

2.2.1.4 System of Systems (SoS)

[return to Fundamental Views](#)

To go beyond simply filling the roles traditionally fulfilled by cash, DIDOs (e.g., cryptocurrencies) need to become part of a fully distributed System-of-Systems (SoS) rather than a single monolithic product offering from a single source. Although the use of OSS helps mitigate the multiple source issue, it is not the panacea some envision because the specification of the system is by definition the current [source code](#) of the OSS.¹⁾ The migration from a single monolithic product offering to an SoS means there should be multiple implementations available for most of the systems (or components) that comprise the DIDO Ecosystem (or Universe). Multiple DIDO implementations greatly reduce the risk of any part of the system being compromised by a single system, subsystem, or component failure, resulting in an even more robust network.

Furthermore, each component must be deterministic in its behavior, meaning that given the same set of inputs, the outputs will always be the same. In other words, not only will all the same implementations of a [node](#) produce the same output, but multiple implementations of a component within a node will provide the same results given the same inputs. In OSS systems, the OSS implementation **is** the reference implementation and provides the baseline definition of behavior resulting in a set of specific outputs for specific inputs. Selecting a component should be left to the individual participants in the network and treated as a business decision based on trust, mutual interests, and history. This means the ideal for all components is to have at least two different implementations, with each implementation acting to provide independent [validation](#) and verification of the other. This is not so different from current systems, where each node within the [blockchain](#) running the same code and getting the same results validates the transaction; differences in the results indicate a potentially compromised node.

For example, a successful DIDO could have worldwide usage (i.e., [Bitcoin](#), [Ethereum](#), etc.). With an SoS approach, each country in which a transaction is executed can have its own implementations of the various components and these implementations can reflect the rules and regulations on reporting and logging of the governing organization. For example, the Swiss might not want to have their transactions reported to the USA, China, Russia or even EU members owing to their unique privacy laws and [Data Residency](#) issues.²⁾ They would therefore select components that meet the needs of the Swiss rather than the world at large.

Technical Standards

- [OMG: Systems Modeling Language \(SysML\)](#)
- [OMG: Unified Architecture Framework \(UAF\)](#)
- [OMG: Unified Modeling Language \(UML\)](#)
- [ISO/IEC/IEEE 15288:2015 Systems and software engineering -- System life cycle processes](#)
- [ISO/IEC 25023:2016 SQuaRE -- Measurement of System and Software Product Quality](#)

de facto Standards

- None at this time

1)

[Section 2.2.1.2](#) covers the Open Source Paradigm in more detail.

2)

Object Management Group (OMG), “Data Residency Challenges and Opportunities for Standardization,” March 2017. [Online]. <http://www.omg.org/cgi-bin/doc?mars/17-03-22.pdf>.

From:

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

Permanent link:

https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:1.2_views:2_tech_views:1_core:1_sos

Last update: **2022/02/09 05:07**

