



Cloud et Système Web

V. SERVERLESS

Stéphane WOUTERS
École WIS 2019/2020



SERVER LESS

"MOINS DE SERVEURS ?"

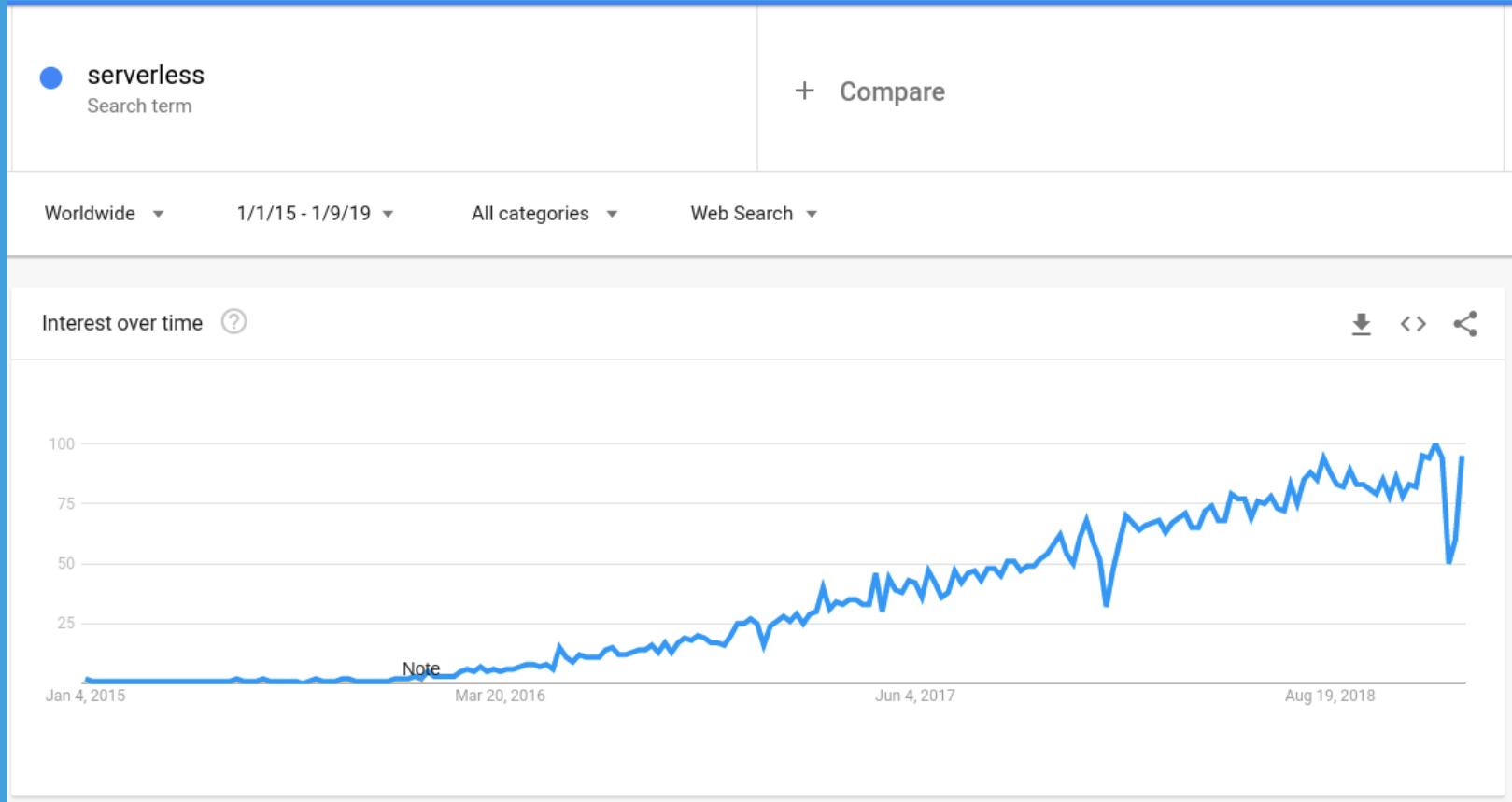
SERVERLESS

Quand la partie serveur d'une application est abstraite à son maximum.

Une manière de développer une application où on ne s'occupe plus du serveur

SERVERLESS

L'un des buzzworld du cloud



Nombre de recherches Google sur le mot "serverless"

Pourquoi un intérêt pour supprimer les serveurs ?

La gestion de serveur a un coût opérationnel :

- La conception de l'infrastructure
- La mise en place de l'infrastructure
- La mise à jour logicielle des serveurs
- Les interventions pour corriger les problèmes
- L'astreinte nécessaire pour superviser l'infrastructure

Le serverless serait une suppression de ces coûts. Ça intéresse les entreprises...

COMMENT FAIRE DU SERVERLESS ?

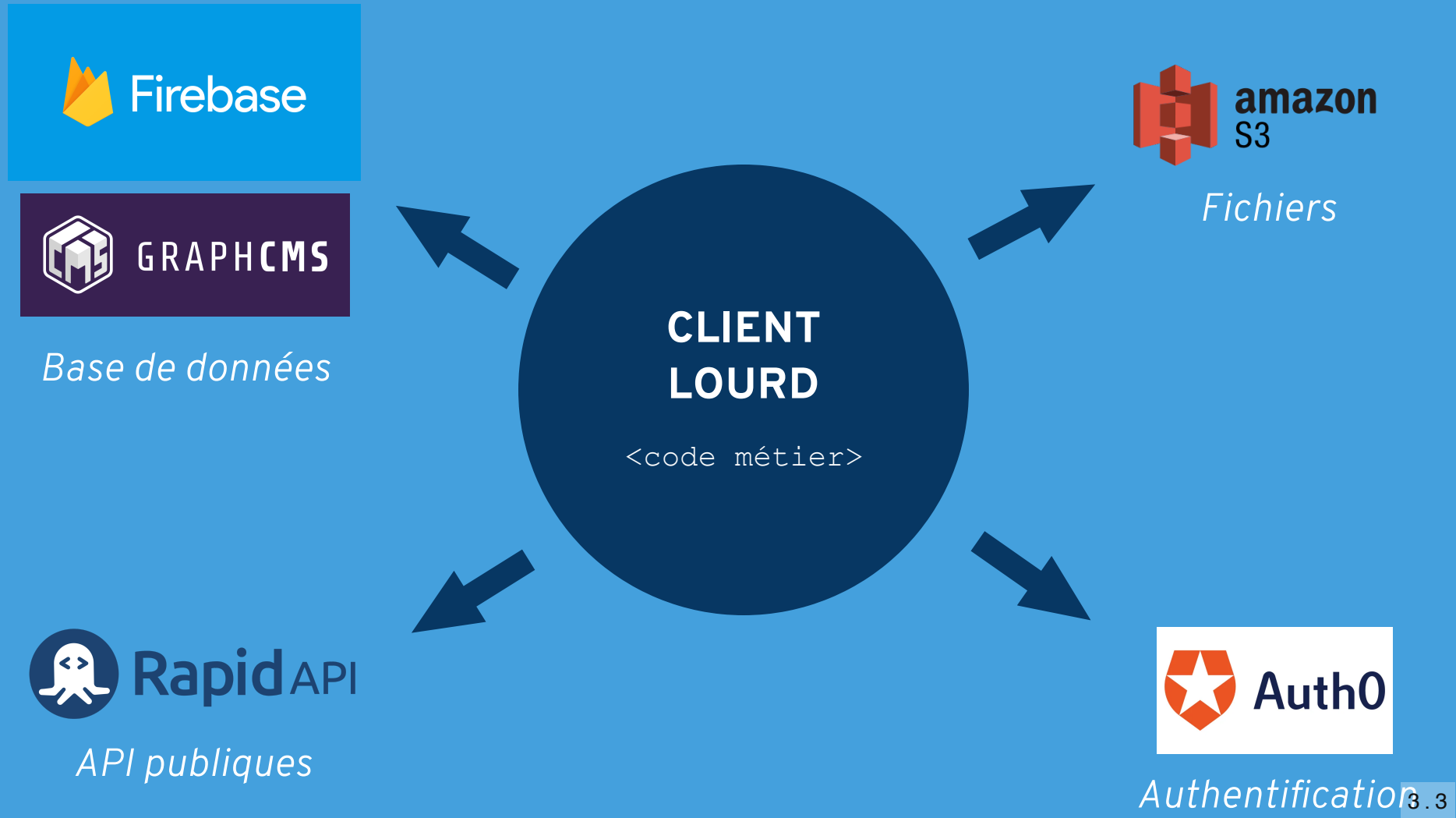
2 MÉTHODES CONNUES AUJOURD'HUI :

1. EN CLIENT LOURD CONNECTÉ À DES SERVICES DE DONNÉES HÉBERGÉS DANS LE CLOUD

On utilise des services en SAAS dans le cloud
pour générer des API, stocker de la donnée,
gérer l'authentification...

Le code métier est côté client.

1. EN CLIENT LOURD CONNECTÉ À DES SERVICES DE DONNÉES HÉBERGÉS DANS LE CLOUD



2. AVEC DU FUNCTION AS A SERVICE

On développe l'application côté serveur.

le code est cloisonné dans des fonctions qui
seront exécutées dans des mini conteneurs

Leur exécution est assurée par le prestataire.

➡ **C'est du FAAS (Fonction As A Service)**



Authentification



Base de données



Vue graphique



Fichiers

FAAS

FUNCTION AS A SERVICE

FAAS revient à exécuter du code backend sans
gérer ses propres serveurs

Contrairement au PAAS, l'utilisateur ne s'occupe
même pas du dimensionnement

De manière générale, qu'est ce qu'une fonction ?



En programmation, une fonction est un bout de code qui prend des paramètres et qui retourne un résultat

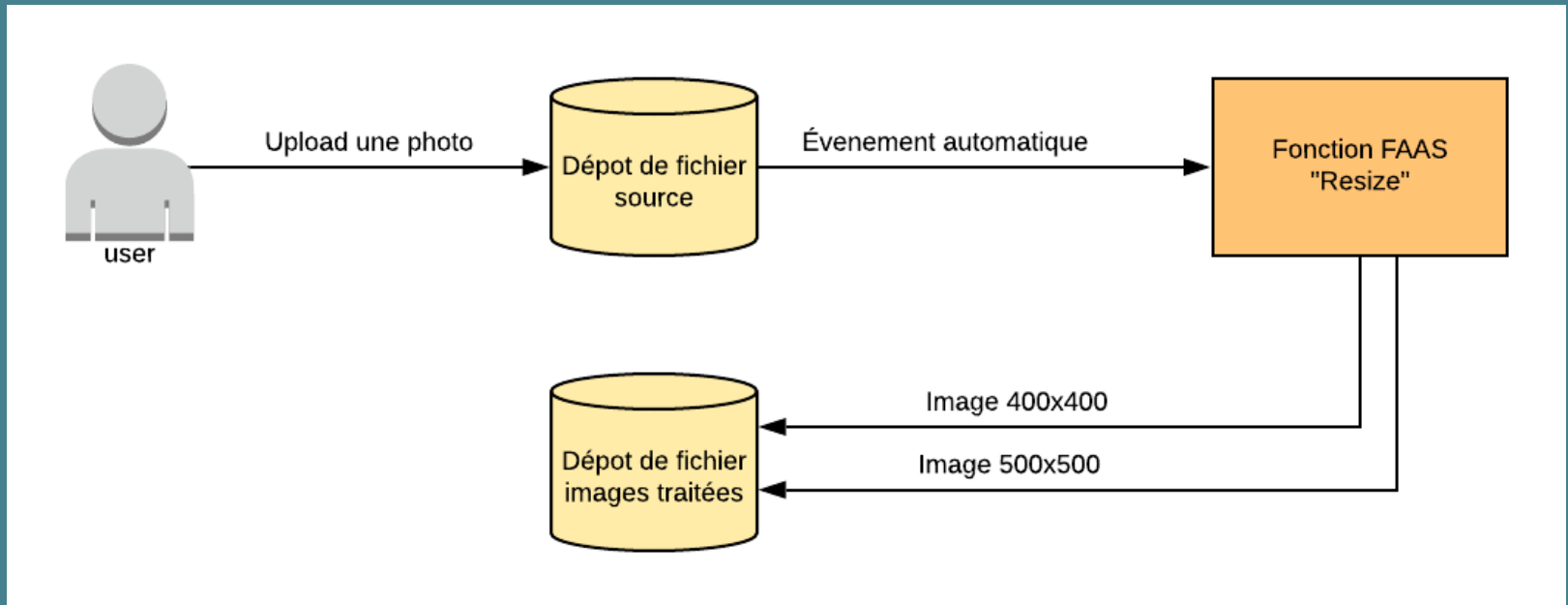
DÉCLENCHEMENT

Les fonctions peuvent être configurées pour être déclenchées de manières différentes :

- Déclenchées par **un évènement** (dépôt de fichier par un utilisateur, exécution périodique...)
- Par **requête HTTP** (permet de créer une API)

DÉCLENCHEMENT

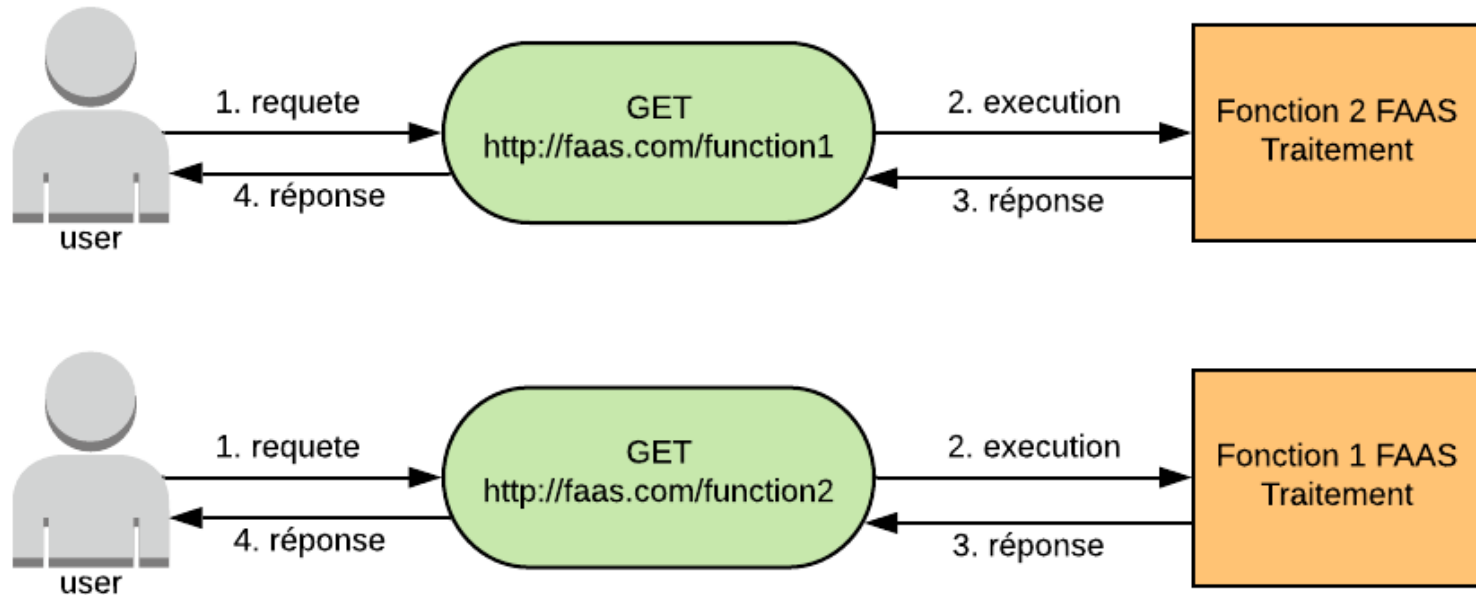
Exemple avec un évènement "dépôt de fichier"



Pour redimensionner automatiquement des images

DÉCLENCHEMENT

À une URL HTTP pour créer des webservice :



DIFFÉRENCES DANS LE DÉVELOPPEMENT

Ne nécessite pas de langage ou de framework particulier.

Par exemple, sur AWS Lambda on peut envoyer des fonctions en Javascript, Python, Go, Java...

La différence réside dans le découpage du code

Une fonction = Un code source indépendant

DÉPLOIEMENT

Le déploiement est différent car il n'y a pas de serveur applicatif.

Le code source est directement envoyé sur le FAAS

AUTO SCALING

La mise à l'échelle est totalement automatique et gérée par le prestataire.

L'utilisateur n'a pas connaissance du nombre de machines utilisées

Afin de respecter ses engagements, l'opérateur définit souvent des quotas.

100 requêtes max / secondes, 30 secondes d'exécution, 6 MO en sortie maximum...

La facturation ne se fait pas au nombre de ressources consommées mais au nombre d'appels

Sur AWS : 0,20 USD / Millions de requêtes

C'est la différence principale avec un PAAS

TEMPS DE DÉMARRAGE

Les fonctions sont hébergées dans des mini-conteneurs

Afin d'optimiser leurs coûts, les prestataires
"endorment" les containers qui ne sont pas
appelés souvent

Quand une fonction est appelée, il faut le temps
que le conteneur se réveille

C'est le "Cold start"



Peut impliquer des temps de réponse > 5
secondes pour une fonction, même simple

LES PRESTATAIRES

Deux exemples :



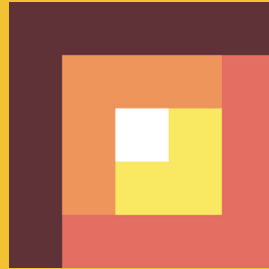
Ecosystème AWS complet

(Multi-langages, Web services,
Stockage de fichiers, événements,
BDD...)



Plus limité mais
plus simple en utilisation

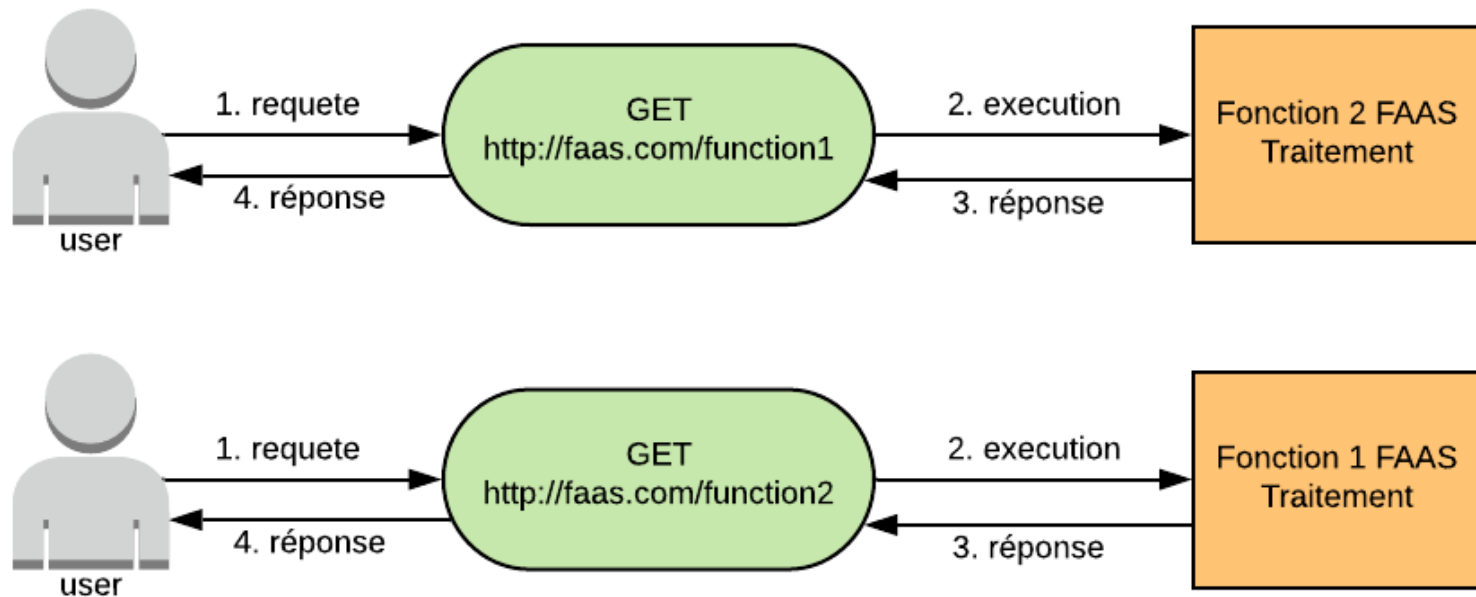
(Uniquement pour créer
des webservices en Javascript)



FAAS EN PRATIQUE

AVEC WEBTASK

Webtask permet de créer des fonctions reliés à un endpoint HTTP

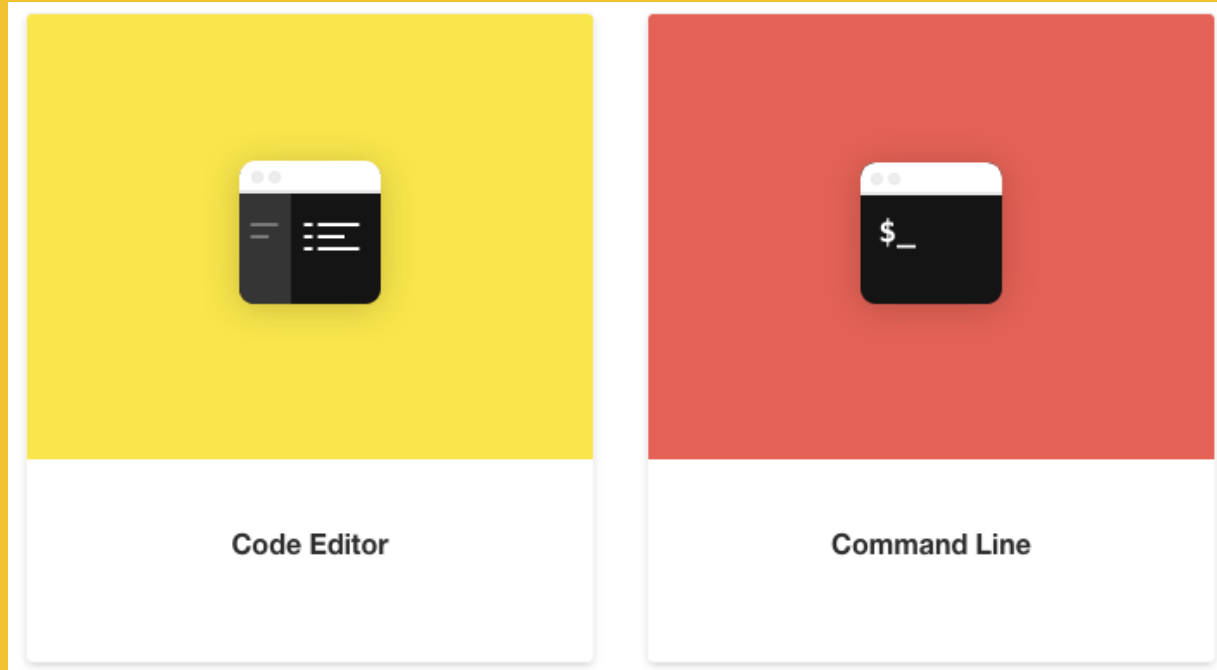


Webtask n'est compatible qu'avec un langage :
NodeJS



(Javascript Côté serveur)

Sur Webtask, on peut écrire le code de deux manières :



**Directement en ligne
dans son navigateur
avec l'éditeur de texte
Webtask**

ou

**Sur son poste local avec
son éditeur de texte
préféré, puis en uploadant
le code avec une
commande**

UNE FONCTION MINIMALE AVEC WEBTASK

```
1 module.exports = function(context, cb) {  
2     cb(null, 'Bonjour');  
3 }
```

Cette fonction retourne "Bonjour" quand on l'appelle

Exemple

RÉCUPÉRER LES PARAMÈTRES D'ENTRÉES

Les paramètres d'entrées sont dans la variable "context".

```
1 module.exports = function(context, cb) {  
2   cb(null, "Hello " + context.query.name);  
3 };
```

<https://wt-2ac738535a8e33dfc96d78da0842507e-0.sandbox.auth0-extend.com/hello-me?name=WIS>

Plus d'infos sur le contenu de context

STOCKER DE LA DONNÉE

Webtask intègre un système pour stocker des données et les réutiliser plus tard.

```
1 module.exports = function(ctx, cb) {  
2   ctx.storage.set("Bonjour");  
3   cb(null, "OK");  
4 }
```

Écrire

```
1 module.exports = function(ctx, cb) {  
2   ctx.storage.get(function (error, data) {  
3     cb(null, data);  
4   });  
5 }
```

Lire

TP 5

Un webservice pour créer
une TODO list avec Webtask

<http://bit.ly/36yj5jj>

Crédits

- <https://martinfowler.com/articles/serverless.html>