



## 1. Introduction

---

In August 2020, participants from Austria, Germany and Switzerland discussed the features and properties of a “Next Generation Learning Management System in a workshop (Adams Becker et al., 2017) “organized by CampusSource, the research focus project D2L2 at FernUni Hagen, and ETH Zurich (Dröschler & Kortemeyer, 2021). A major result of this workshop was that there will likely be no monolithic Next Generation Learning System, but a dynamic Learning Ecosystem with pluggable or otherwise interoperable components and services. A follow-up workshop on Open-Source Education Platforms, as well as a subsequent workshop on Open Educational Resources in November 2021, sponsored by SwissUniversities, stressed the importance of self-sovereignty and privacy in managing educational data.

A strong emphasis was put on non-traditional, lifelong learners, who carry their own data from institution to institution, where they “plug into” the local campus systems; this resulted in a proposed user model for these migrant learners (Kortemeyer, Dröschler, Riegler, & Koslowski, 2021). Models for the handling of shared LOs will be discussed here. The main use case is that instructors can select these LOs from repositories, organize them using their respective LMSs, and make them available to learners, without having to worry about establishing interoperability or enforcing licensing terms – or, in the second model presented here, privacy.

Some critics would argue that a LO thus far has never been an “object” in the sense of object-oriented programming: it traditionally lacks the methods to serve the content or interact with it. For example, the XML source of an assessment problem cannot be deployed without the methods to display and grade the problem.

LOs in the proposed models are comprised of both the content and the environment to bring this content to life. The fundamental idea of a “contentware” ecosystem (Haddad & Draxler, 2002) has already been proposed nearly three decades ago, when in 1993 the Educational Object Economy (EOE) was spearheaded by the US-American National Science Foundation (Gaible, 2004). Back then, the mission was to develop a simple end-user authoring tool for educational content as an alternative to publishing CD-ROMs. These efforts coincided with the emergence of the World Wide Web and soon after Java’s promise of “write-once-run-anywhere”. As a model and predecessor to a larger economy, authors could post Java applets and, if desired, their source code, into a repository. The model included the idea of a “socially-enriched learning community,” which was expected to ignite once a “critical mass” of repository content was reached (Shum, Sumner, & Spohrer, 1998; Gaible, 2004).

In the late 1990s, running functionality server-side emerged as another way of sharing interactive educational content across platforms, however, in a monolithic, closed environment (Kortemeyer & Bauer, 1999); others have likened such objects to Lego bricks in a Next Generation Digital Learning Environment (Brown, Dehoney, & Millichap, 2015). Notably, all of these developments occurred at a time when LOs were mostly associated with distance education (Gunawardena & Mclsaac, 2013); by contrast, the field was further advanced by repositories such as Schulcloud (Kremer et al., 2019), WirLernenOnline

(WirLernenOnline 2022), CampusContent, and edu-sharing (Klebl et al., 2010; edu-sharing, 2022), which implement varying degrees of integration into local learning platforms with support for blended scenarios.

The tool-interoperability version of the content model proposed in the first part of this report is designed to work with and around current technology, and with the explicit goal of low-threshold adoption. The self-sovereign model proposed in the second part relies on a corresponding user model to be in place (Kortemeyer, Dröschler, Riegler, & Koslowski, 2021), where both user and content model combined form the foundation of a Next Generation Learning Ecosystem.

## 2. Status Quo

---

### 2.1 Learning Management Systems

Currently, most higher education institutions make use of Learning Management Systems (LMSs) such as Moodle, Blackboard, ILIAS, D2L/brightspace, Stud.IP, LON-CAPA, OLAT, Canvas, etc. – and those are only some of the systems currently in use. Going from one system to another is connected with high switching costs. This reality gives rise to a few assumptions for our proposed model regarding LMSs:

- A reasonable assumption appears to be that there will not be a one-size-fits-all LMS, and that any model for cross-institutional learning content management and sharing should not rely on the idea of there being one single, monolithic LMS that will eventually gain market dominance; instead, it should take the dynamics and variety of the overall ecosystem into account.
- It is also illusionary to assume that all LMSs will implement particular common functionality and standards beyond some small subset of those put forward by the IMS Global Learning Consortium (IMS, 2021a) – and even with the most straightforward of these standards, in reality, there are bugs and idiosyncrasies when attempting to implement anything beyond minimal functionality, because there are diverse implementation flavors. Different LMSs – and thus the materials created within these different LMSs – will essentially remain incompatible.

Most of the current LMSs are built around the concept of a course container: content is created, served, and interacted with inside of courses, which are their basic unit of content granularity. Typically, it is up to the instructor to fill these course containers with content, either produced by themselves, shared with or “inherited” from colleagues, or gathered from repositories of shared educational content or from the web-at-large. Particularly when it comes to downloading content from the web, instructors frequently struggle with the concept of copyright.

## 2.2 Repositories

Typical repositories for downloading shared educational content came and, in some cases, went over the last decades; examples include the National Science Digital Library, Merlot, comPADRE, and OER Commons. Content pieces, e.g., texts, PDFs, web pages, images, videos, etc., are available in a searchable catalogue, typically built around Dublin Core (Weibel, S., Kunze, J., Lagoze, C., & Wolf, 1998) or IEEE Learning Object Metadata (LOM; IEEE, 2020). These static metadata like title, subject, description, language, type, etc., are useful for instructors, who know the field and thus also know what to look for, and who are then able to assess the discovered content with respect to correctness, level of presentation, pre-requisite knowledge, claimed and realistically achievable learning goals, appropriate depth, granularity, or pedagogical context (Krämer, 2005). There are no established educational taxonomies for the latter features, neither within nor beyond certain areas of study, so an expert is needed to remix and sequence the content resources. Efforts to establish recommender systems or AI-agents to assist with content discovery have thus far not had the same level of success as the corresponding systems in e-commerce, largely due to the lack of usage data. The selected repository materials are in many cases covered by Creative Commons licenses (Creative Commons, 2021)), which might allow for local modifications or corrections, but frequently these improvements do not make it back into the repositories.

Resources that have been downloaded from a repository need to be uploaded into the LMS. There are some problems with this otherwise easy to implement approach, most notably interoperability: even if content is downloaded as IMS Common Cartridge (IMS, 2021b), any interactive content is limited to the least common denominator of combined LMS and SCORM (ADL, 2021) functionality, leaving little room for innovation; besides, different LMSs export different flavors of cartridges. Finally, any corrections or improvements to the materials usually do not make their way from the LMS into the repository or vice versa. In summary, downloading content from a repository is a one-time, one-way street (Kortemeyer, 2013).

Linking to identified resources via hyperlinks embedded in the LMS course content oftentimes means linking to the websites which contain them, which presents learners with all of the proprietary context of these sites, e.g., menus and links to other resources; as a result, courses can lose coherence and learners can get lost. In addition, results from any interactive content, in particular assessment performance data, cannot flow back to the LMS. This can be remedied using plug-ins, as systems like edu-sharing (Klebl et al., 2010) successfully demonstrate – this means, however, that the plug-in has to be installed within the local LMS, which might present an adoption hurdle.

Deep-linking to identified resources, frequently using the IMS Learning Tool Interoperability (LTI) standard (IMS, 2021c) and possibly Single Sign-On (SSO), provides another way for incorporating shared LOs. LTI allows for data being sent back-and-forth between the external site and the LMS, so for example, de-personalized identities can be sent to the remote site, and assessment performance data can be sent back. LTI was originally designed to couple learning tools, not necessarily resources, but in combination with deep-linking to a particular resource can be used to embed content into LMS user interfaces within inline frames (“iframes”) or oEmbed (oEmbed, 2021). Particularly if the external site

allows for suppression of any of its internal menus and contextual information, this can make for a virtually seamless, transparent integration of external content, though not to the same level as plug-ins (Klebl et al., 2010).

## 2.3 Identity Providers

Most services and tools, including LMSs and interactive LOs, are built around the concept of sessions. Authentication is either handled by service-specific username-password combinations or increasingly by Single-Signon (SSO) identity providers, either within institutions, within federations of institutions (e.g., Swiss edu-ID), or world-wide by third parties (e.g., Google, LinkedIn, or Facebook). As the name suggests, SSO allows for the user to log in once and then seamlessly jump between systems recognizing the same central identity provider without having to log in again. While convenient and eliminating the hassle to remember individual username-password combinations, the mechanism also allows for users to be tracked across services.

## 3. The Proposed Models

---

We propose two distinct models of moving beyond the status quo, which are at opposite ends of the spectrum in terms of the extent of changes to the existing ecosystem. The first model is based on tool interoperability, which has been conceptualized and rolled-out before the advent of LTI (Krämer, & Zobel, 2008). The second model is based on user self-sovereignty, which circumvents the challenges of connecting the underlying platforms, yet is based on a vastly different paradigm of users and services (Preukschat, & Reed, 2021). Figure 1 illustrates these fundamentally different approaches to interacting with learning resources: in a tool-interoperability model, users are hosted on systems that connect behind the scenes, and their data is stored in databases on those systems (folder icon); in a self-sovereign model, the users themselves are the connecting elements, and their data stays with them in the form of so-called Verifiable Credentials (VCs) inside their wallets.

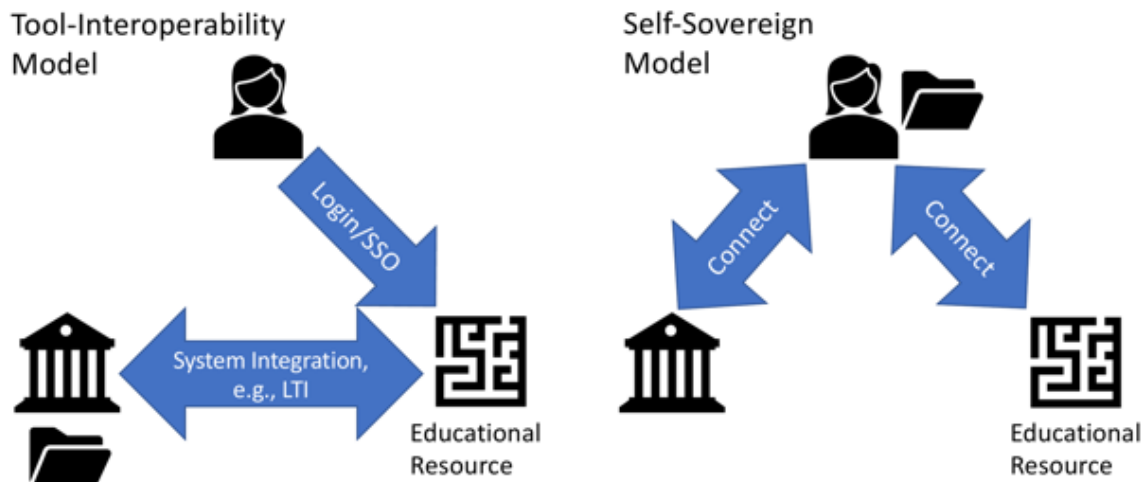


Figure 1: Different approaches for integrated learning resources. The left panel makes use of tool interoperability, while the right panel shows a self-sovereign approach.

### 3.1 Tool Interoperability

Standardized tool-interoperability is a gamechanger, compared to content-compatibility approaches. In many respects, it represents an extension of the concept of Remote Rendering Services (Hupfer et al., 2012). Learning Objects (LOs) are deployed and functioning in-situ with feedback to the institutional LMS instead of relying on the institutional LMS to properly deploy their downloaded components.

#### 3.1.1 Boundary Conditions

The tool-interoperability model does not expect a new ecosystem, like it would be required by the self-sovereign approach. The boundary conditions are:

- The system should not require any modifications (besides possible configuration changes) to the LMSs running at institutions.
  - For closed-source system, this is a straightforward assumption beyond documented APIs, but even for open-source systems, it is generally not a good idea to create forks.
  - This also excludes plug-ins, which require efforts in systems engineering and are frequently associated with security risks.
- The model works best in conjunction with a central identity provider and SSO.
  - Users are seamlessly transferred from one system to another, where automatically sessions are established.
- The system should enable deployment of content from an extensible set of authoring environments.

- The repository should provide an interface for instructors to discover and select LOs.
  - Instructors need to be provided with the necessary information to use these LOs within their LMSs.
- The repository should provide an interface for providers (e.g., individual authors, projects, software developers, publishing companies) to upload content.
  - Content can be associated with existing environments to create the Los
  - Pricing mechanisms can be selected.

Unfortunately, at least at first glance, the boundary conditions lead to a complex system, which requires a number of components.

### 3.1.2 System Components

Figure 2 shows an overview of the proposed tool-interoperability model, which is designed to manage and serve LOs in our sense of object, i.e., content and necessary environment. The components of the system are:

- **Institution:** a school, college, university, or education provider running its own local LMSs. These local LMSs manage courses (including communication, forums, gradebooks, etc.) and users, and it connects to the local administrative systems. Beyond that, in our model, the LMS mostly acts as the “glue” for LOs found in the repository.
- **Repository:** a central or federated repositories for LOs, including the content, environments to serve the content, and infrastructure for content management. The repository would be maintained and operated by a company, organization or institution, and it would accommodate not only not-for-profit, but also for-profit partners.

The repository should likely be divided into “domains” for different institutions, publishers, and organizations, similar to LON-CAPA (Kortemeyer, Kashy, Benenson, & Bauer, 2008), which allow for easier management of LOs and their access rights. In edu-sharing, this concept is implemented in the form of “contexts” and “organizations.”

- **Content:** these are content packages, which have a unique and persistent identifier that is assigned by the repository infrastructure. Content may be free, e.g., Open Educational Resources, covered by Creative Commons or open-source licenses, or it may be commercial.

Content packages can be versioned, nested, and forked. The packages include all resources (e.g., images, script libraries, etc.) necessary to serve the piece of content when called upon by its identifier, as well as static and dynamic metadata (including analytics).

The metadata also points to the required environment to serve the content (type and version). For non-interactive content, only minimal environments may be needed, while for highly interactive, user-modifiable content such as coding exercises, the content or components of the content may need to be copied into a persistent instantiation of the environment.

- **Environment:** the functionality needed to serve types of content, likely implemented as containers or virtual machines. These environments may contain anything from video streaming services and homework engines to complete, configured LMSs (for example, a Moodle with a particular set of plugins) – the only requirement is that LTI provider functionality is provided, which in many cases will be the built-in provider of the LMS or service. In some cases, the environment might be the main component of the LO, for example, a new type of discussion forum or annotation software, in which case the content of the LO is minimal. Environments can be free, e.g., covered by open-source licenses, or they may be commercial.

Some environments might be stateless and can be instantiated and destroyed again at will, while others may need to manage data or be persistent in one way or the other. For example, the environment may require to store data under the depersonalized ID of the user; these could be small entries like previous attempts on a homework problem, but go as far as complete, modified, or graded instantiations of the original content (e.g., modified versions of code templates for computer science assignments). The repository infrastructure needs to manage these environments through hypervisors, using launching and processing information from the content and environment metadata. Finally, the environment metadata contains information which flavor LTI to use.

- **LTI Frontend:** the LTI interface through which external LMSs access the LOs. Ideally, this would be one single, well-designed and adaptive interface, but in reality, it may have to be more than one interface for different LMSs and LTI

“flavors,” each with different addresses. The LTI Frontend receives access data, including a User-ID; in a well-designed client system, the institution will make this ID non-identifiable (i.e., depersonalized), but it will be unique and persistent over time. The LTI Frontend also enforces authorization and triggers the subsequent processes to serve the LO. Among other things, it will provide the connection and data adaption to connect to the appropriately flavored internal LTI Consumer.

- **Internal LTI Consumers:** these form the interfaces to the environments. Between the LTI Frontend and these internal consumers, LTI data is transformed between “flavors;” frontend and internal LTI interfaces act as converters.

- **https Frontend:** this is the web service which delivers the iframe-renderings of the content processed by the environment. It may perform some clean-up or mix in additional web content, such as the output of recommender systems. If semantics of the output are known, accessibility features can be embedded at this point.

In addition, the https Frontend can be used for the finalization and adaptation of the content rendering. If appropriate markup is possible within the environment, LOs can include multiple language or notation versions of the same content; content may be available in German, French, or English, and discipline-specific notation differences can be managed at this final point in the output chain. Separating universal business logic from output specifics makes maintenance of LOs more sustainable.

- **App Frontend:** this is the direct login to the repository, which can be used by instructors to locate content, as well as the business interface for content licenses.



- **Authorization:** this component controls access to content and app functionality and will need to work in terms of the depersonalized IDs.
- **E-Commerce:** an important component if the repository is operated by a commercial company, similar to iTunes or Netflix. Content and environments may require licenses for authorized access, so the system will need to provide the corresponding functionality. Different payment mechanisms will need to be accommodated, e.g., free access, micropayments, or vouchers.
- **Search, Recommender, and Analytics:** this component provides search and recommendation functionality for the App Frontend, as well as dynamic content recommendations for embedding into the LTI Frontend. These are mostly based on the static and dynamic metadata of the content and the environment (e.g., Kortemeyer, Dröschler, & Pritchard, 2014), but some mechanism requires users to be tracked across several transactions (e.g., Kortemeyer & Dröschler, 2021). In any case, all required transactional data is available through the LTI Frontend, from where it can be retrieved and appropriately stored in this component; comprehensive content analytics across domains and environments are one of the main advantages of centralized or federated repositories, yet also the source of privacy concerns. An alternative to harvesting this data from the LTI Frontend would be to have the environments deliver it directly via xAPI (xAPI, 2022), but this requires additional functionality within the environment.

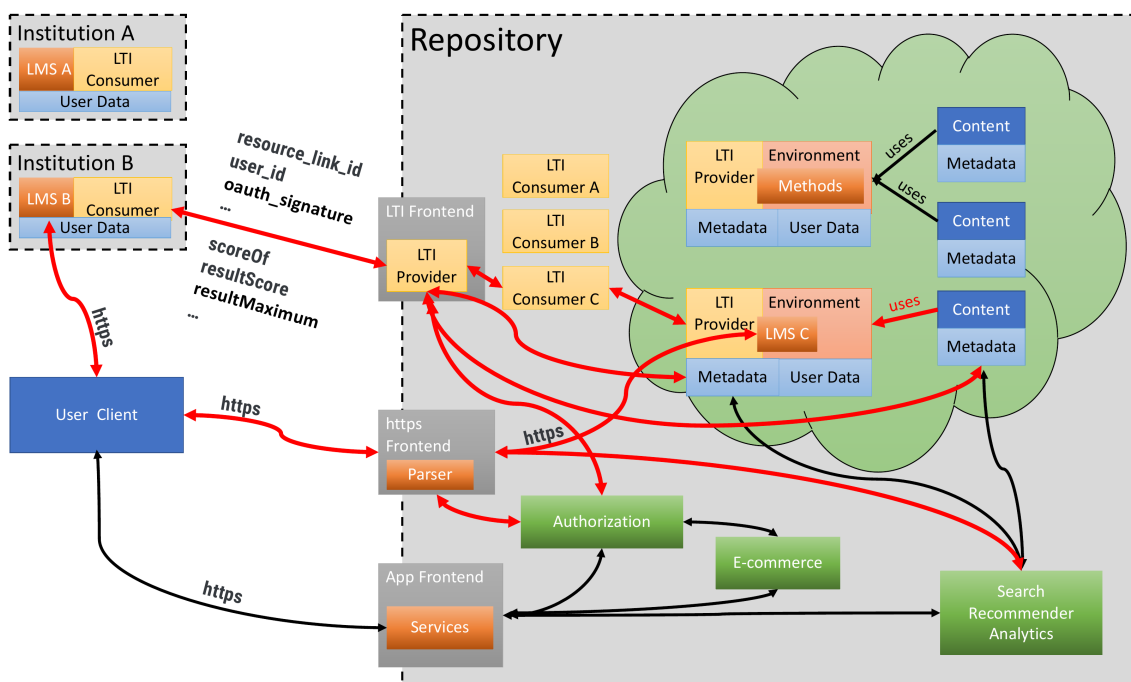


Figure 2: Overview of the proposed system. Red arrows highlight connections related to the user story outlined in Subsection 3.1.3

The system acts like an app store or content store (like iTunes), making available work by authors, publishing companies, and software companies for free or commercial access. The local LMS at the institutions acts like the playback device for these LOs.

This model may not be elegant, but as demonstrated by edu-sharing, it is feasible. Technical hurdles are minimal, as all components of the system are essentially known or relatively easy to build. A lot will depend on the operator, which needs to be a trusted entity. A possible avenue could be a public-private partnership, which may offer the best of both worlds.

### 3.1.3 User Story

To illustrate how these components interact, the following describes a typical user story. The connections required are highlighted in red in Figure 2.

- A learner at Institution B is working in his institutional LMS B, e.g., Blackboard, using his User Client (typically a browser or app). His instructor had previously assigned a LO from the Repository, which she found using the App Frontend and inserted in her course.
- As the learner accesses the LO, the LTI dialog is started between LMS B and the LTI Frontend of the Repository. In addition, an iframe is provided by LMS B for the content rendering, pointing at the https Frontend of the Repository.
- The LTI Frontend checks authorization to access the LO. If the content or the environment require commercial licenses, a paywall is brought up by the e-Commerce component, where the learner can add payment information or use tokens provided and purchased by Institution B.
- Having received authorization, the LTI Frontend locates the expected content package by its identifier. Based on the metadata, the required environment is invoked, which the underlying container or VM system instantiates if necessary. The metadata for the environment identifies the required internal LTI Consumer.
- The content is processed in the Environment, for example containing LMS C, which might be Moodle. If this system requires establishing a user, it is the LTI Frontend's task to establish the account within the Environment under the depersonalized ID provided by the LTI data, using procedures described in the Environment's metadata. User data may be stored by the Environment.
- The web output of the Environment is sent through the https Frontend to the iframe in the User Client. At this point, recommendations from the Recommender may be injected.
- LTI replies are sent through the LTI chain either synchronously or at a later point in time, e.g., after asynchronous grading.
- The LTI Frontend may send additional transactional data to the Search, Recommender, and Analytics component.
- LMS B stores data provided by the LTI Frontend under the learner's personal ID.

Other user stories may include an author or a publishing company uploading content to the Repository, where they can choose from existing Environments; the App Frontend in this case can also provide the editing interface toward the Environment. In addition, authors or software companies can publish new Environments.

## 3.2 Self-Sovereignty

This model, illustrated on the right panel of Figure 1, extends the user model constructed at the workshops (Kortemeyer, Dröschler, Riegler, & Koslowski, 2021) by operating fully self-sovereign. While in the tool-interoperability model, systems connect with each other in ways that are not necessarily obvious and open to the user, in this model, the user is the connecting element: all data-transfer is controlled by him or her – the user is self-sovereign and managing his or her Self-Sovereign Identity (SSI). There is no central or federated repository, instead there is a federated ecosystem.

SSI is built around connections instead of sessions. Entities, be it users or services (including educational resources), connect via Decentralized Identities (DIDs) instead of IP-addresses, and these connections remain persistent across what traditionally might be called sessions. These DIDs are specific to the connection and pseudonymous, which makes the entity behind the DID untraceable across connections with different entities. The end-points of the connections are called Agents, and may be on an individual's personal device ("Edge Agent") or in the cloud ("Cloud Agent").

Data is stored with the user, mostly in the form of Verifiable Credentials (VCs) (W3C, 2021), which are the common language between the systems; where necessary, the integrity of this user-hosted data is verifiable against a crypto-secured Ledger. Alongside their verifiable content ("payload"), VCs have a verifiable Issuer – whoever generated the VC – and a verifiable Holder – whomever the VC was made out to. The Ledger, frequently a blockchain, can be out in the open, and copies are typically widely distributed – this is possible, since the Ledger does not include any information that is useful beyond cryptographically verifying a VC. Holders can present VCs to other entities, who can independently verify the authenticity against the Ledger – in SSI-lingo, the recipient of a VC or other cryptographic proof is then called the Verifier.

A common concern is possible data loss. While centralized databases can be systematically backed up, individual users might lose their digital wallets and thus their private keys and VCs. A possible solution, highly in line with the decentralized nature of self-sovereign systems, is social recovery, where shards of the wallet data are automatically and redundantly shared among groups of users via the cloud, e.g., friends, family, and co-workers. Each shard would be a useless snippet of data, but a wallet could be recovered by contacting a sufficient number of these users (Liu, Lu, Paik, & Xu, 2020).

In the model presented here, LOs are entities. While this may seem counter-intuitive, it is in line with the idea that SSI includes the Internet-of-Things (IoT). The LO can make connections with users and issue VCs, for example, if certain tasks are accomplished. This even includes an LO managing its "own" financials (payment systems have been one of the first places where the combination of blockchains and DIDs took hold, up to the point where it seems normal to pay with a smartwatch)

### 3.2.1 Boundary Conditions

Figure 3 illustrates the necessary layers for such a self-sovereign ecosystem, only half of which are technological. On the crypto-utility layer, open-source will be an essential component to build trust in academic environments; these security-relevant layers need to be open for inspection. This utility layer might best be administered by a public-private partnership.

The upper layer of this ecosystem establishes its governance. In a global environment, where benevolence cannot be taken as a given, one of the most essential tasks of governance is to establish who is and who is not a member of the federation – who is trusted? Likely, this task will be delegated to sub-federations, based on nationalities or industry-sectors.

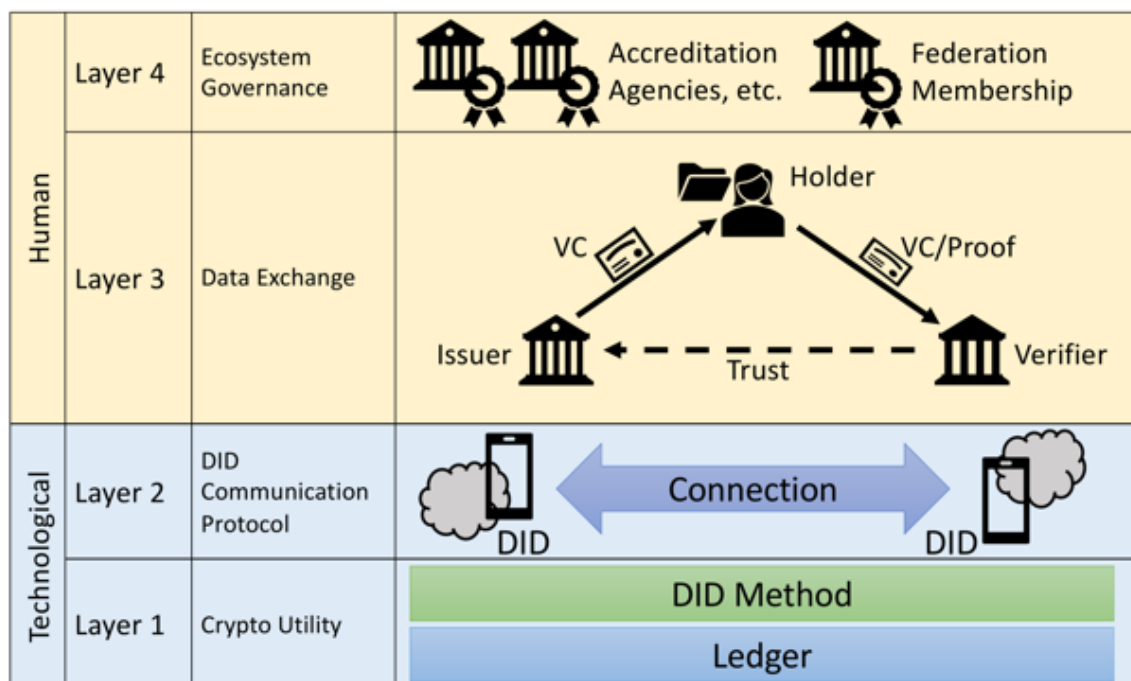


Figure 3: The layers of a self-sovereign ecosystem

In reality, there will be many ecosystems for different industry or public sectors, which may use different modes of governance and technology stacks (Kortemeyer, 2022); the common language between these ecosystems needs to be VCs, where the governances of each ecosystem need to decide which other ecosystems it trusts.

Establishing this upper governance layer is also the make-or-break of any ecosystem: is there sufficient political will or commercial pressure to implement this kind of paradigm shift? In other sectors, though, like payment systems (e.g., Apple Pay) or “ecosystems” like the COVID-certificates, change happened within a couple of years.

### 3.2.2 System Components

There are a number of components needed to connect to LOs in a self-sovereign fashion:

- **User Data Management:** A user is the Holder of his or her own data: VCs and other files, such as portfolios or logs. Their “digital twin” has two components:
  - **Wallet:** this is where typically VCs are stored and managed. Wallets can usually be found on personal devices, such as smartphones, and connect through Edge Agents.
  - **Data Pod:** Data Pods would be located in the cloud, where typically larger files are stored and managed — the virtual equivalent of a safe-deposit box. Data Pods are typically hosted at so-called Agencies and connect through Cloud Agents (in an extension of VCs, files in the Data Pod could be verifiable through storing hashes of the write transactions (full or incremental)); a possible implementation for Data Pods may be found in an extension of Solid (Solid Project, 2022).
- **Ledger:** This would most likely be a federated blockchain, as opposed to a public blockchain such as Bitcoin or Ethereum. An advantage of a federated blockchain is that it does not require an expensive and energy-devouring proof-of-work, but instead only a consensus-based proof-of-stake.
- **Cloud-Agent Implementation:** The environment of LOs will require a Cloud Agent to connect to users. Users typically establish connections between their agent and service’s agent by scanning QR-codes from a welcome screen instead of making accounts and logging in. Typically, at least when first connecting, the Service may request certain VCs or other Proofs, which the user can selectively provide or deny. At a minimum, the Cloud Agent would replace the https Frontend, the LTI Frontend, and the e-Commerce component in Figure 2, while – if no closer integration is possible – still internally using the same LTI Consumer infrastructure as described in Subsection 3.1.2, however, with individual LOs.

Also the local LMS will require a Cloud Agent to accept VCs issued by LOs. However, this is symmetric, as neither the LMS nor the LO is in any way a leading system; in fact, the local LMS is just another LO-entity.

- **E-Commerce:** any payments required to use an LO would be handled by the LO-entity, which does its own “banking.” For site-licenses, the university might issue VCs that the user holds in his or her Wallet and presents; for individual payments, there are already a variety of self-sovereignty compatible payment systems. Note that this does not necessarily entail using crypto-currencies, but might.
- **Search, Recommender, and Analytics:** this component does and should become more cumbersome to implement in a self-sovereign system. By design, users cannot be tracked across interactions with different LOs. With the user being the only cross-connecting entity, the Recommender needs to be a Cloud Agent associated with the user. This “privately-employed” Cloud Agent can connect to services and LOs in the ecosystem for search and anonymous analytics.

Many of the technical system components would already exist in an ecosystem like the one in Figure 3, particularly when it comes to the user, where educational VCs would exist alongside health information, flight tickets, diplomas, driver licenses, etc.

### 3.2.3 User Story

The right panel of Figure 1 shows how a user would interact with an LO to accomplish an assigned task:

- The local LMS provides links to the LO-entities, as assigned by the instructor.
- The learner connects to the LO using a DID.
- Any possible e-commerce would be handled at this point in time.
- As the learner works, artifacts may be stored in their Data Pod.
- After completion of tasks, the service issues a VC.
- The learner subsequently turns in the VC to the local LMS.

Note how at the steps along the way, all interactions are under the control of the user; there is no direct dataflow between the local LMS and the LO, and no central identity provider which could track the user (openLCMS, 2022).

## 4. Conclusions

---

As a result of workshops on Next Generation Learning Management, we presented two distinctly different methods of deploying Learning Objects and getting performance data back: a low-threshold tool-integration approach, and a self-sovereign approach compatible with the self-sovereign user model proposed at the same workshop series.

Neither of these go far beyond the technological status quo (in fact, edu-sharing essentially implements the first model); the real challenge lies in the area of “human” structures that establish sustainable and widely recognized ecosystems around these Learning Objects.

## References

---

Adams Becker, S.; Cummins, M.; Davis, A.; Freeman, A.; Hall Giesinger, C.; Ananthanarayanan, V.: NMC Horizon Report: 2017 Higher Education Edition. The New Media Consortium, Austin, TX, 2017.

ADL (2021). <https://adlnet.gov/projects/scorm/> (last check 2022-06-29)

AWS (2021). <https://aws.amazon.com/> (last check 2022-06-29)

Brown, M.; Dehoney, J.; Millichap, N.: The Next Generation Digital Learning Environment: A Report on Research. ELI Paper, CO: Educause Louisville, 2015.

Creative Commons (2021). <https://creativecommons.org> (last check 2022-06-29)

Dröschler, S.; Kortemeyer, G.: Report from the Next Generation Learning Management System Workshop 2020. In: *elead*, se2021, 2021. urn:nbn:de:0009-5-53233, <https://elead.campussource.de/archive/se2021/5323> (last check 2022-06-29)

edu-sharing. 2022. <https://edu-sharing.com> (last check 2022-06-29)

Farnell, A.: We can't teach in a technological dystopia. In: *Times Higher Education*. March 4, 2021. <https://www.timeshighereducation.com/features/we-cant-teach-technological-dystopia> (last check 2022-06-29) (paywall)

Gaible, E.: The Educational Object Economy: Alternatives in Authoring and Aggregation of Educational Software. In: *Interactive Learning Environments*, 12(1-2), 2004, pp. 7-25. <https://doi.org/10.1080/1049482042000300887> (last check 2022-06-29)

• Gunawardena, C. N.; Mclsaac, M.S.: Distance education. In: *Handbook of research on educational communications and technology*. Routledge, 2013, pp. 361-401. <https://www.taylorfrancis.com/chapters/edit/10.4324/9781410609519-22/distance-education-charlotte-nirmalani-gunawardena-marina-stock-mcisaac> (last check 2022-06-29)

• Haddad, W.D.; Draxler, A. (Eds.): *Technologies for Education: Potential, Parameters, and Prospects*. AED, Washington, D.C., 2002. <https://unesdoc.unesco.org/ark:/48223/pf0000119129> (last check 2022-06-29)

• Hupfer, M.; Krämer, B. J.; Lukaschik, C.; Klebl, M.; Zobel, A.: CampusContent and the Repository Network edu-sharing. In: *elead*, 8, 2012. urn:nbn:de:0009-5-32890 <https://elead.campussource.de/archive/8/3289> (last check 2022-06-29)

• IEEE (2020). IEEE Standard for Learning Object Metadata, IEEE 1484.12.1-2020. <https://standards.ieee.org/ieee/1484.12.1/7699> , 10.1109/IEEESTD.2020.9262118 (last check 2022-06-29)

IMS (2021a): <https://www.imsglobal.org/specifications.html> (last check 2022-06-29)

IMS (2021b): <http://www.imsglobal.org/cc/> (last check 2022-06-29)

IMS (2021c): <https://www.imsglobal.org/activity/learning-tools-interoperability> (last check 2022-06-29)

Kortemeyer, G.; Bauer, W.: Multimedia Collaborative Content Creation (mc3): The MSU Lecture Online System. In: *Journal of Engineering Education*, 88(4), 1999, pp. 421-427. <https://doi.org/10.1002/j.2168-9830.1999.tb00469.x> (last check 2022-06-29)

Kortemeyer, G.; Kashy, E.; Benenson, W.; Bauer, W.: Experiences using the open-source learning content management and assessment system LON-CAPA in introductory physics courses. In: *American Journal of Physics*, 76(4), 2008, pp. 438-444. <https://doi.org/10.1119/1.2835046> <https://aapt.scitation.org/doi/abs/10.1119/1.2835046> (last check 2022-06-29)

Kortemeyer, G.: Ten years later: Why open educational resources have not noticeably affected higher education, and why we should care. In: *Educause Review*, 48(2), 2013. <https://er.educause.edu/articles/2013/2/ten-years-later-why-open-educational-resources-have-not-noticeably-affected-higher-education-and-why-we-should-care> (last check 2022-06-29)

Kortemeyer, G.; Dröschler, S.; Pritchard, D. E.: Harvesting latent and usage-based metadata in a course management system to enrich the underlying educational digital library. In: *International Journal on Digital Libraries*, 14(1-2), 2014, pp. 1-15. <https://dspace.mit.edu/handle/1721.1/104920> (last check 2022-06-29)

Kortemeyer, G.; Dröschler, S.: A user-transaction-based recommendation strategy for an educational digital library. In: *International Journal on Digital Libraries*, 22.2, 2021, pp. 147-157. <https://link.springer.com/article/10.1007/s00799-021-00298-8> (last check 2022-06-29)

Kortemeyer, G.; Dröschler, S.; Riegler, P.; Koslowski, N.: A Model for a Lifelong Learners' Educational Records and Identity in a Next Generation Learning Management System. In: *elead*, se2021, 2021. urn:nbn:de:0009-5-52672 <https://elead.campussource.de/archive/se2021/5267> (last check 2022-06-29)

Kortemeyer, G.: Verifiable Credentials und Hochschulen, *educa Newsletter*, 2022. <https://www.educa.ch/de/news/2022/verifiable-credentials-und-hochschulen> (last check 2022-06-29)

Klebl, M.; Krämer, B. J.; Zobel, A.; Hupfer, M.; Lukaschik, C.: Distributed repositories for educational content. In: *elead*, 7, 2010. urn:nbn:de:0009-5-27748 <https://elead.campussource.de/archive/7/2774> (last check 2022-06-29)

Kremer, A.; Oberländer, A.; Renz, J.; Meinel, C.: Finding Learning and Teaching Content inside HPI School Cloud (Schul-Cloud). In: 2019 IEEE Global Engineering Education Conference (EDUCON), IEEE 2019, pp. 1324-1330. DOI: 10.1109/EDUCON.2019.8725078 <https://ieeexplore.ieee.org/document/8725078> (last check 2022-06-29)

Krämer, B. J.: CampusContent. In: *elead*, 1, 2005. urn:nbn:de:0009-5-940 <https://elead.campussource.de/archive/1/94> (last check 2022-06-29)

Krämer, B. J.; Zobel, A.: Rollout of CampusContent. In: *elead*, 4, 2008. urn:nbn:de:0009-5-14173 <https://elead.campussource.de/archive/4/1417> (last check 2022-06-29)

Liu, Y.; Lu, Q.; Paik, H. Y.; Xu, X.: Design patterns for blockchain-based self-sovereign identity. In: *Proceedings of the European Conference on Pattern Languages of Programs 2020*. 2020, pp. 1-14. <https://doi.org/10.1145/3424771.3424802> <https://dl.acm.org/doi/abs/10.1145/3424771.3424802> (last check 2022-06-29)

oEmbed (2021). <https://oembed.com> (last check 2022-06-29)

openLCMS (2022) Use Cases (case #7). <https://openlcms.org/use-cases/> (last check 2022-06-29)

Preukschat, A.; Reed, D.: *Self-sovereign identity: decentralized digital identity and verifiable credentials*. Manning Publications, Shelter Island, NY, 2021.

Shum, S. B.; Sumner, T.; Spohrer, J.: Educational authoring tools and the educational object economy: Introduction to this special issue from the east/west group. In: *Journal of Interactive Media in Education*, 1998(2). <https://jime.open.ac.uk/articles/10.5334/1998-10/> (last check 2022-06-29)

Solid Project (2022). <https://solidproject.org> (last check 2022-06-29)



US Census Bureau (2021). Annual Services Report, retrieved via Statista. <https://www.census.gov/programs-surveys/sas.html> (last check 2022-06-29)

W3C (2021). Verifiable credentials data model 1.1: Expressing verifiable information on the web. <https://www.w3.org/TR/vc-data-model/> (last check 2022-06-29)

Weibel, S.; Kunze, J.; Lagoze, C.; Wolf, M.: Dublin core metadata for resource discovery. In: Internet Engineering Task Force RFC, 2413(222), 1998, pp. 132. <https://www.hjp.at/doc/rfc/rfc2413.html> (last check 2022-06-29)

WirLernenOnline (2022). <https://wirlernenonline.de> (last check 2022-06-29)

xAPI. <https://xapi.com/> (last check 2022-06-29)