

# OAE : développement d'un widget en NodeJS

## Retour d'expérience

Jimmy Kasprzak -- Université de La Rochelle



# Plan de présentation

## 1. Présentation du projet

1. Résumé
2. Démonstration

## 2. Retour d'expérience sur NodeJS

1. Quelques généralités
2. Mes bonnes raisons d'utiliser NodeJS
3. Erreurs et difficultés les plus communes
4. Les incontournables à connaître

## 3. Quelques mots sur Cassandra

## 4. Conclusion : synthèse de la présentation

# OAE : développement d'un widget en NodeJS



## Présentation du projet



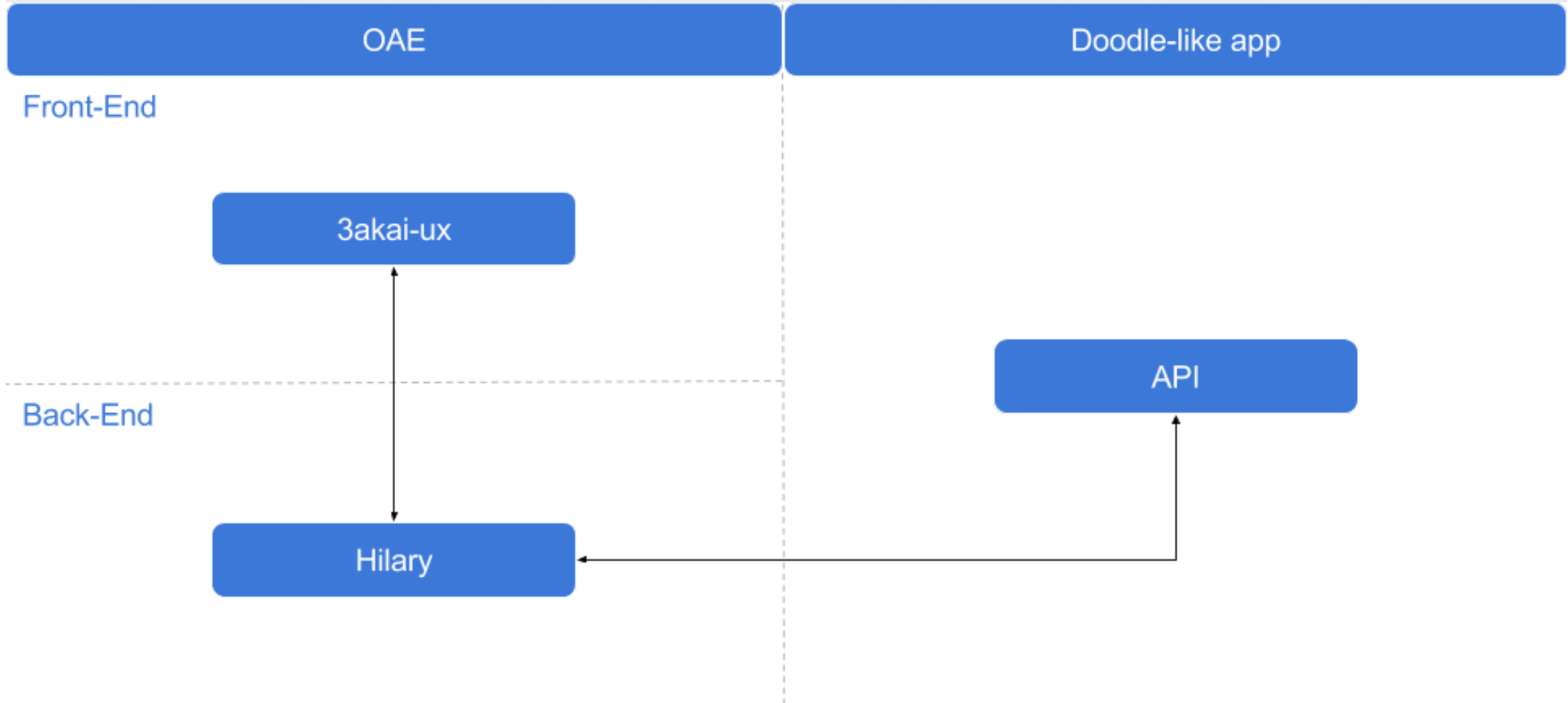
# Présentation du projet

## ➤ Objectifs

Permettre la planification de réunions directement dans OAE.  
Ajout d'une nouvelle fonctionnalité Doodle-like

# Présentation du projet

## ➤ Architecture



# Présentation du projet

## ➤ Caractéristiques fonctionnelles

- \* Inscription, connexion d'utilisateurs
- \* Création, suppression et mise à jour de sondage
- \* Ajout et suppression de créneaux
- \* Mise à jour de ses disponibilités
- \* Ajout et suppression de créneaux
- \* Système de notifications
- \* API Rest





## Démonstration

# OAE : développement d'un widget en NodeJS



## Retour d'expérience



# Retour d'expérience

## ➤ Quelques généralités sur NodeJS

- \* JavaScript
- \* Moteur V8
- \* Licence MIT (2009)
- \* Communauté très active
- \* Gestion des paquets : NPM



# Retour d'expérience

## ➤ Mes bonnes raisons d'utiliser NodeJS

- \* Pas de nouveau langage à apprendre
- \* Comportement asynchrone
- \* Profusion de librairies disponibles aussi bien pour le front que pour le back
- \* Liberté d'écriture du JavaScript
- \* Gestion native du JSON



# Retour d'expérience

## ➤ Erreurs et difficultés les plus communes

### ❖ Difficultés

- \* Liberté du JavaScript
- \* Maîtrise du côté asynchrone
- \* Pas encore de frameworks tels qu'on peut en trouver dans d'autres langages

### ❖ Erreurs les plus communes

- \* Ne pas utiliser d'outils de développement
- \* Exécuter plusieurs fois le même callback
- \* Le « Callback hell »
- \* Ne pas utiliser d'outils de logs
- \* Pas de test



# Retour d'expérience

## ➤ Mes incontournables NodeJS

### ❖ Frameworks

- \* Express (4 404 483 de téléchargements en Janvier 2016)
- \* Hapi (168 473 de téléchargements en Janvier 2016)
- \* Sails (48 720 de téléchargements en Janvier 2016)
- \* Socket.io (2 258 640 de téléchargements en Janvier 2016)

### ❖ Gestion de l'asynchrone

- \* Async (22 904 823 de téléchargements en Janvier 2016)
- \* Vasync (99 861 de téléchargements en Janvier 2016)

# Retour d'expérience

## ➤ Mes incontournables NodeJS

### ❖ Logs

- \* Bunyan (606 451 de téléchargements en Janvier 2016)
- \* Winston (2 619 595 de téléchargements en Janvier 2016)

### ❖ Outils d'automatisation de développement

- \* Nodemon (407 162 de téléchargements en Janvier 2016)

### ❖ Tests

- \* Mocha (2 240 622 de téléchargements en Janvier 2016)
- \* Karma (2 240 622 de téléchargements en Janvier 2016)



# OAE : développement d'un widget en NodeJS



## Quelques mots sur Cassandra



# Quelques mots sur Cassandra

➤ Qu'est ce que c'est ?

- \* Base de données NoSQL
- \* Java
- \* Structure clé-valeur
- \* Open Source depuis 2008
- \* Originellement créé par Facebook pour son système de messagerie
- \* Géré à l'heure actuelle par la fondation Apache



# Quelques mots sur Cassandra

## ➤ Quelles sont ses principales caractéristiques ?

- \* Capacité à gérer un très grand nombre de données
- \* Capacité à gérer tout type de données
- \* Système de noeuds de réplication
- \* A été conçu pour fonctionner en mode cluster
- \* Accès en lecture et écriture très rapide
- \* N'autorise que des requêtes simples
- \* Cqlsh : un sql-like langage

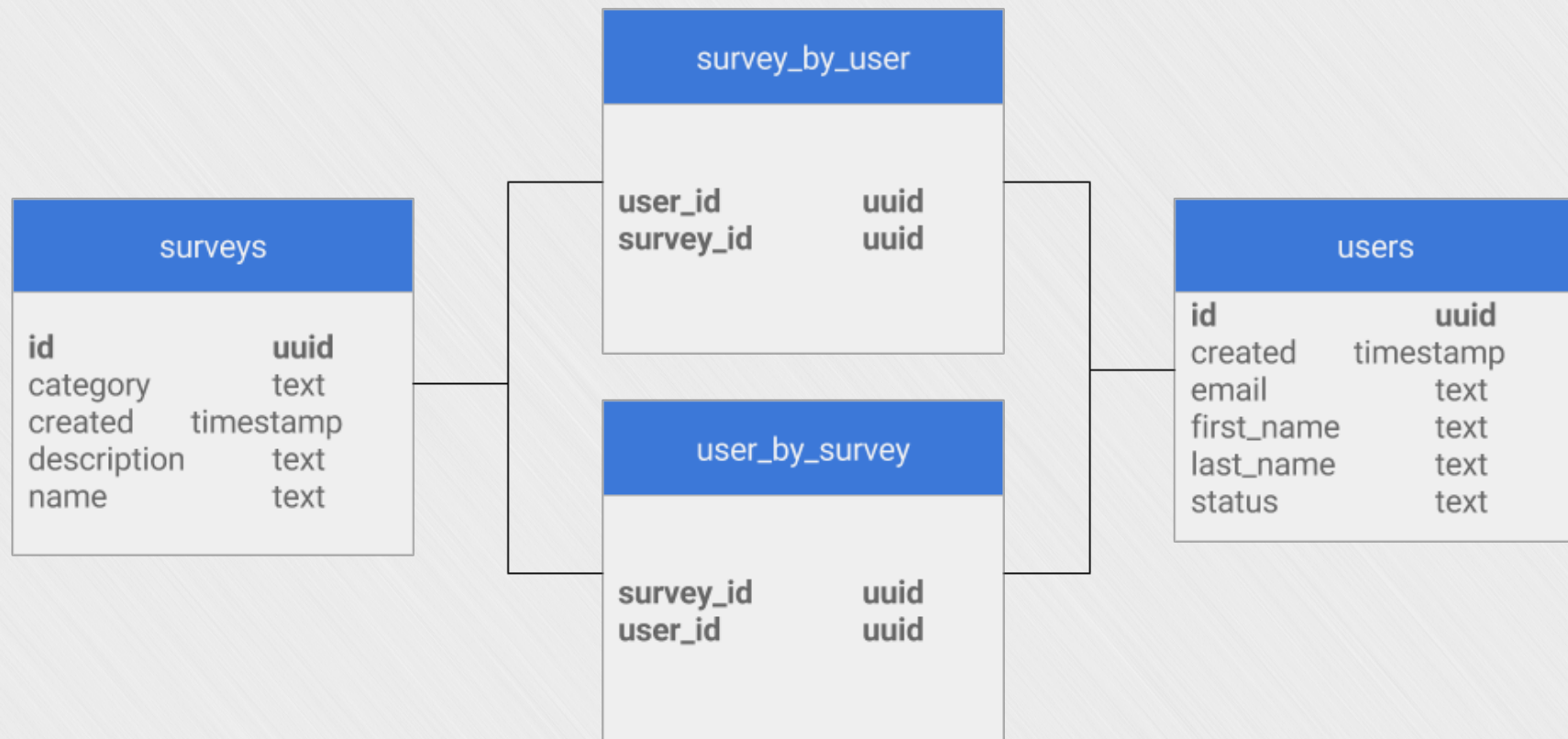


# Quelques mots sur Cassandra

## ➤ Quelles sont ses principales difficultés ?

### ❖ Ses principales difficultés ?

- \* La seule contrainte possible est la clé primaire
- \* Nécessité de créer sa base des requêtes qu'on souhaite effectuer





# Conclusion



## Synthèse de la présentation


# Conclusion

## ➤ Synthèse


- \* NodeJS :
  - Un langage de programmation : JavaScript
  - Un format de données : JSON
  - Un paradigme de programmation : l'asynchrone
  
- \* Si vous rencontrez un problème, il y a 99% de chances pour que quelqu'un ait déjà créé une librairie pour vous :

nuclear pumpkin mocha


[npm On-Site](#)
[npm Private Packages](#)
[npm Open Source](#)
[documentation](#)
[support](#)




find packages

[sign up or log in](#)



npm is the package manager for **javascript**.




**231 725**  
total packages



**142 986 855**  
downloads in the last day



**780 610 476**  
downloads in the last week



**2 939 709 797**  
downloads in the last month

packages people 'npm install' a lot

# Conclusion

## ➤ Synthèse

### \* Cassandra :

- Une base de données NoSQL
- Création de sa base en fonction de ses requêtes
- Accès très rapide aussi bien en lecture qu'en écriture
- Mondialement utilisé :

**GitHub**



**ebay**



**NETFLIX**



**Adobe**



# Conclusion

## ➤ Quelques liens utiles

- \* <https://www.npmjs.com/>
- \* <https://www.joyent.com/developers/node>
- \* <https://nodejs.org/en/>
- \* <http://nodeschool.io/>
- \* <http://docs.nodejitsu.com/>
- \* <http://howtonode.org/>
- \* <http://www.planetnodejs.com/>
- \* <http://cassandra.apache.org/>
- \* <https://academy.datastax.com/demos/brief-introduction-apache-cassandra>

# Sources

- \* <https://www.npmjs.com/>
- \* <https://www.joyent.com/developers/node>
- \* <https://nodejs.org/en/>
- \* <https://academy.datastax.com/>
- \* <http://cassandra.apache.org/>