

Shared Agenda

Votre agenda dans le Cloud!

Introduction

Problématique

Introduction

Problématique

- Stockage distant

Introduction

Problématique

- Stockage distant
- Accéder partout

Introduction

Problématique

- Stockage distant
- Accéder partout
- Collaborer

Introduction

Solution

Introduction

Solution

- Un Agenda centralisé

Introduction

Solution

- Un Agenda centralisé
- Accessible partout avec une connexion internet

Introduction

Solution

- Un Agenda centralisé
- Accessible partout avec une connexion internet
- Sécurisé

Introduction

Architecture Serveur

¹All SVGs are under CC0 from SVGRepo unless stated otherwise

Introduction

Architecture Serveur



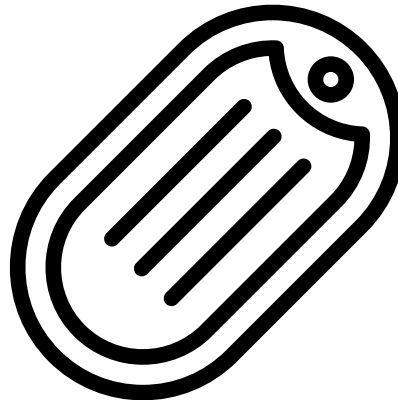
¹All SVGs are under CC0 from SVGRepo unless stated otherwise

Introduction

Architecture Serveur



1



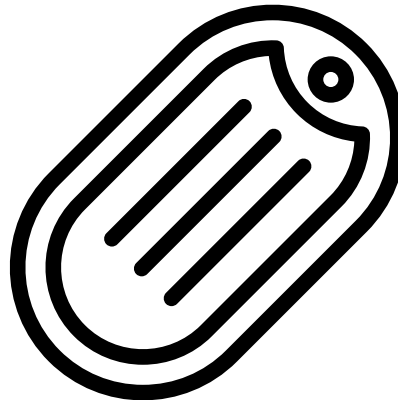
¹All SVGs are under CC0 from SVGRepo unless stated otherwise

Introduction

Architecture Serveur



1



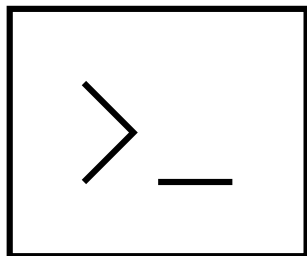
¹All SVGs are under CC0 from SVGRepo unless stated otherwise

Introduction

Architecture Client

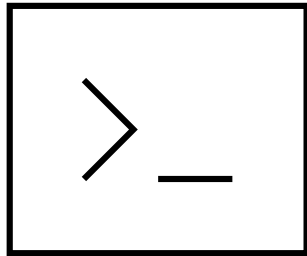
Introduction

Architecture Client

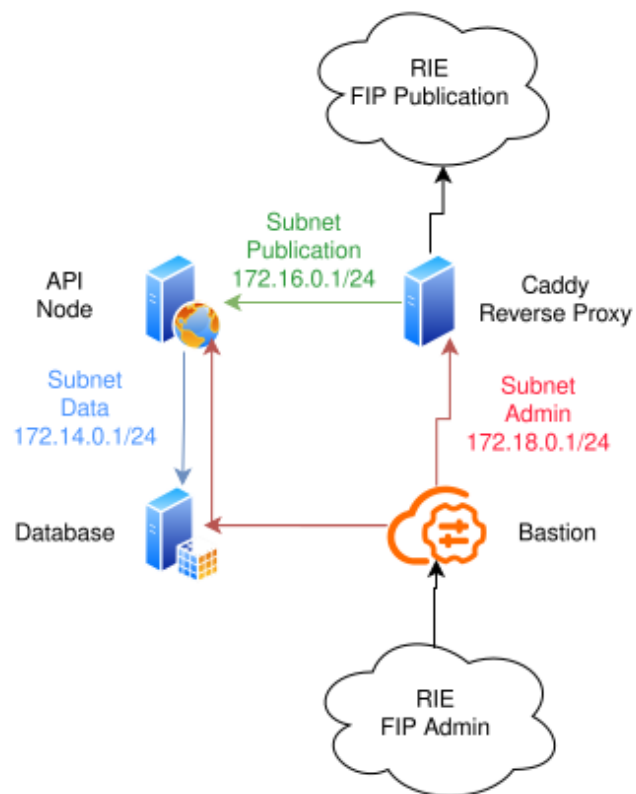


Introduction

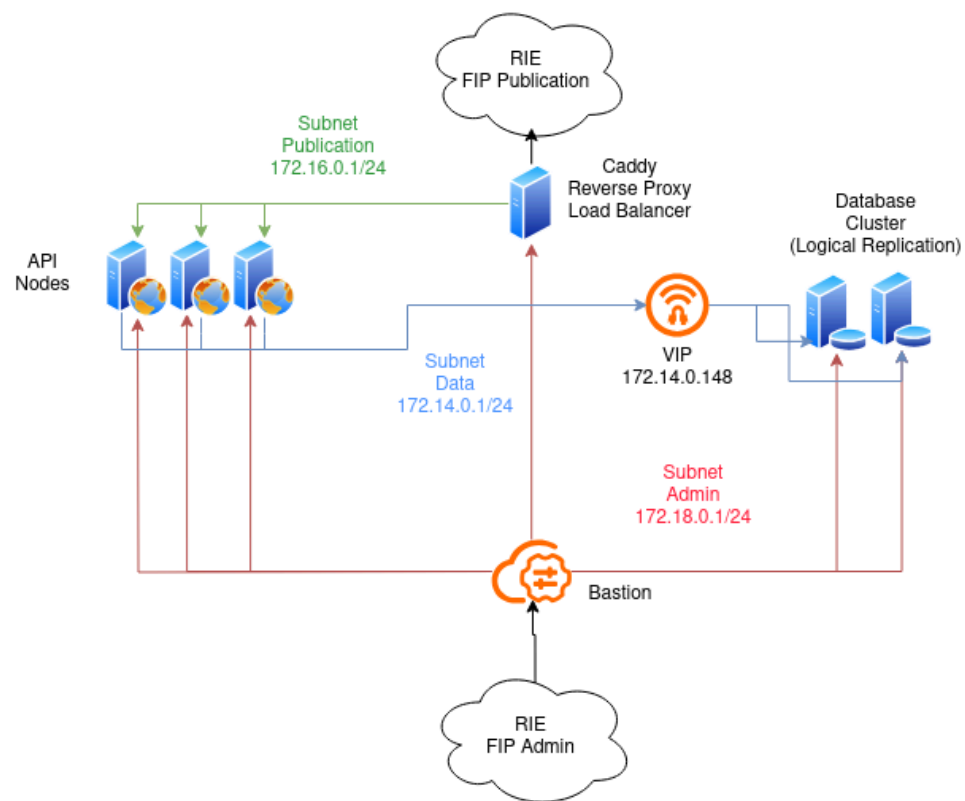
Architecture Client



Infrastructure (V1)



Infrastructure (V2)



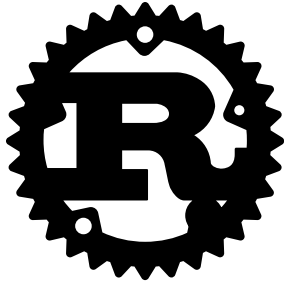
Serveur

Choix Techniques

¹Usage authorised under special licence see <https://uxwing.com/license/>

Serveur

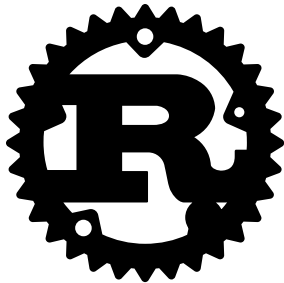
Choix Techniques



¹Usage authorised under special licence see <https://uxwing.com/license/>

Serveur

Choix Techniques

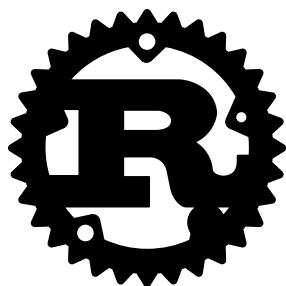


¹

¹Usage authorised under special licence see <https://uxwing.com/license/>

Serveur

Choix Techniques



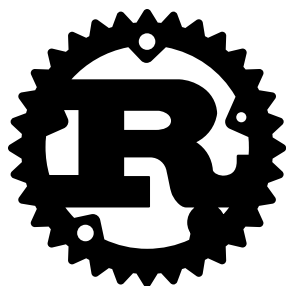
¹



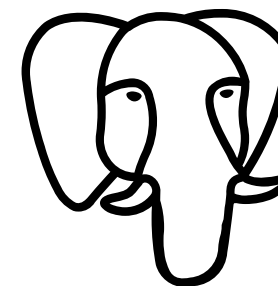
¹Usage authorised under special licence see <https://uxwing.com/license/>

Serveur

Choix Techniques



¹



¹Usage authorised under special licence see <https://uxwing.com/license/>

Serveur

Implémentation

Serveur

Implémentation

- Transactions BDD

Serveur

Implémentation

- Transactions BDD
- API Asynchrone

Serveur

Implémentation

- Transactions BDD
- API Asynchrone
- Codes de retour (ex: 200 OK, 401 Token Expired)

Serveur

Implémentation

- Transactions BDD
- API Asynchrone
- Codes de retour (ex: 200 OK, 401 Token Expired)
- TTL des Tokens: 24h

Serveur

- Fonctions génériques

```
1 pub async fn query<T: QueriedData>(self, sql: &str, args: &[(dyn ToSql + Sync)])  
  -> Vec<T> {  
2     let mut res: Vec<T> = vec![];  
3     match self.connection.query(sql, args).await {  
4         Ok(rows) => {  
5             for row in rows {  
6                 /*[...]*/  
7                 res.push(T::create_from_row(&row))
```



Client (CLI/REPL)

Implémentation

Client (CLI/REPL)

Implémentation

- Rust

Client (CLI/REPL)

Implémentation

- Rust
- Appel API

Client (CLI/REPL)

- Programmation par contrat

```
1 pub trait Answer {  
2     fn code(&self) -> i32;  
3     fn answer(&self) -> String;  
4     fn process(&mut self);  
5 }
```

 Rust

Client (CLI/REPL)

Démonstration