

## 2.3.4.5 Data Lifecycles

[Return to Data Taxonomy](#)

### Overview

[Return to Top](#)

The **Data Lifecycle** covers the stages a particular piece of data transitions through from its initial generation or capture to its eventual archival and/or deletion at the end of its useful life. The [Data Management \(DM\)](#) is the coordination and administration of all data associated with a project, program or effort. Data Management includes the meta data as well as the individual pieces of data as it progresses through the Data Lifecycle. Data management follows a [Data Strategy](#) which also includes the required business rules, especially those captured in any governing Legal Documents such as the Charter, By-Laws, and policy and Procedures. Figure 1 reflects the major stages in the Data Lifecycle:

1. **Create** - The data is created, captured, copied from other sources
2. **Store** - The data is stored into a [Datastore](#) (i.e., [Database](#), [Document](#), Files, etc)
3. **Use** - The Data is actually used, accessed and referenced by an on-going business process. Often a [Data Management Platform \(DMP\)](#) is a non-datastore specific way to access data.
4. **Propagate** - The Data is copied to other data structures or to other nodes within the system. For example, servers, clients, tiers, and alternative formats (i.e., RDBMS to XML or JSON)
5. **Share** - The data is made public and can be shared with other internal or external business processes using any number of mechanisms such as

[HTTP\(S\)](#), [File Transfer Protocol \(FTP\)](#), [SMTP](#), [SMS](#), [IPFS](#), [DDS](#),  
[\]\]dido:public:ra:xapend:xapend.a\\_glossary:b:blockchain\]\]](#), [Distributed Ledger Technology \(DLT\)](#),  
 etc.

6. **Archive** - The data is preserved as part of the system legacy. In the past, the amount of data archived was limited primarily due to the cost of storage, but with the advent of inexpensive offline storage, most data is now preserved.
7. **Destroy** - The data is considered as having no value, become a liability or is required to be destroyed by law (i.e., The Right to Be Forgotten). In traditional data systems, the data is usually overwritten with newer, more german data. For example, old room temperatures are replaced with new ones if there is no requirement to archive the old temperature.

**Note:** Figure 1 also includes a **Plan** Stage, however, since the data does not exist during planning, it is not considered as an actual Data Stage, but this in no way means it is not an important stage for data. It is during this stage the requirements for the data are identified using system architecture and engineering and modeling. Planning usually includes [Use-Cases](#), [Prototypes](#), and the various [Data Models](#) (i.e., Conceptual, Logical and Physical), [legal considerations](#) and the pragmatics of things like the quantity and quality of data. Sometimes during this stage, the “planned” data never comes to fruition.

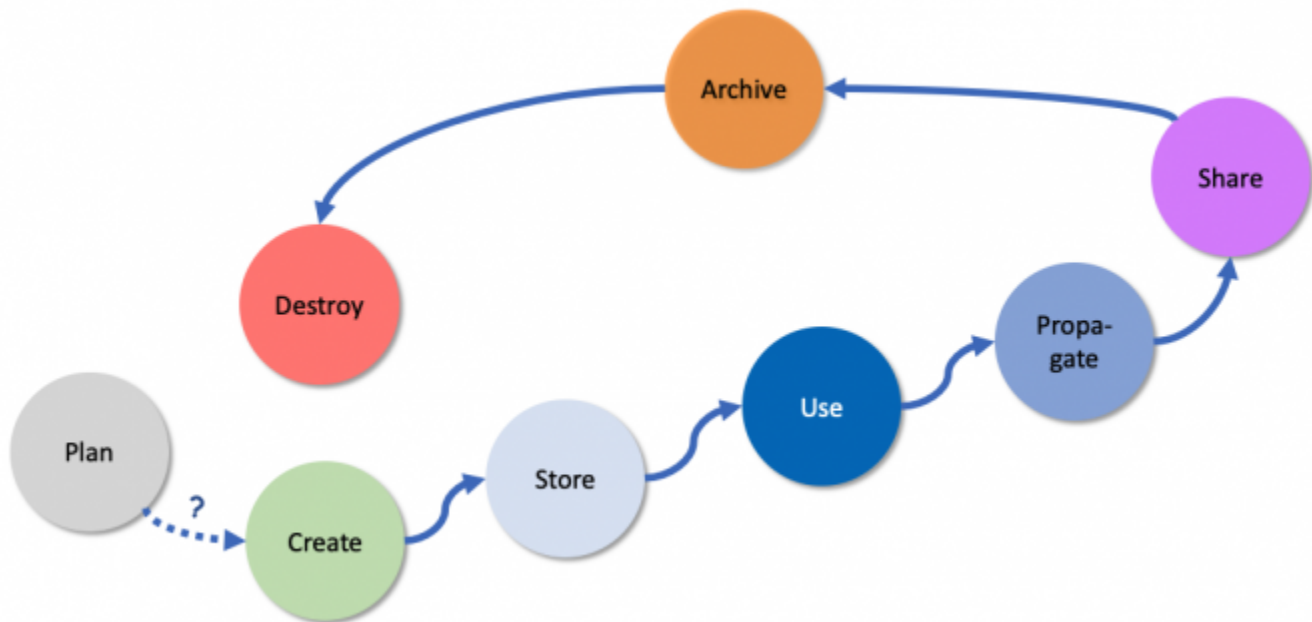


Figure 1: Traditional Data Lifecycle Stages

Wigmore<sup>1)</sup> defines only six stages for the Data Lifecycle:

1. **Generation or capture:** In this phase, data comes into an organization, usually through data entry, acquisition from an external source or signal reception, such as transmitted sensor data.
2. **Maintenance:** In this phase, data is processed prior to its use. The data may be subjected to processes such as integration, scrubbing and extract-transform-load (ETL).
3. **Active use:** In this phase, data is used to support the organization's objectives and operations.
4. **Publication:** In this phase, data isn't necessarily made available to the broader public but is just sent outside the organization. Publication may or may not be part of the life cycle for a particular unit of data.
5. **Archiving:** In this phase, data is removed from all active production environments. It is no longer processed, used or published but is stored in case it is needed again in the future.
6. **Purging:** In this phase, every copy of data is deleted. Typically, this is performed on data that is already archived.

**Note:** These six stages are roughly the same as the ones in Figure {ref>dataLifecycle}} with slightly different names. Wigmore's model combines the **Propagation** and **Sharing** stages into a single **Publication** stage.

**Note:** It is important to have a [Data Retention Policy](#) in place during all the stages of the Data Life Cycle. During **Planning**, knowing the Data Retention Policies governing the data can influence the kinds of information collected and how the data is stored. For example, partitioning the data by its data retention policy allows data with similar governing policies to be **Archived** and **Deleted** at the same time while retaining other less sensitive data.

## DIDO Specifics

[Return to Top](#)

The main differences between a generic Data Lifecycle and a DIDO Data Lifecycle is that in the idealized DIDO Data Lifecycle the **Destory** and **Archive** stages are non-existent or modified. See Figure 3. However, because the data within a DIDO is theoretically immutable and no data is ever lost, the panacea of “unlimited” data storage is being challenged. There are different ways that the size of the ledgers can be managed. One way is to have diferent kinds of nodes with each node containing differing amounts of data. See section [2.3.3 Node Taxonomy](#).

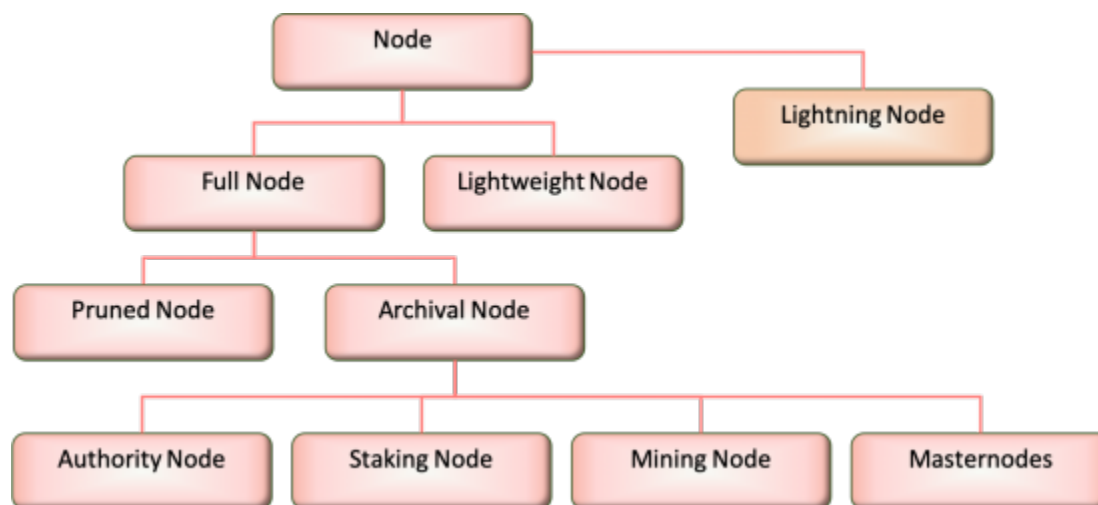


Figure 2: DIDO Node Taxonomy: Node Types

Another mechanism to overcome the “size” issue is to use [Sharding](#). Sharding splits a database (in a DIDO the Ledger) horizontally into [Shards](#) spreading the load across the many Shards. In an Ethereum context, Sharding is intended to reduce network congestion and increase transactions per second by creating new Shards referred to as **Chains**.

Iota is using a form of Sharding called Streams and has an [Request For Proposal \(RFP\)](#) at the [Object Management Group® \(OMG®\)](#) to provide a standardized specification for publishing, subscribing and validating the [transactions](#) on the “Tangle” (Iota's version of [Directed Acyclic Graph \(DAG\)](#)).<sup>2)</sup> <sup>3)</sup>

available and has further classified nodes into Full Nodes and Archival Nodes.

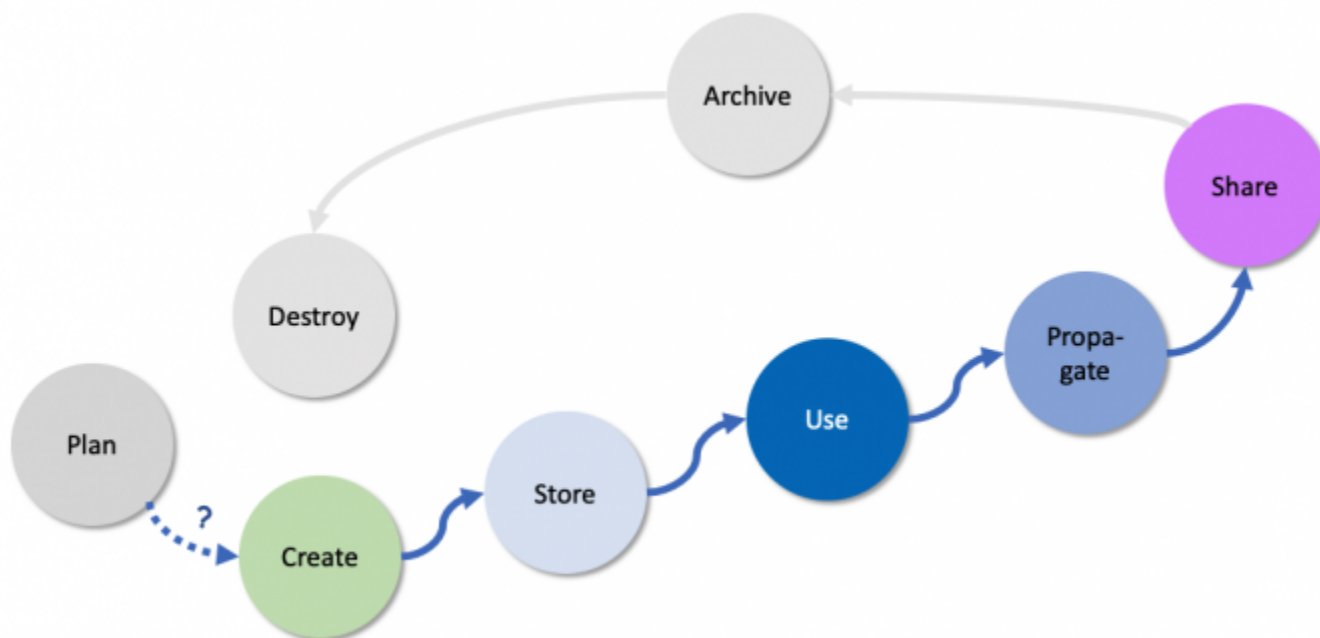


Figure 3: Immutable Data Lifecycle.

<sup>1)</sup>  
Ivy Wigmore, data Life Cycle, TechTarget, July 2017, Accessed: 12 October 2021,  
<https://whatis.techtarget.com/definition/data-life-cycle>

<sup>2)</sup>  
Object Management Group (OMG), Linked Encrypted Transaction Streams (LETS), Accessed: 17 October 2021, <https://www.omg.org/hot-topics/linked-encrypted-transaction-streams.htm>, and the RFP is here:  
<https://www.omg.org/cgi-bin/doc.cgi?mars/20-12-22>

<sup>3)</sup>  
Perry Cohen, Embedding Computing, OMG Issues RFP to Standardize Linked Encrypted Transaction Streams (LETS), January 2021, Accessed: 17 October 2021,  
<https://www.embeddedcomputing.com/technology/security/network-security/omg-issues-rfp-to-standardize-linked-encrypted-transaction-streams-lets>

From:  
<https://www.omgwiki.org/dido/> - DIDO Wiki

Permanent link:  
[https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:1.2\\_views:3\\_taxonomic:4\\_data\\_tax:05\\_lifecycle:start&rev=1634502608](https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:1.2_views:3_taxonomic:4_data_tax:05_lifecycle:start&rev=1634502608)

Last update: 2021/10/17 16:30

