

# **Plateforme d'Automatisation et de Gestion de Contenus Numériques**

## **1. Analyse des Besoins**

- **Besoins fonctionnels**
  - Gestion des fichiers multimédias (téléversement, édition, organisation).
  - Planification et publication automatique sur plusieurs plateformes.
  - Génération de suggestions via un moteur d'IA.
  - Accès à des statistiques avancées.
  - Gestion des équipes et des comptes utilisateurs.
  - Workflows automatisés (ex. publication programmée, notifications).
  - Intégration avec des API tierces.
- **Besoins non fonctionnels**
  - Scalabilité automatique : ajustement dynamique des ressources.
  - Faible latence : temps de réponse optimisé.
  - Sécurité renforcée : authentification et protection des données.
  - Optimisation des coûts : facturation à l'usage.
  - Interopérabilité : connexion fluide avec des services tiers.
  - Haute disponibilité : architecture résiliente aux pannes.

## **2. Identification des Contraintes et Hypothèses**

- **Contraintes techniques et organisationnelles**
  - Techniques
    - Stockage scalable (Amazon S3, Google Cloud Storage).
    - Base de données sans serveur (DynamoDB, Firebase Firestore).
    - Gestion d'événements (EventBridge, Pub/Sub).
    - Authentification et accès sécurisé
  - Organisationnelles
    - Aucune gestion manuelle des serveurs.
    - Simplicité d'utilisation pour les créateurs et entreprises.
    - Possibilité d'ajouter rapidement de nouvelles fonctionnalités.
- **Hypothèses pour les zones d'incertitudes**
  - L'IA utilisera des modèles pré-entraînés (GPT, Vision AI).
  - Les APIs des réseaux sociaux sont accessibles et fiables.
  - Les utilisateurs cibles sont des créateurs de contenu et des entreprises.

## **3. Plan d'Action**

- **Questions ouvertes aux parties prenantes**
  - Quelle est la taille des fichiers multimédias à gérer ?

- Quels formats de fichiers doivent être supportés ?
- Quelle fréquence de publication est attendue ?
- Quels indicateurs clés de performances analytiques sont essentiels ?
- Quel est le niveau de personnalisation des workflows attendu ?
- **Stratégie pour combler les manques d'information**
  - Entretiens avec des créateurs de contenu et marketeurs.
  - Étude des plateformes concurrentes.
  - Analyse des APIs des plateformes cibles.

## 4. Choix d'une Architecture Logicielle

- **Type d'architecture**
  - Serverless
- **Pourquoi ce choix ?**
  - Scalabilité automatique (répond à la charge dynamique).
  - Séparation claire des responsabilités (chaque service gère une tâche spécifique).
  - Coût réduit (facturation uniquement sur l'exécution).
  - Interopérabilité (facilité d'intégration avec des API externes).
  - Déploiement et mise à jour indépendants (agilité accrue).

## 5. Définition de l'Architecture Globale

- **Diagramme d'architecture**
  - Composants principaux
    - Interface utilisateur
    - Services back-end
    - Communication entre les services
- **Principaux modules**
  - Moteur IA de recommandations
  - Assistant vocal
  - Générateur automatique de vidéos courtes et montages dynamiques
  - Intégration avec des plateformes de monétisation
- **Technologies possibles**
  - Front-end : React.js, Vue.js
  - Back-end : Node.js avec
  - Stockage : Amazon S3, Google Cloud Storage
  - Base de données : Firestore, DynamoDB
  - Authentification : Firebase Auth, OAuth 2.0
  - Événements : EventBridge, Pub/Sub
  - Analytics : Google BigQuery, AWS Athena
  - Moteur IA : OpenAI API, Google Vision AI

## Conclusion

L'architecture Serverless apporte :

- Une gestion entièrement automatisée des ressources.
- Une modularité et une extensibilité accrues.
- Une réduction des coûts grâce à un paiement à l'usage.
- Une faible latence et une haute disponibilité.