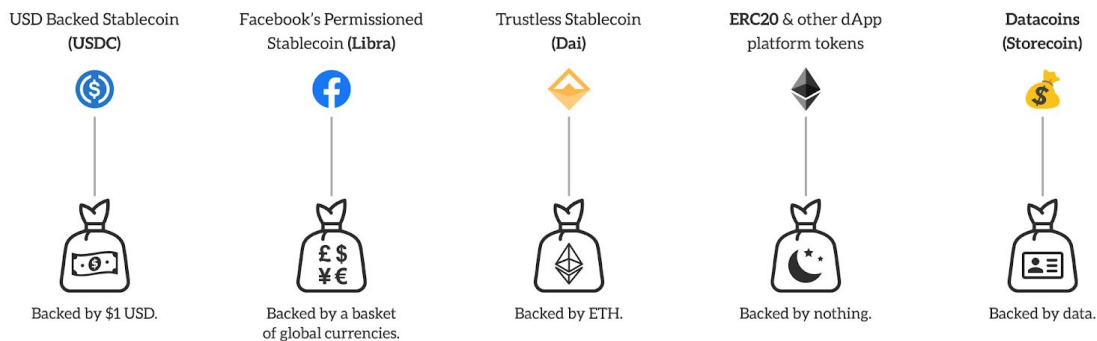


# Datacoins 101

This document answers frequently asked questions about Storecoin's *datacoins*. It discusses what datacoins are, why they exist, how they are issued, and how they are used to transform data into programmable money.

## What are datacoins?

### How datacoins — data-backed cryptocurrency — compare to other tokens



**Datacoins are backed by data, a valuable asset – not by dreamcoins.**

STORE is the unit of account for datacoins making datacoins fungible.  
Datacoins will be a new digital asset issued on top of the Storecoin blockchain.

Datacoins are tokens that represent **units of data** created by data-rich apps hosted on the Storecoin platform. These apps are also called **tokenized apps** (tApps) because they **tokenize their data** into datacoins. For every MB of data created in a tokenized app, 1 datacoin is mined by that app.

**1 MB of data = 1 datacoin**

## Are datacoins premined or mined on a continuous basis?

Datacoins are mined based on the units of data created in a tokenized app, so they are mined on a continuous basis. For example, if a tokenized app creates 1 GB of data on a given day, the app mines 1,000 datacoins on that day. The number of datacoins mined may vary widely on a day-to-day basis, depending on the volume of data created.

## Can an app premine datacoins?

Storecoin platform is designed to support real world, data-rich apps. If an existing app migrates onto the Storecoin platform, it may bring its existing data also on day 1. So, datacoins are mined as soon as the protocol recognizes that the data has been created. For example, if an existing app migrates with 1 TB of data, 1,000,000 datacoins are mined on day 1.

## Do datacoins represent *all* tokenized apps hosted on Storecoin?

No. Datacoin is a noun similar to ERC20 tokens on Ethereum. Each tokenized app hosted on the Storecoin platform issues its own datacoins. For example, an IoT app that monitors activities around volcanoes and produces live data feed will issue its own datacoins (say, ‘volcanocoin’<sup>1</sup>) compared to another AI-app that predicts traffic congestion based on real time traffic data, which will issue its own datacoins (say, ‘trafficcoin’). On an average, volcanocoin app may produce 500MB of data on a given day, so it mines 500 volcanocoins daily. The trafficcoin app, on the other hand, may produce 1 TB of data daily, so it mines 1M trafficcoins daily.

## Do all datacoins from *all* tokenized apps have the same *value*?

No. Since no data is created equal, the value of datacoins issued by each tokenized app will be different. In the above example, the value of 1 volcanocoin may be very different from the value of 1 trafficcoin.

## How do datacoins compare to Ethereum’s ERC20 tokens?

Datacoins have some similarities to ERC20 tokens, but they are different in many ways.

### Similarities:

---

<sup>1</sup> Don’t take the coin names too seriously! They are used here for illustration purposes only.

1. Both are layer 2 tokens.
2. Each tokenized app issues its own datacoins similar to each app on Ethereum issues its own ERC20 token.

#### Differences:

1. **Datacoins are backed by data** created in the tokenized apps. ERC 20 tokens are purely speculative.
2. Datacoins are **mined on a continuous basis** as and when the data is created. ERC20 tokens are created with an initial supply at the launch.
3. **Datacoins are accepted as a compensation by Storecoin miners** if they choose to provide **zero-fee p2p compute resources** to app developers (more on this later.) Ethereum miners only accept ETH, its layer-1 currency.
4. Because datacoins back data, **they have intrinsic value**, if the data is valuable and in demand. As already mentioned above, ERC 20 tokens are purely speculative.
5. Datacoins back valuable resources (data), derive their value from the backed resources, and used to gain access rights to the backed resources.

#### What is the denomination in which datacoins are valued?

**\$STORE** is the **unit of account** on the Storecoin platform. This means, all datacoins are valued and priced in \$STORE. Just like any commodity, the value of data in a tokenized app may fluctuate based on demand and supply, so the value of a datacoin will fluctuate based on the value of the data. 1 volcanocoin may be equal to 2 \$STORE and 1 trafficcoin may be equal to 0.1 \$STORE.

#### What if a tokenized app creates data of different value?

It is possible that some data created by a tokenized app are more valuable than others. For example, the volcano data near an active volcano may be more valuable than the data near a dormant volcano. This need is addressed by **data classification**. The tokenized apps are free to classify their data into multiple *value* buckets and charge each class of data differently. For example, the volcanocoin app may have 3 classes of data, which are valued as follows.

1. Class 1 — 1 MB of class 1 data = 2.25 volcanocoins.
2. Class 2 — 1 MB of class 2 data = 0.5 volcanocoins.
3. Class 3 — 1 MB of class 3 data = 0.25 volcanocoins.

If a buyer wants only class 1 data, the buyer pays 2.25 volcanocoins for every MB of class 1 data.

## How do data classes affect 1 MB of data = 1 datacoin?

**1 MB of data = 1 datacoin** defines how datacoins are mined and how they are paid for in bulk, without requesting a specific class of data. However, if the request is made for a specific class of data as described above, the price per MB of data will be different from 1 datacoin. In the volcanocoin app example above, we can observe the following.

1. 75% of the value of data — 2.25  $\div$  a total of 3 volcanocoins — comes from  $\frac{1}{3}$  of class 1 data
2. 16% of the value of data — 0.5  $\div$  a total of 3 volcanocoins — comes from  $\frac{1}{3}$  of class 2 data, and
3. 8% of the value of data — 0.25  $\div$  a total of 3 volcanocoins — comes from  $\frac{1}{3}$  of class 3 data.

Put simply, if a buyer buys 3 MB of data with 3 volcanocoins, the buyer gets approximately 1 MB of each class of data. However, if the buyer specifically requests class 3 data, the same 3 volcanocoins will buy 12 MB of class 3 data.

## How do app developers decide what their value is worth?

Valuing data is outside the scope of datacoins. There are economic, market, and other forces involved in pricing data. But the following general guidelines apply.

1. Estimate the amount of annual data created by the tokenized app. For example, the volcanocoin app produces 500 MB x 365 = 182,500 MB of annual data. This also means, this app will mine 182,500 volcanocoins annually.
2. Estimate classes of data and their percentage of total data produced. For the volcanocoin app, we assumed 3 classes of data with approximately equal proportions.
3. Estimate demand for each class of data to arrive at the value for each class of data. In the above example, class 1 data is estimated to have 75% of the total value of the data created by volcanocoin app.
4. Determine if the data is sold with exclusive access rights (a specific data set is sold to only one buyer — discussed later) or with shared access. This determines the

potential annual revenue from the data. Let's assume volcanocoin app estimates the annual revenue to be \$5M.

5. Estimate the price for the datacoin based on the estimated revenue and the current price of \$STORE. For example, if the current price of \$STORE = \$1.0, 182,500 volcanocoins issued annually are expected to be worth 5M \$STORE, so the price per volcanocoin is  $5M / 182,500 = 27.39$  \$STORE.

The following verifies the numbers for the volcanocoin app. The app produces 3 classes of data of approximately the same volume. So, each class is approximately 60,833 MB of data. The app estimates that 75% of its revenue would come from class 1 data and the remaining 25% from class 2 and class 3 data.

$$(60,833.33 * 2.25 * 27.39) + (60,833.33 * 0.5 * 27.39) + (60,833.33 * 0.25 * 27.39) = \sim 5.0 \text{ M.}$$

## How exactly does mining the datacoins work?

The process of mining datacoins is a shared responsibility between the tokenized app and the Storecoin platform. When a tokenized app launches on Storecoin, it submits details about its data schema, the estimates as discussed in the previous question, and other related metadata about the app. These details are submitted in a smart-contract, so both the tokenized app and the Storecoin platform are contractually bound by the details submitted in the contract. At runtime, the tokenized app takes the responsibility of creating schema-validated data and the Storecoin platform takes the responsibility of validating the schema conformance and issuing app's datacoins for every 1 MB of schema-enforced data created. Datacoin mining is an automated process once the contract is in place.

## Who *owns* mined datacoins?

The datacoins mined are **deposited into a contract address**, owned by the tokenized app. Any trading on datacoins is facilitated via the contract address. App developers of the tokenized app do not have **direct access to datacoins** mined by the app. In other words, they **can't withdraw or sell datacoins** from the contract address. They have to **first recognize datacoins as their revenue** before they are allowed access to them.

## How does a buyer decide what to buy and how to buy?

The Storecoin platform maintains certain metadata about the data created by a tokenized app. The metadata includes:

1. **Category of data** — data from different tokenized apps are categorized into well published categories. Data buyers can query the categories of data on the Storecoin platform, followed by tokenized apps that contain data for a given category.
2. **Classes of data** — Once the data buyer narrows down to tokenized apps that provide the categories of data they are looking for, the classes of data created in those apps can be queried.
3. **Prices for classes of data** — The metadata about classes also contains the price per MB for each class of data.
4. **Access restrictions on data** — The metadata about classes contains how the data can be accessed. The following access rights can be specified:
  - a. **Exclusive rights for purchased data** — The buyer has exclusive rights to the data they purchase. The tokenized app doesn't sell the same data to another buyer.
  - b. **Shared rights for purchased data** — The data may be accessed by multiple buyers.
  - c. **Exclusive rights for *all* data** — The buyer may acquire exclusive access to both existing and **future data** of a particular class of data.

The price of data depends on the type of access the buyer wants.

The metadata are derived from the smart-contract proposal submitted by the tokenized apps when they are launched on the Storecoin platform. The metadata helps with **data discovery** on the Storecoin platform.

## How does data discovery work?

The Storecoin platform publishes APIs and client libraries for discovering data on the platform. The metadata discussed in the previous question power data discovery APIs. A typical data discovery flow is as follows.

1. Data buyer queries for *data categories* on the Storecoin platform. The data discovery APIs return the available categories and tokenized apps servicing those categories.
2. Data buyer queries for *data classes* of one or more tokenized apps returned in step 1 above. Data classes with price in respective datacoins and access restrictions, if any, are returned. Data classes also include the data schema and sample data to help buyers prepare queries for the data.

3. Data buyer uses the data schema and sample data to prepare queries and runs them. A response containing the size of the data, classes of data satisfying the query, price per class of data, and options to purchase the data are returned. The buyer can refine the queries to buy just a specific subset of data or the entire firehose of data. Note that the data itself is not returned because the buyer hasn't purchased the rights to access the results yet.
4. Once the buyer is satisfied with the query, they use that query to purchase the rights to access the data. This ensures that the buyer gets what they want when the purchase is completed. The purchase flow is discussed next.

Data discovery is free.

## How does the buyer purchase data?

As discussed in the previous question, the buyer prepares a query to narrow down the data they want to purchase. The query result contains the size of data and the price in app's datacoins to purchase that data. The buyer purchases access to the data by paying in app's datacoins. The purchase flow is as follows.

1. The buyer uses a decentralized exchange (DEX), powered by Storecoin APIs to purchase app's datacoins. For example, the buyer wants to buy 10 volcanocoins.
2. Since datacoins are priced in \$STORE, the buyer pays for datacoins in \$STORE. The exchanges may allow purchasing datacoins in fiat currencies also. In this example, the buyer pays  $10 \times 27.39$  \$STORE (or \$273.9, if the DEX supports it) to purchase 10 volcanocoins.
3. Datacoins are purchased from the tokenized app's contract address and deposited into the buyer's wallet.
4. A smart-contract is produced alongside that contains sufficient details for the buyer to exercise the rights to access the data purchased. The contract may have a deadline to ensure that any change in the value of datacoins or \$STORE doesn't affect the amount and quality of data promised to the buyer. The buyer must execute the contract within the stipulated deadline.

## How does the data purchase flow work?

The purchase smart-contract contains the original query created by the buyer where they decided what data to buy. When the buyer executes the purchase smart-contract, the query is re-run, but this time, with the necessary authorization to access the data. So, the

query returns or grants access to the actual data when the purchase smart-contract is executed.

The contract can be run any number of times before its deadline at which the access to data ceases to exist. A purchase contract may be perpetual, depending on the access rights requested.

## How does the purchase contract ensure unauthorized access?

The purchase smart-contract is tied to buyer's public key, so the buyer must sign it with their private key to activate and execute it. So, even if the contract reaches wrong hands, it cannot be executed without access to the buyer's private key.

## What happens to datacoins when purchase contract is executed?

When the data purchase smart-contract is executed for the very first time, the following happens to datacoins, depending on how data access rights are requested in the contract.

1. If **exclusive right** to the data is purchased, the datacoins used to purchase the data are burned permanently. In the volcanocoin app example above, if the buyer requested exclusive access to the data with 10 volcanocoins, the coins are never returned to the circulation again. The total volcanocoin supply is reduced by 10 with such an access request. The dataset served by the query in the purchase contract is protected from access from other purchase contracts to ensure exclusive access rights.
2. If **shared right** to the data is purchased, the datacoins are returned to the circulation. This allows future buyers to purchase the same (or different) data as the current buyer, who executed the purchase contract. When datacoins are returned to circulation, they are deposited into the app's contract address.

## What happens if the buyer doesn't buy the data at all?

The buyer may not execute the data purchase contract, either intentionally or unintentionally. The buyer may simply want to **hodl** datacoins or the buyer may have forgotten to exercise the purchase contract. Depending on the access rights requested while purchasing datacoins, the following will happen.



1. If the purchase contract has a deadline and the deadline expires, the contract is invalidated. The buyer now has only one option — sell datacoins in the open market, at the current market price. The buyer cannot access the data anymore.
2. If the purchase contract doesn't have a deadline, the buyer can execute it at anytime. This is the case where the buyer would have paid a premium to protect against datacoin price variations that affect the amount and type of data they can access.
3. The buyer may instead sell datacoins in the open market, at the current market price, while the purchase contract is still valid. When the datacoins are sold, the purchase contract is automatically invalidated.

### When do app developers recognize the *revenue* from datacoins?

Datacoins deposited into tokenized app's contract address turn into app developer's *revenue* when a buyer purchases datacoins. In the above example where a buyer purchased 10 volcanocoins, the developer of the volcanocoin app recognized 10 volcanocoins as their revenue. The revenue is recognized whether the buyer ends up executing their purchase contract or not.

### How does zero-fee p2p cloud compute work?

Storecoin miners may see certain tokenized apps as more profitable than others. So, they may choose to work with the app developers as early investors and fund their development work. The miners do so with *zero-fee* p2p cloud compute resources where the computational resources such as memory, CPU, storage, and bandwidth are provided at no upfront cost to developers. In exchange for providing zero-fee resources, the miners enter into a profit-sharing agreement with app developers. For example, miners and developers may agree for a 50% - 50% datacoin revenue sharing. In the above example where developers recognized 10 volcanocoins as their revenue, Storecoin miners earn 50% of it (5 volcanocoins) if they provided zero-fee resources to volcanocoin app developers. So, the app developers have a net revenue of 5 volcanocoins.

Note that all tokenized apps may not be eligible for zero-fee resources. In all other cases, app developers pay miners separately for using the computational resources. This is akin to app developers paying AWS to use their resources.

## How does Storecoin prevent a buyer from selling the data in the gray market?

Storecoin **cannot prevent** data reselling because the buyer can copy the data to their local infrastructure, repackage, and resell it. This is very similar to how products are sold in gray markets in other businesses such as electronic items, phones, computers, etc.

What Storecoin **guarantees** is **integrity of data** when it is accessed directly on the Storecoin platform. The data is guaranteed to have passed schema enforcement and other validations and it is guaranteed to be anonymized of any sensitive information. So, if data quality and such guarantees are critical, buyers will stick to legal access to data on the Storecoin platform.

## How is user privacy protected when potentially all types of data can be sold on Storecoin platform?

All data, including sensitive and private information can be sold on the Storecoin platform. So, where is the user privacy then? Tokenized apps are required to annotate sensitive and private data before submitting it to the platform for persistence and datacoin mining. The annotated segment is encrypted automatically at rest, so even an unauthorized access to the data doesn't leak sensitive information. When sensitive information is annotated, some additional metadata is created in the clear text form that describes the classification and categorization of the sensitive data. This metadata assists in data discovery and context-sensitive anonymization, when the data is eventually accessed. If the sensitive information happens to be included in the query executed by a purchase contract, the sensitive data is automatically anonymized with the information provided in the metadata. So, all queries always return **safe data**. This allows all types of data being traded without sacrificing user's privacy.

## What happens if a tokenized app cheats and doesn't annotate users' sensitive information?

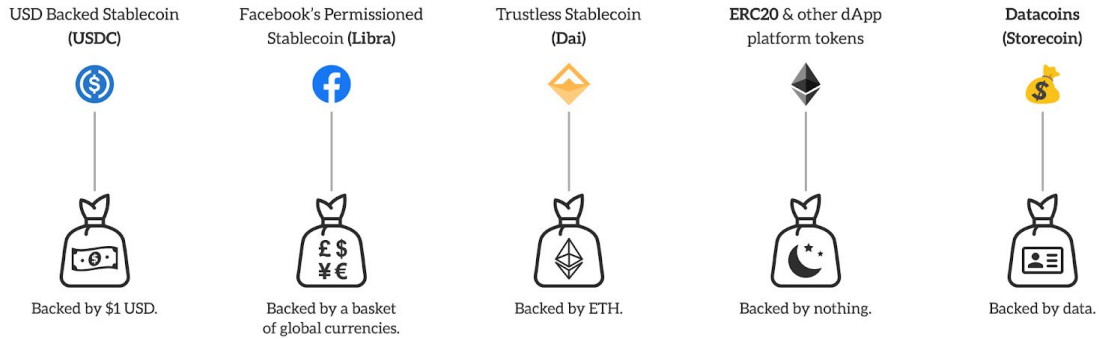
Cheating can happen in any environment and Storecoin is no exception. However, Storecoin's data discovery APIs also describe if tokenized apps annotate any sensitive information and the schema used to annotate sensitive information. This **transparency** is what forces app developers to be honest. If the users of a tokenized app know that sensitive and private information is collected, but don't see that information being

annotated, they know that the app is not treating data privacy as it should. The market decides what should happen to the data (and the resulting datacoins) of that tokenized app.

### **Last question — why do we need datacoins? Why can't buyers buy data with fiat currencies?**

Datacoins serve the same purpose as the tokens used in video arcades, points accumulated with airlines and hotels, and the shares issued by public companies — they allow for finer control on how the economy works within the system. The thesis behind datacoins is that they have intrinsic value if the underlying data is in demand. So, datacoins can be traded without every buyer actually buying data with them, much like how (almost all) ERC 20 tokens are traded purely based on what buyers think of the company, which issued those tokens. So, datacoins can help with valuing companies rather than just data. Lastly, 1 MB of data could be 1/1,000 of a penny in value for a tokenized app, which produces some streaming data where only the real-time data has any value. Allowing and managing access to such data is hard, if not impossible, with fiat currencies. ERC 20 specification proved that a token economy is possible. We are improving on it by saying the following:

## How datacoins — data-backed cryptocurrency — compare to other tokens

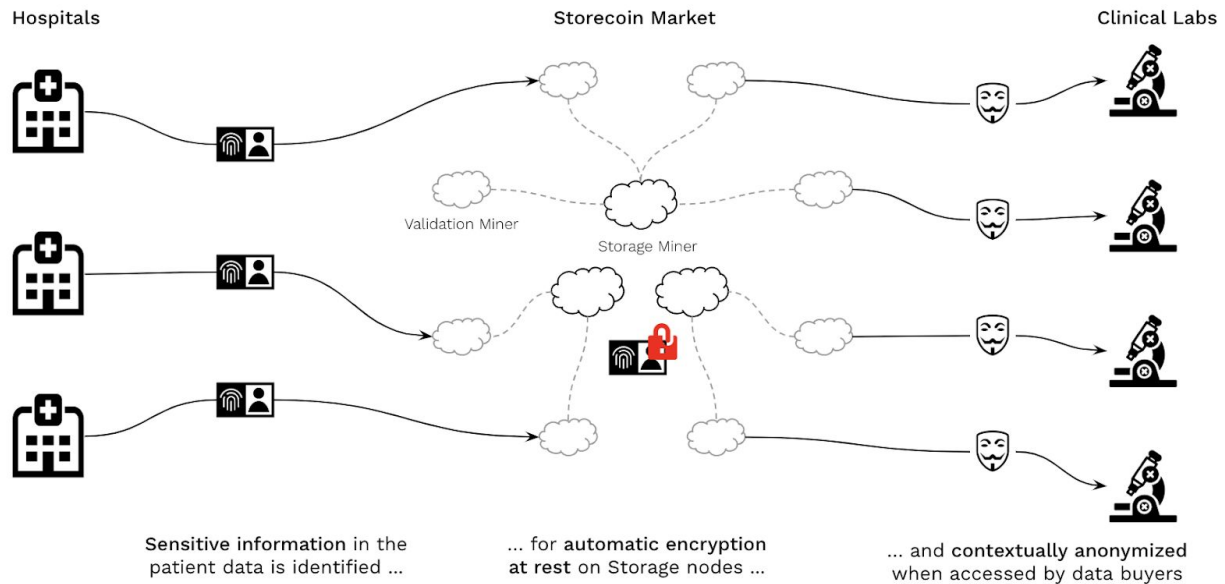


Datacoins are backed by data, a valuable asset – not by dreamcoins.

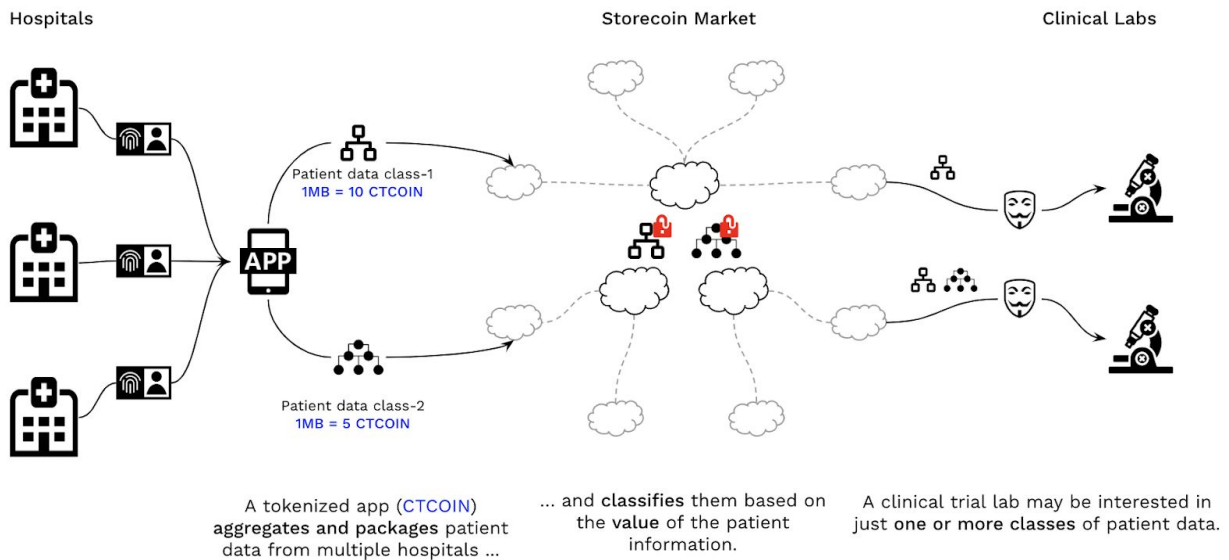
STORE is the unit of account for datacoins making datacoins fungible.  
Datacoins will be a new digital asset issued on top of the Storecoin blockchain.

## Use cases

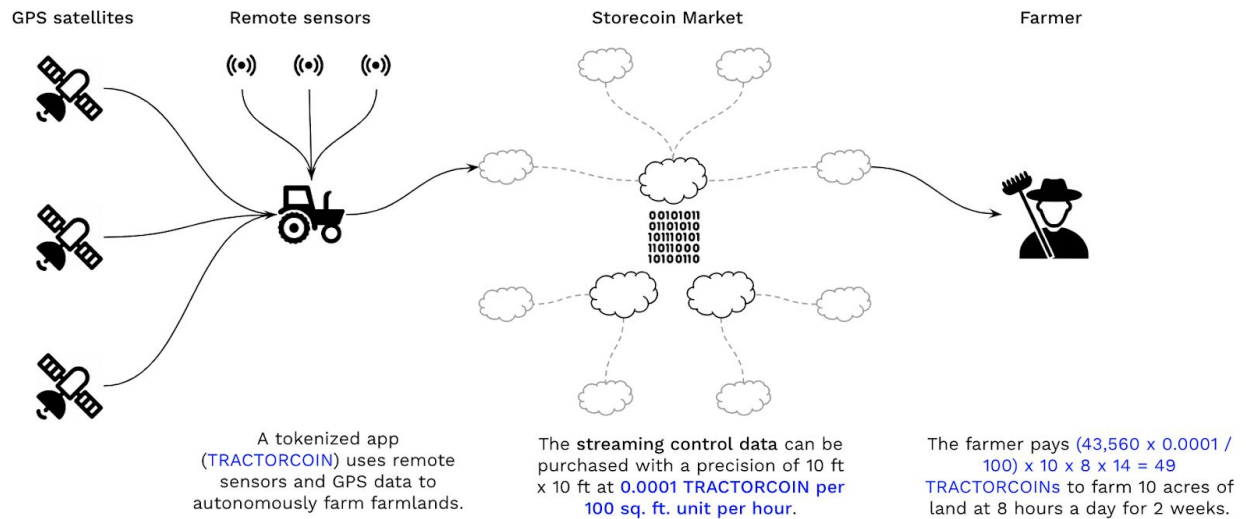
### Use case 1 — Privacy-preserving patient data in clinical trials



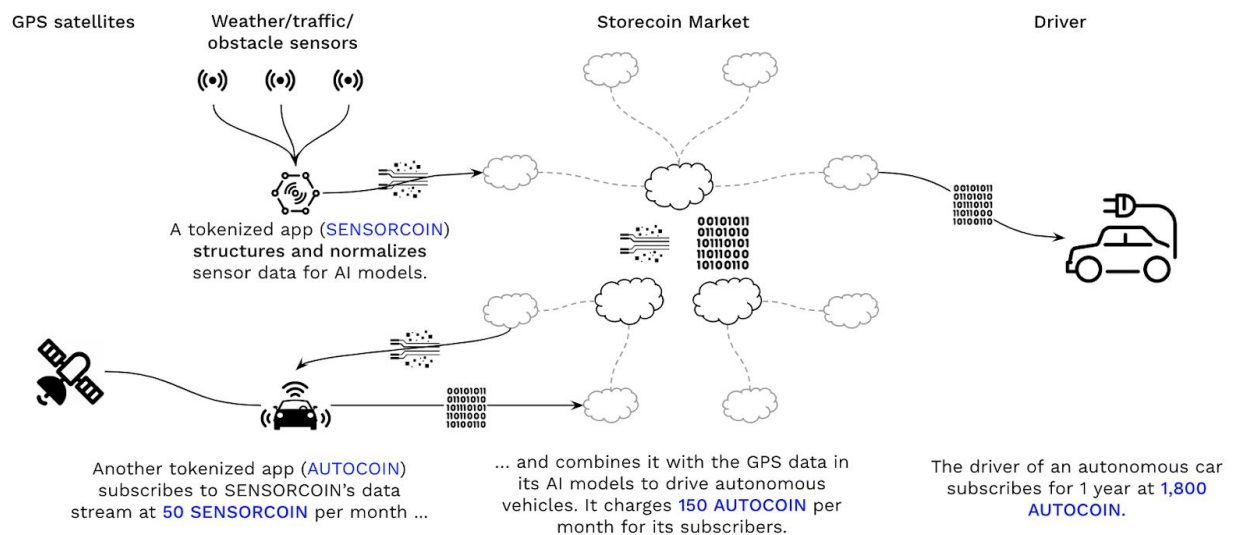
### Use case 1 — Datacoin use for the patient data in clinical trials



## Use case 2 — A farmer subscribing to autonomous farming

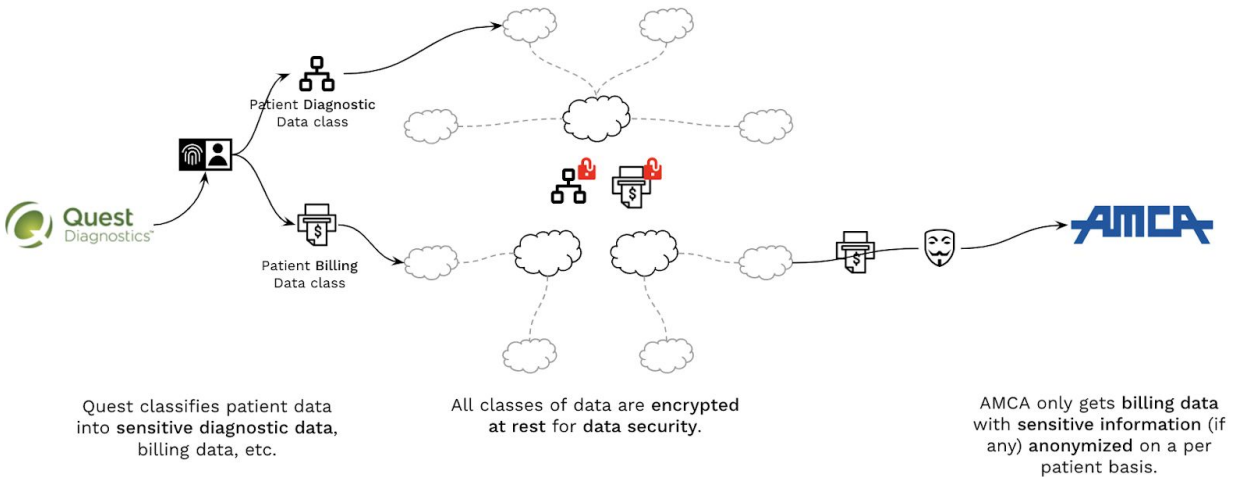


## Use case 3 — autonomous vehicles



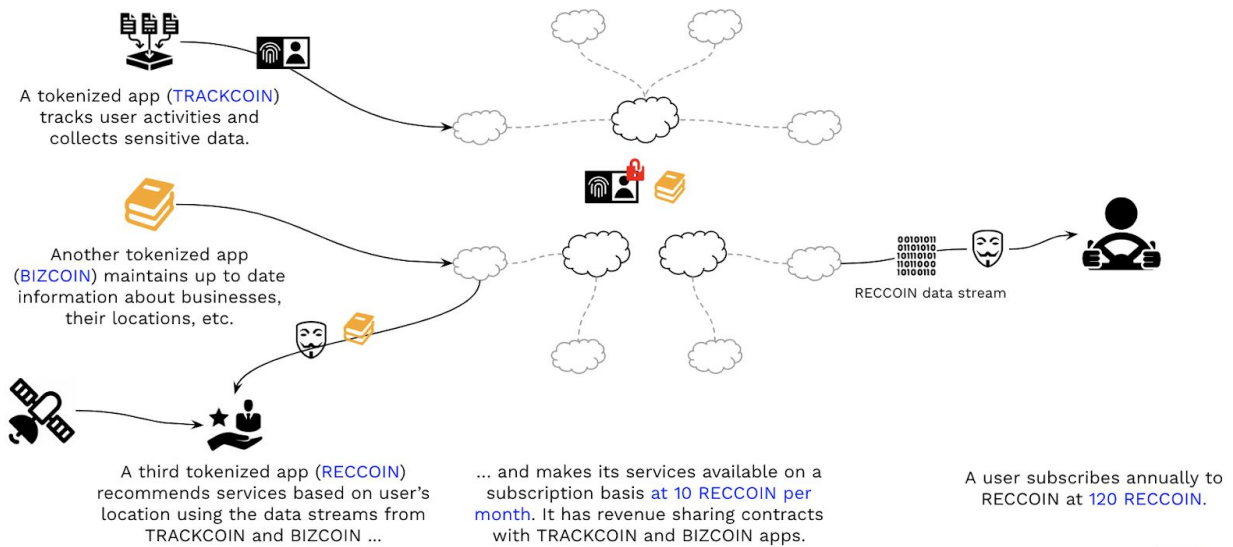
## Use case 4 — Prevent data leaks with partners and customers\*

\* Massive Quest Diagnostics data breach impacts 12 million patients



This use case doesn't use datacoins, but illustrates how sensitive information is stored and shared securely with partners and customers without leaking data when data leaves one infrastructure to another.

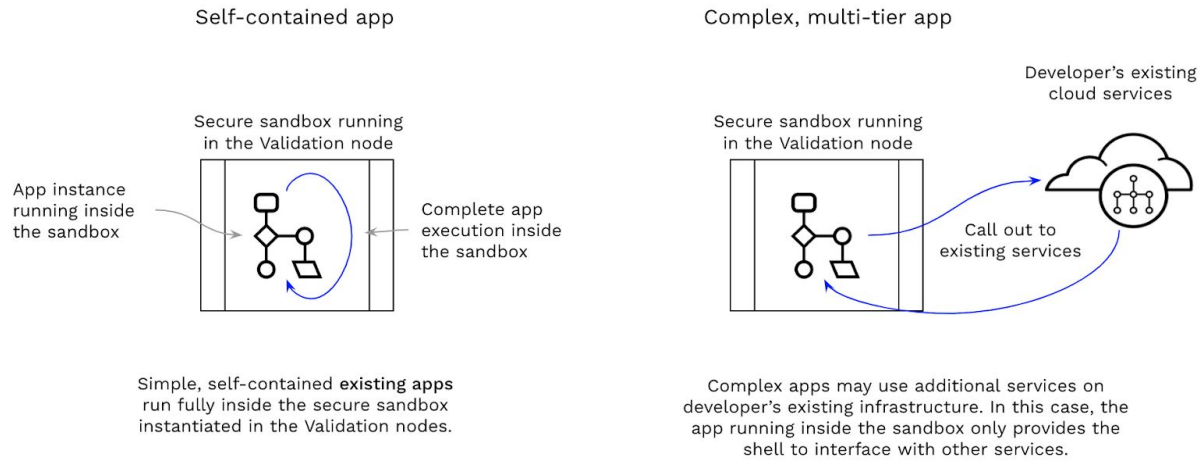
## Use case 5 — Prevent data leaks — ad industry example





## Use case 6 — Migration of an existing app onto Storecoin Platform

Fully on-chain app execution and app data storage may be impractical, especially when a large existing app is migrated to Storecoin Platform with complex app logic and petabytes of data.



## Use case 6 — Migration of an existing app — storage services

Storage can similarly be fully migrated onto Storecoin Platform for small apps or only metadata (*pointers*) to the actual data is stored on Storecoin platform for complex apps with petabytes of data.

