**SOMMAIRE**

[INTRODUCTION 2](#_Toc134973426)

[I. ARCHITECTURE DU PROJET 2](#_Toc134973427)

[I.1. Architecture globale 2](#_Toc134973428)

[I.2. Frontend 2](#_Toc134973429)

[I.3. Backend 2](#_Toc134973430)

[I.4. Base de données 2](#_Toc134973431)

[II. PRESENTATION DES TECHNOLOGIES UTILISEES 2](#_Toc134973432)

[III. MISE EN PLACE DE L’ENVIRONNEMENT DE TRAVAIL 2](#_Toc134973433)

[IV. REALISATION 3](#_Toc134973434)

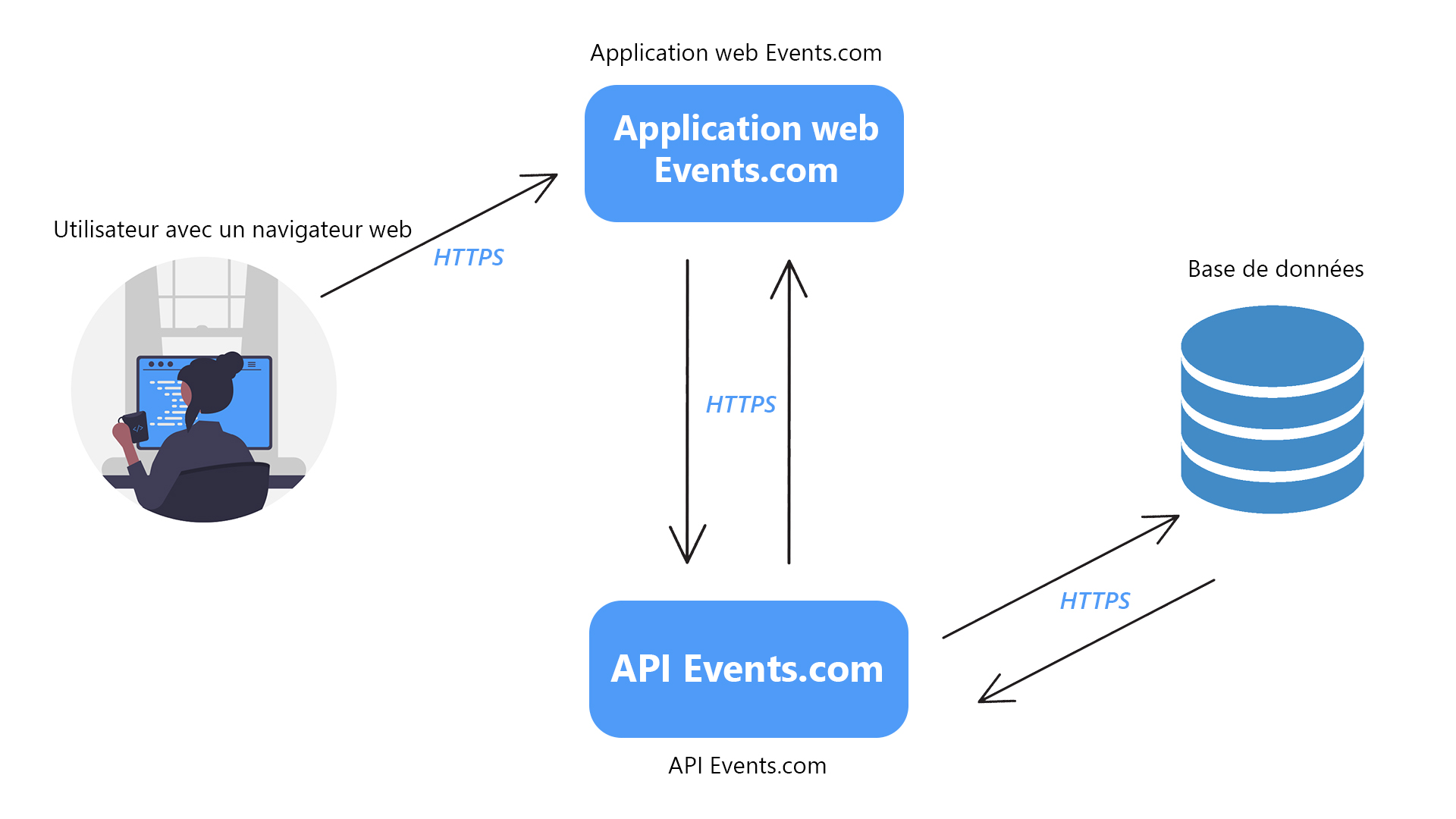
# INTRODUCTION

## ARCHITECTURE DU PROJET

### Architecture globale

L’architecture globale de notre projet de décline comme suit :

Figure 1 : Architecture globale du projet



Pour notre développement, nous utilisons l’architecture distribuée de type trois tiers. L’architecture trois tiers, également appelée architecture à trois niveaux ou à trois couches, est une architecture client-serveur dans laquelle coexistent et sont maintenus des modules indépendants permettant le rendu d'une interface utilisateur, les « process » logiques, fonctionnels et métiers ainsi que l'accès aux données. Notre choix s’est donc porté sur ce type d’architecture car nous voulons rendre notre projet très modulable et facilement maintenable. L’architecture se dessine donc en trois principales parties :

* **La présentation des données** : correspond à l’affichage, la restitution sur le poste de travail, le dialogue avec l’utilisateur. Elle est assurée par un poste client et un navigateur web.
* **Le traitement métier des données** : correspond à la mise en œuvre de l’ensemble des règles de gestion et de la logique applicative. Le traitement métier est géré par un serveur applicatif ;
* **L’accès aux données persistantes** : correspond aux données qui sont destinées à être conservées sur une durée voire de manière définitive. L’accès aux données se fait grâce au serveur de base de données MySQL.

### Frontend

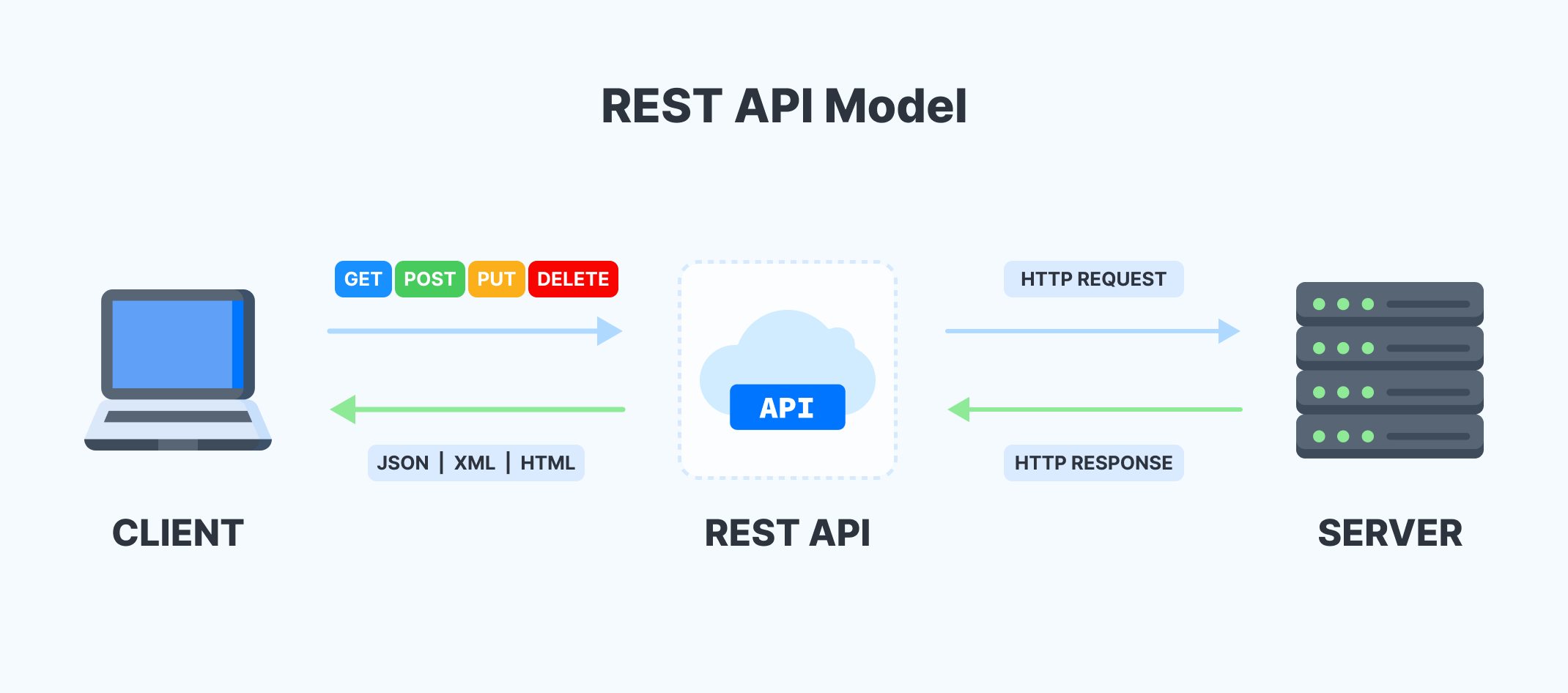
Figure 2 : Architecture du frontend



Le frontend de notre projet consiste en une plateforme web basée utilisant le Framework Laravel. L’architecture utilisée est l’architecture MVC (modèle-vue-contrôleur) qui est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.

### Backend

Figure 3 : Architecture du backend

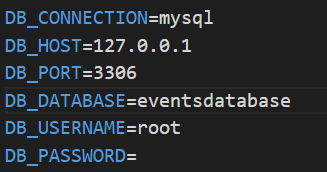


Pour notre backend, nous avons utilisé une architecture REST qui est un ensemble de contraintes permettant de créer des services web. Ces contraintes sont, entre autres, l’exécution de requêtes GET, POST, PUT et DELETE entre les différents composants du système.

### Base de données

Notre base de données est une base MySQL, avec 9 tables.

Connexion à la base de données :



Script de création de la base de données :



* Table Type
* Tale Evènement
* Table Ticket

## PRESENTATION DES TECHNOLOGIES UTILISEES

## MISE EN PLACE DE L’ENVIRONNEMENT DE TRAVAIL

## REALISATION