Smart-Github-Analyzer

CÉCILE GRACIANNE, ALEXANDRE BLUKACZ, PAUL-ARNAUD PY

CERTIFICATION – CEGEFOS

19 MARS 2018

Objectifs du projet

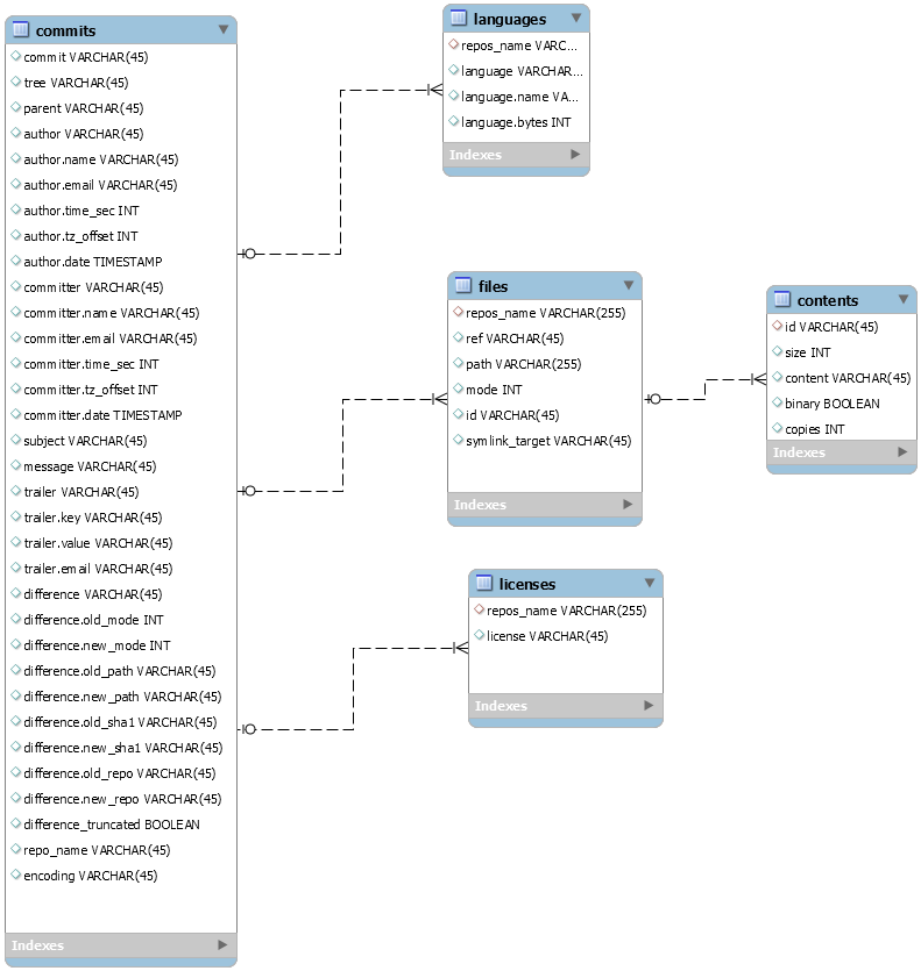
- Utiliser les données open source de la plateforme GitHub pour :
  - Collecter les données et développer une structure de stockage adaptée à la volumétrie et aux modifications constantes de ces données
  - Créer un moteur de recherche interrogeant ces données
  - Identifier les projets qui deviendront populaires à l’avenir
  - Effectuer des recommandations pour les utilisateurs de la plateforme

Description des données sources

- 2 datasets disponibles sur Google BigQuery :
  - **github_repos**
    - 3 TB
    - 9 tables
      - commits
      - contents
      - files
      - languages
      - licenses
      - sample_commits
      - sample_contents
      - sample_files
      - sample_repos

Description des données sources

- 2 datasets disponibles su
  - **github_repos**
    - 3 TB
    - 9 tables
      - commits
      - contents
      - files
      - languages
      - licenses
      - ~~sample_commits~~
      - ~~sample_contents~~
      - ~~sample_files~~
      - sample_repos



Description des données sources

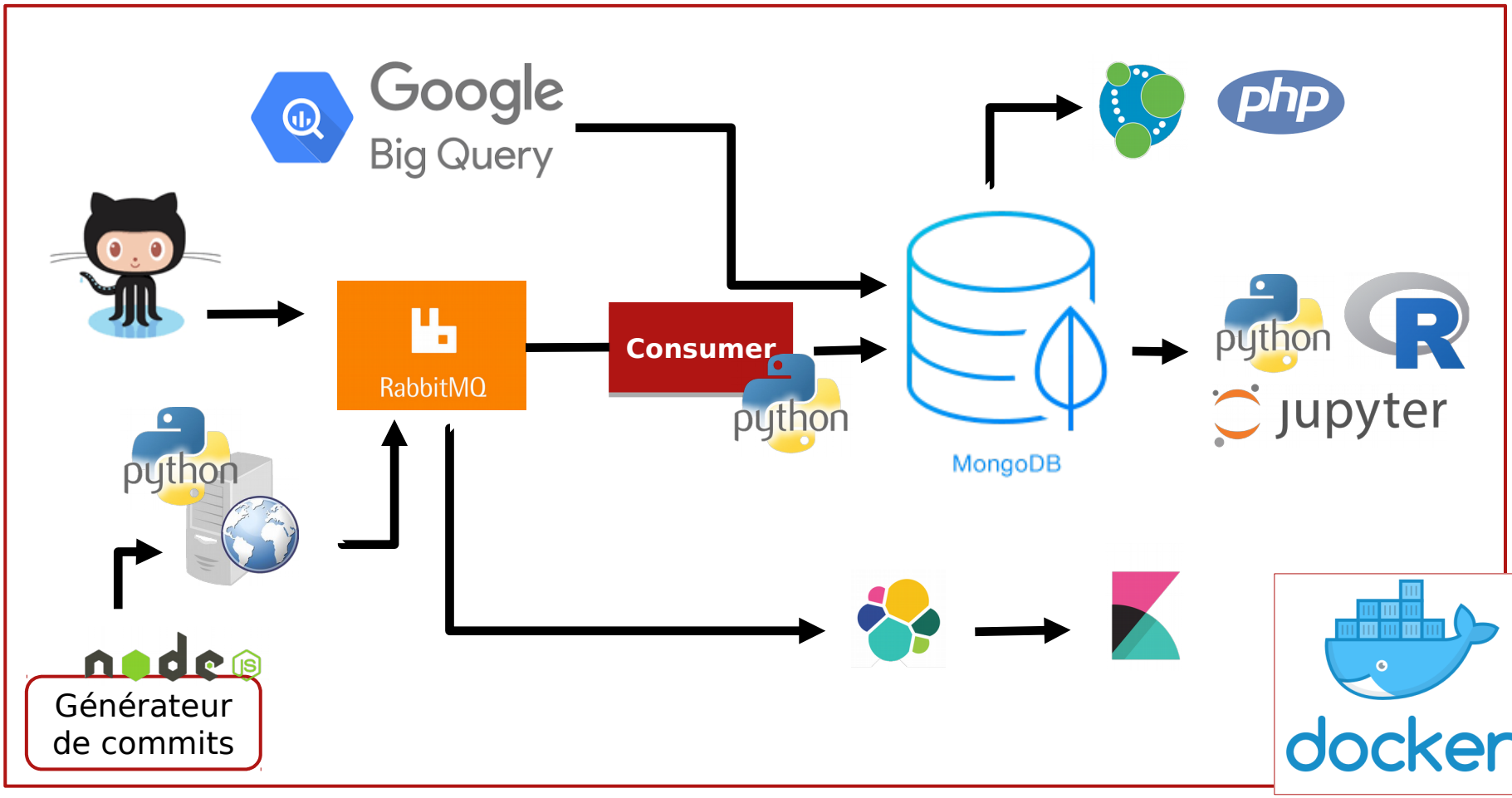
- 2 datasets disponibles sur Google BigQuery :
  - **github_repos**
  - **githubarchives**

Row	type	public	payload	repo.id	repo.name	repo.url	actor.id	actor.login	actor.gravatar_id	actor.avatar_url
1	IssuesEvent	true	{"number":10	null	/	https://api.git	null	null	null	https://secure
2	IssuesEvent	true	{"number":14	null	/	https://api.git	null	null	null	https://secure
3	CreateEvent	true	{"name":"Sid	null	/	https://api.git	null	null	null	https://secure
4	IssuesEvent	true	{"number":96	null	/	https://api.git	null	null	null	https://secure
5	IssuesEvent	true	{"number":73	null	/	https://api.git	null	null	null	https://secure
6	IssuesEvent	true	{"number":18	null	/	https://api.git	null	null	null	https://secure
7	PushEvent	true	{"shas":[["de	null	/	https://api.git	null	null	null	https://secure
8	CreateEvent	true	{"name":"san	null	/	https://api.git	413899	naoty	0031b165c1f8	https://secure
9	GistEvent	true	{"name":"gist	null	/	https://api.git	17495	woodie	0439edc42c4	https://secure
10	CreateEvent	true	{"name":"tuto	null	/	https://api.git	508577	vladvoic	079b0f7167db	https://secure

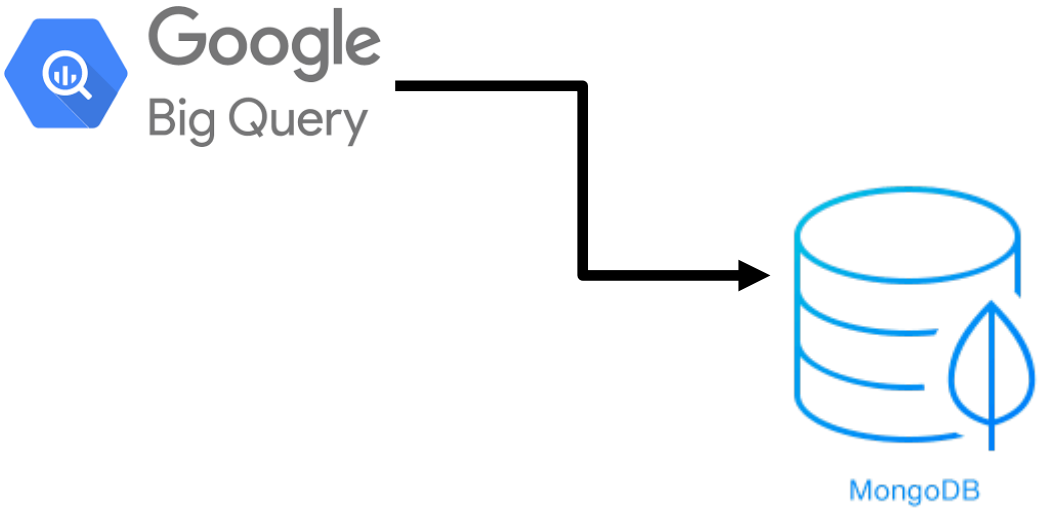
Par jour, mois, années

Row	actor.url	org.id	org.login	org.gravatar_id	org.avatar_url	org.url	created_at	id	other
1	https://api.git	null	null	null	null	null	7:38.000 UTC	1154270429	null
2	https://api.git	null	null	null	null	null	3:38.000 UTC	1154610989	null
3	https://api.git	null	null	null	null	null	5:11.000 UTC	1154218032	null
4	https://api.git	null	null	null	null	null	0:39.000 UTC	1154266629	null
5	https://api.git	null	null	null	null	null	2:36.000 UTC	1154237601	null
6	https://api.git	null	null	null	null	null	9:41.000 UTC	1154923509	null
7	https://api.git	null	null	null	null	null	1:00.000 UTC	1154870299	null
8	https://api.git	null	null	null	null	null	5:56.000 UTC	1154581173	null
9	https://api.git	null	null	null	null	null	5:40.000 UTC	1153898859	null
10	https://api.git	null	null	null	null	null	9:36.000 UTC	1154106861	null

Architecture globale



1. Collecte des données



1. Collecte des données

Google BigQuery

COMPOSE QUERY

Query History

Job History

Filter by ID or label

My First Project

- analyse_comments_author
  - archive
  - **archive2**
  - commits
  - contents_plus1copie
  - files
  - languages
  - licences
  - repository
  - results_20180226_102743
- **bigquery-public-data**
- **githubarchive**
- **Public Datasets**
  - gdelt-bq:hathitrustbooks
  - gdelt-bq:internetarchivebooks
  - lookerdata:cdc
  - nyc-tlc:green
  - nyc-tlc:yellow

New Query

Query Editor | UDF Editor

```
1  SELECT t2.* FROM [analyse_comments_author.repository] t1
2  INNER JOIN
3  (
4    SELECT *
5    FROM
6      [githubarchive:year.2017]
7    WHERE
8      type IN('WatchEvent','ForkEvent', 'IssuesEvent','PushEvent','PullRequestEvent','CreateEvent')
9  ) t2
10 ON
11   t1.repo_name=t2.repo.name
```

RUN QUERY | Save Query | Save View | Format Query | Show Options

Table Details: archive2

Refresh | Query Table | Copy Table | Export Table | Delete Table

Schema | Details | Preview

Row	type	public	
1	ForkEvent	true	{"forkee":{"id":90048621,"name":"big","full_name":"otime/big","owner":{"login":"otime","id":28112241,"avatar_url":"https://avatars2.githubusercontent.com/u/28112241?v=3","gr
2	ForkEvent	true	{"forkee":{"id":86054536,"name":"triqs","full_name":"mailhexu/triqs","owner":{"login":"mailhexu","id":6870380,"avatar_url":"https://avatars2.githubusercontent.com/u/6870380?v
3	ForkEvent	true	{"forkee":{"id":78380310,"name":"Firmware","full_name":"gcrisis/Firmware","owner":{"login":"gcrisis","id":13515569,"avatar_url":"https://avatars.githubusercontent.com/u/13515
4	ForkEvent	true	{"forkee":{"id":82462274,"name":"Firmware","full_name":"goovsgoo/Firmware","owner":{"login":"goovsgoo","id":9588808,"avatar_url":"https://avatars.githubusercontent.com/u/
5	ForkEvent	true	{"forkee":{"id":95787762,"name":"Firmware","full_name":"TEAMIFOR/Firmware","owner":{"login":"TEAMIFOR","id":24207286,"avatar_url":"https://avatars2.githubusercontent.c
6	ForkEvent	true	{"forkee":{"id":97905342,"name":"Firmware","full_name":"abcxyz1111111/Firmware","owner":{"login":"abcxyz1111111","id":24379678,"avatar_url":"https://avatars3.githubusercon
7	ForkEvent	true	{"forkee":{"id":87673408,"name":"Firmware","full_name":"droneman59/Firmware","owner":{"login":"droneman59","id":18642727,"avatar_url":"https://avatars2.githubusercontent
8	ForkEvent	true	{"forkee":{"id":87106820,"name":"Firmware","full_name":"lingxiaw/Firmware","owner":{"login":"lingxiaw","id":19678358,"avatar_url":"https://avatars1.githubusercontent.com/u/1
9	ForkEvent	true	{"forkee":{"id":94336290,"name":"Firmware","full_name":"hyperorbit/Firmware","owner":{"login":"hyperorbit","id":29410447,"avatar_url":"https://avatars1.githubusercontent.com
10	ForkEvent	true	{"forkee":{"id":89977654,"name":"Firmware","full_name":"liulisheng10/Firmware","owner":{"login":"liulisheng10","id":28280901,"avatar_url":"https://avatars3.githubusercontent.
11	ForkEvent	true	{"forkee":{"id":103649820,"name":"Firmware","full_name":"Myeinnovation/Firmware","owner":{"login":"Myeinnovation","id":22832420,"avatar_url":"https://avatars2.githubusercc
12	ForkEvent	true	{"forkee":{"id":79974739,"name":"Firmware","full_name":"tcheehow/Firmware","owner":{"login":"tcheehow","id":5392697,"avatar_url":"https://avatars.githubusercontent.com/u/5

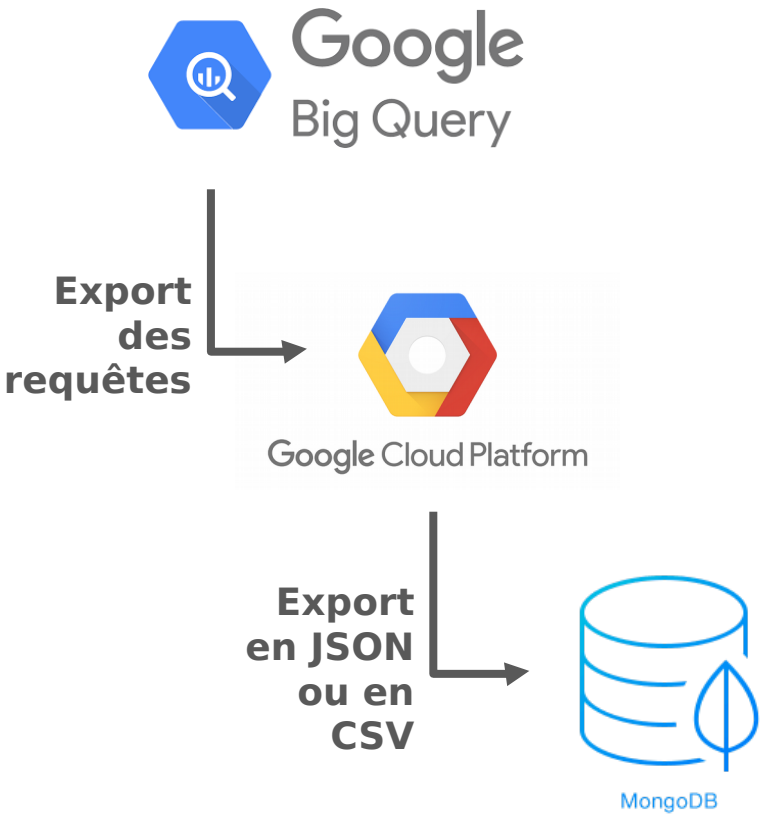
Table | JSON

First < Prev Rows 1 - 12 of 4341654 Next > Last

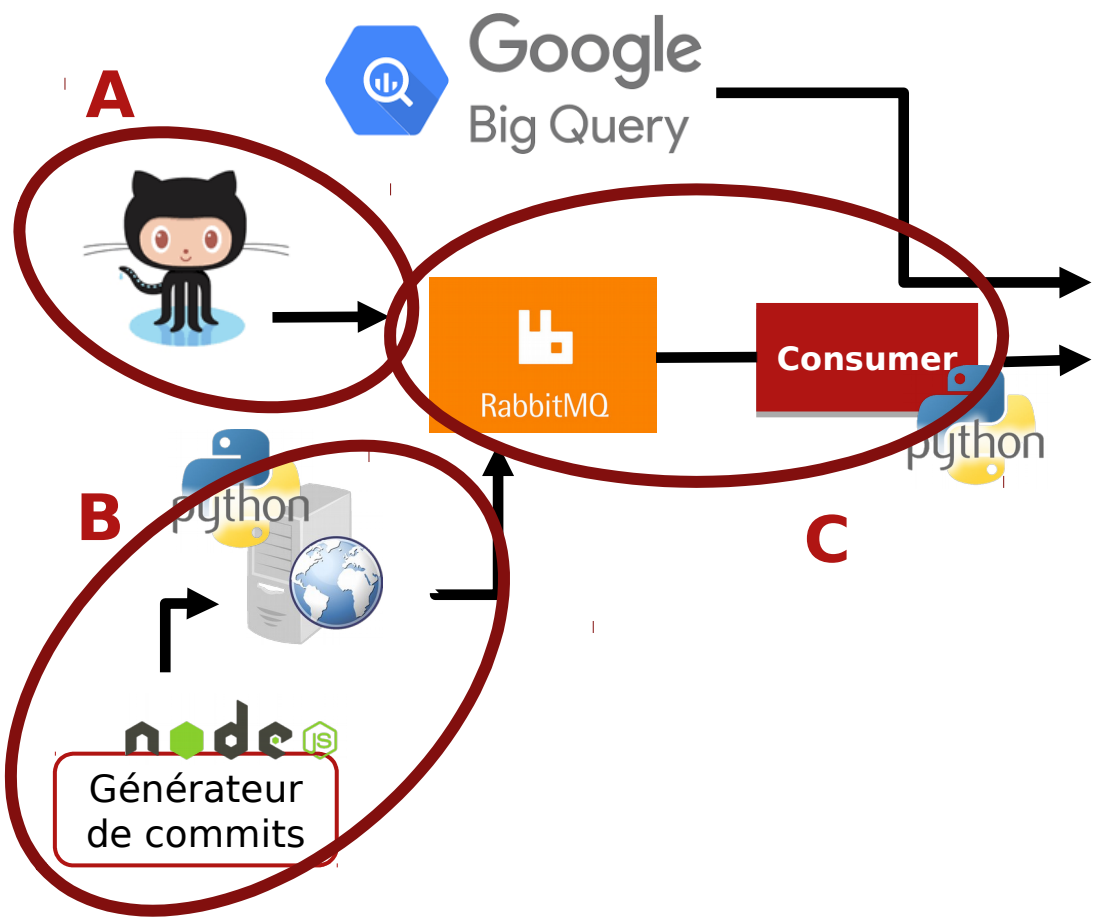
2. Stockage des données

mongoDB

- Pourquoi MongoDB ?
  - Pas de schéma de données prédéfini
  - Format JSON
  - Existence d'une version stand-alone



3. Mise à jour de la base de données



A. **Collecte des données provenant des archives Github**

B. **Ajout de commits dans les données source**

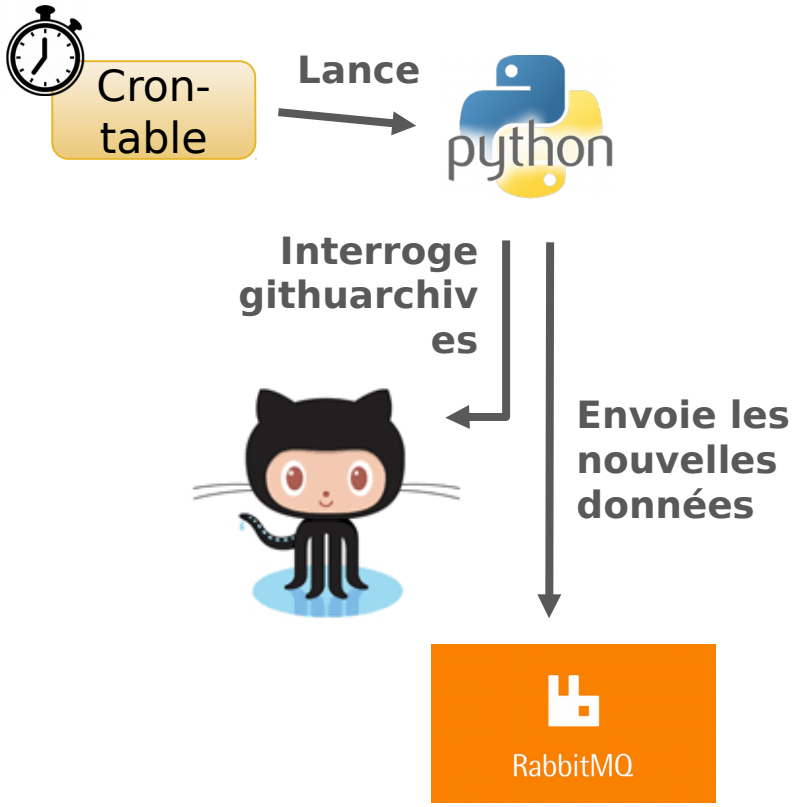
C. **Broker/consumer**

D. **Difficultés rencontrées**

3. Mise à jour de la base de données

A. Collecte des données provenant des archives Github

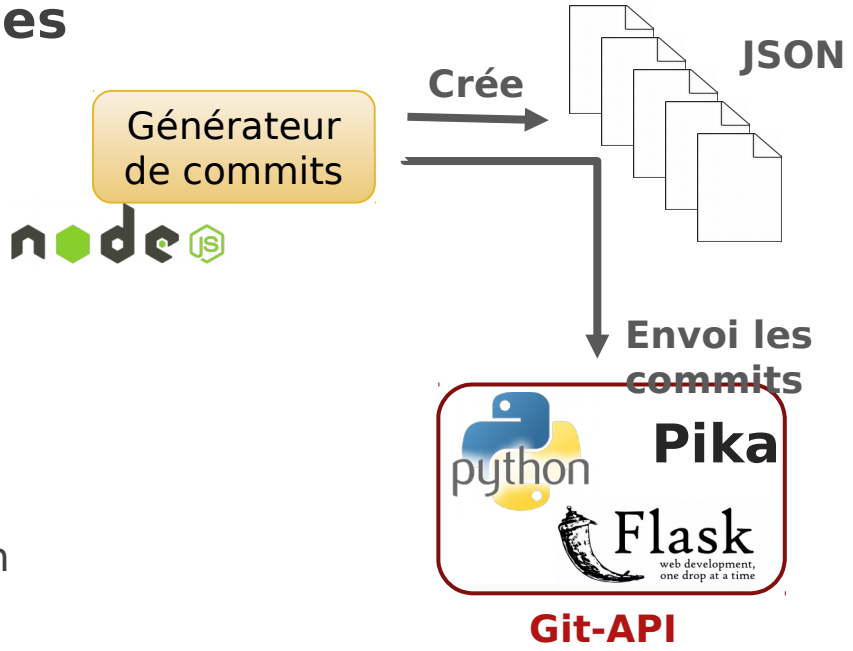
- Cron-table :
  - Mise à jour de la base de données githubarchives toutes les heures
- Script python :
  - Collecte en format JSON les nouvelles données
  - Décompresse les fichiers JSON à la volée
  - Envoi les données dans RabbitMQ



3. Mise à jour de la base de données

B. Ajout de commits dans les données source

- Générateur de commits
  - NodeJS
  - Création de faux commits à partir d'un modèle de commit en format JSON
  - Association de ces commits à un projet fictif
  - Envoi de ces commits sur un serveur web, puis sur rabbitMQ
- Git-API
  - Serveur web

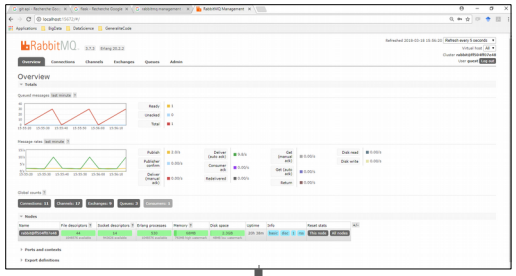


3. Mise à jour de la base de données

c. Broker/consumer

- RabbitMQ (broker)
  - Gestionnaire de file d'attente
  - Grande fiabilité sur forte montée en charge
  - Gain de flexibilité : modèles complexes de file d'attente possible
- RabbitMQ Management plug-in
  - HTTP API
- Consumer
  - Python

http://localhost:15672



Administration et visualisation des données en temps réel

RabbitMQ

Désynchronisation de l'envoi et l'ajout des données dans mongoDB

python **Pika**

pymong

Q Consumer

Envoie les fichiers un à un dans mongoDB

3. Mise à jour de la base de données

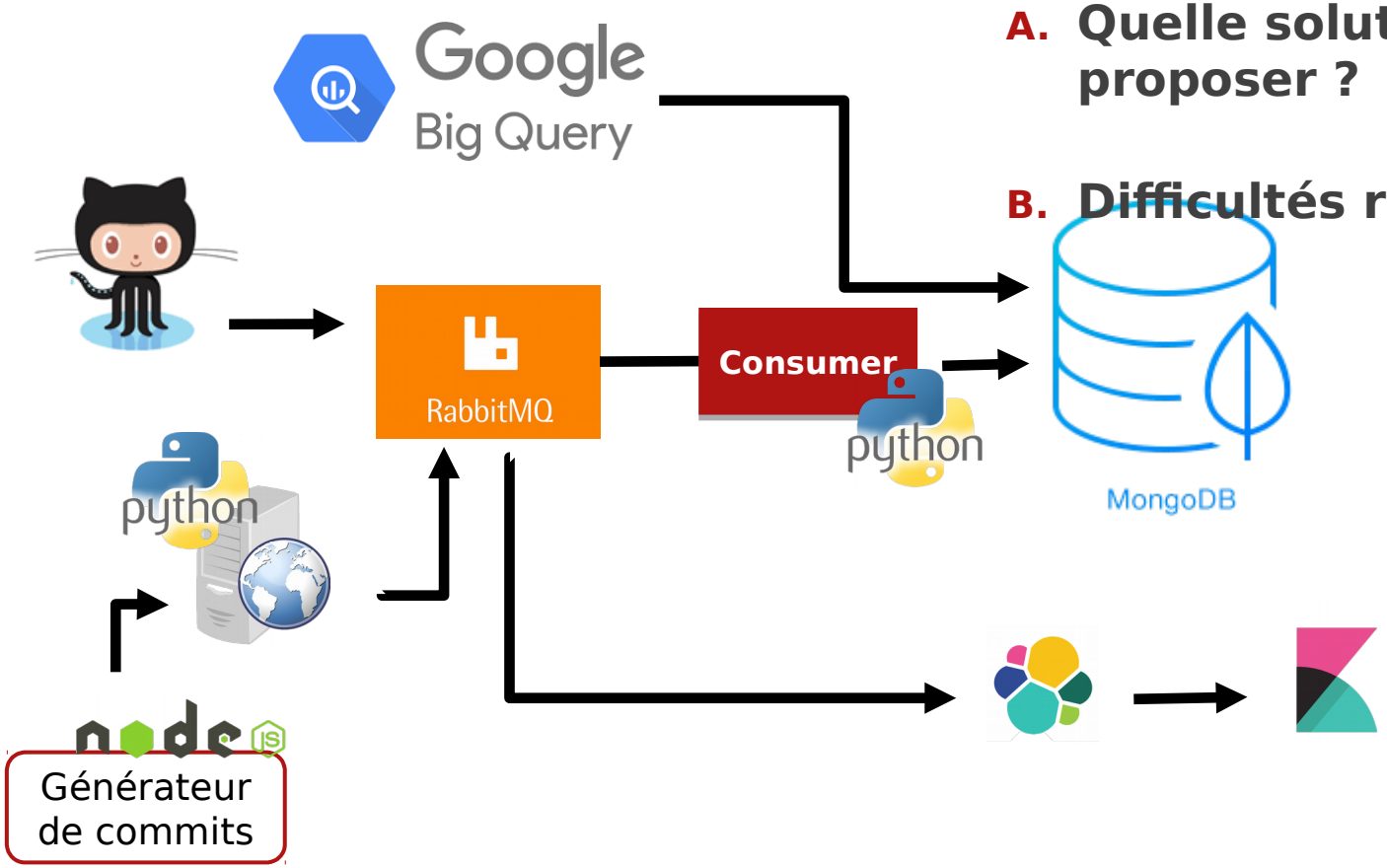
D. Difficultés rencontrées

- Applicabilité de la solution sur Windows
  - Incompatibilités des formats de retour à la ligne Linux/Windows
- Library python Pika
  - Plus de mise à jour depuis 3 ans
  - Library alternative : Celery

4. Rechercher de l'information dans les données

A. **Quelle solution proposer ?**

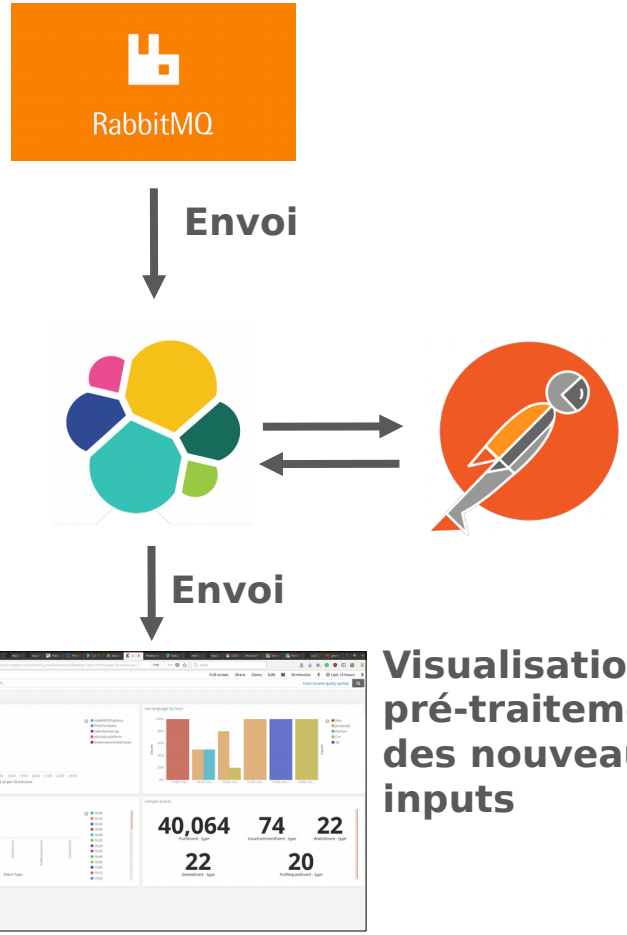
B. **Difficultés rencontrées**



4. Rechercher de l'information dans les données

A. Quelle solution proposer ?

- ElasticSearch
  - Recherche sur tous les champs possible (y compris sur les contenus des commits et des commentaires)
  - Recherche rapide
- Postman
  - Pour tester la bonne marche d'elasticSearch et certaines requêtes
- Kibana
  - Interface visuelle et dynamique
  - Interconnexion native avec ElasticSearch



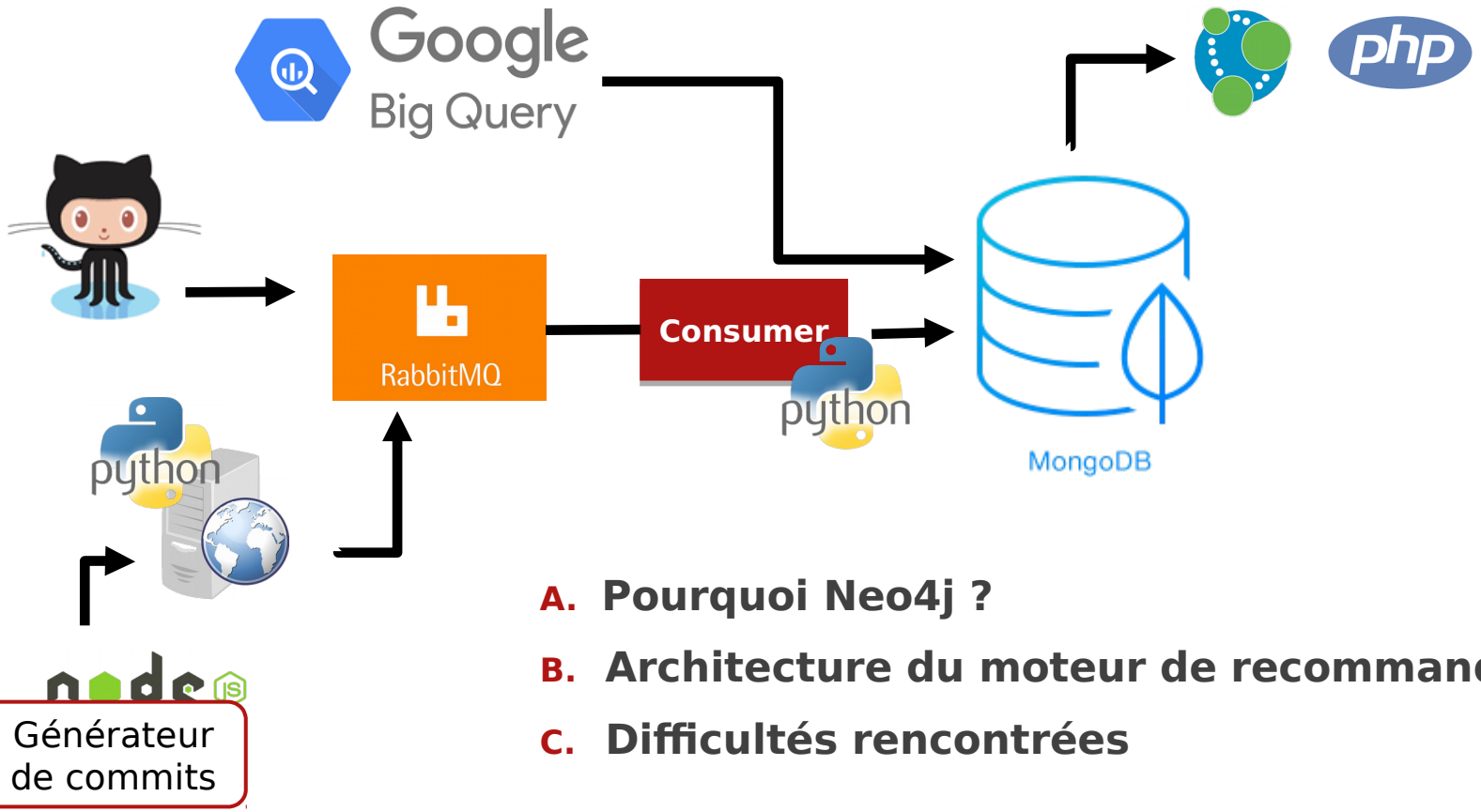
Visualisation et pré-traitement des nouveaux inputs

4. Rechercher de l'information dans les données

B. Difficultés rencontrées

- Conflit ElasticSearch/Kibana - Neo4j
  - Les versions 5.6.8 de Kibana et ElasticeSearch entrent en conflit dans le docker-compose avec Neo4j sous windows
  - Utilisation des versions 6.2.2 pour éviter ce conflit
- Connecteur MongoDB
  - Pas encore de versions disponible gratuitement compatibles avec les versions 6.2.2 d'ElasticSearch et Kibana
  - Manque de temps pour en développer un nous-même

5. Identifier des pistes de travail pour les utilisateurs de Github



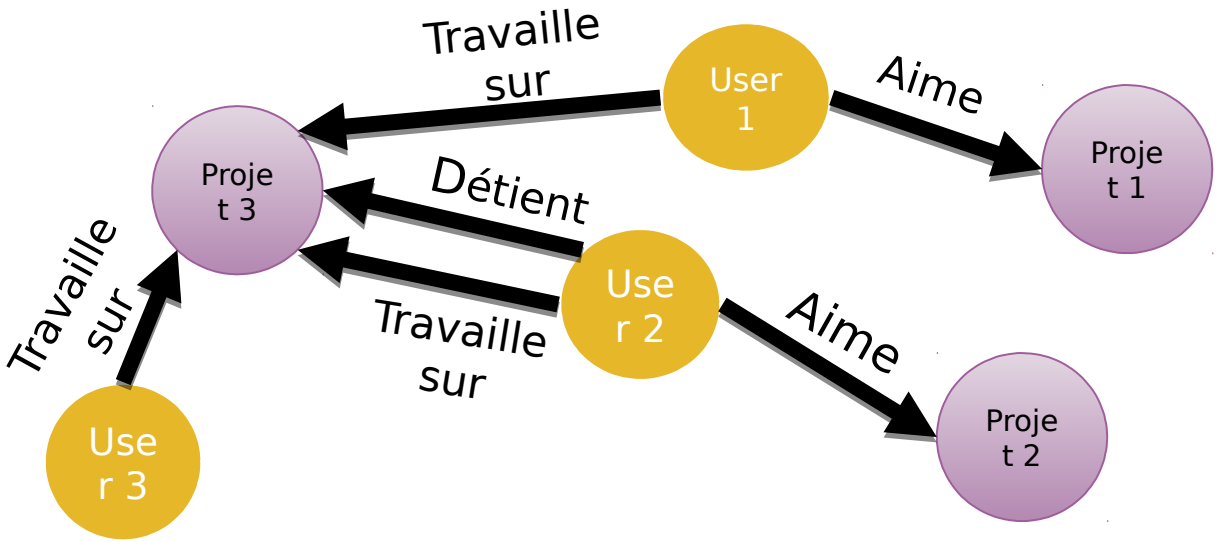
A. **Pourquoi Neo4j ?**
B. **Architecture du moteur de recommandations**
C. **Difficultés rencontrées**

5. Identifier des pistes de travail pour les utilisateurs de Github

A. Pourquoi Neo4j ?

- Interprétation possible des events de github comme des relations

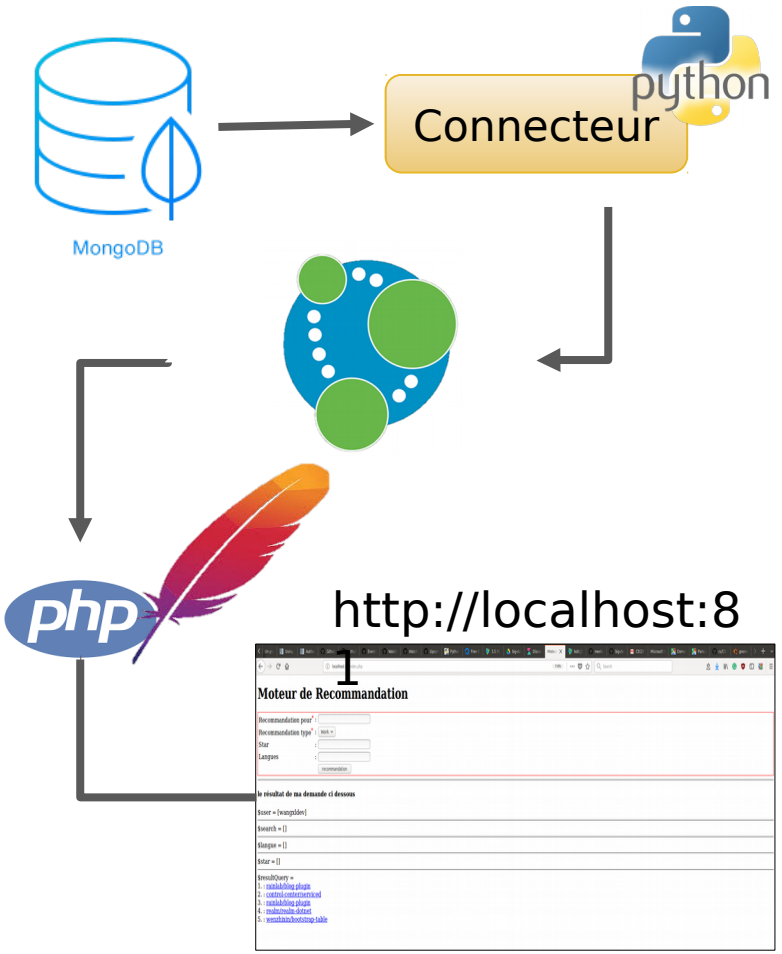
  Recommander à un utilisateur un projet sur lequel il pourrait travailler = comprendre les **relations** entre les projets et les utilisateurs de Github



5. Identifier des pistes de travail pour les utilisateurs de Github

B. Architecture du moteur de recommandations

- Connecteur MongoDB-Neo4j
  - Python
  - Collecte et tri les données dans mongoDB
  - Insertion des données dans Neo4j
- Neo4j
  - Modélisation relationnelle des données
  - Visualisation des résultats
- Serveur Apache
  - php
  - Interrogation de Neo4j et visualisation des recommandations

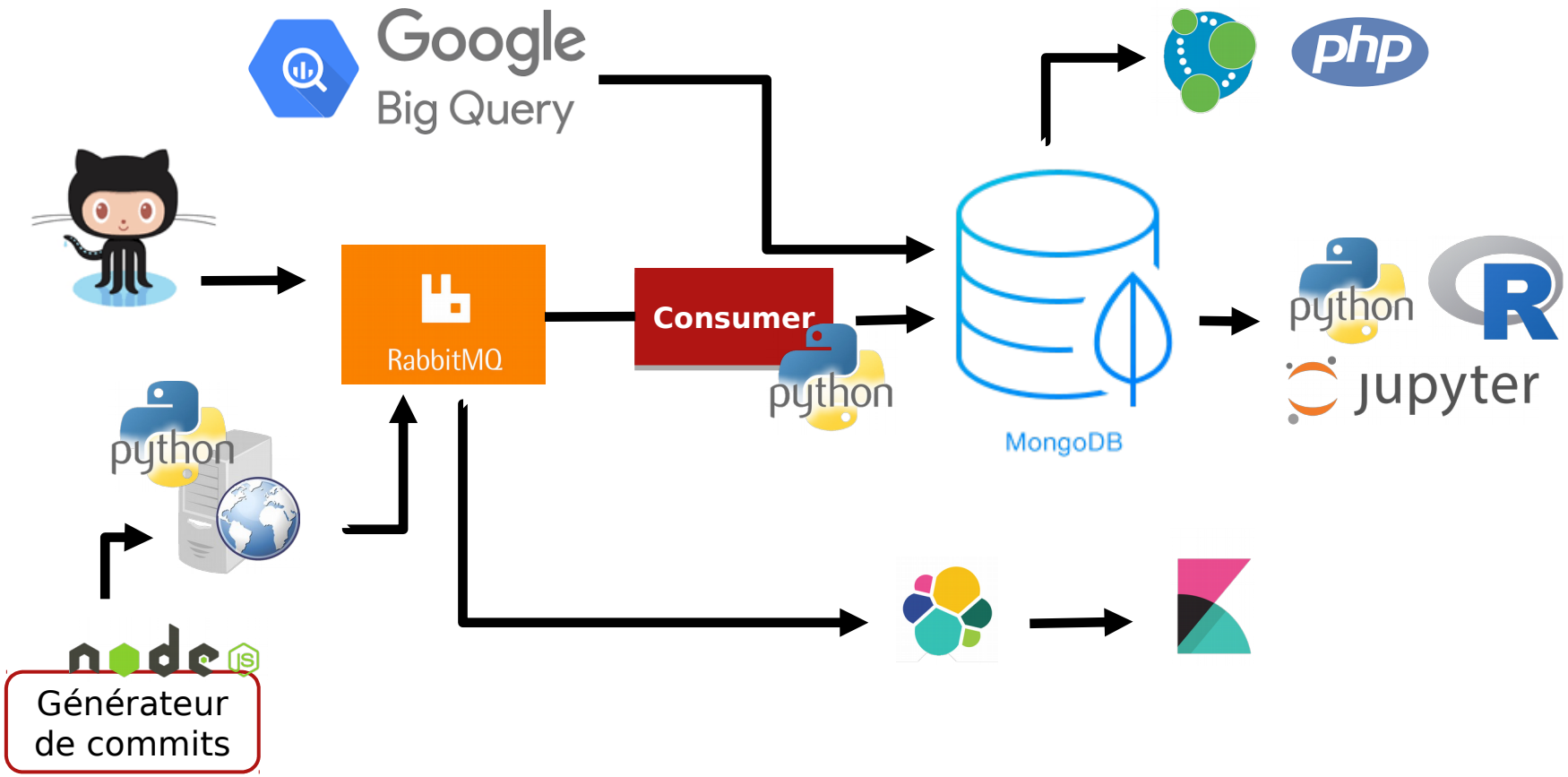


4. Rechercher de l'information dans les données

D. Difficultés rencontrées

- Connecteur MongoDB-Neo4j
  - Mongo connector + Neo Doc Manager
    - Requiert un replicaset d'un cluster mongo pour fonctionner
    - Importe l'intégralité des données des documents JSON
    - Tri possible avec une version payante du connecteur
  - Plug-in Neo APOC en swift
    - Interrogation sur mongoDB extrêmement lente
  - Développement d'un connecteur spécialisé
- Importation d'une fraction seulement des données de mongoDB dans Neo4j

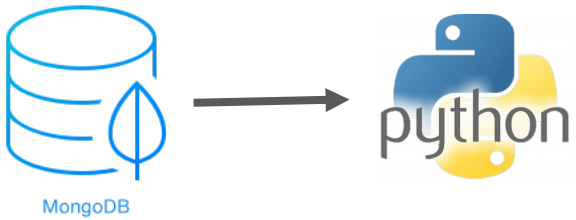
6. Prédire la popularité des projets Github



6. Prédire la popularité des projets Github

A. Transformation des données

- Adoption massive d'un projet = projet populaire
- Qu'est-ce qu'un projet populaire ? Quels indicateurs utiliser ?
  - Activité du repository : nombre de commits, de push event, de fork event etc.
  - Le nombre de stars
  - Le nombre de watchEvent/unité de temps
  - Le nombre de contributeurs
  - La popularité des contributeurs
  - Le langage utilisé
  - Le type de licence du projet

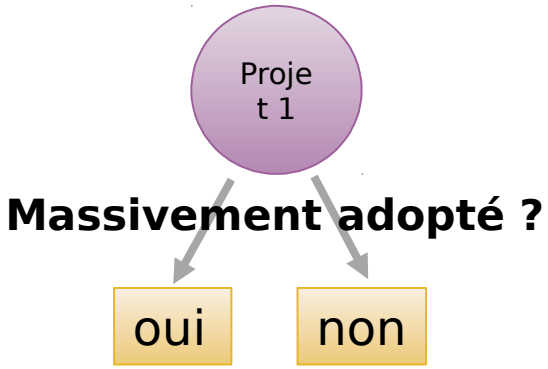


- Nécessité de transformer les variables qualitatives brutes en données quantitatives

6. Prédire la popularité des projets Github

B. Modèle de prédiction

- Un projet est soit massivement adopté, soit il ne l'est pas
  - Apprentissage supervisé : classification
    - Arbres de décisions
    - Random Forest
    - Naive Bayes
    - Support vector machine
    - K-mean
    - etc.
  - Quel seuil (de popularité) considérer ?

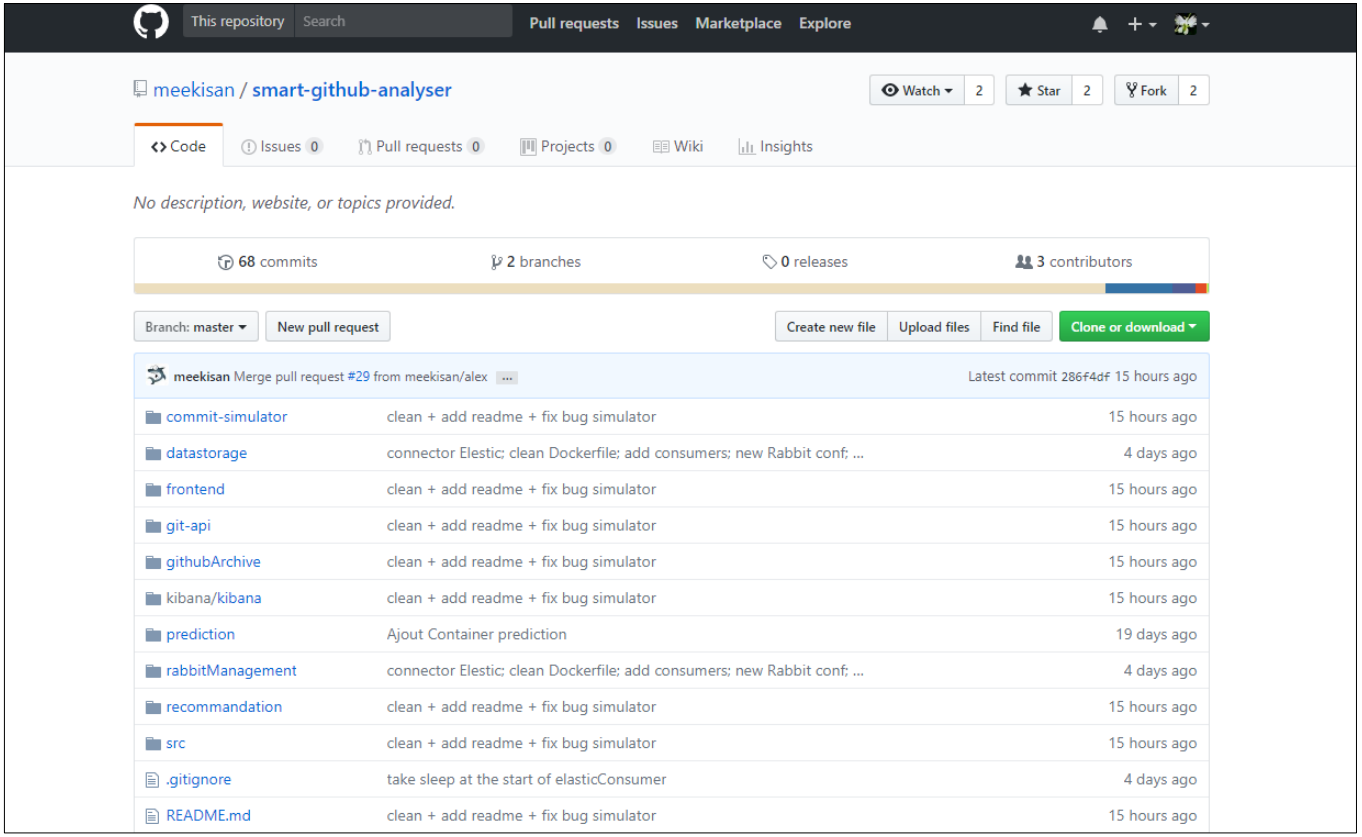


Smart-Github-Analyzer : to be continued...

- Architecture inachevée, mais qui devrait supporter une montée en charge pour une mise en production, sous réserve de :
  - Déployer des clusters MongoDB et ElasticSearch
  - Adapter l'import de données dans Neo4j
- Terminer le projet
  - Finaliser le moteur de recherche
  - Finaliser le moteur de prédictions
- S'appuyer sur des solutions cloud
  - Machines de collecte des données dans AWS

Smart-Github-Analyzer : to be continued...

- **Readme**
- **Only on Github !!**



Merci de votre attention !