



# Spécifications techniques Crypto Viz

👤 Créé par	👤 Stéphane DS
🕒 Heure de création	@28 septembre 2023 10:36
🏷️ Étiquettes	Ingénierie

## Énoncé du problème

Un client, actif dans le domaine des crypto-monnaies, souhaite rester informé en temps réel des développements du marché et prendre des décisions commerciales éclairées. Cependant, la quantité de données générées par les crypto-monnaies est énorme et évolue rapidement. Le client a besoin d'une solution efficace et automatisée pour collecter, analyser et visualiser ces données en temps réel.

## Objectifs

### 1. Collecte de Données en Continu :

- Les données sur les crypto-monnaies sont collectées en continu à partir de la source d'actualités définie.
- Le scraper web fonctionne de manière automatisée et constante, sans interruption.

### 2. Analyse en Continu :

- Les données collectées sont traitées en continu par le constructeur d'analyses en ligne.
- Les analyses sont générées rapidement et efficacement pour refléter l'évolution du marché en temps réel.

### 3. Visualisation Dynamique :

- Le visualiseur dynamique affiche les analyses générées de manière dynamique et se met à jour automatiquement lorsque de nouvelles données sont disponibles.
- Les utilisateurs peuvent accéder aux visualisations en temps réel sans délai significatif.

#### **4. Dimension Temporelle :**

- L'application inclut une dimension temporelle dans les visualisations, permettant aux utilisateurs d'explorer l'évolution des données au fil du temps.
- Les utilisateurs peuvent analyser les tendances passées et actuelles grâce à des fonctionnalités d'exploration temporelle.

#### **5. Architecture Robuste :**

- L'architecture logicielle de l'application est robuste, garantissant la fiabilité et la disponibilité de l'application en temps réel.
- L'application peut gérer la montée en charge en cas de volumes de données élevés.

#### **6. Projet Git :**

- Le projet Git contient l'ensemble du code source de l'application, les fichiers de configuration nécessaires et, le cas échéant, des automatisations pour le déploiement ou la gestion des données.

#### **7. Utilisabilité et Convivialité :**

- L'interface utilisateur de l'application est conviviale et intuitive, permettant aux utilisateurs de naviguer facilement dans les données et les analyses en temps réel.

#### **8. Conformité et Éthique :**

- L'application respecte les lois et règlements relatifs au scraping de données et à la protection de la vie privée.
- L'utilisation de l'application est conforme aux normes éthiques et légales.

#### **9. Formation et Support :**

- L'équipe du client est formée à l'utilisation de l'outil, et un support est disponible pour résoudre d'éventuels problèmes ou répondre aux

questions.

#### 10. Performance :

- L'application est optimisée pour des performances élevées, assurant une réactivité rapide et une expérience utilisateur fluide, même en cas de volumes de données importants.

#### 11. Sécurité :

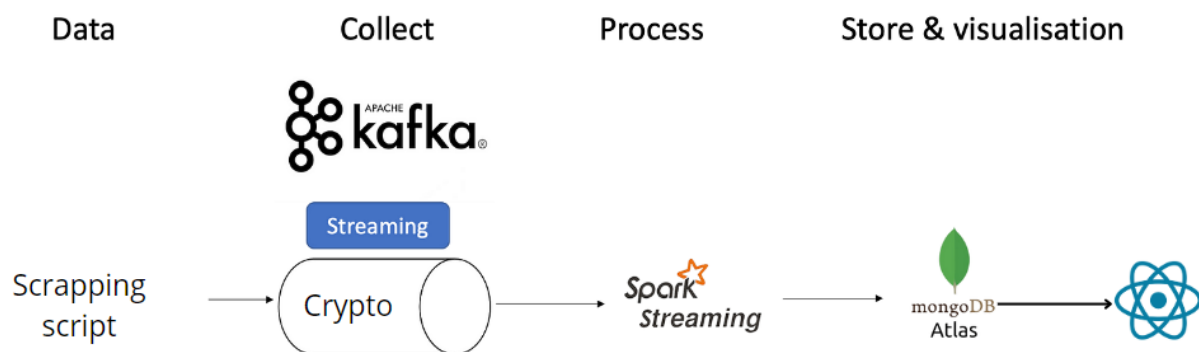
- Des mécanismes de sécurité sont en place pour protéger les données collectées et stockées par l'application contre les menaces potentielles.

#### 12. Extensibilité :

- L'application est conçue de manière à pouvoir être étendue avec de nouvelles fonctionnalités ou sources de données à l'avenir, si nécessaire.

## Solution proposée

### Architecture de l'Application "Crypto Viz"



#### 1. Composant de Collecte de Données en Continu (Online Web Scraper) :

- **Outil** : BeautifulSoup, Requests.
- **Description** : Ce composant collecte en continu des données à partir d'une source d'actualités sur les crypto-monnaies. Il doit être planifié pour s'exécuter régulièrement.
- **Automatisation** : Utilisation de tâches planifiées pour déclencher le scraping à intervalles réguliers (chaque 10s).

- **Stockage Temporaire** : Les données collectées sont stockées temporairement dans une file d'attente Kafka dans le topic Crypto pour le traitement ultérieur.

## 2. Composant de Traitement des Données en Continu (Online Analytics Builder) :

- **Outil** : Spark Structured streaming pour la consommation des données depuis les topics Kafka et le traitement en parallèle.
- **Description** : Ce composant traite en continu les données collectées par le scraper. Il génère des analyses en temps réel à partir des données en occurrence des agrégation par crypto et par intervalle de temps (10s, 1min, 15min, 30min, 1h, 1d, 1s, 1m).
- **Traitement Parallèle** : Utilisation de la parallélisation pour accélérer le traitement des données.
- **Stockage des Analyses** : Les analyses générées sont stockées dans une base de données MongoDB atlas pour un accès rapide.

## 3. Composant de Visualisation Dynamique (Dynamic Viewer) :

- **Outil** : Frameworks JavaScript tels que React pour la création d'interfaces utilisateur dynamiques.
- **Description** : Ce composant affiche les analyses générées de manière dynamique et en temps réel.
- **Mise à Jour en Temps Réel** : Utilisation de WebSockets pour mettre à jour automatiquement les visualisations lorsque de nouvelles données sont disponibles.
- **Exploration Temporelle** : Intégration d'une dimension temporelle permettant aux utilisateurs d'explorer l'évolution des données au fil du temps.

## 4. Système d'Architecture et de Gestion :

- **Infrastructure Cloud** : Utilisation d'une infrastructure cloud (par exemple, AWS, Azure, ou GCP) pour garantir la scalabilité et la disponibilité.
- **Serveurs d'Application** : Des serveurs d'application (par exemple, Nginx, Apache) peuvent être utilisés pour servir l'application web.

- **Gestion des Tâches** : Un outil de gestion de tâches (par exemple, Apache Airflow) peut être utilisé pour planifier et orchestrer les différentes étapes de collecte, de traitement et de visualisation.

## 5. Sécurité et Gestion des Données :

- **Sécurité** : Utilisation de protocoles HTTPS pour sécuriser les communications. Mise en place de mécanismes d'authentification et d'autorisation pour les utilisateurs.
- **Stockage des Données** : Les données sensibles sont stockées de manière sécurisée. Un chiffrement peut être utilisé si nécessaire.
- **Protection contre les Attaques** : Mise en œuvre de mesures de sécurité pour protéger l'application contre les attaques (par exemple, injection SQL, Cross-Site Scripting).

## 6. Monitoring et Maintenance :

- **Outils de Monitoring** : Utilisation d'outils de monitoring (par exemple, Prometheus, Grafana) pour surveiller la santé de l'application et la performance du système.
- **Maintenance** : Planification de la maintenance régulière de l'application, y compris les mises à jour de sécurité et les correctifs.