

# Table of Contents

Single source of truth (SSOT)

.....

3



# Single source of truth (SSOT)

Single source of truth (SSOT) is the practice of structuring information models and associated data schema such that every data element is mastered (or edited) in only one place.

Snippet from [Wikipedia](#): [Single source of truth](#)

In information science and information technology, **single source of truth (SSOT)** architecture, or **single point of truth (SPOT)** architecture, for information systems is the practice of structuring information models and associated data schemas such that every data element is mastered (or edited) in only one place, providing data normalization to a canonical form (for example, in database normalization or content transclusion).

There are several scenarios with respect to copies and updates:

- The master data is never copied and instead only references to it are made; this means that all reads and updates go directly to the SSOT.
- The master data is copied but the copies are only read and only the master data is updated; if requests to read data are only made on copies, this is an instance of CQRS.
- The master data is copied and the copies are updated; this needs a reconciliation mechanism when there are concurrent updates.
  - Updates on copies can be thrown out whenever a concurrent update is made on the master, so they are not considered fully committed until propagated to the master. (many blockchains work that way.)
  - Concurrent updates are merged. (if an automatic merge fails, it could fall back on another strategy, which could be the previous strategy or something else like manual intervention, which most source version control systems do.)

The advantages of SSOT architectures include easier prevention of mistaken inconsistencies (such as a duplicate value/copy somewhere being forgotten), and greatly simplified version control. Without a SSOT, dealing with inconsistencies implies either complex and error-prone consensus algorithms, or using a simpler architecture that's liable to lose data in the face of inconsistency (the latter may seem unacceptable but it is sometimes a very good choice; it is how most blockchains operate: a transaction is actually final only if it was included in the next block that is mined).

Ideally, SSOT systems provide data that are authentic (and authenticatable), relevant, and referable.

Deployment of an SSOT architecture is becoming increasingly important in enterprise settings where incorrectly linked duplicate or de-normalized data elements (a direct consequence of intentional or unintentional denormalization of any explicit data model) pose a risk for retrieval of outdated, and therefore incorrect, information. Common examples (i.e., example classes of implementation) are as follows:

- In electronic health records (EHRs), it is imperative to accurately validate patient identity against a single referential repository, which serves as the SSOT. Duplicate representations of data within the enterprise would be implemented by the use of pointers rather than duplicate database tables, rows, or cells. This ensures that data updates to elements in the authoritative location are comprehensively distributed to all

federated database constituencies in the larger overall enterprise architecture. EHRs are an excellent class for exemplifying how SSOT architecture is both poignantly necessary and challenging to achieve: it is challenging because inter-organization health information exchange is inherently a cybersecurity competence hurdle, and nonetheless it is necessary, to prevent medical errors, to prevent the wasted costs of inefficiency (such as duplicated work or rework), and to make the primary care and medical home concepts feasible (to achieve competent care transitions).

- Single-source publishing as a general principle or ideal in content management relies on having SSOTs, via transclusion or (otherwise, at least) substitution. Substitution happens via libraries of objects that can be propagated as static copies which are later refreshed when necessary (that is, when refreshing of the copy-paste or import is triggered by a larger updating event). Component content management systems are a class of content management systems that aim to provide competence on this level.

[Creative Commons Attribution-Share Alike 4.0](#)

- [Fluid Framework](#)
- [Master data management](#)

[kb, architecture](#)

From:

<https://almbok.com/> - **ALMBoK.com**

Permanent link:

[https://almbok.com/kb/single\\_source\\_of\\_truth](https://almbok.com/kb/single_source_of_truth)

Last update: **2023/08/17 12:12**

