# **Data Types**

**DataItemMetadata** {

*id*: string,

*timestamp*: long,

*owner*: string,

*lat*: double,

*lon*: double

}

**DataItem** {

*metadata*: DataItemMetadata,

*content*: byte[]

}

**DataItemsContainer** {

*owner*: string,

*storageAddress*: string,

*dataItems*: list<DataItem>

}

**DataItemMetadatasContainer** {

*owner*: string,

*storageAddress*: string,

*dataItemMetadatas*: list<DataItemMetadata>

}

# **Process**

## Owner generates random data as a DataItemsContainer

* Input: owner, numItems, startTime, endTime
* Output: DataItemsContainer with random content
* Process:
  + Generate *numItems* data items:
    - Generate metadata:
      * owner = owner
      * timestamp = random timestamp between *startTime* and *endTime*
      * lat, lon = random value between a range (currently LA area)
    - content = random bytes
    - id = hex string representation without "0x" prefix of SHA256(content)

## Owner generates AES/GCM/NoPadding encryption key

## Owner encrypts the DataItemsContainer with AES

* Input: encryption key, DataItemsContainer
* Output: new container with the metadata (private data cleaned) and with encrypted data items
* Process:
  + For each data item:
    - Clone the data item
    - Clear private data (i.e. lat, lon) of the cloned item
    - itemAsBytes = serialization of the original item
    - encrypt itemAsBytes using the encryption key and metadata of cloned item as the associated data
    - set content of cloned item = encrypted data

## Owner submits encrypted data to the storage system

## Owner submit commitment for the encrypted data (as the hash of the encrypted data) to the blockchain

## Owner sends the metadata of **plain** DataItemsContainer to the Curator

## The curator uses the **plain** DataItemsContainer from owners to create a PlainIndex with all plain locations

## The curator submits the PlainIndex to the storage and submits the address to the blockchain

## The buyer gets the last seachable index of the (first) curator from blockchain and retrieves it from the storage

## The buyer finds the DataItemsContainer from the index that match his (currently random) query

## The buyer makes offers to the owner of the matched DataItemsContainer

## Owner sends the encryption key for the offer of the buyer

## The buyer retrieves the matched DataItemsContainer from the storage system

## The buyer decrypts the encrypted DataItemsContainer to retrieve data items