



# Interface 2037

Ready for inquiry

S. Drodre - L. Ortolan - A. Pallas - J. Riffard - K. Savaroche

# Sommaire

- Les objectifs
- Notre approche
- Notre organisation
- Rétrospective





# Les objectifs



- Réalisation d'un webservice : architecture REST
- Automatisation des tests d'acceptation
- Implémentation cache serveur : Squid
- Extension N usagers
- Extension N systèmes expert
- Documentation du service
- Simplicité de lancement





# Notre approche

Importer schéma visio Samuel

L'infrastructure





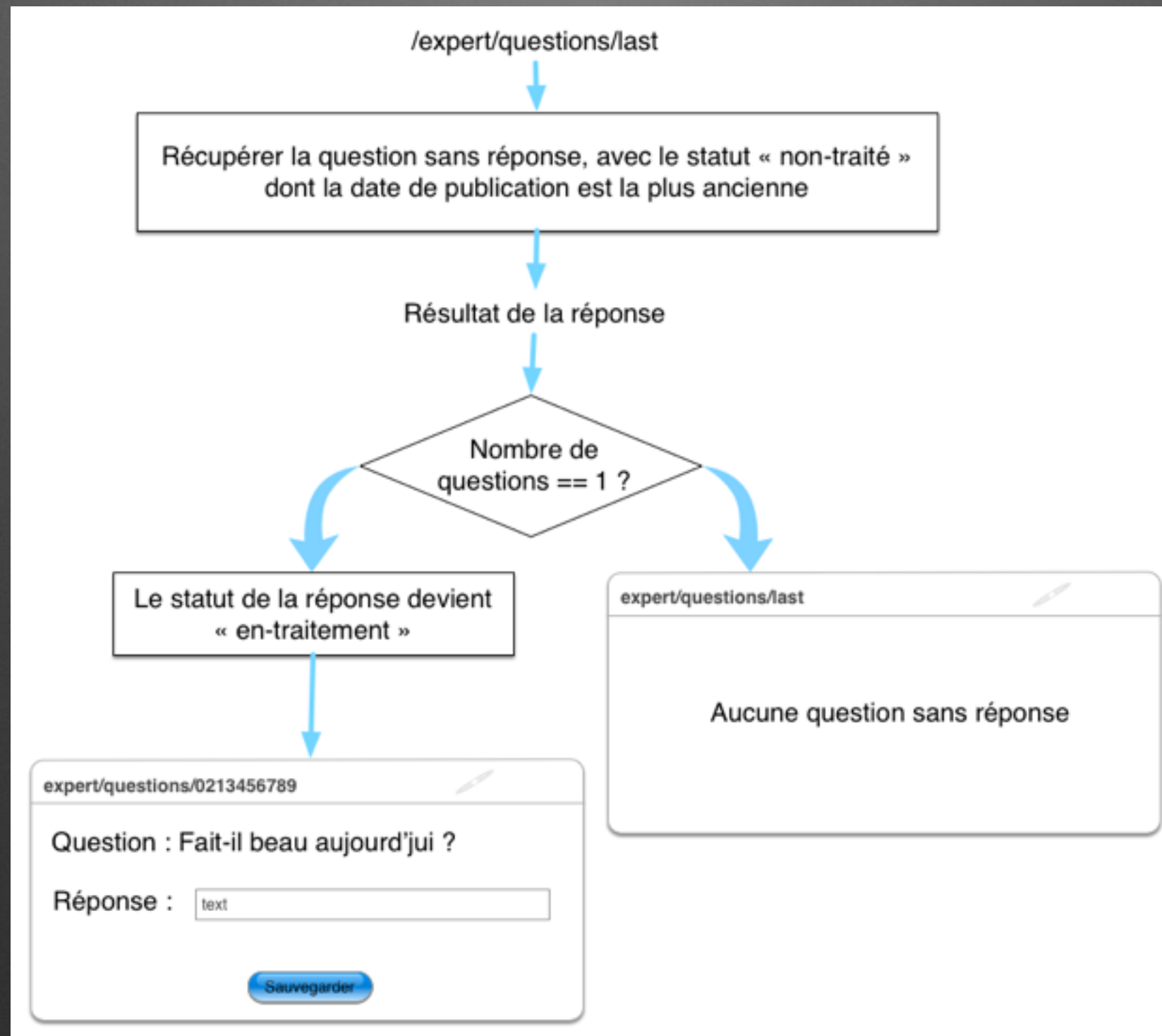
# Extension N usagers

Expliquer l'approche réseau :

Extension N usage avec config SQUID et  
description solution



# Extension N experts





# Automatisation



Un scénario



Test unitaires

Test d'acceptation

**COVERALLS**



```
==== Scénarii ====
Serveur de tests d'acceptation Scénarii lance
Etant donne qu'il n'existe aucune question
Quand le système expert demande la prochaine question au serveur
Alors le serveur indique qu'il n'existe pas de question
Et le système expert se met en veille avant de redemander une question
- Il n'existe aucune question en base
  ✓ Aucune question pour le système expert
Quand l'utilisateur pose une question au serveur
Alors le serveur indique qu'il a enregistré la question
Et il permet à l'utilisateur de localiser la réponse lorsqu'elle sera disponible
- L'utilisateur a posé une question au serveur
- Elle sera disponible à l'adresse : /client/questions/551dc00ead1e4099195e5b03
  ✓ L'utilisateur pose une question au serveur et sait où elle sera disponible
Etant donné qu'il existe une question en attente de réponse
Quand le système expert demande la prochaine question au serveur
Alors il récupère la question en attente
Et la question suivante devient la question en attente
- L'utilisateur a posé une seconde question au serveur
- L'expert récupère la première question posée et la seconde passe en attente
  ✓ Le système expert propose la question en attente
Etant donné qu'un utilisateur a posé une question
Et aucun système expert n'a pas encore traité la question
Quand l'utilisateur demande de consulter la réponse
Alors le serveur indique que la réponse n'est pas encore disponible
- L'utilisateur visualise la question sans réponse.
  ✓ L'utilisateur consulte la question mais elle n'a pas encore de réponse
Etant donné que le système expert a récupéré une question en attente
Et qu'il a trouvé une réponse
Quand il fournit la réponse au serveur
Alors le serveur indique qu'il a enregistré la réponse à la question
Question répondue avec succès ! id : undefined
- La réponse a bien été enregistrée et est disponible à l'adresse :
  ✓ Le système expert fournit une réponse
Etant donné qu'un utilisateur a posé une question
Et un système expert a traité la question
Quand l'utilisateur demande de consulter la réponse
Alors le serveur affiche la réponse du système expert
- L'utilisateur visualise la question avec la réponse du système expert.
  ✓ L'utilisateur consulte la question et le système a bien répondu
Etant donné que le système expert a récupéré une question en attente
Et qu'il a trouvé une réponse
Quand il notifie le serveur de son échec
Alors le serveur enregistre que cette question n'a pas de réponse connue.
Question répondue avec succès ! id : undefined
- La réponse a bien été enregistrée et est disponible à l'adresse :
- L'utilisateur visualise la question qui n'a pu être répondue par l'expert
  ✓ Le système expert indique qu'il ne connaît pas la réponse
```



# Documentation

- Web service : HTTP method option
- /docs/doc\_API.pdf
- Github : README

Method **OPTIONS** URL **http://localhost:5000/client/questions** **SEND**

**Body**

Request Body

**[+] Response**

Response Headers Response Body (Raw) Response Body (Highlight) Response Body (Preview)

1.	Status Code	: 200 OK
2.	Access-Control-Allow-Credentials	: false
3.	Connection	: keep-alive
4.	Date	: Fri, 03 Apr 2015 09:07:20 GMT
5.	Transfer-Encoding	: chunked
6.	X-Powered-By	: Express
7.	access-control-allow-headers	: X-Requested-With, X-HTTP-Method-Override, Content-Type, Accept
8.	access-control-allow-methods	: POST, GET, OPTIONS
9.	access-control-allow-origin	: *

## /client/questions/

### GET

Retourne les trente dernières questions posées.

Liste des codes retours HTTP possibles :

- 200 : succès

#### Paramètres

nom	description	type	défaut
label	paramètre optionnel, indique le contenu de la question que l'utilisateur souhaite rechercher	body	

#### Corps de la réponse

Eléments json	les trente dernières questions posées
Elément json	les questions contenant le label indiqué

La liste des trente dernières question posées en JSON.



# Organisation



- Division du travail en 2 équipes
- Création d'un google drive + Github
- Communication par mail + séances EPSI
- Mise en place de sprints + feedbacks



# Rétrospective



- Création d'une API Restful
- Acquisition de nouvelles compétences techniques
- Organisation de travail efficace et productive
- Expérience d'une équipe hybride : réseau et dev





# Avez-vous des questions



Merci de votre attention :)