

platform.sh



Perdre sa tête

Ori Pekelman

CPO [@platformsh](#)

oripekelman partout github/linkedin/twitter

platform.sh



Perdre sa tête

Ori Pekelman
CPO @platformsh

oripekelman partout github/linkedin/twitter

*Vous pouvez trouver quelques unes de mes présentations API sur
<http://pekelman.com/presentations/>*



Un petit retour d'expérience

4 ans avec du Drupal Headless



L'année est 2014

On commence tout juste à construire un produit qui aura pas mal de succès.

- L'orchestrateur en Python
- Le flux des données en Go

La gestion des comptes et le billing

- Drupal 7.0
- Avec Drupal Commerce

Le UI

- Angular (1.2)



L'architecture des APIs platform.sh

Accounts
Drupal Commerce

Plein de rôles

- Gestion d'utilisateurs
- Authentification
- Gestion des clés SSH
- Gestion des entités (projets, leur configuration)
- Abonnements et paiements récurrents
- API Hypermedia (JSON/Hal)

L'architecture des APIs platform.sh

Accounts
Drupal Commerce

API First

- Mais c'est super agréable d'avoir sous la main une interface d'admin riche avec énormément de fonctionnalités clés-en-main.
- On est trois ingénieurs ... et on construit un *** de cloud

L'architecture des APIs platform.sh

Accounts
Drupal Commerce

Region DE		
Coordinateurs		
Grille Applicative		
Grille de Stockage		

Region EU		
Coordinateurs		
Grille Applicative		
Grille de Stockage		

Region US		
Coordinateurs		
Grille Applicative		
Grille de Stockage		

Region CA		
Coordinateurs		
Grille Applicative		
Grille de Stockage		

Region AU		
Coordinateurs		
Grille Applicative		
Grille de Stockage		

L'architecture des API platform.sh

Accounts
Drupal Commerce

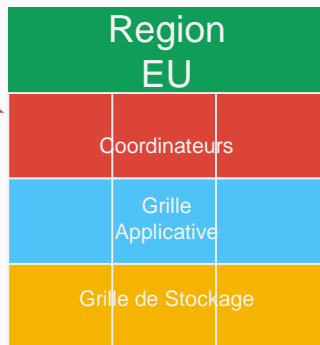


Interface Admin
accounts.platform.sh

Accounts REST API

Partagé.
Parce que: authentication.

UI
eu.platform.sh



Rien de
partagé.

Chaque
région, son
API

php	mysql	git
influx	nodejs	pgsql
php	git	mysql



GIT API

Projet REST API



Rien de
partagé.

Chaque
projet, son
API

L'architecture de platform.sh

UI
eu.platform.sh



Accounts REST API
accounts.platform.sh/api

Projet REST API
eu.platform.sh/projects

Une API

Des milliers, dizaines de milliers d'APIs

Evolution de l'architecture de platform.sh



Déployé sur chacune de nos régions en geodns balancing

Evolution de l'architecture de platform.sh



Déployé sur chacune de nos régions en geodns balancing

- Autant de résilience qu'une solution ou rien n'est partagé.
- Avec des meilleures perf car tout est géo-distribué
- Et à cause de HTTP/2. Multiplexage!

Evolution de l'architecture de platform.sh

- “API first” nous a permis de remodeler l'infra
- De découpler notre système de gestion de comptes et migrer des parties en Drupal 8.0 sans devoir tout refaire, partie par partie.
- D'utiliser des frameworks différents chacun pour ce qu'il fait au mieux.
- Tout en gardant nos marques blanches et nos partenaires sur des bases solides qui évoluent sans casse.

Accounts REST API

accounts.platform.sh/api

Projet REST API

eu.platform.sh/projects

Image Capture API

images.platform.sh/

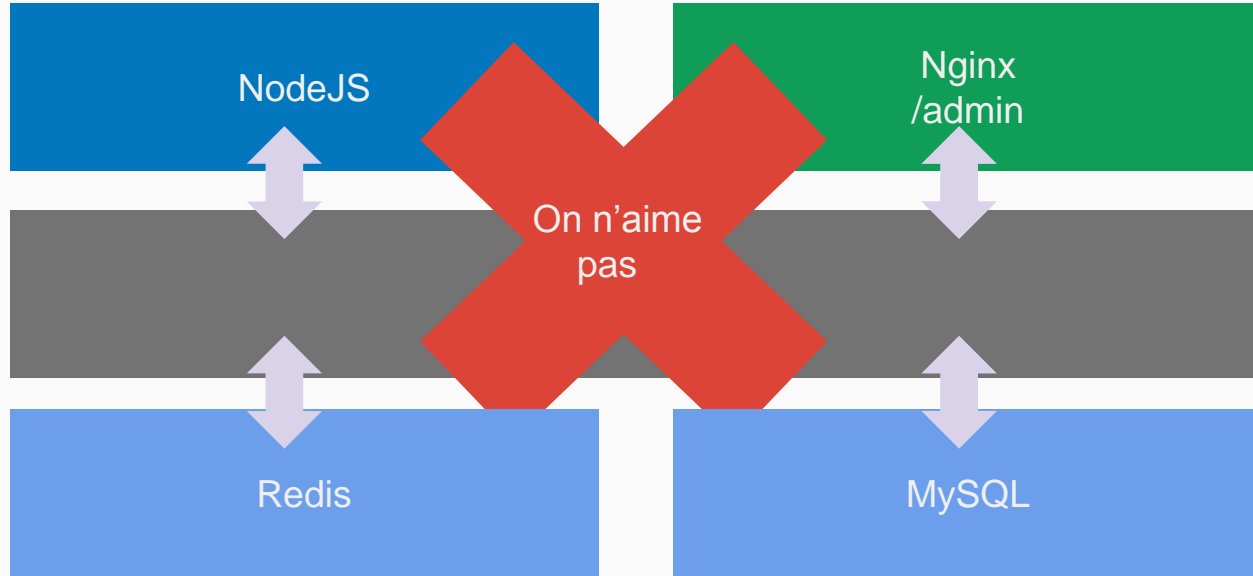
SSH Keys API

ssh.platform.sh/

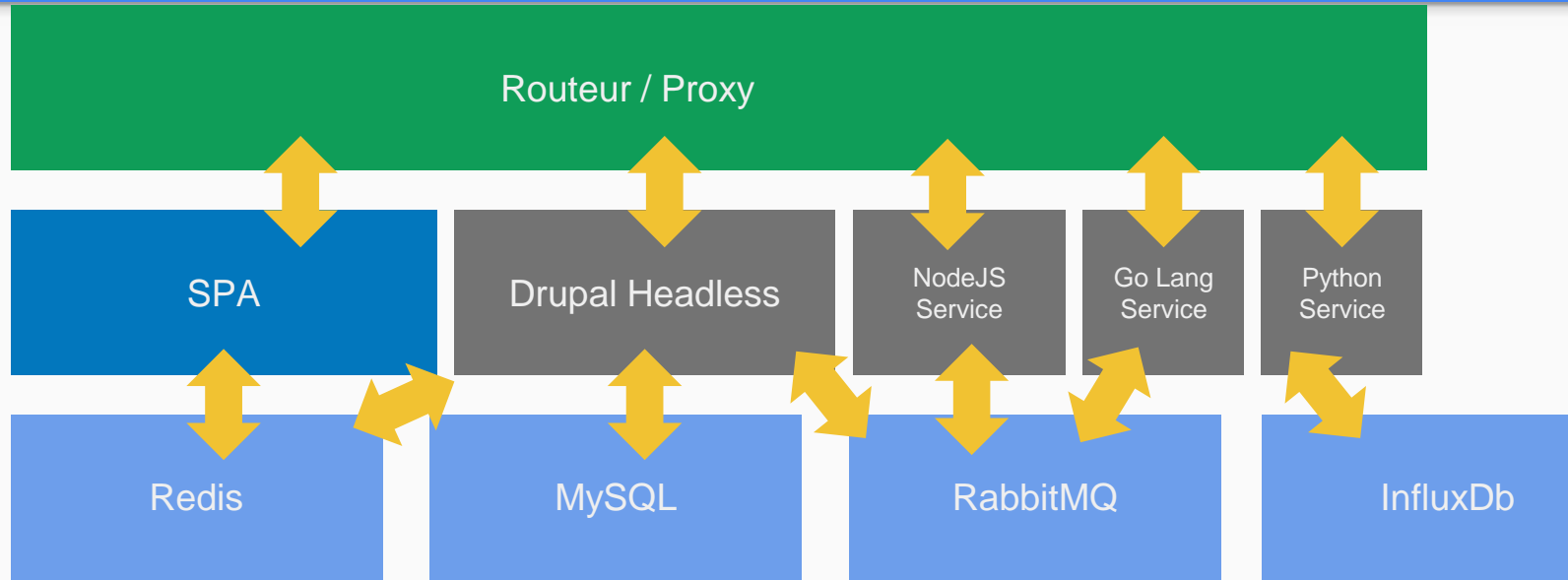
.... API

...api.platform.sh/

Architecture canonique



Architecture canonique



Les problèmes

- Il faut penser à *TOUS* les cas de figure cross-origin. CORS!
- Il faut penser à *TOUS* les cas de figure SSL. On s'est ajouté des problèmes avec le PSL. **Let's Encrypt a beaucoup aidé.**
- Attention à la latence à la queueleuleu et à appeler Drupal en boucle.
- On a commencé en Drupal 7.0 donc REST en modules contribués. On a utilisé *Services* et *Restful*. On a dû ajouter le support Swagger et Hal.
- Mais attention même 8.5.x on a ... des problèmes (content-type negotiation)... et CORS..
<https://www.drupal.org/node/2715637>

Les Astuces

- Let's Encrypt aide beaucoup
- Oauth aide beaucoup
- Cookie-Less authentication (local-storage + bearer token) pour les SPA
- Pensez à Drupal comme à un service. Parmi d'autres. N'hésitez pas à en multiplier. On y prend goût.
- Mais choisissez une forme unique. Quitte à bosser. JSON+HAL c'est pas mal.
- Pensez à ce que vous donne HTTP/2. Gratuitement. Mais il est temps aussi d'être au courant de gRPC. Parfois, parfois, GraphQL.
- Si vous ne connaissez pas API-Platform. Il est temps. J'aurais adoré l'avoir.

- Vaut mieux avoir un hébergeur qui a du support natif pour les micro-services.

- Mais nous, on a platform.sh :)

Demo ?

platform.sh



On va voir si j'ai parlé super lentement ou non. D'habitude je suis assez volubile.

Questions?

platform.sh



Je suis Ori Pekelman
CPO @platformsh

Retrouvez moi sur twitter/linked-in/github :
oripekelman

ori@platform.sh