Tsugi is open source software that makes it practical to quickly build and deploy learning sites, tools, and content that is seamlessly integrated into Learning Management Systems using the latest interoperability standards. [Tsugi](https://www.tsugi.org/) makes it possible for publishers of educational software or content to implement integration with major learning management systems with a relatively low level of developer effort when compared with researching and implementing interoperability standards from scratch.

**Who might benefit from Tsugi?**

* Campus IT staff can deploy an extensible local “App Store” integrated into their local learning management system. The App Store can host Tsugi tools developed locally or install and host Tsugi tools from a wide range of sources including free / [open source tools provided by the Tsugi project](https://github.com/tsugitools).
* A faculty member can find and integrate content and tools from their local App Stores and/or other Tsugi-based Learning Object Repositories anywhere on the Internet into their courses.
* For faculty or campuses producing Open Educational Resources (OER) Tsugi allows easy publishing of reusable and remixable OER content and tools that can seamlessly be imported and/or integrated into learning management systems.
* For book authors, it is possible to produce a web site for the book that includes supporting materials, quizzes, interactive exercises, and other learning content. Content hosted in the Tsugi Learning Object Repository can be seamlessly integrated into LMS systems, giving independent book authors a way to provide their additional materials and software in a form that is as good or better than integrations provided by the (very expensive) publishers.
* Instructional designers can work with local developers to quickly develop interoperable learning tools that meet the pedagogical needs of faculty and students. With the productivity gains of the Tsugi library it is feasible to develop unique applications down to the granularity of a particular course. Developer effort goes into building the tool and not developing and debugging low-level code to support interoperability standards.
* Tsugi is a stand-alone MOOC platform. For a school, individual or even professional society, Tsugi can be deployed on inexpensive hardware to support 100K+ users around the world per course. Tsugi has a built in login, grade-book, badging, and other facilities that allow the quick development and deployment that make it quite simple to share content MOOC-style.
* Edtech startups can save resources and quickly develop a Minimum Viable Product (MVP) using Tsugi, put it in production and begin showing it to their customers and investors. WHile they may chose to ultimately build their own ground up implementation of IMS standards, since Tsugi is open source they have a nice roadmap and test harness for their own implementation.
* For a mainstream LMS vendor, encouraging their customers to use Tsugi means that locally developed code at each of their institutions will be consistent, secure, and reliable. LMS vendors will not each have to separately develop “How to make an LTI Tool” documentation – they can just promote the use of Tsugi and leave all the developer training to Tsugi.

**What is Tsugi?**

Tsugi is a set of flexible “Lego blocks” that work very well together. To meet all of the use cases of Tsugi described above, adopters just pick and choose the elements of Tsugi that are needed.

The primary elements of Tsugi are as follows:

* A set of library routines that implement standards like IMS Learning Tools Interoperability, IMS Common Cartridge, IMS Content Item, and other standards as they become available. The Tsugi library is broken into three layers:
  + The lowest layer is a direct implementation of each of the standards.
  + The second layer is an “opinionated” API layer that prescribes a relational database model, conventions for the use of session data, compensation for slight differences in LMS implementations of various standards, performance improvements, and reliability improvements taking advantage of the underlying data model.
  + The third layer provides an “UI API” that implements the Tsugi style guide so all Tsugi modules look and function similarly from a UI perspective. As we move from 10’s to 1000’s of independently developed tools, consistency in look and feel across tools from multiple sources will be important.
* Tsugi has a [management console](https://github.com/tsugiproject/tsugi) that supports the Tsugi App Store use cases including Tsugi Module installation, providing facilities for API key management as well as the ability to take incoming requests for access keys to hosted Tsugi modules. The management console also makes it so all of the installed modules can be integrated into an LMS using LTI 1.x, LTI 2, or IMS Content Item.
* A faculty member can develop their course content and host it using Tsugi and then simply import that content into their course shell at the beginning of each semester.
* Tsugi can also be embedded into a public-facing web site providing learning management system functionality for the web site. When Tsugi is embedded into a web site, it can publish the content and tools hosted on the site as a set of interoperable learning objects using the IMS Common Cartridge or IMS Content Item specifications. Tsugi can transform a static educational content web site into an interoperable Learning Object Repository and a learning community.
* Tsugi can be further configured to turn a web site into a stand-alone MOOC by supporting login, activity tracking, tool launching, a map of students who are part of the course, and a flexible automated badging system that automatically issues badges that contain the necessary meta data and validation to comply with the Open Badge Initiative (OBI). Examples of this use case include: [Web Applications for Everybody](http://www.wa4e.com/) and [Python for Everybody](https://www.py4e.com/).

**Getting Started with Tsugi**

We are developing a free online course (a.k.a. MOOC) to train Tsugi developers and adopters at www.tsugi.org. The training will be a series of modules with assignments, quizzes and peer-graded activities. As developers complete their training they will be automatically awarded OBI compliant badges showing their progress and accomplishments.

And yes, this is kind of “meta” since we are using Tsugi to build a web site that will train Tsugi developers. Why not? Tsugi is the simplest way to build a stand alone training web site. Oh and of course, since Tsugi is also a Learning Object Repository, anyone can integrate the Tsugi training materials into their local LMS and teach their own Tsugi course. It is indeed “Turtles all the way down” with Tsugi.

There is already a set of exercises with sample code and sample implementations suitable for a workshop or short course on Tsugi.

**Example Tsugi Tools**

There is already an effort within the Apereo foundation to build, review, and share open source Tsugi tools. A number of simple tools are already available in the TsugiTools github repository:

* [A simple attendance tool](https://github.com/tsugitools/attend) – This is kept simple so it can be used used as an example of how to build a Tsugi tool
* [A quizzing tool that supports the GIFT format](https://github.com/tsugitools/gift) pioneered by Moodle. GIFT is a way to author quizzes using a simple plaintext syntax.
* [A course map](https://github.com/tsugitools/map) that allows students in a class to place themselves on a map an optionally release information about themselves on the map.
* [A peer-grading application](https://github.com/tsugitools/peer-grade) the implements a “more social” approach to peer grading. This is quite useful for assessments in MOOCs or social learning situations where there are flexible deadlines and participants are experiencing course material at their own pace.
* A simple way to create [Slack channels for a class](https://github.com/tsugitools/tsugi-php-slack) and invite all the students to the Slack channel.

These tools from the TsugiTools repository can be automatically installed into a Tsugi instance using the Tsugi application management console.

Another set of Tsugi tools are autograders – but these are generally narrowly focused on one topic and not part of the tsugitools repository.

**What Tsugi “Is Not”?**  
Tsugi is not an enterprise Learning Management System like Sakai, Moodle, Blackboard, Canvas or Desire2Learn. Tsugi works with and enhances all of those LMS’s.

An Enterprise LMS is a “walled garden” shielding course material from anyone except those students and faculty associated with a particular course session. An important use case for an Enterprise LMS is to create a “course shell” for every course section that is taught on a campus. Those course shells are important while the class is in session but become archival as soon as the session is completed. Many schools “clean out” course shells that are older than a certain date. All the courses in an Enterprise LMS are at the same URL – the courses exist “within” the LMS. Faculty / instructional designers must design their student experience using whatever limited facilities the LMS has in terms of content authoring.

When Tsugi is embedded into a web site the use cases are quite different. With Tsugi, each course or set of learning content has its own URL and can be separately hosted (this is why a single Tsugi course can easily scale to >100K users). While Tsugi is highly opinionated as to how tools look and feel, when Tsugi is embedded into a web site, the core content of the web site that surrounds Tsugi is simply web content and can be authored and styled in any way that the owner likes. The web site can “inject” style into Tsugi tools hosted by and used by the web site. It means learning content web sites can be beautiful, engaging, and easily searched. The web site can make use of any of the latest HTML/CSS/JavaScript technologies. Tsugi merely “plugs into” and adds value to the beautiful educational content on the web site.

A Tsugi web site is intended to make its materials available “forever” – long after the end of one semester. A Tsugi web site becomes a source of materials that might be in use in hundreds of classes around the world. Search engines can “find” Tsugi learning content sites.

**Programming Languages Supported**

Given that Tsugi is intended to level the playing field and make it possible for thousands of new developers to learn to build sophisticated learning tools without requiring a lot of programming experience, Tsugi implements most functionality in PHP first. Choosing PHP also allows small to medium Tsugi installations to run in the commonly available low-cost hosting plans with minimum hardware requirements. Choosing PHP also broadens the developer community that can contribute to, understand and/or expand the core functionality of Tsugi.

But PHP is less popular among professional programmers so there are Tsugi implementations that support multiple programming languages and web environments. Each of these is at a different level of functionality and will be expanded as there is interest.

* [Tsugi Java](https://github.com/tsugiproject/tsugi-java-servlet) shares its low-level API implementation with Sakai. It supports LTI 1.x and IMS Content Item. Tsugi Java also has the second-level “opinionated API” implemented for LTI 1.x launches.
* [Tsugi Node](https://github.com/tsugiproject/tsugi-node-sample) has support for the LTI launches with much of the the LTI 1.x standard implemented at both the low level and the “opinionated level”.
* There is a simple bridge to allow one to use the Tsugi PHP libraries in [Laravel](https://github.com/tsugiproject/tsugi-laravel-sample) applications.

All these implementations depend on the PHP-based management console for services beyond LTI 1.x launches. If there is interest we can build management consoles for other languages.

There is emerging interest and discussions in developing versions of Tsugi for Python/Tornado, Python/Flask, and Rails but these projects have not yet started.

**Where is Tsugi Going Next?**  
Tsugi is still a relatively small open source project and so the roadmap is defined by those who are using Tsugi as they have new needs and interests. That means that those who join the project early will have the ability to influence the early directions of the project. At the same time, Tsugi is continuing to evolve to meet those needs so adopters will need to stay in touch with the Tsugi project to know the latest news about changes, bug fixes, and improvements to the project.

Some of the general areas that Tsugi is likely to expand include:

* Better support for analytics standards like IMS Caliper and xAPI.
* More tools like (a) YouTube view tracking, (b) simple scalable threaded discussion tool, (c) a markdown authoring tool for “pages”, (d) a simple “clicker” tool to allow quick polls during a live class, (e) a video quizzing application, (f) a collaborative paper reading / commenting tool, …
* Of course the support for languages beyond PHP needs to be improved and more programming languages need to be supported.

**Tsugi is an Instance of the Next Generation Digital Learning Environment (NGDLE)**  
The core principles of Tsugi are openness, interoperability, and modularity. Tsugi is not a single product or a single library. It is a set of building blocks that can be composed in a myriad of ways to allow us to take a more flexible, distributed, and engaging approach to building and using learning resources.

Tsugi is trying to build the underlying infrastructure to enable teaching and learning innovation to happen many places independently as part of an ecosystem rather than looking for the “one product” that will solve every problem. While Tsugi is a “learning management system”, a “learning object repository” and an “app store” it takes a distributed approach to all these use cases. This makes it possible to have as many learning-oriented web sites, MOOCs, learning object repositories, and app stores as we like.

Ov course once we make it possible to have thousands of sources of outstanding and easily integrated learning content, we will need to solve the problem of searching and discovery. Which will be a good problem to solve.