Everybody needs a web site these days. Whether you’re building one for a company, for a service organization, or for personal reasons, you need one. And when trying to decide how to build it, you’ve probably found a dizzying array of choices running on a dizzying array of platforms. So how do you go about choosing which platform is best?

First of all, if Liferay Portal isn’t on your list, you should put it at the top right away. Liferay Portal is a Java-based open source portal, containing an unprecedented number of features, which will help you to implement your site in as little time as possible. And once you have Liferay on your list, let me respectfully submit that your search can end with Liferay Portal, which is hands down the best platform upon which to build a web site.

I can hear your objection now: “Of course you’ll say that—you work for Liferay!”

Ah, but I did not always work for Liferay. I was a Liferay user for some time before I wound up working for them. So yes, I took the red pill,¹ so to speak, but I’ve also experienced Liferay from the outside, and so I know what it’s like to be doing that search for a platform. I can tell you from experience that you’re going to find working with Liferay to be a pleasure, and you’ll be happy to know that using the platform that Liferay offers you will free you from limitations. Using Liferay as a platform will speed up your development cycle and give you features that you likely wouldn’t have had time or the inclination to build yourself. Most of the time, potential Liferay users focus on Liferay as a product—because it boasts such a huge range of features—but they don’t stop to consider the rich development platform it offers. Liferay as a development platform encourages you to take advantage of everything Liferay has to offer. By the end of this green paper, you’ll have a good understanding of what Liferay is all about and what it can do for your web site. And I have no doubt that you’ll find many reasons to choose Liferay for your next development project.

¹ From the 1999 film The Matrix.
Choosing Liferay is also safe: You’re putting yourself in a group with some of the largest organizations (and the largest web sites) out there that have also chosen Liferay as the platform for their web sites. So if I can give you any advice, it would be to end your search with Liferay and begin learning how you can leverage the platform to build the site of your dreams.

This paper will go over several important topics. I’ll show why Liferay calls itself a “portal,” what a portal used to be, and how Liferay pioneered getting past its early limitations—giving you the freedom to use the platform for what it was meant to be: a robust fast track to implementation. We’ll then take a helicopter ride over Liferay’s feature set to see what it can do at a high level. After this, we’ll delve into how Liferay helps you structure a web site. You’ll also get to see what Liferay looks like by default and how you can navigate around it. And finally, using all the information we’ve presented, I’ll show you how you can begin to imagine how your site might be implemented using Liferay Portal.

But first, to get our bearings, let’s start by exploring why Liferay calls itself a portal and what that term has come to mean in the industry historically.

**The Java portal promise: from disappointment to fulfillment**

Liferay calls itself a portal. What do you commonly think of when you hear the word portal? As a big fan of sci-fi and fantasy, I tend to think of a doorway to another dimension or time like the portal that Kirk and Spock went through, chasing after McCoy to stop him from doing whatever he did to change the timeline. I’ll tell you right away: Liferay Portal isn’t that elaborate (but you’ve likely already figured that out). So why do we call it a portal? Let’s start with the so-called official definition of a portal.

<table>
<thead>
<tr>
<th><strong>Portal</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>A portal is a web-based gateway that allows users to locate and create relevant content and use the applications they commonly need to be productive.</td>
</tr>
</tbody>
</table>

That comes from a bullet on a slide I’ve used to teach Liferay to prospective users. I might even have written that bullet, but I’m not sure. Generally, the reaction I get is a narrowing of the eyes, some pursed lips, and then heads begin nodding up and down. This tells me that people want me to think that what I’ve just said makes sense, but they’re being kind and reserving judgment on my teaching abilities, because it actually made no sense at all.

The problem with definitions like that is that they try to say too much in one sentence. Liferay is many, many things, and you can’t capture it all in one sentence. But just for fun, let’s try it again.

<table>
<thead>
<tr>
<th><strong>Portal</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>A portal is designed to be a single web-based environment from which all of a users applications can run, and these applications are integrated together in a consistent and systematic way.</td>
</tr>
</tbody>
</table>

That one’s a bit closer when viewed in the context of the web. When we talk about Liferay as a development platform, that’s exactly what we mean. At its base, Liferay is a container for integrated applications. Those applications are what make the difference between Liferay and competing products. You’re free to use the applications you like, write your own, and disable the rest. And this is what sets Liferay apart. Figure 1 shows how you can easily mix and match your applications with Liferay’s.
As an analogy, think back to the eighties and early nineties. If you bought a computer and you needed to use it to write something, you also bought a word processor. If you then decided you wanted to calculate numbers with it, you bought a spreadsheet. And if you needed to store and retrieve data of some kind (perhaps for a mailing list), you bought a database. (Nobody created electronic slides back then; they used an overhead projector. And yes, I am dating myself.)

Most of the time, people would pick what was considered the best of whichever program they wanted. One vendor had the best word processor. Another had the best spreadsheet. A third had the best database. So if you had to perform all three functions, it was likely that you had three separate programs written by separate entities, but individually they were the best.

Pretty soon, people wanted to create graphs in their spreadsheets that they would insert into a word processing document that they would send to a mailing list stored in the database. The problem with that was that all of these programs were created by different vendors, and they didn’t always work together all that well. Much effort on users’ parts had to be spent on trying to get them to work together well.

You know the rest of the story. We wound up with office suites, consisting of programs written on the same platform that were designed to work together. Not only did this save us all some money (because buying the separate programs cost a fortune), it also gave us a level of integration that had so far been unavailable.

The same thing is happening with software that can be used as an engine for running web sites. Liferay consists of that engine as well as the many applications that run on that engine. When you use this platform, your applications can have a level of integration with the rest of Liferay’s applications that will make your users’ experience seamless and smooth. Why? Because we believe the integrated experience is far better than the nonintegrated experience. This is the difference that makes Liferay stand above all the other portals out there. I have to say that because there are some who view the word portal with disdain, and sometimes this is with good reason. Let me explain further.

The Java portal disappointment

When Java portals were first announced, they were hailed as the solution to many of the problems facing enterprises and other solution architects. The web had grown up. Instead of proprietary interfaces to everything, everybody had finally standardized on TCP/IP networking and open protocols such as HTTP, IMAP, SMTP, SOAP, and the like. Services, applications, and email operated on these open protocols, and products that once had relied
on proprietary protocols had now opened up to the web. Those products that didn’t (or whose vendors had delayed it) were relegated to the dust bin of history. And once we had all of these siloed services speaking the same language, we needed something to bring it all together for the end user.

Enter the Java portal. The release of the Java portal specification came with the promise of bringing all of these services together in a single unified “web desktop.” Not only would it unify everything for a corporation’s internal applications, it would also be the hub of all B2B (business to business), B2C (business to consumer), B2E (business to employee), and even G2P (government to public) communication. It would be the presentation layer for the brand-new service-oriented architecture that you finished (or were in the process of) implementing. It could also be a platform for new applications. And it could finally bring together your static web sites and your applications, which resided on separate application servers.

Do you think too much was promised? How’s that old saying go? if it seems too good to be true, it probably is. Well, you’re right. What happened? At least three issues emerged that prevented Java portals from achieving widespread acceptance.

For one, it turned out to be difficult to develop solutions using a portal. The initial Portal API turned out to be something like getting to the least common denominator. Instead of providing all the features developers would need to bring all this stuff together, it defined what seemed like the absolute minimum that all the vendors could standardize on and then left everything else up to the individual vendors. This meant that developers had to spend more time implementing features that should have been part of the platform in the first place. One example of this is that the initial standard did not include any way for portlets to communicate with each other.

Second, the portal servers themselves were too big and complex (not to mention hideously expensive), often taking days to get set up. And for the developer trying to get a development environment going, it was sometimes even worse. I can remember trying to work with one of the first portals (sorry, can’t tell you which one it was) and finding it impossible to get a development environment properly configured on my laptop. At the time, I was a team lead and was trying to get this install process to a repeatable procedure for the rest of the developers on my team. My solution? I went to a conference, grabbed one of the presenters after his talk, and made him help me install the development environment on my machine. When he heard of my plight, he understood completely and told me everybody was having this problem and that they had to make this process easier.

Third, other things were happening in the industry at the same time. The Web 2.0 concept was beginning to get popular, and the portlet specification had left no room whatsoever for enabling a rich, client-side experience for the end user. In order to compete, portal vendors started to implement their own proprietary extensions to the portlet specification. We all know what this leads to: vendor lock-in, which is precisely what defining a standard is supposed to avoid.

At the same time Java portals were getting a bad rap, sites like Facebook and MySpace came out and pretty much implemented what portals were meant to do all along. And as they got more and more popular, suddenly other sites like Amazon.com and other software like Jira began to implement the social collaboration features Facebook and MySpace had, along with their slick, AJAX-enabled user interfaces. What powered all of these new and improved web sites? What enabled them to implement such rich features for the end user so quickly? You guessed it: open source.

Open source solves a lot of the problems inherent in the old Java portal paradigm. Open source projects don’t wait around for committees to decide on things; they tend to implement what the users want as fast as possible. There are no barriers to entry with open source; the development tools and the software are made available for free. Open source products also tend to be lighter weight; you don’t need a large, dedicated server to start building your solution. Development goes faster, because developers don’t have to learn the entire architecture to be effective. And you don’t need a huge initial investment to get started using an open source solution. You can start small (free) and then grow your application and hardware as your needs grow. Facebook is the perfect example of this: Implemented using PHP (which is an open source web development platform), the site has grown organically as its user base has grown. This is what the market really wanted. And as you’re about to see, Liferay Portal provides the same kind of open source platform that has allowed many organizations to do the same.
**Liferay keeps the Java portal promises**

From the beginning, Liferay Portal has been an open source project. Its whole purpose for existence was to level the playing field so that smaller organizations such as nonprofits, small businesses, and open source projects could take advantage of its platform without having to incur huge expenditures for either software or hardware. So right out of the gate it was doing things differently. An open source project doesn’t have the luxury of making it difficult for developers to work on the platform. Instead, developers need to find the platform to be easy to work with, or the project will have major hurdles to community gestation. And if an open source project can’t foster the birth and growth of a vibrant community, it’s dead. So right away, Liferay was (and continues to be) easy for developers to use, adapting to many different development styles, and not requiring any specific tools to be installed beyond what is already in any Java developer’s toolset.

This same philosophy translates to its size. Open source projects also don’t have the luxury of being too big or taking up too many system resources; they may be running on new hardware or five-year-old hardware that was donated to a nonprofit that can’t afford anything else. Liferay Portal is much smaller and simpler to configure than its competitors. Can you run Liferay on big hardware with a proprietary Java application server? Sure you can. Can you run it on a shared server with a small servlet container like Tomcat? Absolutely. Liferay Portal is provided as a standard .war file—only 75 MB in size—which can be installed on any application server, or as a “bundle,” preinstalled in your open source application server of choice. You don’t have to go through long installation routines and complex command-line incantations to get it working. If you use a bundle, installing Liferay is as easy as unzipping an archive and editing a text file to point it to your database.

And guess what? Instead of giving you by default an empty portlet container into which portlet applications can be installed, Liferay Portal comes with over 60 portlet applications included. These applications cover about all of the “standard” functionality you’re likely to need in a web site: content management, forums, wikis, blogs, and much more—leaving you to implement only the features specific to your site. And for developers, your setup time will be measured in minutes, not hours. You also don’t have to know everything about the architecture to be effective—it’s really easy to get started.

Open source software also has to be innovative in order to compete with its proprietary competition. Liferay Portal was the first portal to implement that slick, Web 2.0 interface, back in 2006. The first time I saw a portlet being dragged across the browser window and dropped into another spot on the page, I was blown away, because I was used to the old, proprietary solutions that hadn’t implemented that yet. Because Liferay Portal was open source, it could respond to market demands faster than the other guys, using the same standards they were using. You’ll continue to see that in Liferay Portal, because the open source paradigm works. What users demand gets implemented, without sacrificing adherence to standards.

As far as standards go, Liferay is also based on widely used, standard ways of doing things. It adheres to the JSR-286 portlet standard. In addition to that, though, it includes utilities such as Service Builder to automatically generate interfaces to databases (something not covered by the standard). Under the hood, Service Builder is just Spring and Hibernate—widely used by Java developers everywhere. So you get the benefit of using the platform to get your site done faster, while still taking advantage of standards that keep your code free.

Now that I’ve spent so much time extolling the virtues of this magical, mystical thing known as Liferay Portal, you’re probably anxious to see what this wonderful specimen I’ve described looks like.

**Getting to know Liferay**

Liferay Portal is an open source project that uses the LGPL open source license. This is the GPL license you know and love with one important exception: Liferay can be “linked” to software that is not open source. As long as you use Liferay’s extension points for your custom code, you don’t have to release your code as open source if you don’t wish to. You can keep it, sell it, or do whatever you want with it; it’s yours. If, however, you make a change to Liferay itself by modifying Liferay’s source code and want to redistribute the product thereafter, then you need to contribute that change back to Liferay. So you get an important exception with the LGPL: You can still use Liferay as a base for your own product and either open source the result or sell it commercially if you wish. Or, if you want to change Liferay directly, you can contribute to the open source project. It’s entirely up to you. You can download the open source version of Liferay Portal for free from Liferay’s web site.
Alternatively, Liferay sells an Enterprise Edition of Liferay Portal. This is a commercially available version of the product that comes with support and a hot-patching system for bug fixes and performance improvements. There are web sites running on both versions of Liferay Portal, and both are perfectly appropriate for serving up your site.

In this section, we’ll take a quick tour of some of the things you can do quickly with Liferay to begin building a web site. We’re going to play around with the interface a bit so you can get to know it a little better. So first, we’ll show the default. Figure 2 shows the Liferay Portal 6 user interface.

![Figure 2 Liferay Portal 6, as it looks the first time you start it. It presents you with a basic interface at first, but as you’ll see, you can easily jazz it up.](image)

Figure 2 Liferay Portal 6, as it looks the first time you start it. It presents you with a basic interface at first, but as you’ll see, you can easily jazz it up.

Okay, I agree; it doesn’t look like much, does it? But there’s an awful lot of power hidden in the humble interface that Liferay shows you by default.

**Liferay is an application aggregator**

We’ve been saying that Liferay Portal is not just a product; it’s a platform. This platform runs applications, and these applications are integrated in ways that separate applications can’t be, by virtue of their shared platform.

What this means is that we can take that default Liferay page and load it up with integrated applications. Liferay makes doing something like that easy. First, we have to log in as the default administrative user, whose user name is test@liferay.com and whose password is test. This will display the Dockbar (see figure 3) at the top of the page, which gives us access to several other functions.

![Figure 3 Running your mouse over the Add function in the Dockbar brings up a menu of options. To see the full list of available applications, select More.](image)

Figure 3 Running your mouse over the Add function in the Dockbar brings up a menu of options. To see the full list of available applications, select More.

We’ll come to all of the things you can do with the Dockbar in a moment. For now, however, all we want to look at is applications, which can be accessed under the Add menu. Some commonly used applications appear directly in the menu, but if you want to see the whole list, click the More link. This will pop up a fully searchable, categorized view of all the applications that have been installed in your Liferay Portal by default.
We’re going to fill this page with applications, so you can see how Liferay can aggregate them. You can browse
the applications by opening the categories to which they’re assigned. Or if you know the name of the application
you’re looking for, you can search for applications by using the search bar at the top of the Applications window.
Let’s pick some cool applications to add to our page. Note that in a real-world web site, you’d likely never put all of
these on one page—we’re doing an experiment here to show the concept. To the left column, add Navigation, Activities Dictionary, and Translator. To the right column, add Message Boards, Wiki, and Calendar. You can add an
application to a specific column by dragging the application off the Applications window and dropping it into the
appropriate column, as you can see in figure 4.

![Figure 4](image)

Figure 4 Most of the applications have been added. This screen shot was taken while dragging the Wiki application into the column on the right.

Now we have a single page with a whole bunch of applications on it. These applications can perform a lot of
different functions.

The Message Boards application is a complete implementation of web-based forum software. If you’re planning
to have discussion forums on your web site, Liferay already has them built in. And the cool thing about it is you
don’t have to integrate anything. They already work with Liferay’s user management and security features, as do all of Liferay’s applications.

You’ve also added a Wiki application to the page. Again, this is a full-fledged wiki that you can use for whatever
purpose suits you. As with the Message Boards application, the Wiki is integrated with Liferay’s user management
and security. But (and this is the cool part) the Wiki also is integrated with Liferay’s Message Boards application,
because it borrows functionality from that application to provide comment threads at the bottom of Wiki articles.
Those threads will use your users’ profile information (including pictures) in the threads to uniquely identify them in
a consistent way throughout your site, which is yet another level of integration.

What about the Calendar? Again, it’s totally integrated, complete with email notifications and more. And it’s a
full-fledged calendar application that supports export and import of calendar data from other applications.

The other applications are smaller, and I don’t want to gloss over them, but you’re probably getting the picture
at this point. Let me point out one other thing, though, and that’s the Activities application. Notice in figure 5 what
it says.
Figure 5 Every application in Liferay that uses the Social API can capture activities unique to that application. The Activities portlet displays those activities. Did you really create a new wiki page?

By virtue of adding the Wiki application to your page, you created a top-level Wiki page, which is by default called FrontPage. Because the Wiki application uses Liferay’s Social API to capture its unique activities, the Activities application can report on what you did (and even provide an RSS feed of activities). Liferay has an API that lets you tap into this social capability. This opens up all kinds of possibilities for your own applications, doesn’t it?

NOTE
Because Liferay is a portal, its applications are called portlets. I’ve been careful so far to refer to them only as applications, but from now on, I’ll use the terms portlet and application interchangeably.

Naturally, we’d never in the real world create a page such as this. Your users would throw conniption fits if they had to navigate such a thing. You might do better by your users if you put some of these applications on different pages. I just wanted to illustrate how integrated Liferay’s applications are.

In addition to providing a development platform and a slew of applications out of the box, Liferay is also a powerful content management system (CMS).

Liferay is a content manager
If you have lots of web content that you wish to publish, and you wish to publish that content using a workflow, or on a schedule, statically or dynamically, to staging or production, with templates or without, then you might want to check out Liferay’s CMS.

You can access the web content functions from the same Applications window you’ve already seen; they’re in their own category. But the quickest way to do it is to select Web Content Display, right from the Add menu. It’s in the top level of the menu for convenience—if you’re building a content-rich web site, you’ll use it a lot. Once it’s added, you can drag it to whatever position on the page you want. Figure 6 shows this portlet added to the right column on the page.
The Web Content Display portlet does what its name implies: It displays web content. But for it to do its job, you’ll have to create some web content. You can do that quickly by clicking the Add Web Content icon, which is the icon at the bottom right. You’re then brought to a page where you can add content. Figure 7 shows this page.

For now, don’t worry about all the options on the right side of the screen (Structure, Template, Workflow, and so on.). For basic content management, all you have to do is start adding content. Give your piece of content a name and a description, and type some content into the editor. Notice in figure 7 that you can apply all sorts of formatting in the editor: fonts, tables, bullets, colors, and images.

Once you’ve finished adding your content, notice the buttons at the bottom of the page. Though there is a whole workflow process you can go through, you’re logged in as the portal administrator. This means you can
short-circuit the workflow process by clicking the button marked Save and Approve. Doing that brings you back to your original portal page, and the Web Content portlet will contain the content you just added.

We could go further with Liferay’s Web Content Management system, but suffice it to say that it’s sufficiently powerful for whatever content needs you may have. For example, you can create your own structures with custom fields for your content, as well as templates to go along with your structures to display your content in exactly the way you want it displayed. You can stage your content on a staging server and have it published on a schedule of your choosing. You can write powerful, scripted templates in XSL, Velocity, or Freemarker.

You’ve seen so far that Liferay can be a platform or a UI for your applications, and it can also manage your site’s content. The last ingredient that you need Liferay provides in spades, and that’s a way for your users to find and collaborate with each other.

**Liferay is a collaboration tool**

Liferay Portal is ideal for setting up collaboration environments among workgroups. Whether you call these environments communities or virtual team rooms, you can use Liferay to help your team get their work done. It does this by providing applications geared specifically toward document sharing and communicating with one another.

One of the portlets you can add to a page in Liferay is the Document Library. This application provides a facility for sharing documents with your entire team. It keeps a complete version history of all of your documents and is integrated with Liferay’s permissions system. This integration allows you to grant access to shared documents or prevent some of your users from accessing sensitive documents. And if your users need an easier way to access the documents than the web interface provides, the Document Library supports WebDAV, allowing documents to be uploaded and downloaded through their operating system’s familiar interface. Figure 8 shows both of these interfaces.

![Figure 8 Accessing the same folders in the Document Library in the operating system via WebDAV or using the browser interface](image)

Documents are one thing, but what about communication? Liferay’s portlets allow for communication right in context, so your users can keep all the relevant information in the right place. The Document Library allows your users to create discussion threads next to the documents they need to talk about. The Wiki portlet does the same thing. Liferay also provides both chat and email applications, allowing currently logged-on users to communicate in real time, no matter what their physical distance is from one another.

Need a group calendar? The Calendar portlet can be used for either individuals or groups. In addition, your users can all have their own individual blogs on their own pages, which are then aggregated using the Blogs Aggregator to the community page. This enables you to display a “blog of blogs,” allowing your team to stay updated on what everyone is doing. Combining this with Activities makes for a consistent, rolling list of what the team is up to.

All of the functionality I’ve mentioned so far is what is built into Liferay (and there’s more we haven’t touched on). But Liferay is extensible too.
Liferay is anything you want it to be and any way you want it to look

Liferay offers a level of customization that’s unparalleled, because you can modify anything in Liferay, from simple functionality changes all the way to making your own product out of it.

You can write your own portlets so that your applications can be added seamlessly to your Liferay-powered web site’s pages in a way that’s indistinguishable from the built-in portlets. You can also customize Liferay’s layout templates so that your page layouts can be what you want them to be. You can also use hooks, which let you customize Liferay by substituting your own classes and JSPs in the place of Liferay’s. And finally, you can use Ext plug-ins, which let you override anything in Liferay with functionality of your own.

No discussion of customizing Liferay would be complete without covering themes. Using themes, you can transform Liferay’s look and feel to make it look any way you want it to (see figure 9). In short, Liferay can be anything you want it to be, and it can look any way you want it to look. This gives you the power and flexibility you need to build your own custom site, with the functionality you need to get it done in a timely fashion.

Figure 9 Liferay with a custom theme applied. This is one of many themes in Liferay’s community repository.

Liferay provides you the freedom to make your site look the way you want it to look, using skills you already have. Themes are nothing more than custom HTML and CSS applied to the page. You’ll have the same ability to design your site as you would have if you were writing the whole thing from scratch—except that you’ll have less work to do, because of Liferay’s built-in functionality and rich development platform.

Liferay also comes connected to two repositories of ready-made plug-ins, which extend Liferay’s functionality. One of these is a Liferay-provided repository, and the other is a community repository. Figure 9 is an example of a theme provided by Liferay’s community through the community repository. Liferay’s repositories make it easy to both distribute and install new software that runs on Liferay’s platform, as you can see in figure 10.
As you can see, a lot of functionality is built in to Liferay Portal, and it’s also extremely easy to add functionality to Liferay Portal. You can rest assured that the software you create on Liferay’s platform will be easily installed by your users. Let’s take a step back now so you can see what we’ve accomplished so far with a few clicks.

So, what has this little exercise accomplished?

Hopefully you see the power that Liferay gives you. In about 10 minutes—and without any additional software—we’ve created a web site that contains web content, forums, and a wiki; displays users’ activities; shares documents; and has a custom look and feel. We didn’t have to get separate applications to do all of that—instead, all that functionality (and more) is already included in Liferay. And because we didn’t have to use separate applications to implement what we wanted, we didn’t have to spend any time integrating those applications. Users get the experience of being able to sign into your site once and then navigate to the content they have access to, and you don’t have to do anything to make that work.

Pretty awesome, isn’t it?

Obviously, this only scratches the surface. You’re going to need ways of organizing and granting permissions to all those users you’re going to have. To do this, you’ll need to understand the support beams and structures Liferay gives you to organize that portal full of users.

How Liferay structures a portal: the high-level view

Every portal is different in the way users, security, and pages are handled. Because these aspects of a portal are not covered by the JSR-286 standard, every portal vendor has implemented these concepts differently. So if you’re going to start developing on Liferay’s platform, you’ll need to understand how a Liferay portal is configured and organized. Don’t worry; it’s not all that complicated, though it may look that way at first. Once you start using the system, you’ll quickly get the hang of it.
At its most basic level, a Liferay server consists of one or more portals. Portals have users, and these users can be categorized into various collections. Some of these collections can also have web pages that compose a portion of your site.

You can define many portals per portal server, and each portal has its own set of users and user collections. Figure 11 displays this graphically.

![Figure 11 A single Liferay Portal installation can host many different portals, all with separate users and content.](image)

As shown in figure 11, each portal has users, and those users themselves can be organized into several different types of collections: Roles, Organizations, Communities, User Groups, or any combination of those collections within that portal. Table 1 lists the collection types Liferay offers.

<table>
<thead>
<tr>
<th>Collection type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>Collects users by their function. Permissions in the portal can be attached to roles.</td>
</tr>
<tr>
<td>Organization</td>
<td>Collects users by their position in a hierarchy. Organizations can be nested in a tree structure. You would use organizations to represent things like a company's organizational chart.</td>
</tr>
<tr>
<td>Community</td>
<td>Collects users who have a common interest. They're single entities and can't be grouped hierarchically. By default, users can join and leave communities whenever they want, though you can administratively change it so that users are assigned to communities (or invited) by community administrators.</td>
</tr>
<tr>
<td>User Group</td>
<td>Collects users on an ad hoc basis. Defined by portal administrators</td>
</tr>
</tbody>
</table>

- Roles are inherently linked to permissions. You’d use a role to collect users who have the same permissions. A good example of this would be a Wiki Administrator role. This role would contain users who have permission to administer wikis.

- Organizations are hierarchical collections of users. Users can be members of one or many of them, up and down the hierarchy. Membership in organizations gives users access to the pages of that organization. If you picture a hierarchical structure that represents a company called Inkwell, a user might belong to Inkwell, Sales Department, Mid-Atlantic Region. This would not only denote that employee's position in the company, but it would also give that employee access to the content they need to do their job.
- Communities are ad hoc collections of users. Users can join and leave communities, and membership in communities gives them access to the pages of the communities of which they're members. You might have a community called Photography. Users of your site could join this community to share pictures.

- User Groups are defined by portal administrators. They can be used to collect users for purposes that tend to cut across the portal. For example, you might want to grant some users the ability to create a blog on your site. You would then create a User Group called Bloggers and create a page template for them that contains Liferay’s Blog portlet. Regardless of these users’ membership in other collections (as part of a hierarchy of organizations or as having joined several communities), User Groups provide a separate way of granting specific access to functions that don’t depend on membership in other collections or on specific portal permissions.

That’s the high-level view of a Liferay portal structure. Although this describes a powerful system for building your web site, it’s only scratching the surface of all that Liferay provides. I don’t want to delve too much into these concepts at this stage. While you let all of that sink in, let’s turn to something a little more concrete: how to navigate around Liferay.

**Getting around in Liferay**

Liferay’s user interface has a philosophy behind it: Get out of the way of the user. For that reason, it hides a lot of power behind what looks like a simple interface. One of the main UI elements is the Dockbar.

You’ve already been introduced to some of the functionality of the Dockbar, so let’s see what other functions it provides. Figure 12 shows the Dockbar in full.

![Figure 12 Liferay's Dockbar, which appears at the top of every page when a user is logged in](image)

**Pin icon**

At the far left is a pin icon, which does what you would expect it to do: It pins the Dockbar to the screen so that no matter how far down you scroll, it stays at the top of the screen. This can be helpful if you’re working with long pages and need to use the Dockbar’s functionality to add portlets to the bottom of the screen. This is a toggle switch, so you can unpin the Dockbar by clicking the icon again.

Next in the Dockbar is the Add menu.

**Add menu**

You’ve already seen most of the functionality of this menu for adding applications to the page. It can add pages too. If you click Add > Page, a new page will be added next to the page you’re on, and a field will appear, allowing you to name the page. There’s a much more powerful page administration screen, but this function allows you to quickly add pages to your web site as you’re working on it.

The next item in the Dockbar is the Manage menu.

**Manage menu**

Use the Manage menu to manage pages, page layouts, and more. This is where you get access to the interface, which lets you group your pages in the order you wish as well as nest them into subpage levels. You can also apply themes to whole layouts or to single pages. The Manage Pages screen is shown in figure 13.
Figure 13 The Manage Pages screen allows you to nest your pages, change the display order by dragging and dropping them, change themes, and more.

Perhaps the most important item in the Manage menu, however, is Control Panel. Liferay’s Control Panel is the central location where you can administer just about everything. The Control Panel is easy to navigate. On the left side is a list of headings with functions underneath them. The headings are in alphabetical order, but the functions are in a logical order. Figure 14 shows the Control Panel, which purposefully uses a different theme from the default pages, so you can instantly tell where you are.
Figure 14 Liferay’s Control Panel, with Web Content selected. Notice that the content is scoped by the Guest community, but this can be changed, depending on the user’s privileges.

- **User Name**—The first heading is named for the logged-in user (Test Test in figure 14) and is used to manage the user’s personal space. Here, you can change your account information and manage your own personal pages.

- **Content**—The Content section contains links to all of Liferay’s content management functions. You can maintain web content, documents, images, bookmarks, and a calendar; administer a message board; configure a wiki; and more. These links are scoped for the particular community from which you navigated to the Control Panel, but this can be changed using the select box at the top of the screen. Figure 14 shows the Control Panel with Web Content displayed.

- **Portal**—The Portal section allows portal administrators to set up and maintain the portal. This is where you can add and edit users, organizations, communities, and roles as well as configure the settings of the portal.

- **Server**—The Server section contains administrative functions for configuring portal instances, plug-ins, and more.

**Toggle Edit Controls**

The next function in the Dockbar is not a menu; it’s a toggle for the edit controls on the portlets. As an administrator, you get to see some icons in the title bars of the portlets on a page. These correspond roughly to the icons you might see in your operating system. There’s an icon for closing a portlet, for minimizing it, and for the configuration menu. If you’re composing a page and would like to see something that more closely resembles what your users will see, you can use the Toggle Edit Controls link to turn off these controls.

---

2 From the 1999 film *The Matrix*. 

©Manning Publications Co. Please post comments or corrections to the Author Online forum: [http://www.manning-sandbox.com/forum.jspa?forumID=642](http://www.manning-sandbox.com/forum.jspa?forumID=642)
Toward the end of the Dockbar is the Go To menu (shown in figure 15).

**Go To menu**

Use the Go To menu to navigate to the various Community and Organization pages to which you have access. Each page name appears, along with its public and private layouts, if they have them.

![Go To menu](image)

> Figure 15 The Go To menu displaying public and private layouts for three communities: the default Guest community, Dog Lovers, and Cat Lovers. Notice that the default community has only public pages, so only one link appears.

The final link in the Dockbar takes you to your user account information in the Control Panel.

**User Account**

The User Account menu item opens a page where you can change your name and email address, upload a profile picture, and maintain all information about you. You can also sign out of the portal from here.

As you can see, Liferay packs a lot of power in a deceptively simple user interface. The intent of this small tour was to give you an idea of where you can go in Liferay and how to get there as you begin to build your site.

Even though we’ve now touched on several of the constructs that provide you with the building blocks you’ll use to build a web site in Liferay, it’s sometimes difficult to begin imagining how your site could be built using these building blocks, because many of the concepts are new and unique to Liferay. So let’s spend a little time figuring out how you can imagine your site running in Liferay Portal.

**Imagining your site in Liferay**

Portal design is best done by breaking up your site into small chunks and then designing each chunk individually. That way, you don’t get overwhelmed by the largeness of your task, and before you know it, breaking it up into smaller chunks has enabled you to design the whole site!

In this section, we’ll walk through a design process that I’ve used with success to design many portals. We’ll break out the design process into three main portal chunks:

- User Groupings
- Organizations and Communities
- Content

**Asking the right questions**

The first thing we want to do is figure out how we can get all your ideas divided up into neat, organized chunks that can then be focused in on in more detail. Ask yourself the following questions:

- Will users be given freedom to sign up on the site?
- Will your user groupings be ad hoc, static, or both? (If your user groupings will be ad hoc, you know you’ll
be creating communities for your users to join and leave.

- Will some regular users have access to things other users won’t? (If so, you know you’ll be using Roles.)
- Will you be delegating administrative tasks to some users? (If so, then you may have Community or Organization administrators.)

Once you’ve answered these questions, go ahead and brainstorm the groupings or collections of users you may have.

**Defining and categorizing collections**

Don’t worry about trying to define users as User Groups, Communities, Organizations, or Roles yet. Start figuring out some groupings. Some examples are anonymous visitors (potential customers), customers, community members, and specific groupings based on your web site. For example, if you’re building a web site for do-it-yourselfers, you might come up with categories such as carpentry, plumbing, model rocketry, or even old computers.

At this point, you should have a good list of your groupings. Now combine that list with your answers to the previous questions. Will any of the groups require pages? If so, you know which ones are Communities or Organizations. Are the groupings associated in any way? If so, how? You’re now beginning to identify a possible Organization hierarchy.

Are there some groupings that cut across the entire portal (such as a bloggers group)? If so, that’s a likely candidate for a User Group, and you can begin thinking about whether these users should have page templates defined for them. Or it may be a good candidate for a Community, if the grouping should have its own set of pages. Once you’ve categorized your collections of users categorized into Organizations, Communities, User Groups, and Roles, you can begin designing your content.

**Designing content**

Pages can be part of Organizations or Communities. By default, each can have public pages, which everyone can see, and private pages for authenticated members of that Organization or Community. Take each Organization and Community you’ve identified and determine the page hierarchy that will exist for each one. This may even help you to further define your Roles and User Groups.

When you’ve finished with this process, you should have a nice, high-level design for your web site. You may have something simple, like Liferay’s default: one community called Guest for everyone to use. Or you may have something more complex.

**Summary**

Liferay Portal is an ideal choice for building your web site. Using the unique constructs that the platform gives you, you can design a site that can handle any situation you can throw at it. In addition to that, Liferay Portal offers you an unbeatable platform for building your web applications, as well as a ton of applications that are already implemented, in order to help jumpstart the creation of your site.

In addition, Liferay Portal frees you from the limitations of the old Java portal standard. As an open source project, it enables you to be as lightweight or as heavyweight as you want to be. And because it provides a multitude of tools and utilities for increasing developer productivity, you’ll be able to get your site done faster.

Liferay gives you a powerful paradigm for organizing your users and getting them access to the content they want to see. You can use Communities, Organizations, Roles, and User Groups to make sure that the right content gets to the right people and that restricted content is protected so that only the proper users can view it.

Because Liferay is so easy to use, you can create complex web sites quickly. Because all of the common applications you need to run a web site are already included, it’s a simple matter to pick the applications you need and drop them onto your pages. Because no further work is needed to integrate these applications, your time is freed up to focus on the applications you need to build that are unique to your web site.

*Liferay in Action* shows you how to customize Liferay to make it look the way you want it to look, act the way you want it to act, and host the applications that you design and write. I hope you’ll come along and take the red pill with me—it’s going to be an exciting ride.