

TIER 101

- [Basic and Background Information](#)
 - [What is TIER?](#)
 - [What is TIER supposed to do?](#)
 - [How do I know if TIER fits with my institution?](#)
 - [Is this for schools of any size?](#)
 - [What do I have to do to use TIER?](#)
- [Resources to help understand TIER](#)
 - [Where TIER Came From](#)
 - [TIER Reference Architecture](#)
 - [TIER Model](#)
 - [TIER DevOps Cycle](#)
 - [Container Packaging](#)

Basic and Background Information

What is TIER?

TIER is both an open-source toolset and a campus practice set. It provides a set of identity and access management components (Shibboleth, COmanage, and Grouper) and APIs packaged in a container with a regular release schedule. TIER is built to work with the InCommon Federation, and also has a community-defined set of common practices to ensure common approaches and ease of collaboration.

The TIER Program grew out of campus discussions and the desire to align efforts to form an open-source, community-developed identity and access management suite, with coordinated campus practices to ensure interoperability.

A general overview is on the [Internet2 TIER website](#).

What is TIER supposed to do?

Over the years, the identity and access management (IAM) community has developed a set of software components, and these have become critical parts of some campus infrastructures. In particular, Shibboleth, COmanage, and Grouper have grown up separately. TIER will integrate these components together, using APIs, data structures, and common development schedules. In addition, TIER will use Docker containers to house the software components, which will be configured to work well together and with the InCommon Federation. The goal is an IAM suite built by and for research and education.

At the same time, TIER will include a community-defined set of practices (such as those involved with multifactor authentication and attribute release) to ensure seamless access to services for researchers, faculty, staff, and students.

How do I know if TIER fits with my institution?

You may already use one or more of the TIER components. If so, TIER is probably right for you. The intent is to make configuration easier and consistent, as well as to make upgrades and enhancements easier to obtain and install.

In addition, the TIER focus on common campus practices and configurations will benefit institutions with individuals that collaborate with their colleagues at other institutions and regularly use collaboration tools and other services that are hosted elsewhere.

Is this for schools of any size?

Yes. In fact, we believe that TIER will make it easier for schools with smaller IT staffs to adopt the integrated software components. Making things easier is also the main rationale for using Docker containers to contain already-configured software components.

What do I have to do to use TIER?

The [TIER Package Delivery](#) wiki page includes links to the latest releases of the software, plus some background information about the structure of the container-based files. At this point in the TIER cycle, we are looking for campuses to download and test the VM images and provide feedback to the developers. *The TIER versions of these packages are not recommended for production deployment at this time.*

Resources to help understand TIER

Where TIER Came From

TIER grew out of a series of workshops to capture community input and needs. You can read the [200 campus stories](#) and the resulting [60 community requirements](#) that kicked off the TIER development work.

TIER Reference Architecture

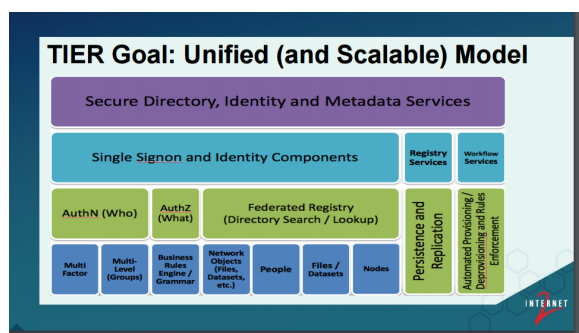
The [TIER Reference Architecture](#) explains the functional components for identity and access management in research and education, and how the components relate to one-another. This is a very understandable high-level overview, complete with diagrams representing how things fit together. You will also find links to the current and planned TIER components, along with links to real-life use cases (called "narrative walkthroughs").

Related Blogs

A blog post provides a [quick introduction of the Reference Architecture](#) from TIER developers.

How does InCommon fit with TIER? Read "[InCommon and TIER: Better Together](#)"

TIER Model



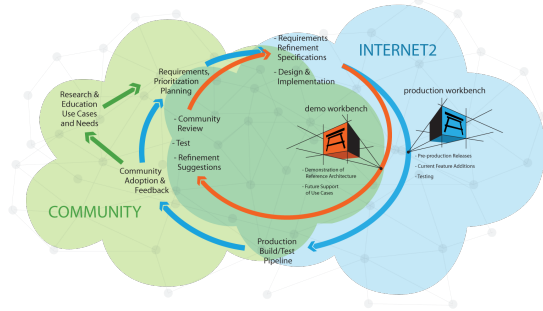
This chart represents the conceptual model initially outlined for the TIER Program's communication with the community. This includes most of the functionality contained within a full-featured Identity and Access Management (IAM) system. The top three blocks represent the functions, services, and processes that are involved when a user (or entity) needs to acquire access to various secured resources. The bottom block represents additional activities pertaining to a particular identity, and to what level of authority or privilege the various resources will be made available to that identity.

The TIER Program's goal is to provide the core software components enabling campuses to quickly implement the primary functions of a well-defined IAM service while providing a fully-developed architecture, and a set of recommendations and guides to both scale and optimize those services. More detailed information can be found in the TIER Reference Architecture (above).

TIER DevOps Cycle

"DevOps" (development/operations) is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support. The graphic to the right shows the TIER community's DevOps cycle.

One major feature of the TIER DevOps cycle is the lack of "big bang releases," in favor of continuous peer-reviewed and approved deliverables. In short, when an improvement or new feature is available, it will be moved into production as soon as practical without waiting for a major release.



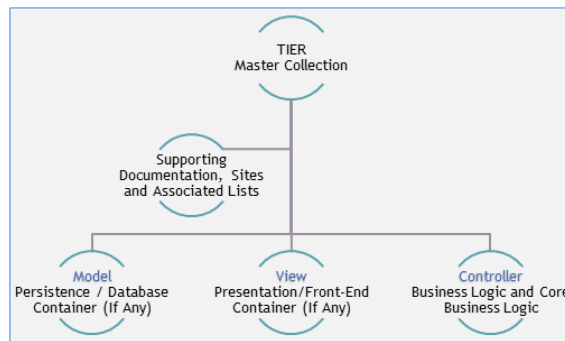
Related Blogs and Resources

Read ["The Landscape of DevOps within TIER"](#) for more background.

Read ["InCommon and TIER: Better Together, Part II, or InCommon in the TIER DevOps Environment"](#)

Container Packaging

The TIER versions of the software components (Shibboleth, Grouper, and CManage) are packaged with APIs in Docker containers. This container-based strategy will provide a consistent presentation of the components, and allow the developers to work in a coordinated consistent fashion to rapidly evolve the components and the APIs. The graphic to the right provides a simplified look at the container strategy.



Handy Links

[TIER on Internet2 website](#)

[TIER Questions, Comments and Feedback](#)

[TIER Package Delivery wiki](#)