

---

# **PDP : ROUTAGE VERS TROU NOIR PILOTÉ À DISTANCE**

---

2 février 2018

Université de Bordeaux

BRISSET Rémi  
CHAUVEAU Pierre  
MASSAMIRI Michel  
PERUZZETTO Enzo

<https://services.emi.u-bordeaux.fr/projet/git/blackholepdp>

# Table des matières

1	Présentation du projet . . . . .	2
1.1	Intitulé . . . . .	2
1.1.1	Définitions . . . . .	2
1.2	Le routage vers trou noir [Sys05] . . . . .	3
2	Analyse de l'existant . . . . .	5
2.1	Erco.xyz [Did15] . . . . .	5
2.2	ExaBGPmon . . . . .	6
2.3	ExaBGP . . . . .	7
2.4	Meteor.JS [Met17] . . . . .	7
2.5	MongoDB [Mon] . . . . .	8
2.6	Dynamips . . . . .	8
2.7	NEMU . . . . .	9
3	Cahier des Charges . . . . .	10
3.1	Besoins fonctionnels . . . . .	10
3.1.1	Administrateur . . . . .	10
3.1.2	Administrateur et client . . . . .	11
3.2	Besoins non fonctionnels . . . . .	11
3.2.1	Organisation . . . . .	11
4	Schéma de structure . . . . .	12
4.1	Structure de la base de donnée . . . . .	12
5	Diagramme de cas d'utilisation . . . . .	14
6	Diagramme de séquence . . . . .	15
7	Maquette de l'interface web . . . . .	19
8	Diagramme de Gant . . . . .	21
	Bibliographie et Références . . . . .	22

# 1 Présentation du projet

## 1.1 Intitulé

L'objectif de ce projet est de développer un outil permettant à un administrateur réseau de définir à distance à partir d'un client Web, des routes<sup>1</sup> menant vers des trous noirs pour dévier des attaques réseaux. Ces routes seront envoyées à un serveur de route qui les diffusera auprès de tous les serveurs BGP du domaine. Le logiciel devra être implémenté en Javascript et du côté serveur il devra piloter le logiciel ExaBGP écrit en Python. L'application Web devra être de type RESTful et elle s'appuiera éventuellement sur un framework JS. Elle devra supporter le routage vers trou noir par la destination, par la source et par la communauté BGP.

### 1.1.1 Définitions

Les **routes** sont définies par : une adresse IP de destination, l'adresse IP du prochain routeur (NextHop) puis une liste des Systèmes Autonomes traversés (AS).

Une **Communauté** est un système Autonomes :

*"Un Système Autonomes est un ensemble cohérent de réseaux et de routeurs sous la responsabilité d'une autorité administrative<sup>2</sup>."*

Un **routeur BGP**<sup>3</sup> est un routeur qui utilise le protocole BGP (Border Gateway Protocol). BGP s'appuie sur le protocole TCP, sur le port 179. Lorsque deux routeurs BGP forment une connexion TCP entre eux. Ces routeurs sont des routeurs homologues (voisins).

BGP est un protocole de type " Path Vector ". Les routeurs s'échangent des informations du type :

Adresse IP du réseau de destination		Adresse IP du prochain routeur(next hop)		Liste des AS traversés pour atteindre le réseau
-------------------------------------	--	--	--	---

#### Annoncer une route :

Les routeurs qui forment une connexion TCP afin d'échanger les données de routage BGP, sont des paires('peers') ou voisins('neighbors'). Les BGP 'peers' partagent alors les informations de routage BGP(Table de routage).

Normalement, BGP 'peers' mettent à jour leurs table de routage en utilisant la technique **UPDATE Message**, cela garantit que tous les routeurs BGP aient la même version de la table de routage, et donc le numéro de version change pour tout changement effectué auprès de n'importe quel 'peer'.

---

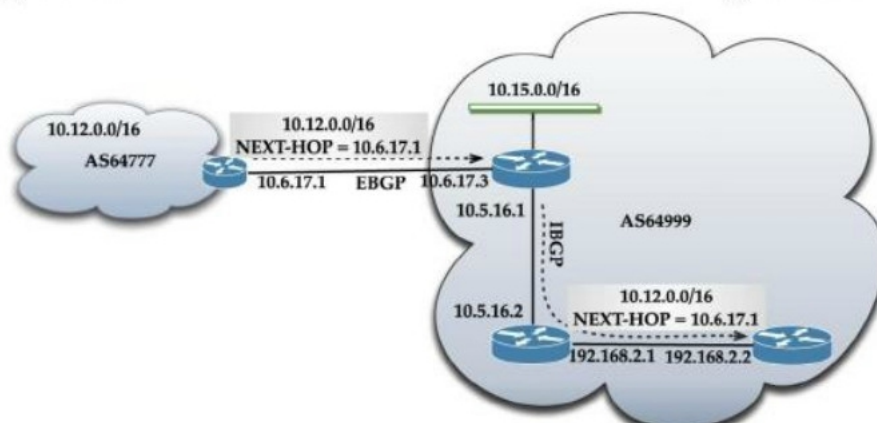
1. route BGP [BGP07]

2. Définition AS : <http://www.linux-france.org/prj/edu/archinet/systeme/ch65s02.html>

3. BGP <https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/26634-bgp-toc.html>

**Next-Hop[DM17]** : Ce terme désigne l'adresse IP du prochain routeur auquel le paquet IP devrait être transmis afin que ce paquet IP puisse atteindre sa destination.

Afin d'illustrer le fonctionnement de l'attribut Next Hop, nous détaillons ce dernier sur un exemple :



Dans ce scénario, si la destination qu'on souhaite atteindre est le réseau : "10.12.0.0/16", donc, tout trafic allant à cette adresse, doit s'orienter vers le routeur "10.6.17.1". Cela veut dire que le prochain routeur (Next-Hop) du "10.12.0.0/16" est le "10.6.17.1".

L'adresse 10.12.0.0/16 est annoncé par le EBGP (External BGP) 10.6.17.1, et donc l'information est passé de la communauté "AS64777" à la "AS64999". Ensuite le UPDATE Message est passé aux voisins par le iBGP(internal BGP) afin que tous les BGP peers gardent la même version de la table de routage.

Une API compatible **REST** (Representational State Transfer) est une interface de programmation d'application qui fait appel à des requêtes HTTP pour obtenir(GET), placer(PUT), publier(POST) et supprimer(DELETE).

## 1.2 Le routage vers trou noir [Sys05]

La déviation des routes vers un trou noir, aussi appelée "Remotely-Triggered Black Hole (RTBH)" en anglais, est une technique qui permet de suspendre un trafic provenant d'une source étant indésirable, avant que ce dernier puisse entrer dans un réseau protégé.

Le routage vers trou noir est essentiellement utilisé pour défendre ou proprement dit pour atténuer les attaques DDoS ("Distributed Denial Of Service", attaque dont le but est de surcharger le réseau attaqué afin de le ralentir ou de le mettre hors d'usage). Les trous noirs sont placés principalement dans un réseau pour lequel, on peut dévier et/ou suspendre le trafic lorsque le système détecte une attaque.

Pour que le système puisse dévier des router, il se base sur l'adresse IP de la destination ou bien sur l'adresse IP source. Il existe donc deux méthodes :

- **Destination-Based Remotely Triggered Black Hole Filtering** : On rend l'adresse IP de la destination inaccessible, en déviant toutes les routes allant à cet adresse vers le trou noir.
- **Source-Based Remotely Triggered Black Hole Filtering** : Dans ce scénario, si le trafic provenant d'une adresse IP est susceptible d'être une attaque, alors, tout trafic lié à cet adresse IP serait suspendu. Cela veut dire que selon l'adresse source IP, cette dernière ne peut pas avoir accès à sa destination. En outre, on fait tomber tous les chemins partants d'une adresse IP source précise.

## 2 Analyse de l'existant

Dans cette partie, nous présenterons les technologies étudiées et leurs qualités. //TOADD//

### 2.1 Erco.xyz [Did15]

Erco est un outil initialement développé à l'Université de Lorraine facilitant la configuration des routes réseau avec Exabgp en réécrivant une partie du fichier de configuration d'Exabgp. Erco fournit une API RESTful et une interface web utilisateur. L'interface web permet de facilement : annoncer un nouveau réseau ou IP, modifier ou supprimer une route, envoyer des commandes à Exabgp (reload, show routes show neighbors et version).

L'esthétique et les fonctionnalités de cette interface web sont utiles pour notre projet. En effet, les boutons "Relancer Exabgp" et "Exabgp fonctionne" sont exactement ce dont on a besoin. De plus notre section "lancer commande" ressemblera à celui de Erco mais avec plus de commande exécutable et un retour de message de Exabgp plus précis pour chaque commande. Nous prendrons aussi exemple sur Erco pour exécuter les commandes avec arguments sous forme de formulaire, et afficher les routes sous forme de tableaux, de liste. Pour essayer la démo en ligne : <https://erco.xyz/demo/>

#### Erco: Exabgp Routes Controller



API Relancer Exabgp

reload

Exabgp fonctionne

Nettoyer la sortie des commandes Lancer la commande

#### Annoncer un nouveau réseau ou une IP

Adresse IP ou réseau en notation CIDR

Next hop 198.51.100.42 (zoidberg.example.org)

Préférence locale

Communautés 1337:1984 (Planet express community)  
42:1984 (Nude Beach Planet)

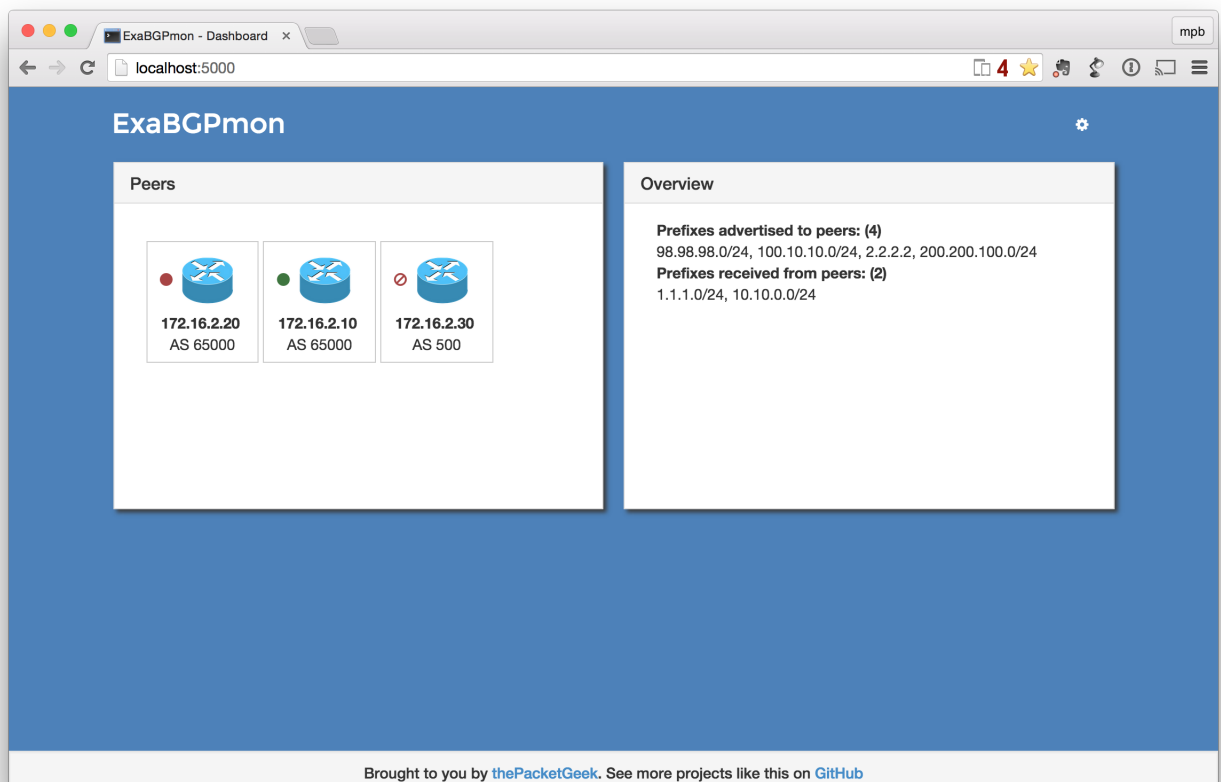
Ajouter

## Sous-réseaux annoncés ↻

#	Réseau	Next hop	Préférence locale	Communautés	Créé le	Modifié le	Actions
1	192.168.85.0/24	198.51.100.42 (zoidberg.example.org)		42:1984 (Nude Beach Planet)	lundi 21 septembre 2015 15:42	Ø	<a href="#">✎</a> <a href="#">✕</a>
2	192.16.45.2/32	203.0.113.42 (bender.example.org)		1337:1984 (Planet express community)	lundi 21 septembre 2015 16:18	Ø	<a href="#">✎</a> <a href="#">✕</a>
3	12.0.0.0/8	198.51.100.42 (zoidberg.example.org)		42:1984 (Nude Beach Planet)	mardi 22 septembre 2015 14:41	Ø	<a href="#">✎</a> <a href="#">✕</a>
4	12.12.12.12/30	203.0.113.42 (bender.example.org)		1337:1984 (Planet express community)	lundi 28 septembre 2015 17:01	jeudi 18 janvier 2018 21:21	<a href="#">✎</a> <a href="#">✕</a>
5	10.9.8.0/24	203.0.113.42 (bender.example.org)		1337:1984 (Planet express community)	jeudi 1 octobre 2015 18:50	Ø	<a href="#">✎</a> <a href="#">✕</a>
6	10.0.0.1/32	203.0.113.42 (bender.example.org)	100	1337:1984 (Planet express community)	jeudi 18 janvier 2018 16:12	Ø	<a href="#">✎</a> <a href="#">✕</a>
7	147.78.2.3/32	198.51.100.42 (zoidberg.example.org)	45	1337:1984 (Planet express community)	vendredi 19 janvier 2018 14:35	Ø	<a href="#">✎</a> <a href="#">✕</a>

## 2.2 ExaBGPmon

ExaBGPmon est une interface web sur le même principe que Erco.xyz, mais nous n'allons rien retenir de ce site car il est moins intuitif et ergonomique que Erco et propose les mêmes fonctionnalités.



## 2.3 ExaBGP

ExaBGP est un outil open source écrit en Python qui permet d'interagir avec les réseaux BGP. Le logiciel peut injecter des routes annoncés dans les réseaux, ainsi il permet d'annoncer des routes (configurer des routes). ExaBGP offre un API contenant plusieurs commandes afin de manipuler les routeurs BGP. Notre application Web est censé donc d'envoyer des requêtes HTTP au serveur ExaBGP en Format JSON. Ensuite, ExaBGP prend en entrée le fichier JSON envoyé par notre applciation Web contenant la configuration et les commandes à exécuter par ExaBGP.

Enfin, ExaBGP renvoie le message de retour en format JSON pour qu'on puisse le traiter dans l'application Web.

On peut aller voir la liste des commandes dans l'API d'ExaBGP :

<https://github.com/Exa-Networks/exabgp/wiki/Controlling-ExaBGP---interacting-from-the-API>

Et apprendre à l'utiliser avec les tutos suivants :

<https://thepacketgeek.com/series/influence-routing-decisions-with-python-and-exabgp/>

## 2.4 Meteor.JS [Met17]



Meteor.JS est un framework open-source javascript, Node.JS qui permet l'élaboration d'une application web de type RESTful. Elle permet de développer le client et le serveur de l'application web avec le même langage.

Notre client voulait une interface web développé en JavaScript de type Restful. Pour un déploiement et implémentation plus rapide de cette interface nous avons décidé d'utiliser le framework open-source Meteor.js qui réponds à ses besoins. De plus avec la possibilité d'utiliser des packages (extensions) de meteor.js nous pouvons implémenter certaines choses plus rapidement.

Les packages utilisés sont :

- twbs :bootstrap
- ian :accounts-ui-bootstrap-3
- iron :router

Voici quelques liens pour pouvoir s'informer plus sur Meteor.js :

<http://meteortips.com/first-meteor-tutorial/>



<http://meteortips.com/second-meteor-tutorial/>  
<http://www.meteor-tuts.com/>  
<https://www.meteor.com/tutorials/react/creating-an-app>  
Documentation : <http://docs.meteor.com/#/full/>

## 2.5 MongoDB [Mon]

MongoDB est une base de données NoSQL open source orientée documents qui fournit de hautes performances, une haute disponibilité, et mise à l'échelle automatique. L'API MongoDB est compatible avec Meteor. Les données manipulées sont structurées au format BSON (pour "Binary JSON" lui même acronyme de "JavaScript Object Notation"). Un tel document est une structure de donnée composée de paires champs/valeur. Les valeurs de ces champs peuvent être d'autres documents, des tableaux, ou des tableaux de documents.

Lien de la documentation avec tuto : <https://docs.mongodb.com/>

Lien d'un tuto de la documentation de Meteor détaillant la gestion des données avec MongoDB : <https://docs.meteor.com/api/collections.html>

## 2.6 Dynamips

Dynamips est un émulateur de routeurs Cisco écrit par Christophe Fillot. Dynamips peut émuler plusieurs versions de routeur Cisco, ainsi il peut lancer différentes images OS Cisco. Dans le contexte de ce projet, on utilise l'image '7200'.

Afin d'installer Dynamips, on aura besoin d'installer certaines bibliothèques : "libpcap", ou "winpcap" en fonction du système. ces bibliothèques sont utilisées pour fournir les fonctionnalités de "bridging router interfaces" utilisés par les cartes réseaux physiques.

Ensuite, il suffit de télécharger Dynamips depuis GitHub :

<https://github.com/GNS3/dynamips>

Afin d'optimiser et de configurer Dynamips, on peut également utiliser Dynagen. Dynagen est un outil écrit en python permettant de paramétrer le fonctionnement de Dynamips en se basant sur des fichiers textes. De plus, Dynagen simplifie le lancement et la configuration de réseaux virtuels. Cependant, nous n'envisageons pas de l'utiliser dans un premier temps. En effet, il ne nous sera utile que dans un but d'optimisation des performances.

## 2.7 NEMU

NEmu(Network Emulator for Mobile Universes) est un environnement de réseaux virtuels distribués. NEmu gère un parc de machines virtuelle QEMU(un émulateur de matériel<sup>4</sup>) hébergé sur un ou plusieurs hôtes physiques afin de construire une topologie virtuelle.

Nemu fournit un interpréteur python afin de créer facilement des réseaux virtuels distribués. En ce qui nous concerne, on va utiliser cet outil pour pouvoir créer et virtualiser des topologies virtuelles. Alors pour installer Nemu sur linux, on suit les instructions décrites dans ce lien : <http://nemu.valab.net/index.php?static27/tuto-setup-nemu-linux>.

Concernant la documentation de Nemu, on peut également la trouver dans le lien suivant :

<http://nemu.valab.net/index.php?static2/doc>

---

4. site officiel : <https://www.qemu.org/index.html>

## 3 Cahier des Charges

Après avoir analysé les outils existants et les besoins du client, on s'est rendu compte que le client aura besoin d'une application Web de type RESTful pour pouvoir interagir avec ExaBGP.

Par conséquent, notre application Web s'appuiera sur le framework Meteor JS.

### 3.1 Besoins fonctionnels

#### 3.1.1 Administrateur

1. S'identifier à l'aide d'un identifiant et un mot de passe
  - les informations de connexion doivent être stockées dans la base de donnée
2. Vérifier le bon fonctionnement ExaBGP :
  - Envoyer un message d'avertissement(popup)quand ExaBGP est en panne ou ne tourne pas.
3. Annoncer une route :
  - Envoyer une requête HTTP composé du fichier JSON au serveur ExaBGP, pour exécuter la commande 'announce route'.
  - Annoncer un réseau ou une adresse IP à tous les BGP 'peers' afin qu'ils prennent en compte la configuration de la route entrée par l'administrateur réseau.
  - Attribuer une communauté à une adresse IP ou à un réseau lors de l'ajout.
  - La route sera stockée dans la base de données.
  - En cas de succès, un message s'apparaît à l'utilisateur pour confirmer l'ajout. Dans le cas échéant, un message d'erreur est affiché à l'écran contenant le code d'erreur.
4. Pouvoir supprimer une route :
  - L'administrateur réseaux pourrait supprimer une route qui a été déjà annoncée.
  - La route sera donc supprimée de la base de données.
  - En cas de succès, un message s'apparaît à l'utilisateur pour confirmer la suppression. Dans le cas échéant, un message d'erreur est affiché à l'écran contenant le code d'erreur.
5. Dévier une route vers un trou noir :
  - La route sera déviée vers le trou noir selon son adresse IP source, sa destination ou bien la communauté à laquelle elle appartient.
  - Le Next-Hop de la route sera l'interface null0.
  - En cas de succès, un message s'apparaît à l'utilisateur pour confirmer que la route n'est plus accessible selon le préfixe définit(Source IP ou destination IP ou bien la communauté). Dans le cas échéant, un message d'erreur est affiché à l'écran contenant le code d'erreur.
6. Relancer ExaBGP en cas de panne :
  - Un bouton permettant d'envoyer une requête HTTP au serveur ExaBGP.
  - La requête consiste à exécuter la commande 'reload' de l'API de ExaBGP.
  - La commande 'reload' relance la configuration d'ExaBGP sur le serveur.

- En cas de succès, un message de confirmation s'apparaît pour l'utilisateur. Dans le cas échéant, un message d'erreur s'apparaît portant le code d'erreur.
7. Exécuter les différentes commandes de l'API de ExaBGP
    - L'application web devrait pouvoir exécuter la totalité des commandes de L'API ExaBGP.
    - Les commandes à exécuter seront envoyées au serveur ExaBGP en format fichier JSON.
    - En cas d'erreur, l'application web devrait récupérer le code d'erreur de l'exécution provenant du serveur ExaBGP.

### 3.1.2 Administrateur et client

1. Lister les routes :
  - Lorsque qu'une route est modifiée, ses informations sont mises à jour dans la base de donnée
  - L'utilisateur peut voir toutes les routes qui existent dans la base de données.
  - Le résultat sera découpé en plusieurs pages web pour faciliter la lisibilité.
2. Rechercher des routes selon leurs préfixes, (adresse IP, communauté, destination) :
  - Une page web dans l'application dédiée à effectuer la recherche des routes selon leurs préfixes.

## 3.2 Besoins non fonctionnels

- Certains packages de meteor.js
  - twbs :bootstrap
  - ian :accounts-ui-bootstrap-3
  - iron :router
- Synchronisation du serveur web avec ExaBGP
- Interface différente pour l'admin et l'utilisateur anonyme
- Ecriture des scripts en Python et Javascript (imposé par l'utilisation de ExaBGP et Meteor)

### 3.2.1 Organisation

En effet, pour qu'on puisse commencer la phase de développement, on a besoin de virtualiser la topologie complète de réseaux et donc d'installer l'environnement technique sur nos machines (Dynamips, Nemu, et ExaBGP).

Concernant l'ordre de priorité, nous avons étudié les besoins du client et on en avait déduit l'ordre suivant :

1. Installer l'environnement technique dans le but d'avoir une topologie virtuelle.
2. Réaliser les tâches qui concerne l'administrateur en partant de la tâche '1' jusqu'à la tâche '5'.
3. Réaliser les tâches qui concerne l'administrateur et le client dans l'ordre indiqué.
4. Enfin, réaliser la tâche '6' et '7' concernant l'administrateur.

## 4 Schéma de structure

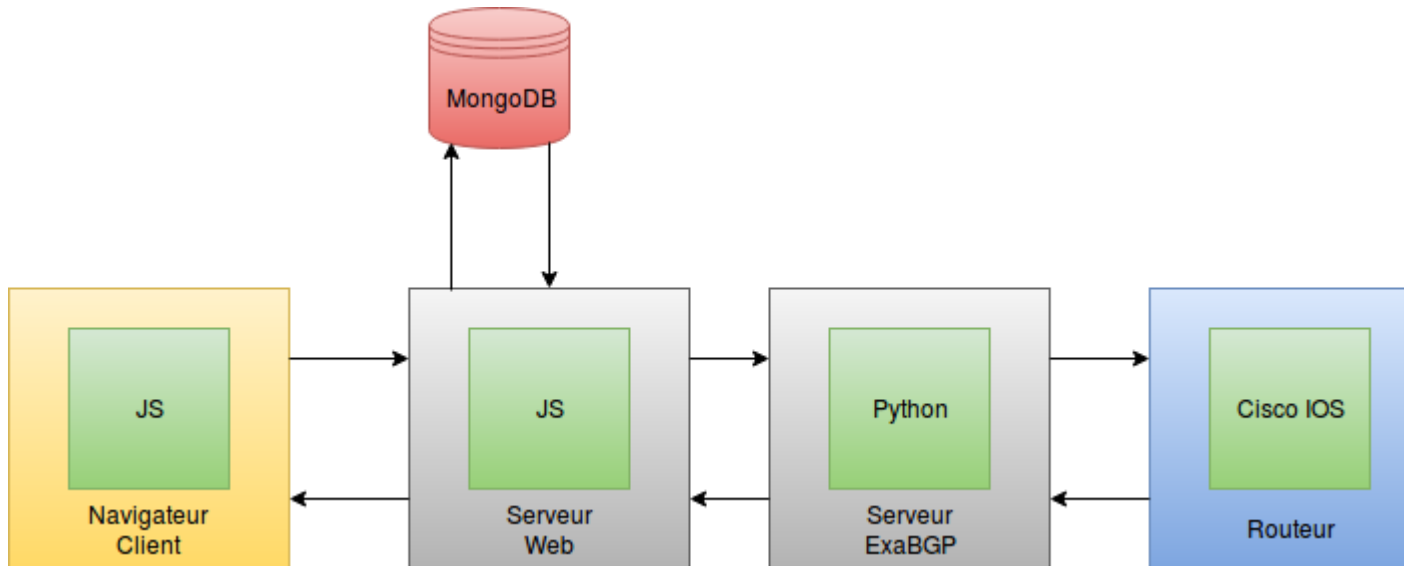


FIGURE 1: Schéma de structure

### 4.1 Structure de la base de donnée

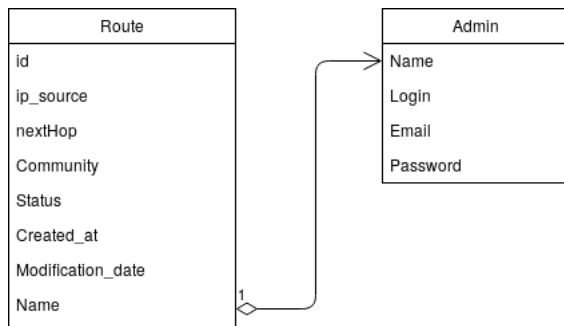


FIGURE 2: Uml de la base de donnée

La table "Route" permettra de stocker toutes les routes annoncées avec plusieurs champs spécifique :

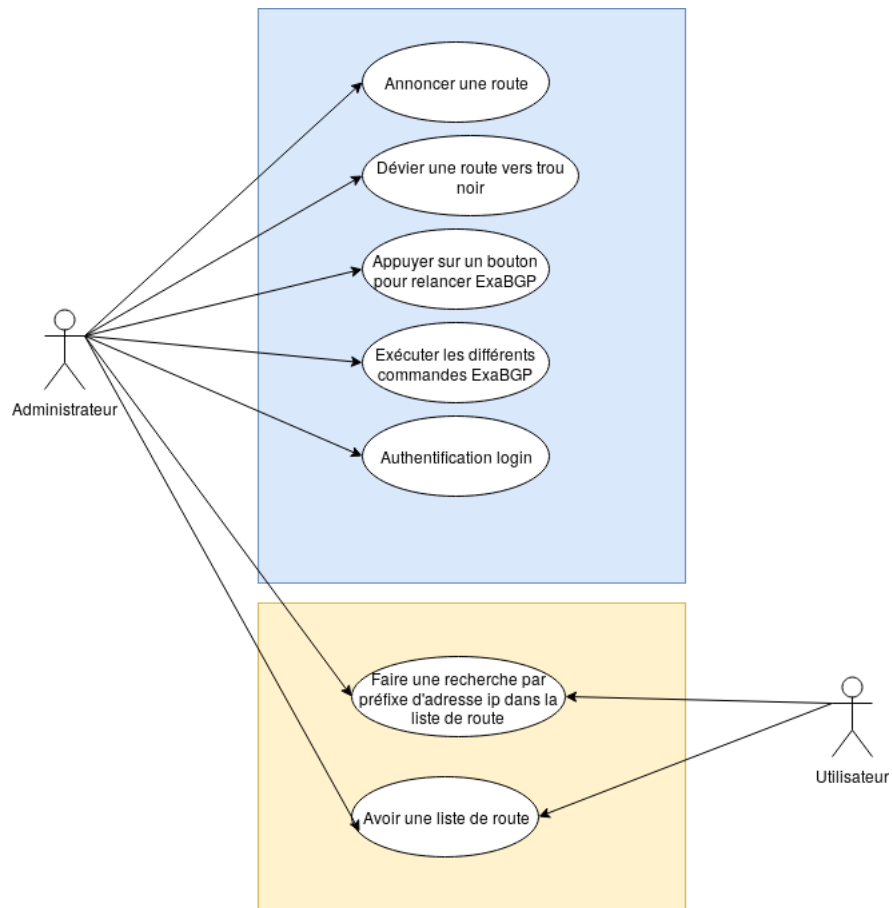
- id : Une clé unique représenter par un entier positif.
- Ip source : une adresse ip sous forme ipv4/ipv6, c'est-à-dire pour ipv4 sous la forme de quatre nombres entiers séparés par des points allant de 0.0.0.0 à 255.255.255.255. pour ipv6 8 groupes écrit en hexadécimale séparés par deux-points.
- NextHop : une adresse ip comme ip source.
- Community : Un tag décrivant la communauté du routeur une chaîne de caractère.

- Status : Une chaîne de caractère qui est soit "Activate" ou "Desactivate".
- Created at : la date de création de la route sous format <YYYY-mm-ddTHH :MM :ss> Y :année  
m :mois d :jour h :heure M :minute s :seconde.
- Modification date : la date de la dernière modification sur cette route, peut être vide.
- Name : le nom de l'administrateur qui la modifier ou ajouter représenter par une chaîne de caractère.

La table Admin permettra de stocker tout les administrateurs avec les champs suivants :

- Name : nom de l'administrateur représenter par une chaine de caractere.
- Login : pseudonyme de l'administrateur représenter par une chaine de caractère.
- Email : une adresse mail valide de l'administrateur.
- Password : un mot de passe.

## 5 Diagramme de cas d'utilisation



**FIGURE 3:** cas d'utilisation

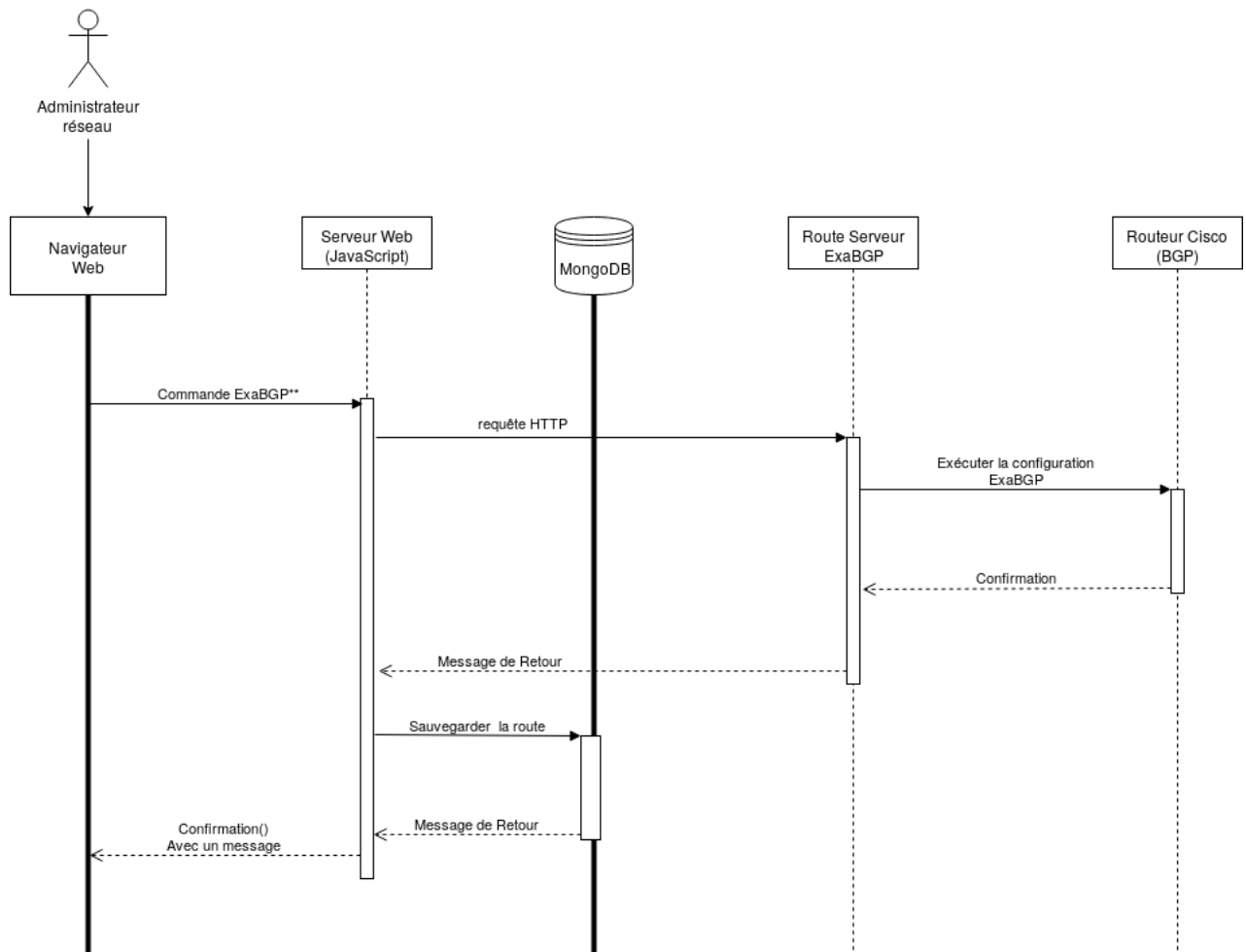
## 6 Diagramme de séquence

Nous représentons le scénario de la déviation d'une route vers un trou noir en utilisant le diagramme de séquence.

L'utilisateur de l'application web dans cet exemple est l'administrateur réseaux. On suppose que l'administrateur réseaux veut dévier une route par son adresse IP source.

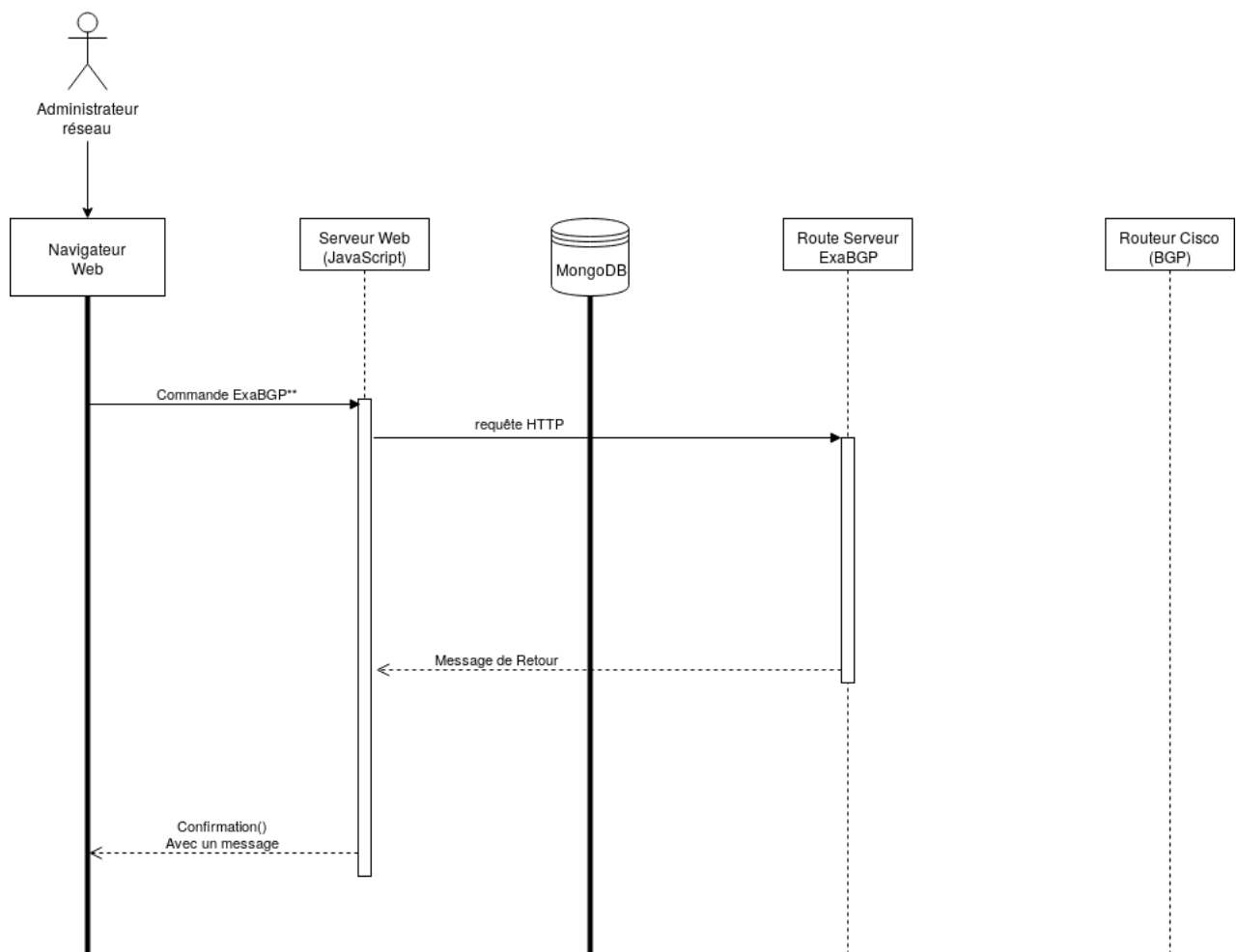
Lorsque l'opération est terminée, donc l'application web confirme que c'est fait, la route est ensuite serait stockée dans la base de données afin que le client puisse aller consulter les routes qui ont été déviées.





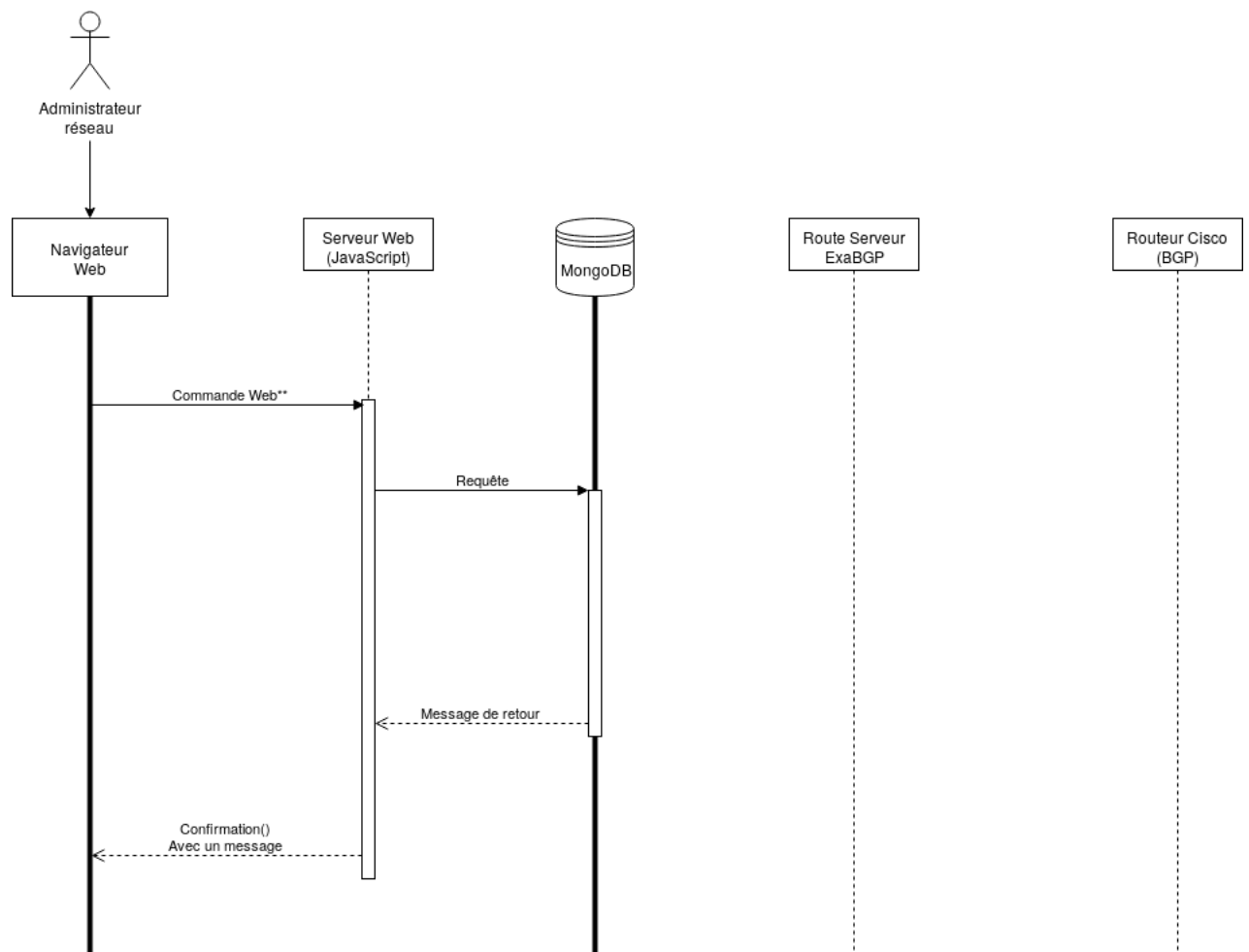
#### Commandes ExaBGP :

- show neighbor [optional neighbor ip] summary
- show neighbor [optional neighbor ip] extensive
- show neighbor [optional neighbor ip] configuration
- show ajd-rib in [extensive]
- show ajd-rib out [extensive]
- announce watchdog
- announce route
- announce eor
- announce flow
- announce operational
- announce vpls
- announce route-refresh



Commandes ExaBGP :

- shutdown
- reload
- restart
- version
- teardown



Commandes Web :

- Recherche route
- Connections

## 7 Maquette de l'interface web

The wireframe shows a web interface layout. At the top left is a header area labeled 'TITRE'. To its right is a blue button labeled 'Connexion'. Below the header, there are three buttons: 'API' (grey), 'Statut ExaBGP' (green), and 'Relancer ExaBGP' (white). On the left side, there is a pink button labeled 'Liste de commandes'. Below this button is a large white rectangular area labeled 'Formulaire d'envoi'. To the right of this area is a yellow button labeled 'Lancer la commande'. Below the 'Formulaire d'envoi' area is a white rectangular area labeled 'Message de retour'. At the bottom left, there is a small white button labeled 'Recherche'. Below this button is a large yellow rectangular area labeled 'Liste des routes'.

Le bouton connexion permet à l'administrateur de se connecter en entrant son login/email et son mot de passe.

Le bouton "API" sert à rediriger vers l'API de ExaBGP à l'url suivant :

<https://github.com/Exa-Networks/exabgp/wiki/Controlling-ExaBGP--interacting-from-the-API>

La fonctionnalité "Statut ExaBGP" sert à afficher si ExaBGP fonctionne ou non.

Le bouton liste de commande permet de sélectionner la commande d'ExaBGP voulue.

En fonction de la commande sélectionnée un formulaire spécifique est affiché. Cette partie sera affichée dans les cas où la commande a besoin de paramètre.

Le bouton "lancer commande" permet d'exécuter la commande sur ExaBGP ou sur la base de donnée.

Une zone dédiée au retour de message de la commande lancé. Le message devra être le plus détaillé

possible surtout pour les messages d'erreur d'une fonction qui s'est mal exécutée.

Une barre de recherche qui permettra à l'utilisateur ou à l'administrateur d'effectuer une recherche par préfixe d'adresse ip.

Une zone d'affichage qui listera la liste des routes. Initialement cette liste affichera les plus récentes routes stocké dans la base de donnée, mais si l'utilisateur effectue une recherche, c'est cette recherche qui sera affichée dans cette zone.

## 8 Diagramme de Gant

	semaine 5 : 5 fév / 9 fév	semaine 6 : 12 fév / 16 fé	semaine 7 : 19 fév / 23 fév	semaine 8 : 26 fév / 2 mars	semaine 9 : 27 fév / 3 mars	semaine 10 : 5 mars / mars	semaine 11 : 12 mars / mars	semaine 12 : 19 mars / mars	semaine 13 : 26 mars / mars	semaine 14 : 2 avr / 6 avr
Mise en place de machine virtuelle pour test	X	X	X							
connecter ExaBGP au serveur web			X							
état de ExaBGP				X						
Connexion administrateur				X						
Afficher le message de retour					X					
annoncer une route					X					
supprimer une adresse IP ou réseau					X					
Lister les routes					X					
dévier une route					X	X				
Recherche IP, Route par préfixe						X	X	X		
Relancer ExaBGP								X		
Exécuter les différentes commandes d'ExaBGP								X	X	X

# Bibliographie

- [BGP07] Le routage dynamique avec bgp. <http://www.linux-france.org/prj/edu/archinet/systeme/ch69.html>, 2007. [Accessed January].
- [Did15] Luc Didry. Erco : Exabgp routes controller. <https://erco.xyz>, 2015. [Accessed January].
- [DM17] Karthik Ramasamy Deep Medhi. *Network Routing Algorithms, Protocols, and Architectures*. Morgan Kaufmann, 2017.
- [Met17] Meteor.js. <https://www.meteor.com/>, 2017. [Accessed January].
- [Mon] MongoDB, documentation officielle. <https://docs.mongodb.com/manual/introduction/>. [Accessed January].
- [Sys05] Cisco Systems. Remotely triggered black hole filtering— destination based and source based. [https://www.cisco.com/c/dam/en\\_us/about/security/intelligence/blackhole.pdf](https://www.cisco.com/c/dam/en_us/about/security/intelligence/blackhole.pdf), 2005. [Accessed January].