The **Data In The Open (dito):** is data structured in files, each is optionally encrypted for groups of readers (gpg --group), where the data is in the open (i.e. shared in public boxes/folders/drives/url) and each of its files is renamed and located by any field of its locator, being a record in the (originally named) locators.csv encrypted file

    being specific per root and additionally held by each of the participants (involved peer), while
    being synced/refreshed/notified upon its cid changes, where
    each of its locator records is defined such that
        nid (as in node id) is of 64 char of 32 digit pairs coded
            in base32 for index<1024 per each layer<31, as the last pair indicates the node's version;
        cid (as in content id) equals hash(content), optionally with some hashcash;
        name(data)=cid(data).type(data), as its original name is moved to its locator (for revalidating in
            the open as being unchanged and matching content by type from the node until its root);
        stand(data)=(nid(data), name(data));
        h(data) is the host function to download or upload the data, eg. dedicated 5G on gdrive; and
        locator(data) =(h(data),nid(data), cid(data), type(data), cid(stand(data)),
                original-name(data), age(data), list of signatures upon cid(stand(data))).

Dito use case: Assuming anonymous can only download or add files only to area under digesting dito node, such that the file is to be automatically removed when does not comply with specific conditions to be checked in specific period of time (such as per cycle, it must be younger than max age or permitted by...), such that when accepted the file can be moved to specific participant/s (optionally mapped to iiaom) to be sitting in the corresponding dito-node, where one or more of the participants is functioning as dito-gate-keepers optionally peeking data from the pool before it's expired.

   This, after acceptance by the dito-gate-keepers, can be stored additionally to log of distributed records (in messages and/or files), as dito tree of db>>table>>column>>sorted-field, where each node has name, type and description including how to use it.

  For to reduce spamming risk, the node owners can specify excepted formats up-to max size, lifetime etc and other conditions, with iiaom and public key etc.

The evolutionary naming in dito LogChain: being a p2p and communal bookkeeping of transactions and/or sharing information eg music, while using, instead of mining, signatures and optionally iiaom of authenticators. where
    authenticators are or representing those who made the transactions or bring the information and
    the cid(stand(data)) is digitally signed by, and optionally with the iiaom of, authenticators granting the missing coherence, which is missing from the regularly made activity to be by most of the peers, such as matching hashes of the origin until root, or checking balances/resources against statements etc.

Conclusion: Per each time-block consisting data and its locator, the longer the chain is, the harder the forgery in it, the maturer it's. The locator in size is small and in its name encapsulates, when conformation is reliable, the whole chain of blocks from the root until the current. When the conformation by peer/authenticators is given for the matching of the contents to their (hashed) names, it is given for the order from the root of the blocks with their timestamps or the match in of balance and movements and is not given for whether the data or its metadata tells true or not, only conforming the telling itself.

….and if all others accepted the lie which the party imposed – if all records told the same tale – then the lie passed into history and became truth. Do not let the **1984**'s party slogan becoming true: "who controls the past, controls the future: who controls the present controls the past."