



## Fiche **TSORAGE**

UNE PLATEFORME DE GESTION DE DONNÉES (I)IOT OPEN SOURCE

Vous êtes une entreprise à la recherche de technologies fiables, rapides et rentables concernant le déploiement d'applications basées sur des séries chronologiques à grande échelle? TStorage est la solution idéale pour gérer vos flux IoT!

Après la description de TStorage et de ses différents atouts, nous évoquons les concepts nécessaires à la compréhension de cette plateforme et expliquons son architecture. Nous terminons en vous présentant un cas d'étude.

### Prérequis

- Pour la lecture de ce document, connaissance élémentaire de l'IoT et de ses enjeux, de Kubernetes, de Cassandra.
- Pour tester TStorage,
  - un cluster Kubernetes doit être installé, par exemple grâce à [minikube](#).
  - Maîtrise de HTTP et/ou MQTT, JSON, PromQL, Docker.

### Partenaires

## Fiche **TSorage**- Une plateforme de gestion de données (I)IoT open source

### Pour quoi faire?

À l'heure actuelle, les entreprises industrielles font face à une concurrence mondiale de plus en plus agressive. L'Internet industriel des objets (IIoT) est perçu comme une opportunité clé pour renforcer la position de ces entreprises ou pour gagner en compétitivité grâce à une meilleure productivité, à l'amélioration de la qualité des produits et à un meilleur contrôle des processus (voir notre [fiche sur la gestion des réseaux de capteurs](#)).

Cependant, la gestion de la vitesse et du volume des données de séries chronologiques fournies par ces IIoT représentent un défi dans lequel la valeur des solutions traditionnelles de gestion des données pour l'industrie tend à diminuer. Le monde industriel recherche des outils et des approches capables de répondre aux nouveaux besoins relatifs à l'évolutivité, à la disponibilité, au problèmes de réutilisation, d'intégration et de tarification, entre autres.

Afin de répondre à ces besoins, nous proposons une solution innovante pour la gestion des séries chronologiques appelée **TSorage**.

TSorage est une plateforme open source, évolutive et résiliente, qui permet la collecte, l'ingestion, le traitement et le stockage de séries chronologiques (ou séries temporelles) générées par l'IoT (industriel ou non) comme les détecteurs, les sondes et autres capteurs au sens large du terme.

Cette plateforme propose une collection de services intégrés pour gérer les séries temporelles, à grande échelle, de tout type, avec un horodatage avec une résolution temporelle d'une milliseconde et avec aussi peu de contraintes techniques que possible.

Une série temporelle est définie comme une collection de valeurs, triées par un horodatage associé à chaque valeur. Dans TSorage, une valeur peut représenter n'importe quel concept tant que celui-ci peut être représenté sous forme d'un objet JSON.



## Fiche TStorage- Une plateforme de gestion de données (I)IoT open source

### ✦✦ Les atouts de TStorage

TStorage présente les avantages suivants :

#### 1 Une disponibilité et un passage à l'échelle avant tout (évolutivité)!

TStorage s'appuie sur du matériel standard et moyen de gamme (commodity hardware) pour garantir un service évolutif et résilient aux pannes. Étant nativement une solution distribuée et décentralisée, ses capacités peuvent être étendues en ajoutant simplement plus de ressources sur un cluster TStorage. Lorsqu'il est déployé sur plusieurs sites, TStorage offre des performances de lecture et d'écriture locales tout en prenant en charge de manière transparente la réplication et la synchronisation entre sites, dans le monde entier. Lorsqu'un site se remet d'un problème de connexion, il se re-synchronise automatiquement avec les autres sites du groupe.

#### 2 Pas de vendor locking mais des technologies standardisées, ouvertes, et pérennes

La technologie évolue extrêmement rapidement, surtout le domaine (I)IoT où de nouvelles façons de gérer et d'exploiter les capteurs émergent chaque année. Afin d'atténuer le risque de faire des choix technologiques qui s'avéreraient inappropriés à l'avenir, TStorage est composé de modules indépendants et basés sur des technologies open source. Avec une telle approche, les mises à jour sont beaucoup plus faciles lorsqu'une technologie vient en remplacer une autre. Tous les services de TStorage sont disponibles via une API REST qui offre un moyen standardisé de s'abstraire des technologies sous-jacentes. Cette plateforme favorise également l'intégration avec n'importe quelle source ou consommateur de données, faisant de TStorage une plateforme de choix pour vos applications IoT.

#### 3 Une flexibilité des données

La plupart des capteurs mesurent un signal continu, tel qu'une température ou une pression. Cependant, les séries chronologiques couvrent également de nombreux autres types de données, tels que les positions géographiques, les transactions commerciales et pratiquement tous les événements répétitifs. TStorage gère nativement des types de données couramment utilisés et est conçu pour être facilement étendu afin de prendre en charge vos types de données spécifiques. Si vous pouvez représenter vos mesures sous forme d'objets JSON, TStorage peut les gérer!

Fiche **TSorage**- Une plateforme de gestion de données (I)IoT open source

4

#### **Une adaptation rapide et de manière prévisible**

L'ajout d'une nouvelle source de données (comme un capteur) doit être aussi simple et rapide que possible afin de ne pas étouffer toute innovation. Commencez simplement à alimenter TStorage avec un nouveau flux de données et administrez-le dans un second temps, soit via une application Web dédiée, soit par programmation. Chaque valeur peut être soumise avec des propriétés arbitraires (appelées "tags" dans la terminologie TStorage, nous détaillons cela dans la section suivante) qui permettent d'interroger et de gérer plus efficacement les sources de données. En fin de compte, les utilisateurs ne font plus référence à un identifiant de source unique, mais interrogent, comparent et agrègent les sources en fonction de leurs tags.

Au fur et à mesure que vous intégrez de plus en plus de sources de données à la solution, vos besoins de traitement augmentent. TStorage s'appuie sur une architecture élastique qui exploite efficacement les ressources de votre infrastructure. Commencez avec un petit nombre de services conteneurisés et étendez-les à volonté en exécutant simplement plus de nœuds de travail.

5

#### **Une adaptation à votre infrastructure, prête pour le Cloud**

Lorsque TStorage est utilisé pour gérer des données sensibles, le déploiement sur site peut être préféré à l'utilisation d'une solution d'hébergement à distance. Pour d'autres cas d'utilisation, un déploiement sur un Cloud public ou privé est une meilleure option. Dans les deux cas, TStorage est fourni avec des scripts de déploiement et de surveillance qui réduisent la charge de déploiement et de maintenance de la solution.

## Fiche TStorage- Une plateforme de gestion de données (I)IoT open source



### Quelques concepts

Voici les différentes notions nécessaires à la bonne compréhension de la plateforme TStorage :

- **Une métrique**

Concept fondamental de TStorage, une *métrique* est une entité abstraite associée à des mesures classées chronologiquement. Chaque *mesure* est également appelée un point de données ou une *observation*.

Dans TStorage, tous les points de données appartenant à une métrique représentent le même phénomène physique, numérique ou logique, et ont donc typiquement le même type de données (bien que ce ne soit pas une limitation technique de la plateforme). Le type de données d'une observation détermine la manière dont TStorage la stocke et la présente, ainsi que les transformations auxquelles cette observation peut être soumise.

- **Les tags**

Les *tags* sont des propriétés associées aux points de données. Leur utilité est d'aider l'utilisateur à comprendre la signification d'une observation particulière ou à interroger des points de données ayant une signification particulière. Concrètement, un tag est un texte arbitraire (la *clé*), associé à une valeur textuelle arbitraire (la *valeur*). Il y a deux types de tags :

- les tags dynamiques : ils sont directement attachés à un point de données,
- les tags statiques : ils sont attachés à une métrique et sont automatiquement hérités de tous ses points de données.



- **Les tagsets**

L'ensemble de tags associé à un point de données est appelé *tagset*. Un tagset combiné à une métrique constitue une série chronologique dans TStorage. Il existe deux manières typiques d'utiliser les tagsets :

- 1) Une métrique identifie un capteur (au sens large du terme), tandis que les tags clarifient le statut du capteur (nom du fabricant, position géographique, etc.) ou certains des points de données générés (qualité de la donnée, état de fonctionnement, etc.).
- 2) La métrique fait référence à une propriété d'intérêt (utilisation du processeur, par exemple), tandis que les tags font référence à l'élément à l'origine des points de données (serveur numéro 5, par exemple).

## Fiche **TSorage**- Une plateforme de gestion de données (I)IoT open source

Bien que mélanger ces deux approches soit techniquement possible, nous vous recommandons de choisir l'une d'entre elles et de vous y tenir.

Nous vous conseillons d'utiliser les tagsets pour appliquer une politique de sémantique, c'est-à-dire une ontologie. Au lieu de placer des informations dans le nom de la métrique, en utilisant certaines conventions spécifiques à l'entreprise qui sont peu respectées et qui souffrent donc de nombreuses exceptions, le nom de la métrique peut être dénué de sens et la sémantique peut être explicitée à l'aide du tagset.

- **Organisation hiérarchique**

Veillez noter que rien ne vous empêche d'organiser les métriques de manière hiérarchique, comme c'est souvent le cas avec les réseaux de capteurs industriels, car les clés de tag peuvent être utilisées pour définir les niveaux hiérarchiques de l'organisation (Vous en découvrirez davantage via la [documentation officielle](#) ou via cet [article de blog](#)).

Les séries chronologiques sont organisées de manière hiérarchique : les points de données et les valeurs agrégées des différentes séries chronologiques ayant la même métrique peuvent être combinés afin de produire une série chronologique plus générique. Bien que les technologies impliquées dans TStorage sont assez efficaces et passent bien à l'échelle, sachez que l'extraction et la fusion de plusieurs séries chronologiques peuvent mener à une consommation importante des ressources disponibles. Par conséquent, le nombre de séries chronologiques qui doivent être fusionnées pour satisfaire une requête de données doit rester raisonnable afin de maintenir la pression sur la base de données à un niveau acceptable.



- **Les messages**

Afin d'offrir de meilleures performances, les communications TStorage sont basées sur le concept de *message*. Un message est essentiellement un ensemble de points de données liés à la même série chronologique. En d'autres termes, les messages sont un moyen de soumettre plusieurs points de données à la fois, tout en ne soumettant le nom de la métrique, le tagset dynamique et le type de données qu'une fois par message. Chaque message doit contenir les éléments suivants :

- *metric* : l'identifiant de la métrique pour laquelle de nouveaux points de données sont fournis.
- *tagset* : l'ensemble de tags dynamiques associés à tous les points de données décrits dans le message.



## Fiche TStorage- Une plateforme de gestion de données (I)IoT open source

- *type* : le type de tous les points de données décrits dans le message. Bien que l'utilisation du même type pour tous les points de données relatifs à une métrique soit généralement considérée comme une bonne pratique, le type associé à une métrique (ou à une série chronologique) peut changer d'un message à un autre.
- *values* : une liste de points de données. Chaque point de données est constitué de deux éléments : une représentation de l'horodatage associé au point de données et la valeur du point de données.

D'un point de vue technique, un message est représenté par un objet JSON (il peut également être représenté par un message Protobuf, cliquez [ici](#) pour plus d'infos).

Le schéma JSON d'un message TStorage est le suivant :

```
{
  "$id": "be.cetic.tsorage.messageschema.json",
  "type": "object",
  "properties": {
    "metric": {
      "type": "string"
    },
    "tagset": {
      "type": "object",
      "additionalProperties": {
        "type": "string"
      }
    },
    "type": {
      "type": "string"
    },
    "values": {
      "type": "array",
      "items": [
        {
          "type": "array",
          "items": [
            {
              "type": "string",
              "pattern": "^(?! (?:(?:[1-9][0-9]*)?[0-9]{4})-(1[0-2]|0[1-9])-(3[01]|0[1-9]|[12][0-9])T(2[0-3]|[01][0-9]):(?:[0-5][0-9]):(?:[0-5][0-9])(\\.([0-9]+)?(\\.[0-9])?(1,3))?$"
            },
            {}
          ]
        }
      ]
    }
  },
  "required": [
    "metric",
    "type",
    "values"
  ]
}
```

## Fiche **TSorage**- Une plateforme de gestion de données (I)IoT open source

Comme illustré ci-dessus, l'attribut *tagset* doit être un dictionnaire de chaînes de caractères. Chaque valeur (qui représente un point de données) est un tableau contenant l'horodatage et la valeur du point de donnée, dans cet ordre. L'horodatage est représenté par une chaîne de caractères au format ISO 8601.

La valeur elle-même peut être n'importe quel objet JSON valide. Son schéma réel dépend du type de donnée spécifié. Il existe plusieurs types de données proposés d'office par TStorage, et des types de données supplémentaires arbitrairement complexes peuvent être ajoutés à volonté.

L'extrait ci-dessous est un exemple de message valide décrit à l'aide du format JSON :

```
{
  "metric": "my-temperature-sensor",
  "tagset": {
    "quality": "good",
    "owner": "myself"
  },
  "type": "tdouble",
  "values": [
    [ "2020-01-02T03:04:05.678", 42.1337 ],
    [ "2020-01-02T03:04:06.123", 654.72 ]
  ]
}
```



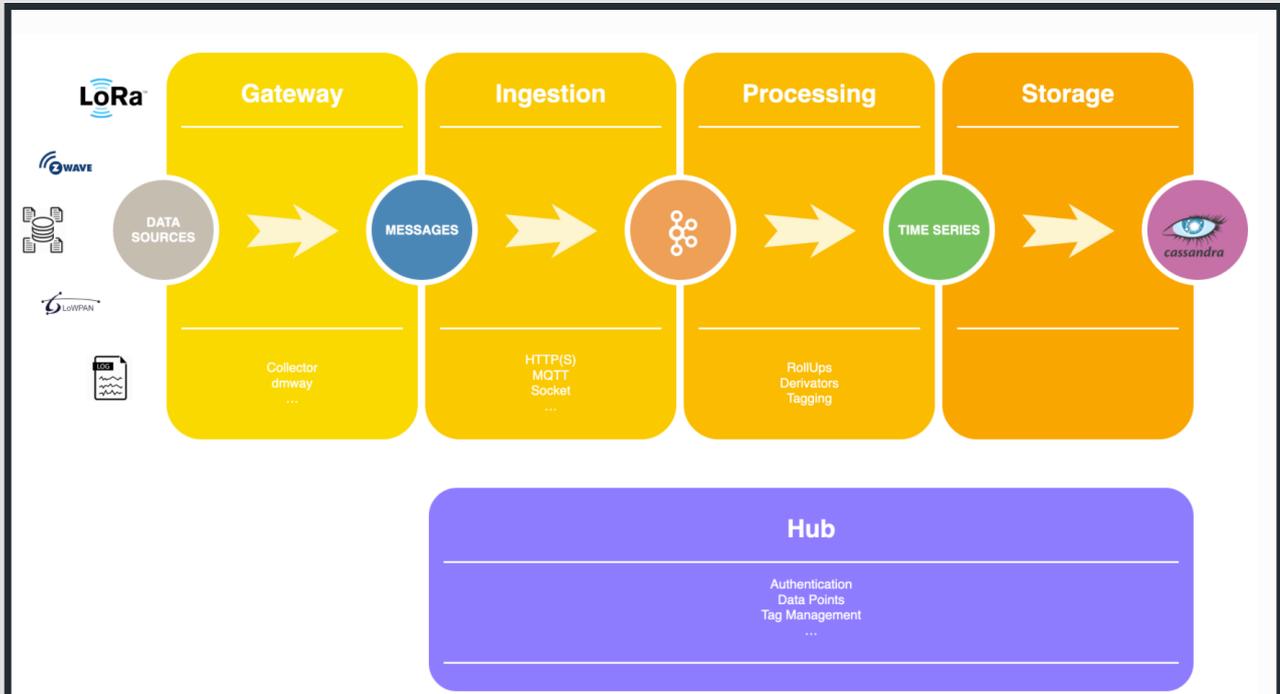
### Architecture

Le projet TStorage est basé sur une architecture modulaire, tous les modules étant conçus pour être exécutés dans des conteneurs Docker distincts. Cela fait de TStorage une solution portable, avec des étapes de déploiement simples et standardisées. Il offre également la possibilité de placer les composants sur différentes machines physiques et virtuelles, le rendant disponible sur une large gamme de plateformes et de services.

De plus, le (re)dimensionnement d'une architecture conteneurisée est plus facile, puisqu'un composant peut être déplacé vers une plateforme offrant plus de ressources. Sous certaines conditions, les conteneurs peuvent être dupliqués afin d'augmenter les performances des modules sous-jacents.

Fiche TStorage- Une plateforme de gestion de données (I)IoT open source

La figure ci-dessous donne un aperçu de l'architecture TStorage.

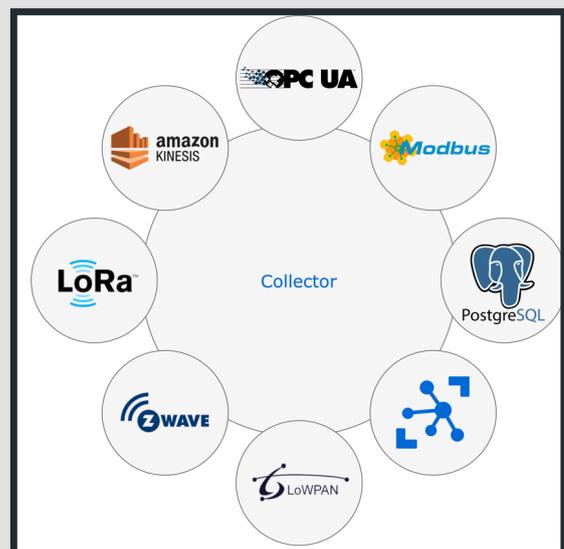


1 La passerelle (Gateway)

Le traitement des séries temporelles commence par la couche Gateway, qui contient des composants ad hoc pour collecter ou extraire des valeurs de séries chronologiques à partir de diverses sources de données.

Cette couche prend en charge diverses technologies de communication industrielles, y compris Modbus, OPC-UA et MQTT.

Les bases de données distantes peuvent également être consultées pour la collecte de données historiques. Les valeurs de séries chronologiques collectées sont standardisées et temporairement stockées dans un tampon avant d'être finalement transmises à la couche d'ingestion à l'aide des protocoles HTTP ou MQTT.



Fiche **TSorage**- Une plateforme de gestion de données (I)IoT open source

## 2 La couche d'ingestion

Le système d'ingestion des données est polyvalent et se base sur des technologies standardisées (HTTP, MQTT, etc.) afin de faciliter l'intégration de TStorage dans votre système informatique! Cette couche est celle du point d'entrée des valeurs de séries chronologiques : la conformité des messages reçus est vérifiée, les messages non autorisés sont rejetés. À partir de là, les entités décrites sont considérées comme des composants internes de TStorage, ce qui signifie essentiellement qu'elles sont gérées par le cluster TStorage. La couche d'ingestion est composée de différents modules d'interface, chacun d'entre eux fournissant un moyen spécifique pour une source de données de soumettre de nouvelles valeurs de séries chronologiques. Les messages acceptés sont poussés vers un topic Apache Kafka<sup>12</sup>, qui agit comme une file d'attente de messages pour les services internes de traitement de TStorage.

## 3 La couche de traitement (processing)

A ce stade, un ensemble d'applications de traitement gère le flux de messages Kafka. C'est le langage de requêtes de séries chronologiques de Prometheus, PromQL, qui a été implémenté afin d'exploiter les données collectées. Ce langage permet de réaliser des agrégations temporelles, des combinaisons de valeurs, et de filtrer des données tout en prenant en compte la sémantique associée aux capteurs." Il permet d'effectuer des agrégations et des transformations de données en temps réel.

- Les rollups de données (data rollups) sont des agrégations des valeurs au fil du temps. De cette façon, des analyses de haut niveau couvrant de longues périodes peuvent être effectuées plus facilement. Les rollups facilitent également l'analyse de séries chronologiques, car leurs valeurs sont alignées chronologiquement.

- Les transformations de données sont les modifications des valeurs, des noms et des tags des séries chronologiques, en temps réel et selon des fonctions métier, afin d'enrichir les informations traitées.

## Fiche **TSorage**- Une plateforme de gestion de données (I)IoT open source

### 4 La couche de stockage (storage)

Apache Cassandra est utilisé pour stocker des valeurs de séries chronologiques de manière permanente. Cette base de données distribuée et décentralisée offre une évolutivité linéaire, tandis que la réplication automatique des données entre différents sites permet à la solution de rester opérationnelle et de se remettre automatiquement de la défaillance d'un nœud ou même d'un data center entier. De cette façon, la pérennité, la survie et la disponibilité des données sont assurées. TStorage offre également une résilience aux pannes grâce à un mécanisme qui assure la migration automatique des différentes fonctionnalités de la solution d'un serveur défaillant vers un autre fonctionnel, garantissant ainsi une haute disponibilité des différents services.

Le modèle de données conçu respecte la nature des données manipulées et des requêtes typiques qui se rapportent à une série chronologique ainsi qu'à un intervalle de temps précis. En plus des valeurs de séries chronologiques, la base de données stocke également les métadonnées qui aident à atteindre plus efficacement les données souhaitées. Par exemple, une liste de toutes les partitions associées à une métrique particulière, et ayant un tagset particulier, est tenue à jour afin d'améliorer le temps de réponses des requêtes.

### 5 La couche hub

Il s'agit des services qui aident l'utilisateur à interagir avec les autres couches. Les services déployés dépendent des besoins de l'utilisateur mais ils incluent généralement la gestion des tags, le requêtage de données et les représentations de ces dernières sous forme de tableaux de bord. Grafana est l'outil proposé pour concevoir et déployer facilement des tableaux de bord, alimentés par les valeurs de séries temporelles ingérées, sous la forme de visualisation SCADA. La couche Hub offre une implémentation de PromQL, le langage de requête de séries chronologiques de Prometheus. Les outils d'analyse et de visualisation supportant ce langage s'intègrent donc aisément avec TStorage.

Fiche **TSorage**- Une plateforme de gestion de données (I)IoT open source



Un exemple de cas d'étude

Le CETIC a évalué TStorage dans le cadre du projet de recherche industrielle SW-ARTEMTEC. L'objectif était de fournir des outils analytiques modernes et avancés issus des technologies liées au big data et à la réalité augmentée afin d'améliorer la maintenance des sites industriels répartis géographiquement dans le monde entier.

Le partenaire du CETIC, Safran Aero Boosters (SAB), a apporté des flux données provenant de divers capteurs mesurant l'activité d'équipement de test aérospatial. Ces flux de données ont été enregistrés et traités avec TStorage afin de détecter toute anomalie. Le résultat attendu de ce projet est le développement de services innovants répondant aux attentes du marché de SAB.



Pour aller plus loin...

La plateforme TStorage a été développée par le CETIC dans le cadre de l'Industrie 4.0, avec l'ambition de développer un portefeuille d'outils de gestion de données modernes et conformes aux enjeux introduits par l'Internet Industriels des Objets.

Vous trouverez la documentation technique ici, n'hésitez pas à contacter le CETIC pour plus d'informations.

Vous pouvez également générer des séries temporelles paramétrables grâce à TSimulus pour ensuite les stocker et les visualiser grâce à TStorage. Notez que TStorage peut aussi être déployé à l'aide de FADI, une plateforme "cloud native" dédiée au Big Data (une fiche sur celle-ci est prévue pour bientôt!)