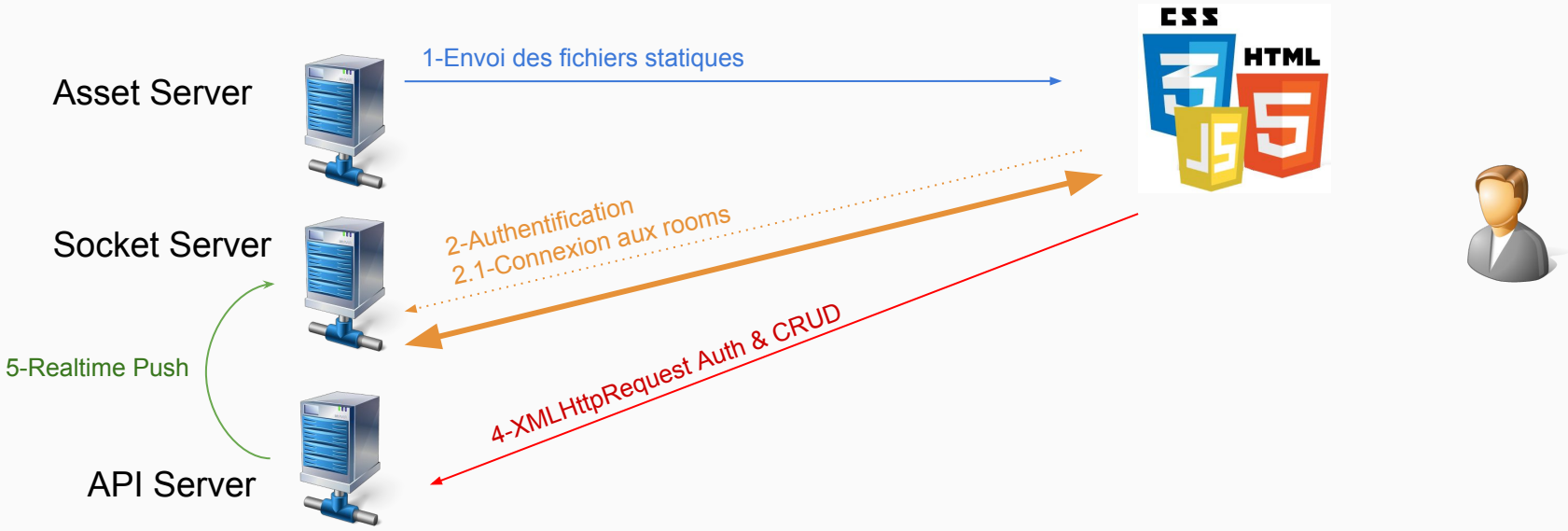
1. Envoi de données / Téléchargement de fichiers par refresh complets
   => FORM POST/GET, HREF (requête client)

2. Envoi de données / Téléchargement de fichiers en arrière plan
   => XMLHttpRequest (requête client)

3. Echange de données full duplex
   => WebSocket (requête client, puis push serveur possible !)


snom

Use Case principal



Implémentations - protocole ws:// & wss://

- PyWebSocket
- jWebSocket
- phpWebSocket
- Socket.io

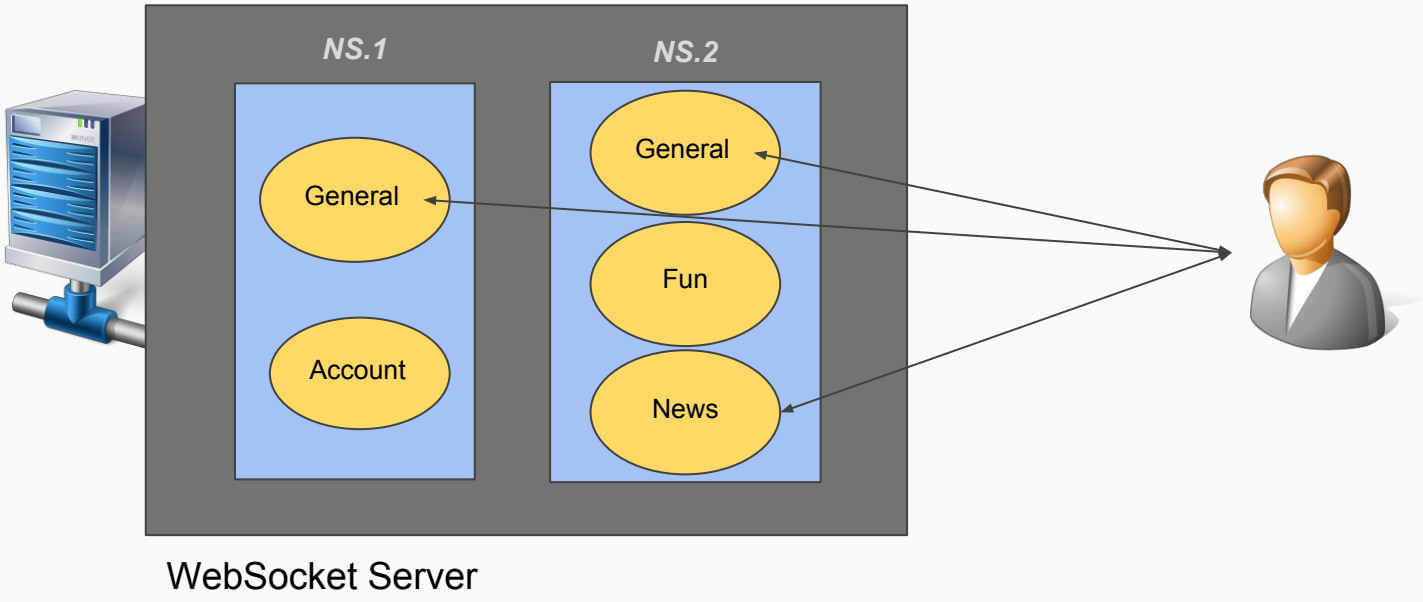
- Dernière version de RubyOnRails
- Même C# a un truc pour ça !

Socket.IO

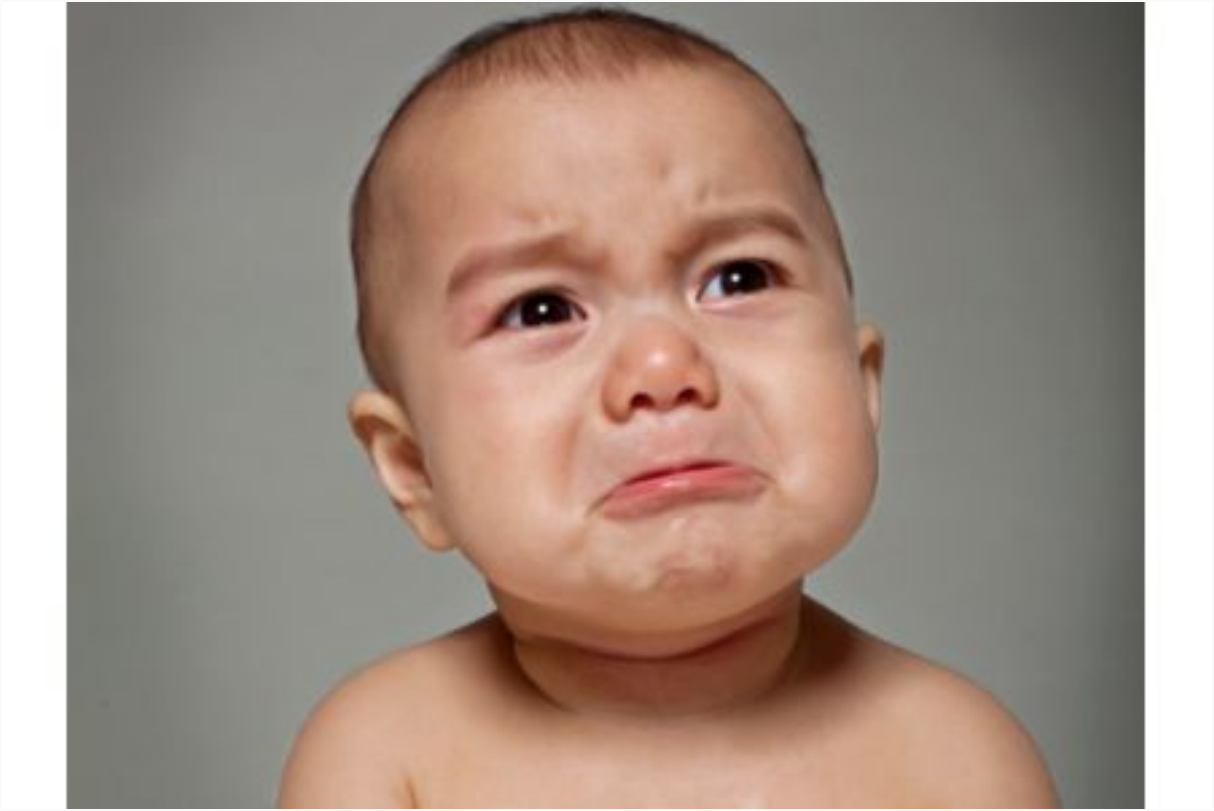
- Très grosse communauté
- Gestion des namespaces
- Gestion des rooms
- Support du streaming binaire depuis la version 1.X
- Serveur excessivement simple à écrire. (merci au Javascript)
- Excellente stabilité (merci à PM2)
- Excellente tenue en mémoire (merci à NodeJS)

- Parfois un peu lourd à interfacer si la librairie n’existe pas.

Namespaces / Rooms - WTF ?



Avant NodeJS...



Quand je devais gérer des sockets avant...

Socket.IO

Using with Node http server

Server (app.js)

```
var app = require('http').createServer(handl
er)
var io = require('socket.io')(app);
var fs = require('fs');

app.listen(80);

function handler (req, res) {
  fs.readFile(__dirname + '/index.html',
  function (err, data) {
    if (err) {
      res.writeHead(500);
      return res.end('Error loading index.ht
ml');
    }

    res.writeHead(200);
    res.end(data);
  });
}

io.on('connection', function (socket) {
  socket.emit('news', { hello: 'world' });
  socket.on('my other event', function (data
) {
    console.log(data);
  });
});
```

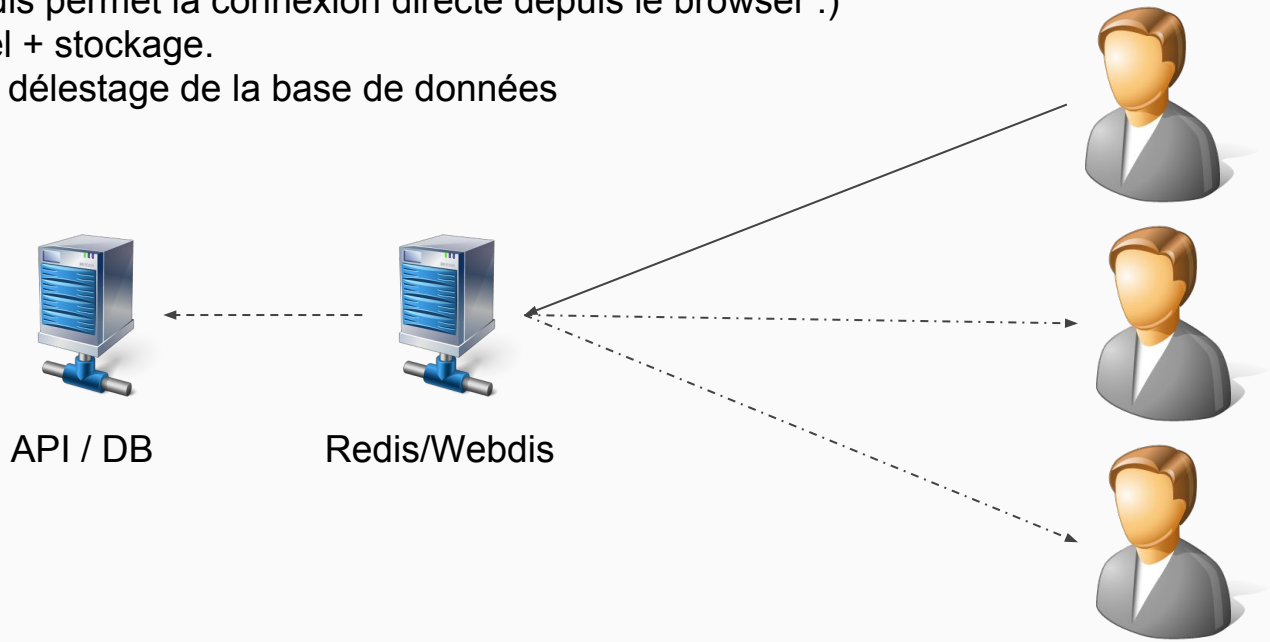
Client (index.html)

```
<script src="/socket.io/socket.io.js"></scri
pt>
<script>
  var socket = io('http://localhost');
  socket.on('news', function (data) {
    console.log(data);
    socket.emit('my other event', { my: 'dat
a' });
  });
</script>
```

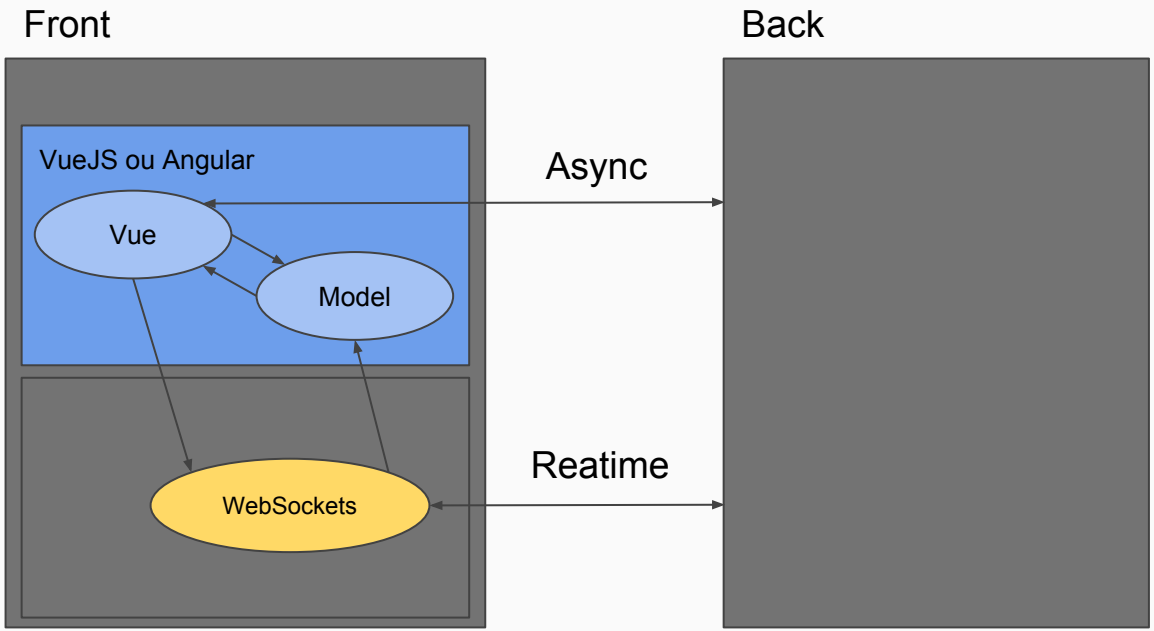
Whooooohooooo : J’ai écris un serveur de socket !!!

Et sinon ? Redis ! (enfin Webdis)

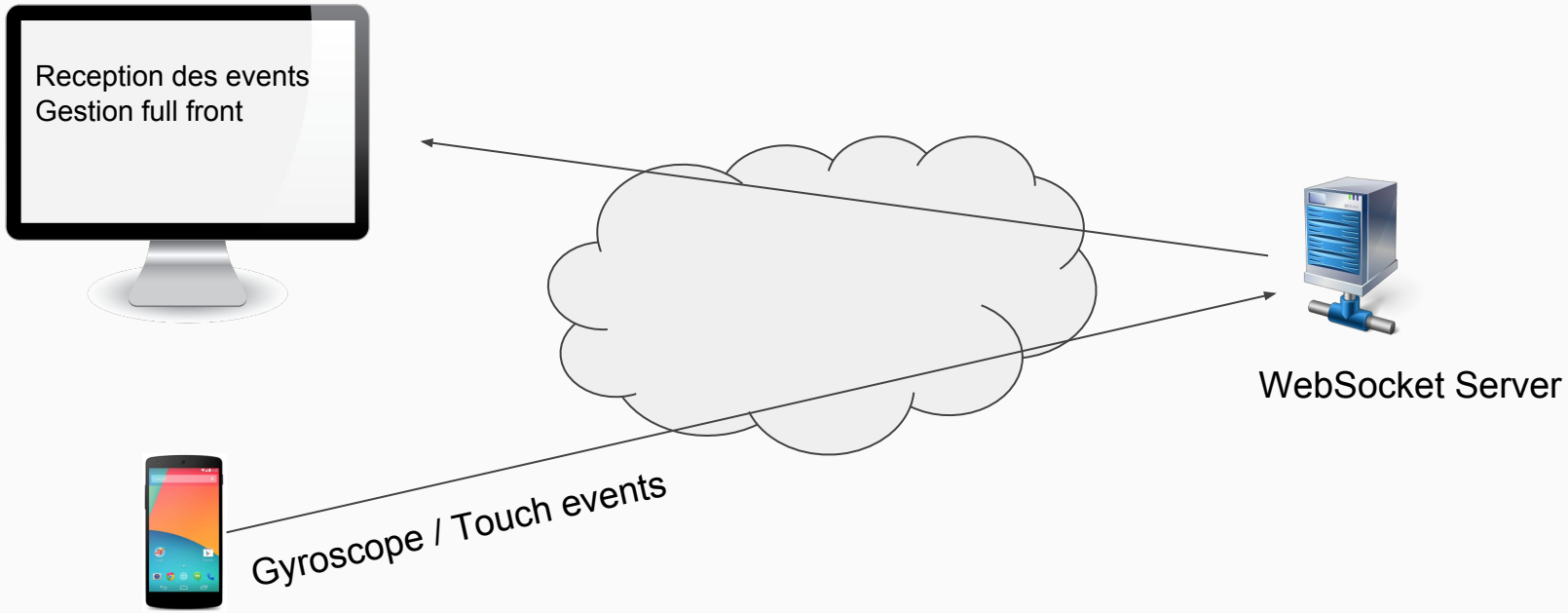
- Serveur de stockage de clés !
- Extrèeeeeeeeemement rapide !
- Fork Webdis permet la connexion directe depuis le browser :)
- Temps réel + stockage.
- Permet un délestage de la base de données



Ma grand mère aussi elle aime le tricot.



Et bim !


Reception des events
Gestion full front
WebSocket Server
Gyroscope / Touch events