Architecture du système

Love Mirroring

|  |  |
| --- | --- |
| Classification | non classifié |
| État | en cours d'élaboration |
| Nom du programme | Love Mirroring |
| N° du projet | 1 |
| Chef de projet | Tim Allemann |
| Version | 0.1 |
| Date | 07.04.2020 |
| Mandant | Stéphane Gerber |
| Auteur/Auteurs | Tim Allemann, Sébastien Berger, Hans Morsch |
| Distribution |  |

Suivi des modifications

| Version | Date | Modification | Auteur |
| --- | --- | --- | --- |
| 0.1 | 26.03.2020 | Création du document et identification des points importants | Sébastien Berger |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tableau 1: Contrôle des modifications

Description

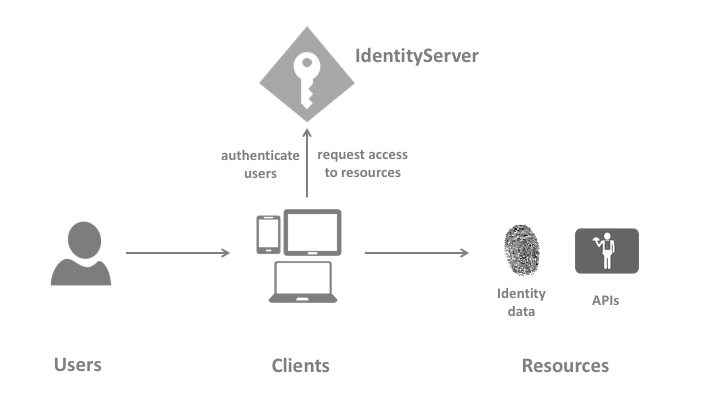
L’architecture du système divise ce dernier en sous-systèmes et en leurs composantes. Elle décrit la structure et les interfaces du système. Elle en donne un aperçu complet. Suivant le résultat et l’étendue du projet, elle contient plusieurs éléments et modèles d’architecture, par exemple le modèle des processus métier, le modèle des fonctions (p. ex. avec cas d’utilisation et récits d’utilisateurs), l’architecture ou le modèle des données, l’architecture de sécurité. Elle contient également la documentation informatique ou renvoie à la documentation du producteur. Elle se fonde sur les prescriptions des organes de prescription et de contrôle de gestion.

# Structure du système

## Aperçu du système

Voir documents annexes : LM\_UserCase.docx + diagramme

## Sous-systèmes et composantes



Le système est composé en 3 applications distinctes :

* Un composé de identity server 4
  + Gestion des accès, authentification, token
* Une pour les API
  + CRUD sur les données de LoveMirroring (liste des utilisateurs, etc.)
* Une pour le client (webApp)

## Architectures / Modèles

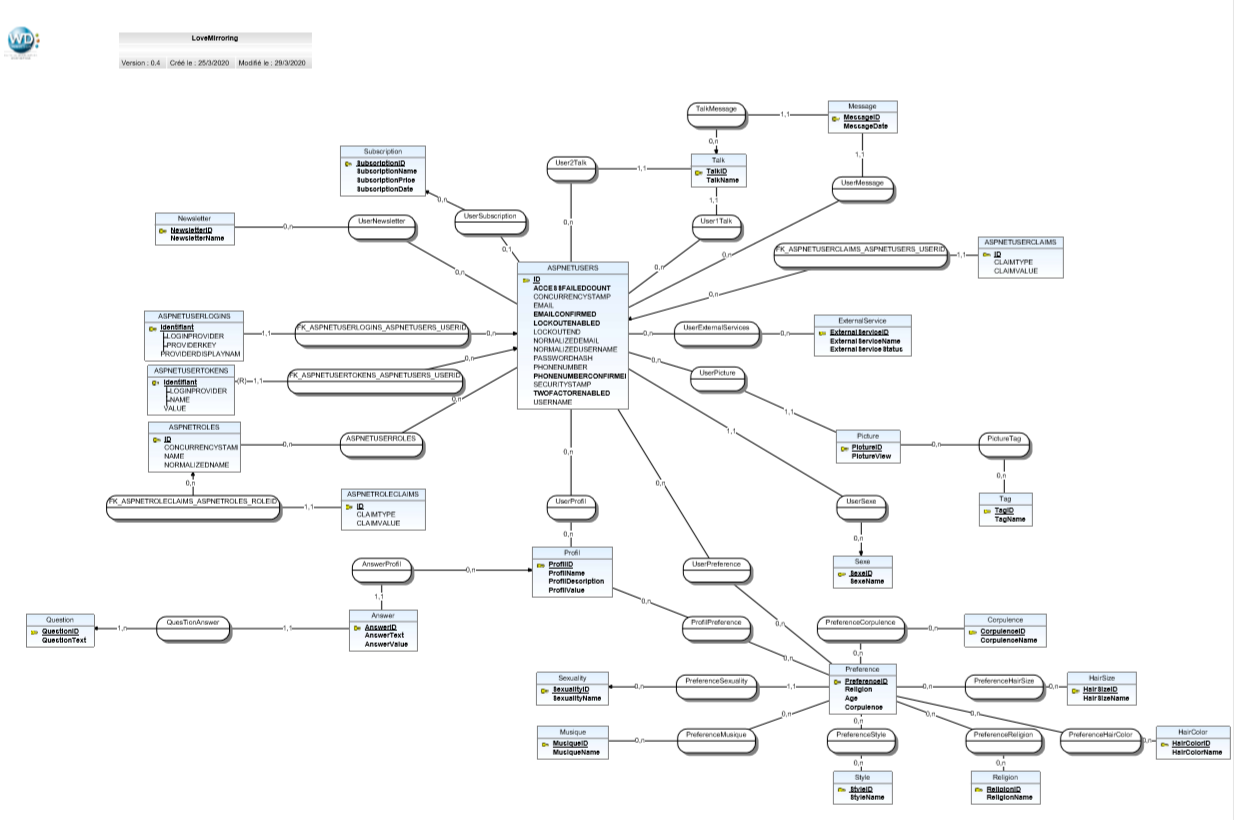
Référence aux études détaillées

Selon le résultat du projet : par exemple modèle des processus métier, modèle des fonctions (p. ex. avec cas d’utilisation / récits d’utilisateurs), architecture des données / modèle des données, etc.

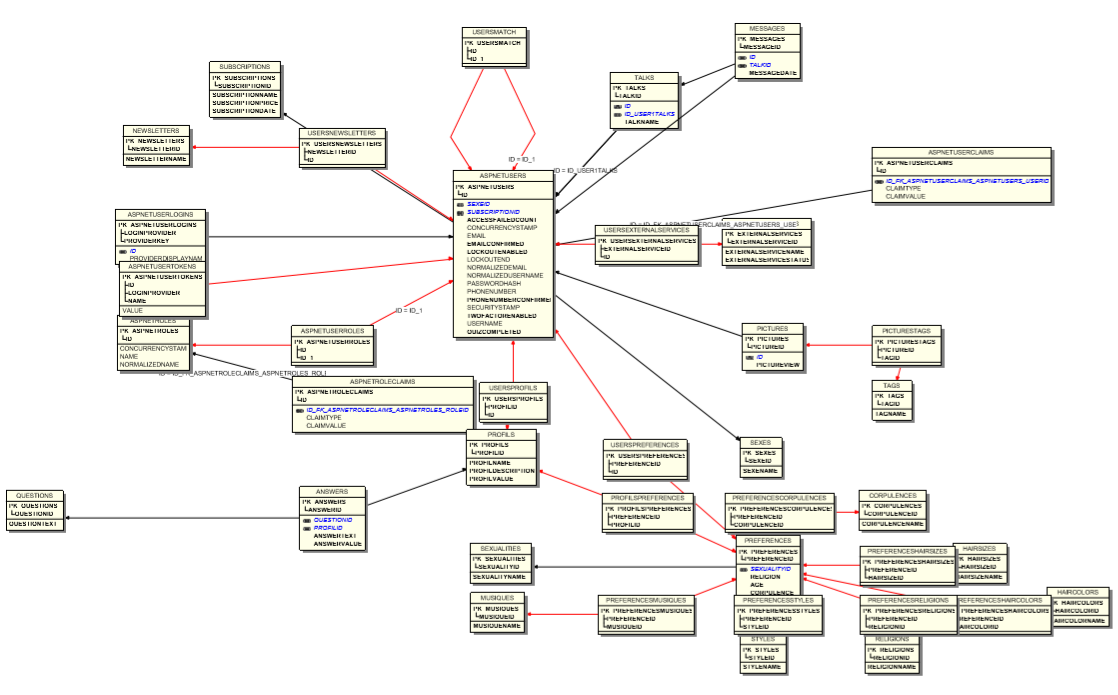
Architecture technique du système informatique

Produits utilisés, versions

|  |  |  |
| --- | --- | --- |
| **Produit** | **Version** | **Description** |
| Asp.net Core | 3.1 | Framework Web |
| Identity Server 4 | 3.1.0 | Service en tant qu’authentification permettant de gérer des api sécurisées |
| Sendgrid | 3.0 | Plateforme de communication permettant l’envoi de mail. |
| Twilio | 5.1.4 | Plateforme de communication permettant l’envoi de sms. |
| Stripe | 6.5.2 | Plateforme de communication permettant des paiements en ligne. |



Voir en annexe le MCD pour plus de détails.



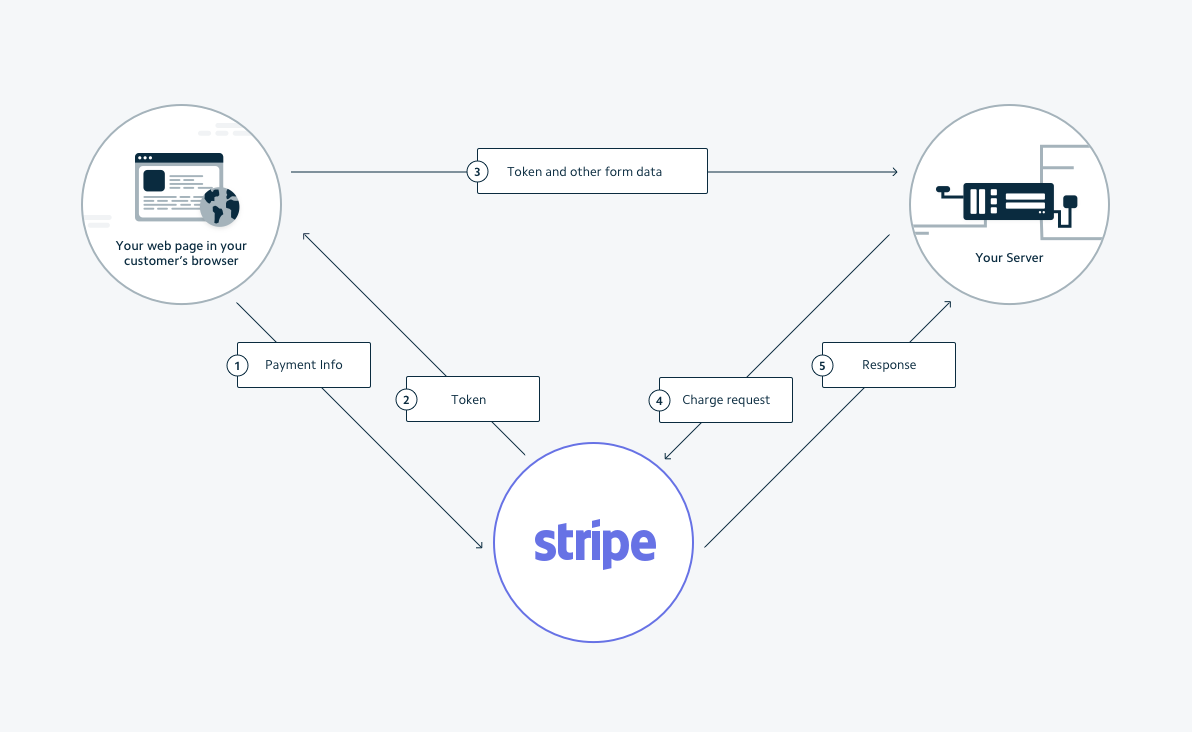
Voir en annexe le MLR pour plus de détails.

# Interfaces et délimitation

## Interfaces avec les systèmes environnants

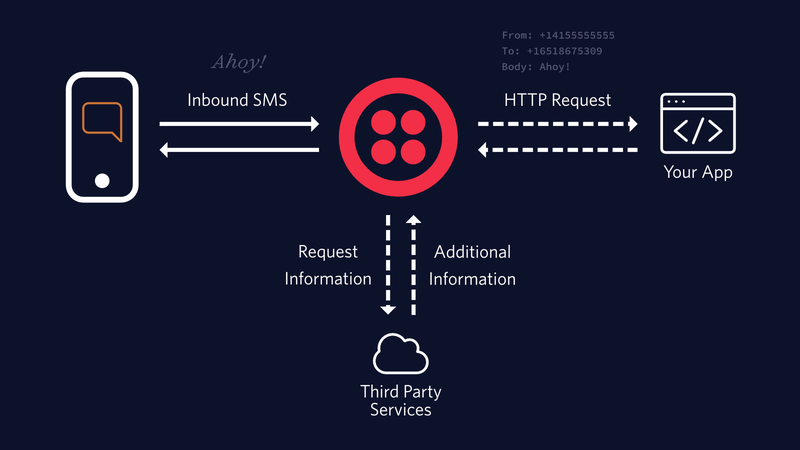
### API Paiement

Nous allons implémenter une interface propre à notre système pour les API des différents fournisseurs de service. Par exemple pour le paiement nous utiliserons l’API de Stripe qui sera couplé à notre interface. Cet Api fonctionne de la manière suivante : un formulaire est créé du côté serveur, l’utilisateur rempli et confirme le formulaire, l’information part chez le fournisseur du paiement et une fois confirmé renvoie à l’application la confirmation de paiement. Afin que cela fonctionne il faut créer un webhook du côté de l’application.



### API SMS

[Afin](https://dev.smsfactor.com/en/api/sms/getting-started) de valider les comptes par sms, l’application va utiliser le fournisseur Twilio. L’utilisateur recevra un code et devra le confirmer sur l’application. Si le numéro de téléphone n’a pas déjà été validé, le compte est validé.



### API Email newsletter

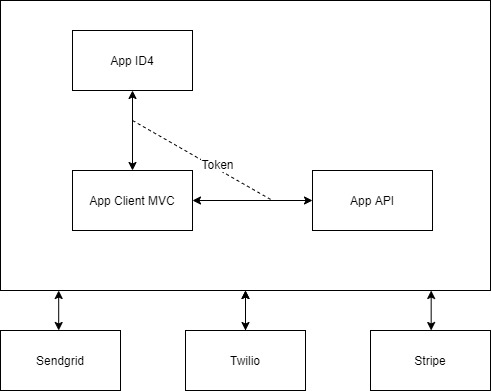
Afin d’envoyer des mails massivement, l’application va utiliser le fournisseur SendGrid.

## Référence au concept d’intégration

N/A

## Délimitation

Texte, éventuellement diagramme



Les trois app peuvent utiliser les Api Sendgrid, Twilio et Stripe.

# Évaluation de la faisabilité

Évaluation de la faisabilité, référence à la documentation pour la réalisation du prototype

# Conformité avec les prescriptions

N/A

# Attribution et respect des exigences

Références aux exigences concernant le système

| ID | Exigence | Attribution au sous-système / au composant | Évaluation de la couverture de l’exigence |
| --- | --- | --- | --- |
| A1 | Création de compte  L’utilisateur doit pouvoir se créer un compte sur l’application. | Fonctionnalité |  |
| A2 | Authentification  L’utilisateur doit pouvoir se connecter sur l’application. | Fonctionnalité |  |
| A3 | Gestion du compte  L’utilisateur doit pouvoir modifier différentes informations liées à ses centres d’intérêts ou informations d’authentification. | Fonctionnalité |  |
| A4 | Suppression du compte  L’utilisateur peut supprimer son compte et effacer toute information le concernant sur l’application. | Fonctionnalité |  |
| A5 | Récupérer données personnelles  L’utilisateur doit pouvoir récupérer ses données personnelles en tout temps. | Fonctionnalité |  |
| A6 | Visualisation des profils  L’utilisateur doit pouvoir afficher les profils des autres personnes utilisant l’application | Fonctionnalité |  |
| A7 | Algorithme de match  L’algorithme doit prendre diverses données en compte pour pouvoir proposer une liste de profil | Fonctionnalité |  |
| A8 | Formule d’adhésion  Proposer différentes formules d’abonnement. | Fonctionnalité |  |
| A9 | Paiement sécurisé  L’utilisateur doit pouvoir effectuer un paiement sans mettre en péril ses données. | Fonctionnalité |  |
| A10 | Vérification des comptes  Vérification effectuée pour assurer l’authenticité de l’utilisateur | Fonctionnalité |  |
| A11 | Communication  Offrir aux utilisateurs différente canaux de discussions | Fonctionnalité |  |
| A12 | Campagne d’e-mail (newsletter)  Garder une interactivité avec l’utilisateur. | Fonctionnalité |  |
| A13 | Gestion des rôles  Pouvoir gérer les rôles et accorder des privilèges à certains comptes pour l'administration. | Fonctionnalité |  |
| A14 | Support client  Rendre possible pour la clientèle de nous contacter et nous faire des propositions. | Fonctionnalité |  |
| A15 | Etablissement du profil  Etablir le profil de l’utilisateur en fonction de plusieurs critères | Fonctionnalité |  |
| A16 | Langues  Ajouter différentes langues au site pour toucher une population plus grande. | Fonctionnalité |  |
| A17 | Administration  Espace pour gérer l’application | Fonctionnalité |  |
| A18 | Bot  Utiliser des bots pour offrir un service propre aux utilisateurs | Fonctionnalité |  |
| A19 | Localisation  Evaluer les possibilités en fonction de la localisation de chaque utilisateur. | Fonctionnalité |  |
| A20 | Demande de réinitialisation  L’utilisateur doit pouvoir faire une demande pour récupérer son mot de passe via email. | Fonctionnalité |  |
| A21 | Traçabilité  Pouvoir avoir des informations sur la connexion de chaque utilisateur en récupérant leur adresse IP et leur temps de connexion. | Fonctionnalité |  |

Tableau 2: Liste des exigences, avec attribution et évaluation

Abréviations et glossaire

| Abréviation / Terme technique | Explication |
| --- | --- |
| HERMES | La méthode de gestion de projets et de programmes HERMES 5 est une norme eCH. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Tableau 3: Abréviations et glossaire

Table des matières

Suivi des modifications 1

Description 1

1 Structure du système 2

1.1 Aperçu du système 2

1.2 Sous-systèmes et composantes 2

1.3 Architectures / Modèles 2

2 Interfaces et délimitation 2

2.1 Interfaces avec les systèmes environnants 2

2.2 Référence au concept d’intégration 2

2.3 Délimitation 2

3 Évaluation de la faisabilité 2

4 Conformité avec les prescriptions 2

5 Attribution et respect des exigences 3

Abréviations et glossaire 4

Table des matières 4

Table des tableaux 4

Table des tableaux

Tableau 1: Contrôle des modifications 1

Tableau 2: Liste des exigences, avec attribution et évaluation 3

Tableau 3: Abréviations et glossaire 4