**NDIAYE EL HADJ PATHE**

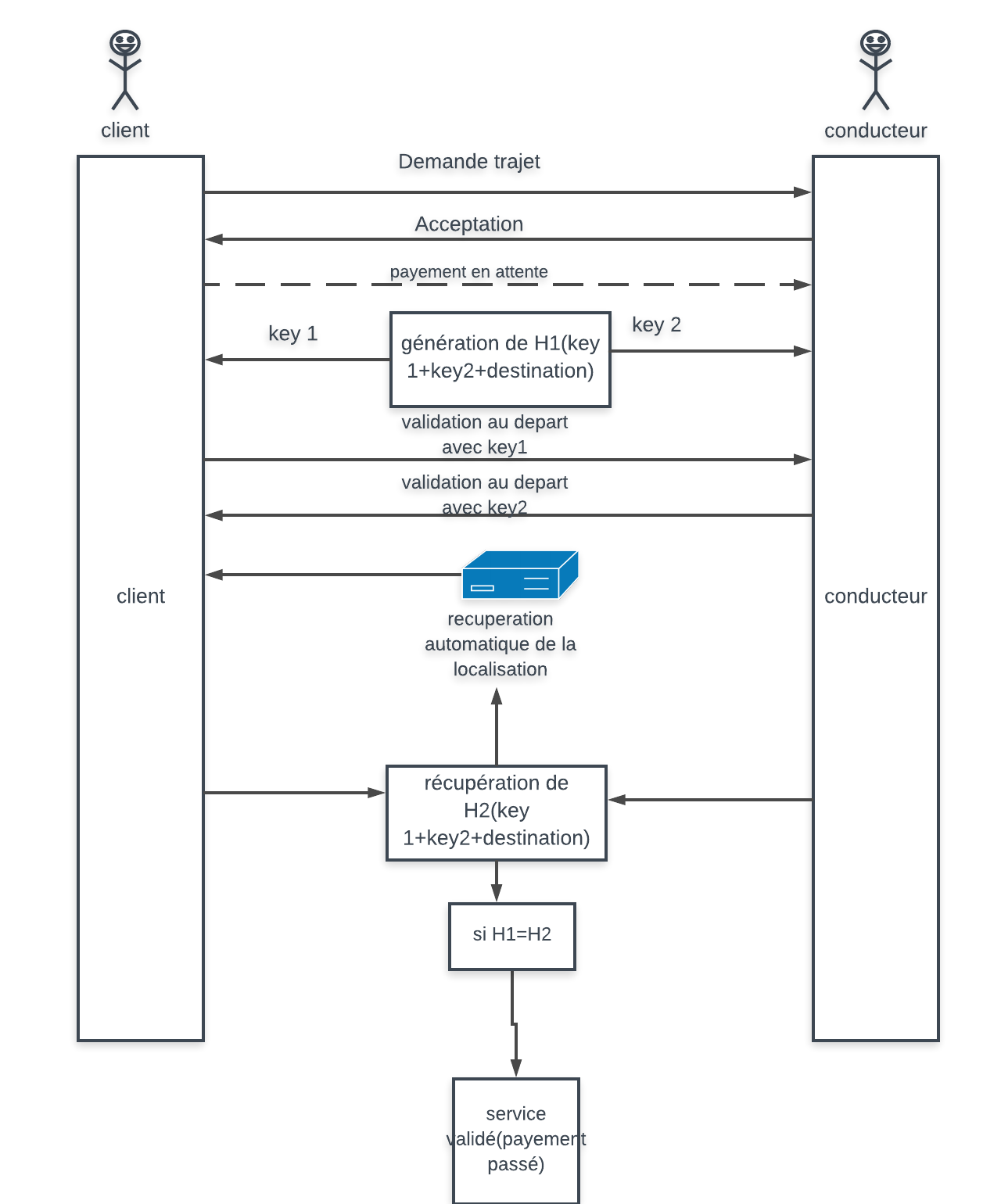
**ADIL EL HANKARI**

**NAJAT BAKKI**

**HABIL SANNI**

**SAAD EDDINE TAHALI**

**Validation d’un trajet effectué**



Après l’acceptation du trajet et le paiement qui sera mise en attente, le système va générer deux clés aléatoires, pour plus de sécurité on va générer un hash des deux clés plus la destination du client.

Pour la validation deux étapes seront effectuées:

* Validation du départ qui se fait avec l’échange des deux clés
* Validation à l’arrivé qui se fait avec la récupération et la comparaison et des deux hashs plus le hash de la localisation.

S’il y a une correspondance avec le hash de départ et le hash d’arrivée le service est validé et le paiement sera effectué.

**Implémentation**

L’idée est de génerer deux nombres aléatoires **nonce,** et attribuer un premier au client et un deuxieme au chauffeur, ces nonces vont être utiliser pour la validation du service dans un premier temps , après on va récuperer aussi la position géographique du client pour valider le paiement d’une façon definitive.

Pour génèrer ces clés on va utiliser des librairies comme :

[PHP Nonce Library](http://fullthrottledevelopment.com/php-nonce-library)

[OpenID Nonce Library](http://shupp.org/openiddoc/__filesource/fsource_OpenID__OpenIDNonce.php.html)

Pseudo-code utilisé (qui peut être developper) pour la génération des clés :

getNonce() {

$id = Identify Request

$nonce = hash('sha512', makeRandomString());

storeNonce($id, $nonce);

return $nonce to client;

}

verifyNonce($data, $cnonce, $hash) {

$id = Identify Request

$nonce = getNonce($id);

removeNonce($id, $nonce);

$testHash = hash('sha512',$nonce . $cnonce . $data);

return $testHash == $hash;

}

getNonce() : Cette fonction génere un nonce aléatoire unique en lui associant un identifiant.

VerifyNonce() : Permet de verifier que les 2 hashes sont identiques et supprime les nonces génerés pour le trajet en question.

Ensuite on calcule le Hash de la concatenation des deux nonces et l’adresse de destination entrée lors de la réservation, et on la compare avec le hash de ces derniers et la localisation du client a la fin du trajet.

On a choisit d’utiliser le hash pour la validation du trajet pour raison de :

-Respecter la confidentialité du client , cela sera possible puisque les 2 hashes sont calculés d’une façon complexes qui augmente la difficulté de récuperer l’adresse exacte.

-Même si on arrive à divulguer une fausse addresse cela sera toujours impossible de trouver le hash identique .

Ce qui permet de garder de respecter les conditions demandés pour la mise en place de l’application (confidentialité , système décentralisé).