

DOCUMENTATION CRYPTO - VIZ

Introduction

L'univers des cryptomonnaies connaît une évolution fulgurante, marquée par une volatilité extrême et une influence directe des actualités et des tendances médiatiques. Chaque annonce, chaque rumeur ou fluctuation de sentiment sur les réseaux sociaux peut impacter en quelques minutes la valeur d'un actif numérique. Dans ce contexte d'information en continu, la capacité à **collecter, analyser et visualiser** ces données en temps réel devient un atout stratégique pour les investisseurs, analystes et décideurs.

C'est pour répondre à ce besoin que le projet **CRYPTO VIZ** a été conçu. Il s'agit d'une application web big data permettant de **surveiller, comprendre et interpréter** l'actualité du marché des cryptomonnaies de manière dynamique. En offrant une vision globale et instantanée des informations disponibles, l'outil aide ses utilisateurs à identifier rapidement les tendances, anticiper les mouvements du marché et prendre des décisions éclairées.

L'application repose sur une **architecture distribuée** bâtie autour du paradigme **producteur/consommateur**, garantissant une collecte et un traitement continus des flux d'informations. Cette architecture modulaire permet d'assurer la performance, la fiabilité et l'évolutivité du système.

Trois composants principaux forment la base de **CRYPTO VIZ** :

- **Online Web Scraper** : collecte en continu les actualités issues de sources spécialisées dans les cryptomonnaies ;
- **Online Analytics Builder** : traite, analyse et transforme les données brutes en indicateurs exploitables ;
- **Dynamic Viewer** : offre une interface interactive pour visualiser les résultats analytiques en temps réel, avec une représentation claire de leur évolution dans le temps.

En somme, **CRYPTO VIZ** vise à devenir un outil de référence pour la **veille et l'analyse en temps réel du marché des cryptomonnaies**, alliant performance technique, simplicité d'utilisation et pertinence des analyses. Ce document présente l'architecture de l'application ainsi que les principaux choix techniques et méthodologiques ayant guidé sa conception.

Objectif du projet

L'objectif du projet **CRYPTO VIZ** est de fournir une plateforme web big-data complète et opérationnelle qui transforme, en temps quasi-réel, des flux d'actualités sur les crypto-monnaies en indicateurs exploitables et visualisations interactives. Ci-dessous une description précise et détaillée des objectifs fonctionnels, non-fonctionnels, des livrables attendus, des métriques de succès et des contraintes.

1. Objectif principal

Fournir aux analystes, traders et décideurs un outil capable de **collecter**, **traiter**, **analyser** et **visualiser** automatiquement les actualités liées aux crypto monnaies, avec une latence faible, une haute fiabilité et une interface qui permet d'explorer l'évolution temporelle des signaux (tendances, sentiment, fréquence, corrélations).

2. Objectifs fonctionnels (par composant)

2.1 Online Web Scraper (producteur)

- **Collecte continue** : ingérer automatiquement articles, titres, résumés et métadonnées (source, auteur, date, URL) depuis plusieurs flux (APIs, RSS, pages web) 24/7.
- **Déduplication** : détecter et éliminer doublons (même contenu/URL ou très similaires).
- **Normalisation** : produire un schéma standardisé pour chaque événement (timestamp UTC, source, langue, texte brut, tokens/entités extraites).
- **Enrichissement léger** : extraction d'entités nommées (ex : noms de tokens, exchanges), détection de la langue, horodatage unifié.
- **Publication** : pousser chaque événement vers un broker/message queue (Kafka, RabbitMQ ou équivalent) en tant que producteur, avec topics partitionnés par source/type.

2.2 Online Analytics Builder (consommateur / producteur)

- **Consommation temps réel** : consommer les messages produits par le scraper et effectuer des traitements stream.
- **Analyses immédiates** : calculer en continu :
 - score de sentiment (par document et agrégé par période),
 - fréquence d'apparition d'entités (tokens, noms de projets),
 - détection d'événements/spikes (pic de volume ou sentiment),
 - agrégats temporels (ex : counts, moving averages, rate of change).
- **Stockage des outputs** : écrire résultats dans des stores adaptés (time-series DB comme Influx/Timescale pour séries temporelles; document store ou OLAP pour métadonnées).
- **API/Publication** : expose les résultats via API / push vers topics dédiés pour que le front (Dynamic Viewer) s'abonne en temps réel.
- **Pipeline de ML optionnel** : possibilité d'enrichir avec modèles (classif. d'événements, détection d'anomalies) en mode online/near-line.

2.3 Dynamic Viewer (consommateur)

- **Visualisation en temps réel** : affichage dynamique des analytics (charts temporels, heatmaps, word clouds, tableaux de métriques) qui se mettent à jour automatiquement à la réception de nouveaux data points.
- **Dimension temporelle** : capacité d'explorer l'historique (zoom, window sliding, comparaison de périodes).
- **Interactivité** : filtres (source, token, sentiment), zoom temporel, drill-down sur articles originaux (clic sur un pic → voir les articles associés).
- **Latence utilisateur** : mise à jour visible sous un objectif (voir section non-fonctionnelle).
- **Export & alerting** : exporter graphiques/données, configurer alertes (webhook, email) sur règles (ex : sentiment < -0.5 pour un token).

3. Objectifs non-fonctionnels (exigences de qualité)

3.1 Performance & latence

- **End-to-end** (article ingéré → analytics visible dans le viewer) : ≤ 5 secondes en condition nominale; **objectif cible** $\leq 2s$ pour la plupart des événements critiques.
- **Throughput** : supporter au minimum **500 événements/s** (scalabilité horizontale prévue pour monter à $> 5k$ événements/s).
- **Temps de réponse API** : $p95 < 200$ ms pour requêtes d'agrégats récents.

3.2 Disponibilité et fiabilité

- **Disponibilité cible** pour les composants critiques : **99.9%** (production).
- **Tolérance aux pannes** : consommateurs/producteurs re démarrables; retry/backoff pour ingestion; persistance des messages jusqu'à traitement.

3.3 Scalabilité

- Architecture distribuée, scale-out pour le scraper (par source), le broker (partitions) et le builder (consumers group), ainsi que pour le frontend (stateless servers, CDN).

3.4 Exactitude et qualité des données

- **Taux de déduplication visé** : $> 95\%$.
- **Précision du sentiment (baseline)** : $> 75\%$ sur un jeu d'évaluation initial, améliorable via réentraînement.
- **Surveillance de la qualité** : métriques de data loss, erreurs de parsing, latence de traitement.

3.5 Sécurité & conformité

- **Sanitisation** des contenus (XSS) avant affichage.
- **Contrôle d'accès** pour l'interface (authentification, rôles).
- **Logs & audit** pour traçabilité (qui a vu quoi / quand).
- **Respect de la propriété intellectuelle** : stockage des URL et métadonnées, gestion des limitations de scrapping selon robots.txt et TOS des sources.

4. Livrables & fonctionnalités mesurables

- **Pipeline ingestion** : scrappers pour N sources (configurable), avec monitoring.
- **Broker** : infrastructure message (Kafka ou équivalent) déployée localement/conteneurisée.
- **Processing stream** : jobs/consumers effectuant sentiment, NER, agrégats.
- **Stores** : base time-series + base de recherche/analytics.
- **API REST / WebSocket** : endpoints pour consommations temps-réel et historiques.
- **Front-end** : dashboard réactif avec graphiques temporels et filtres.
- **Documentation** : architecture, diagrammes, procédures de déploiement et runbooks.
- **Tests & CI** : tests unitaires, tests d'intégration, scripts d'automatisation (docker-compose / k8s manifests).

5. KPIs & critères de succès

- **Latence moyenne E2E** $\leq 5s$ (objectif $\leq 2s$).
- **Disponibilité** $\geq 99.9\%$ pour les composants critiques.
- **Throughput supporté** ≥ 500 evts/s (scalable).
- **Taux de couverture des sources** : capacité à intégrer au moins 10 sources initiales et ajouter de nouvelles en ≤ 1 jour de configuration.
- **Exactitude sentiment** $\geq 75\%$ initialement.
- **Taux d'erreurs d'ingestion** $< 1\%$ sur un mois de fonctionnement.
- **Adoption** : test utilisateur — 80% des testeurs métier trouvent les visualisations utiles pour la prise de décision.

6. Contraintes et hypothèses

- **Hypothèses** : accès légal aux flux/APIs choisis ; ressources infra conteneurisées disponibles (CPU/RAM/disque).
- **Contraintes** : respect des limites d'APIs (rate limits), obligation d'appliquer robots.txt quand on scrape des pages, et budget/ressources pour hébergement si déployé en production.

- **Données historiques** : l'objectif initial est de traiter données live ; l'ingestion backfill historique est optionnelle mais recommandée pour analyses comparatives.

7. Roadmap technique (haut niveau)

1. **Phase 1 — MVP (0–4 semaines)** : scrapper 3 sources, pipeline Kafka local, consumer simple calculant fréquences + sentiment baseline, dashboard avec séries temporelles.
2. **Phase 2 — Robustesse (4–8 semaines)** : déduplication, monitoring, tests de charge, stockage time-series, alerting.
3. **Phase 3 — Amélioration (8–12 semaines)** : modèles ML (anomalies, classification), authentification, règles d'alertes avancées, orchestration (k8s).
4. **Phase 4 — Production & scalabilité (12+ semaines)** : optimisation, HA, sauvegardes, documentation et automatisation complète.

8. Résumé / formulation concise des objectifs

- Livrer un pipeline **end-to-end** de collecte → traitement → visualisation temps-réel des news crypto.
- Garantir **faible latence**, **scalabilité**, **fiabilité** et **qualité** des analyses.
- Offrir une interface interactive avec **dimension temporelle** et capacités d'alerte/exports pour la prise de décision.

Contexte et problématique

Depuis plusieurs années, le marché des **cryptomonnaies** s'impose comme l'un des secteurs les plus dynamiques et les plus imprévisibles de l'économie numérique. Son évolution est fortement influencée par les **informations en ligne** : annonces officielles, décisions réglementaires, innovations technologiques, ou encore réactions sur les réseaux sociaux. Ces flux d'actualités, souvent massifs et instantanés, façonnent directement la perception du marché et, par conséquent, la valeur des actifs numériques.

Cependant, cette abondance d'informations s'accompagne d'un défi majeur : la **surinformation**. Chaque minute, des centaines d'articles, de tweets et de publications apparaissent sur le web, rendant presque impossible pour un utilisateur humain d'en extraire rapidement les éléments réellement pertinents. Les investisseurs et analystes se retrouvent donc face à un volume de données considérable, dispersé, et difficile à interpréter sans outils adaptés.

À cela s'ajoute la **vitesse de réaction exigée** dans l'univers des cryptomonnaies. Les variations de cours peuvent se produire en quelques secondes, en réaction à une simple rumeur ou à un communiqué de presse. Dans un tel environnement, disposer d'un outil capable de **collecter les informations en temps réel**, de **les analyser automatiquement** et de **les visualiser de manière claire et dynamique** devient indispensable pour conserver un avantage compétitif.

C'est dans ce contexte que naît le projet **CRYPTO VIZ**. L'objectif est de concevoir une application web capable de :

- centraliser les actualités liées aux cryptomonnaies provenant de multiples sources fiables ;
- transformer ces informations en **indicateurs analytiques pertinents** (tendances, sentiment, volume d'occurrences, etc.) ;
- et permettre leur **visualisation interactive dans le temps**, afin d'observer l'évolution du marché et d'anticiper ses mouvements.

Le projet s'inscrit donc dans une **problématique double** :

1. **Technique** – Comment concevoir une architecture capable de collecter, traiter et restituer des données massives en continu, tout en garantissant performance, fiabilité et évolutivité ?
2. **Analytique et décisionnelle** – Comment transformer ces flux d'informations non structurées en analyses compréhensibles et exploitables pour les utilisateurs finaux ?

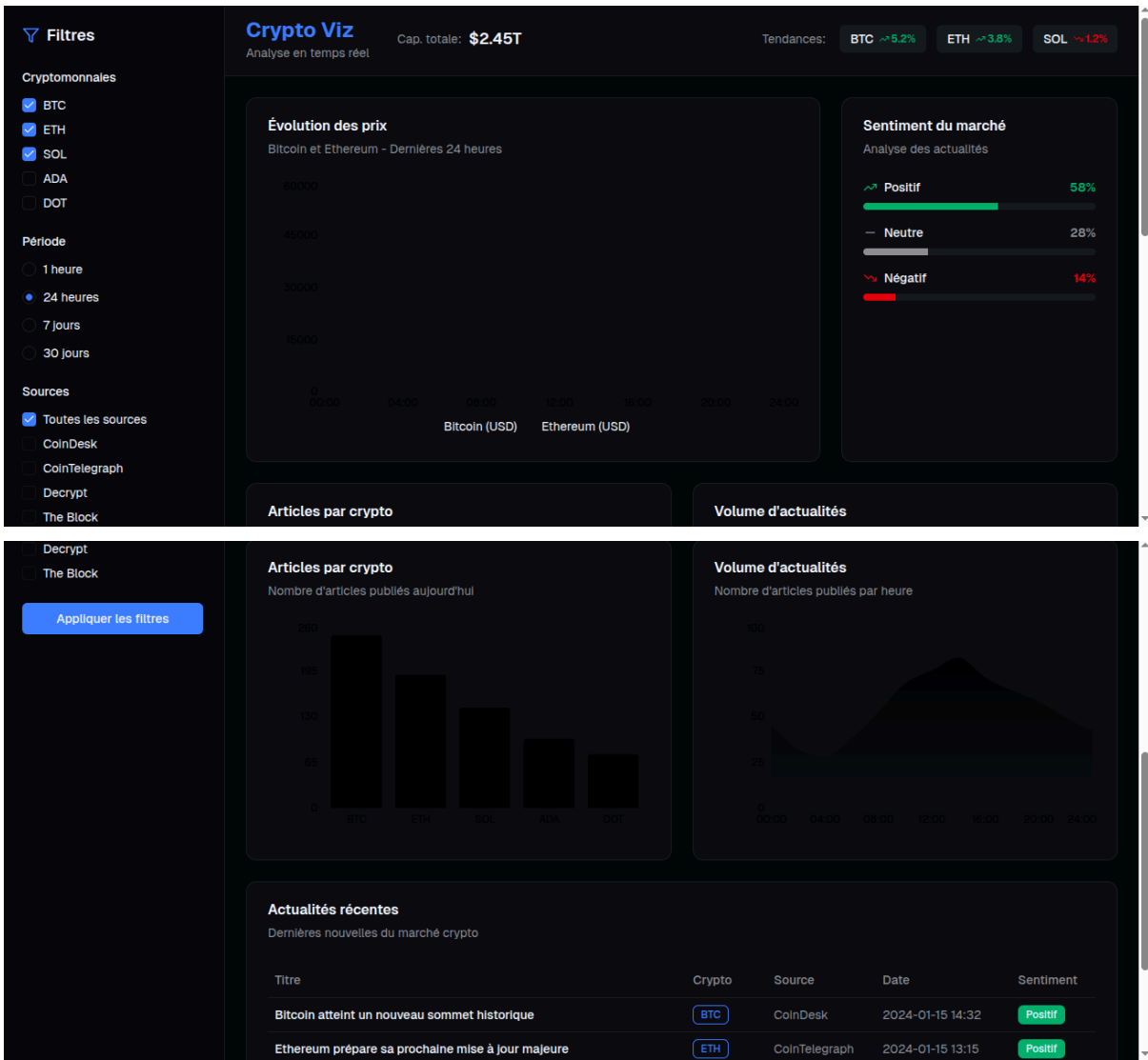
CRYPTO VIZ se positionne ainsi comme une réponse concrète à ces enjeux, en combinant les principes du big data, du streaming temps réel et de la visualisation interactive pour offrir une vision claire, instantanée et intelligente du marché des cryptomonnaies.

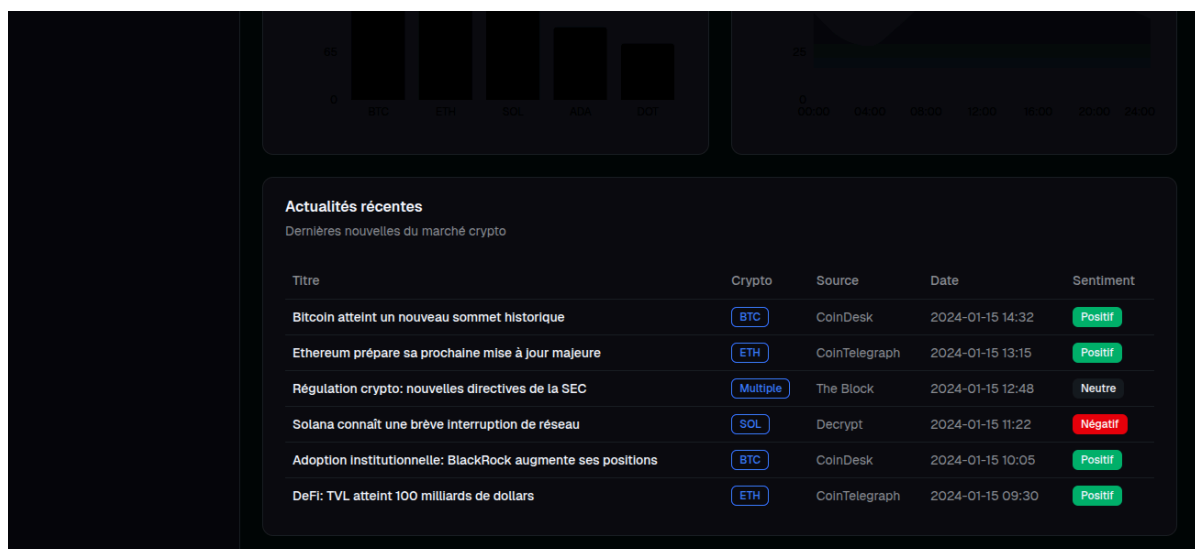
Présentation du projet:

Crypto viz est une plateforme qui permet d’analyser et de visualiser les données liées à la cryptomonnaie. Cette plateforme permet de suivre l’évolution du marché en temps réel sur la base de certains indicateurs tels que: le prix, les tendances du marché, les sentiments des actualités afin d’aider les utilisateurs à prendre des décisions éclairées sur leurs investissements.

La base du projet est la collecte en temps réel des données textuelles et numériques, leur traitement ainsi que leur visualisation via un tableau de bord intuitif.

Maquette de l'utilisateur





Perspectives

Exploration des ressources

La réalisation de ce projet implique l'utilisation de diverses ressources.

Outils

Go - un langage de programmation extrêmement rapide

Le langage de programmation Go nous a permis de développer le crawler utilisé pour collecter les news crypto. Notre choix s'est porté sur Go car il offre de nombreux avantages comme : la rapidité, la concurrence et une bonne gestion de la mémoire. Il est largement utilisé dans le développement de crawler et le de bots de trading.

<http://go.dev/>

Solutions

CoinMarketCap Community

La plateforme communautaire de coinmarketcap nous a servi de feed crypto pour extraire les dernières nouvelles du marché. Cette plateforme fournit des centaines d'articles par minutes et permet d'être au courant de chaque changement capable d'impacter le marché.

<https://coinmarketcap.com/community/>