

Realtime  
publishers

# Intelligently Reducing SharePoint Costs through Storage Optimization

Don Jones

sponsored by

 **AvePoint**<sup>®</sup>  
Unleashing the Power of SharePoint<sup>™</sup>

---

Chapter 3: Optimizing SharePoint Storage for External or Legacy Content.....	36
What Is “External Content?” .....	36
Databases .....	36
Files in Shared Folders .....	37
Media Files.....	37
Files from the Cloud.....	38
Today: Data Chaos.....	38
Traditional Approaches for Getting External Content into SharePoint.....	40
Integration.....	40
The BDC.....	40
Business Connectivity Services or “Two-Way BDC” .....	41
Migration .....	43
How It’s Done .....	43
Benefits of Migrating Content.....	43
Downsides of Migrating Content.....	44
Goals for External Content.....	45
Keeping the Content External .....	45
While Surfacing the Content in SharePoint.....	45
And Making External Content a Full SharePoint Citizen .....	46
Creative Approaches for Getting External Content into SharePoint.....	48
Content “Connectors” .....	48
Special Considerations for Media Files .....	50
How Do You Do It? .....	51
Coming Up Next.....	52

## Copyright Statement

© 2010 Realtime Publishers. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers its web site sponsors. In no event shall Realtime Publishers or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at [info@realtimepublishers.com](mailto:info@realtimepublishers.com).

[**Editor's Note:** This book was downloaded from Realtime Nexus—The Digital Library for IT Professionals. All leading technology books from Realtime Publishers can be found at <http://nexus.realtimepublishers.com>.]

## Chapter 3: Optimizing SharePoint Storage for External or Legacy Content

---

Shared folders. External databases. Even media files—audio, video, and so on. We want it all in SharePoint so that it's version-controlled, secured, and searchable—but can we afford to bloat the SharePoint database with that much content? Adding all that content will not only result in a pretty sizable SharePoint database but also take up a lot of expensive SharePoint storage. However, with the right tools and techniques, you can bring that content “into” SharePoint, while keeping it stored “outside,” helping to optimize your SharePoint storage and maintain a smaller, more manageable SharePoint content database.

### What Is “External Content?”

Today's businesses not only have a ton of data but they have it spread all over the place. Ideally, we could find a way to get *all* our content into *one* place, where users could search for it, read it, and potentially even submit changes as appropriate and allowed. SharePoint can't quite get you to the point where *all* your data is in once place—but with the right tools and techniques, it can come remarkably close to that goal.

### Databases

Businesses keep more data than ever in databases. In fact, it seems as if every major corporate application written these days requires a database of some kind. We don't necessarily want people to be able to *use* that data from within SharePoint, but a lot of the information living in databases *can* be used to great effect in SharePoint-based dashboards and other visualizations. For example, why not display charts and goals right on SharePoint home pages, keeping employees connected with the company's top-level goals and letting them know how well they and the rest of the company are doing in achieving those goals?

Of course, you certainly wouldn't want to copy all that business data into SharePoint just to create charts and graphs, or to use the data in other ways. Ideally, you want SharePoint to be able to *access* the data where it sits—in a secure fashion—so that SharePoint doesn't become a means for users to gain access to data that they shouldn't be looking at.

## Files in Shared Folders

The first corporate local area networks (LANs) were used primarily to connect users to each other—and to use each others' machines as file servers. The first server applications were primarily focused on sharing files. In fact, the first collaboration software—such as Microsoft Mail and cc:Mail—were essentially just elaborate file-sharing services. Shared files and folders have, in other words, been a key means of sharing information on the network since the very beginning of modern networks.

As the amount of data we store continued to grow, however, shared folders became less and less efficient. We had to map an increasing number of network drive letters to increasingly-complex shared folder hierarchies in order for our users to find all of that data. And keeping it organized was (and is) often a nightmare.

Shared folders don't typically support versioning, workflow, and all the other great features present in SharePoint. In fact, it would be great if we could just migrate all of that shared folder content into SharePoint. Doing so, however, isn't always feasible. The sheer amount of data may make a complete migration away from file servers impractical; in some cases, other processes depend on files being located on file servers, not within SharePoint.

In other cases, migrating shared folders into SharePoint can seem, well, wasteful. After all, they're just files, right? Why put them into a database, which is stored on a file system, when we could just store the files on the file system directly? Of course, but *not* having that content in SharePoint, we don't have access to features like versioning, workflow, advanced permissions, and so on—so there definitely *are* benefits to having the content “in SharePoint” somehow. However, perhaps a straightforward migration isn't the best means of achieving that goal.

## Media Files

Media files, more than other kinds of shared files, can highlight the worst about file servers *and* be the most difficult to bring into SharePoint. Media files—audio and video files, primarily—are difficult to index when they're just living on a file server. After all, a search engine can't (yet) “listen” to an audio file and extract keywords that can be used in searching. The file system doesn't, unfortunately, provide us with many ways to manually add those keywords as metadata, either. File servers make it tough for users to find the media files they want and need.

SharePoint could improve the situation with its more-robust metadata capabilities and search features, but moving media files—which tend to be pretty large—into SharePoint can exponentially increase storage utilization and even reduce database performance. Moving media into SharePoint can even reduce the performance of the media itself: Most media files are meant to stream, something that can be done quite readily from the file system. Streaming data from a database, however, goes pretty much against everything that our databases are designed to do. It can be burdensome for a database to retrieve streaming content properly, and when the database can't keep up, media playback is affected.

Again, we're faced with a bit of a dilemma: It would be nice to have media in SharePoint for versioning, workflow, indexing, and searching, but it seems as if doing so would negatively impact storage utilization and might even result in poorer user experiences.

### Files from the Cloud

Recently, businesses have started storing more and more data in a variety of cloud-based repositories. Those repositories take the form of simple FTP servers to cloud-based databases, and from Web sites to dedicated cloud-based storage. We're basically distributing our old file server model across the Internet—and we're not doing much to make it easier to find and manage that data.

But can we just pull all of that cloud-based data into SharePoint? Doing so would make it easier to access and manage, but it seems as if that would defeat the whole point of putting the data into the cloud. If we're just going to pull the data back into our SharePoint data center, then we really don't need the cloud at all, right? Again, a dilemma: How can we make that cloud-based data appear to live in SharePoint without moving it out of the cloud?

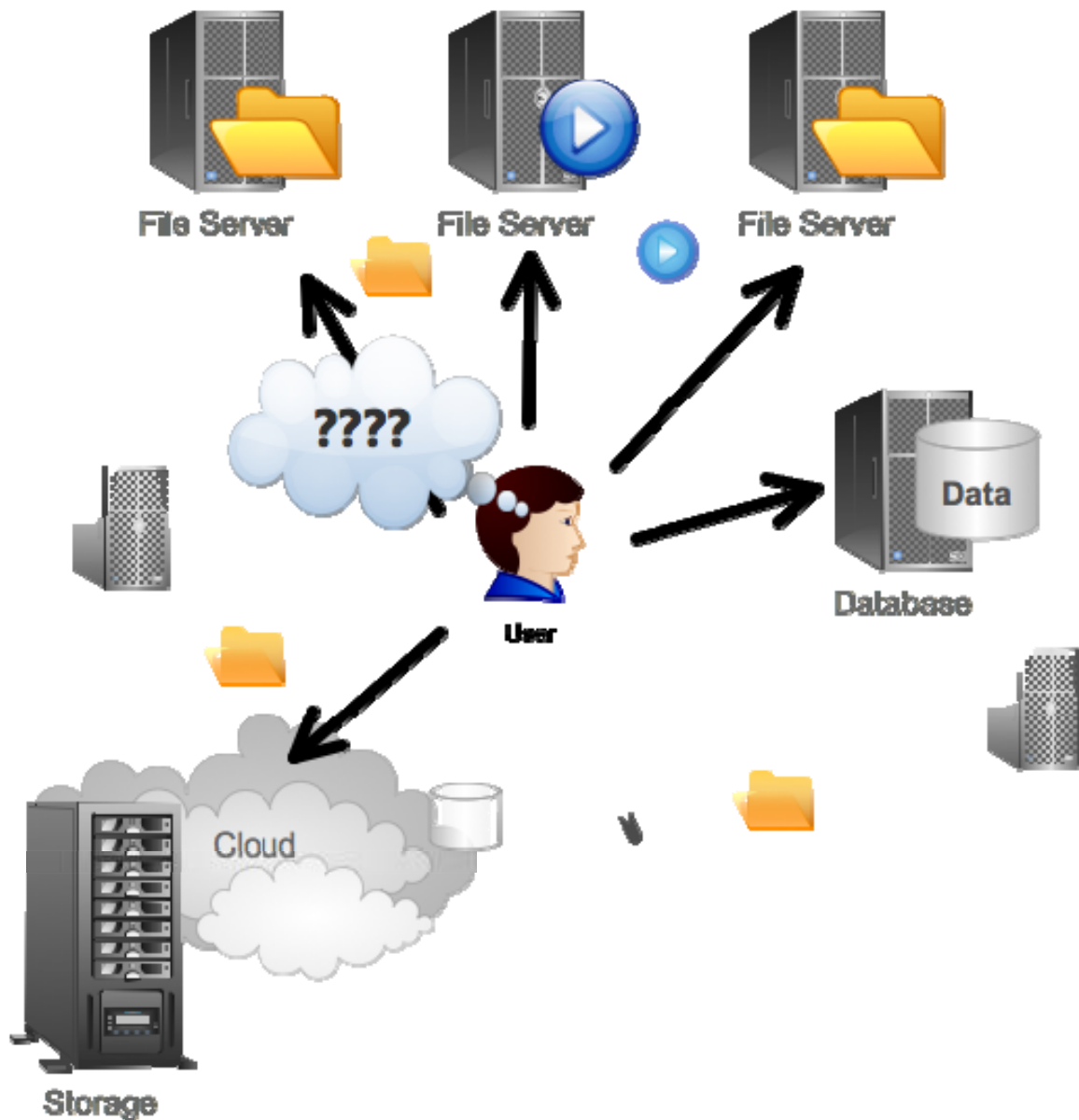
#### Note

You're probably thinking, "SharePoint in the cloud!" as a solution. Not quite. Although it's true that many companies offer SharePoint hosting, their SharePoint installations can only access data living in the hosting company's data center—just as *your* SharePoint servers would seem to be able to access only data on *your* network. The ideal would be to have data that could live in one part of the cloud, yet still be "in" SharePoint—either in your own SharePoint servers or perhaps even those you've outsourced to another part of the cloud.

### Today: Data Chaos

Frankly, it's amazing that our users can find anything—and it's no surprise that so much disk space is given over to the copies of data that users make into their "home" folders—the only place they can organize data in a way that makes sense to them. We've created "data chaos," illustrated in Figure 3.1, where our users are running all over the place (usually virtually, but not always) looking for files, media, data, and more.

SharePoint seems to offer a solution—a way to organize data that is accessible to everyone who needs it: rich metadata, tagging, and search capabilities, and the ability to version-control information, rollback to prior versions, and even apply workflow rules to information updates. But to really benefit from SharePoint's capabilities, we need *as much of our information as possible* inside SharePoint. Simply migrating everything, however, could have a negative impact on expensive SharePoint storage resources, and on SharePoint's own performance. It's possible that migrating might be the perfect choice for some types of information but less than perfect for others. But what can we do *besides* migrate the information, if our goal is to take advantage of SharePoint's features?



**Figure 3.1: Data chaos—our users are forced to look everywhere for the information they need.**

What's the answer? There are actually a few approaches, and they differ depending on exactly what kind of content you're working with.



## Traditional Approaches for Getting External Content into SharePoint

Let's start by looking at some of the traditional ways of incorporating external content into SharePoint. None of these techniques are bad, but they're not always the *best*. They offer compelling features and they help to meet certain business goals, so it's important that we consider them as potential tactics.

### Integration

When you're dealing primarily with data that lives in external databases, you're *not* going to migrate it into SharePoint unless your plan is to create a SharePoint-based application and eventually eliminate that external application (and its database) entirely. For this discussion, I'll assume you have data that you want to *keep* separate from SharePoint, but that you want some means of accessing that data from within SharePoint. Traditionally, Microsoft's answer—introduced with SharePoint Server 2007—has been the Business Data Catalog (BDC).

### The BDC

The BDC is a shared service within SharePoint, enabling SharePoint to grab business data from back-end server applications—ideally without any coding on your part (prior to the BDC, integrating back-end data almost always required some form of custom programming, which could be expensive to both create and maintain). The BDC allows data to be integrated into user profiles, custom applications, Web Parts, lists, and even search. It also provides support for displaying data drawn from both databases and from Web services—such as SAP, Siebel, or other line-of-business (LOB) applications.

Figure 3.2 shows how back-end data can be integrated into SharePoint portals. Here, marketing campaign and other information is drawn from a back-end database or Web service.

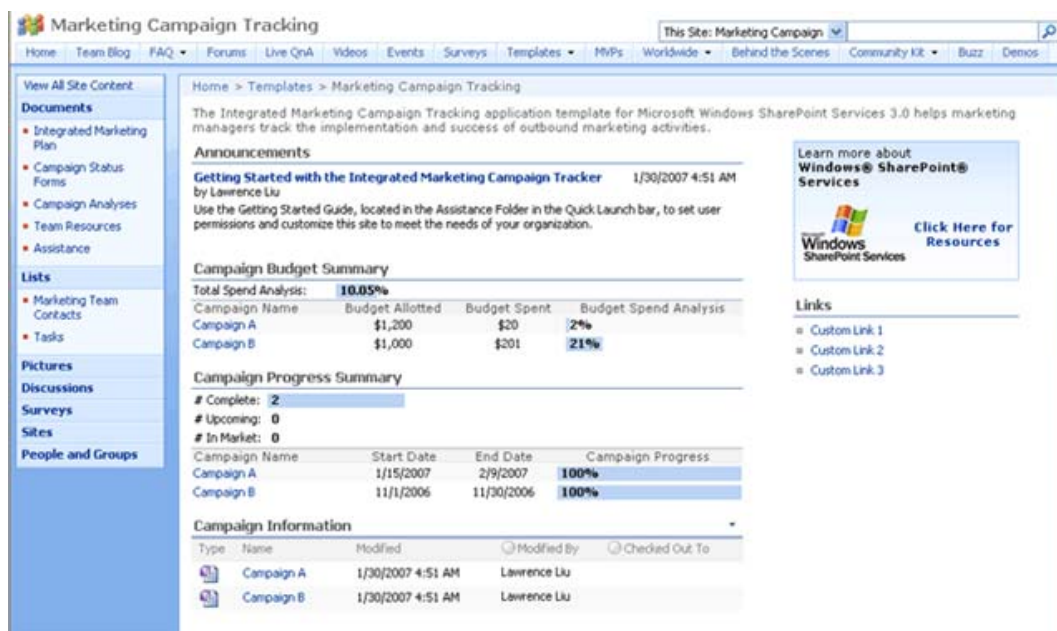
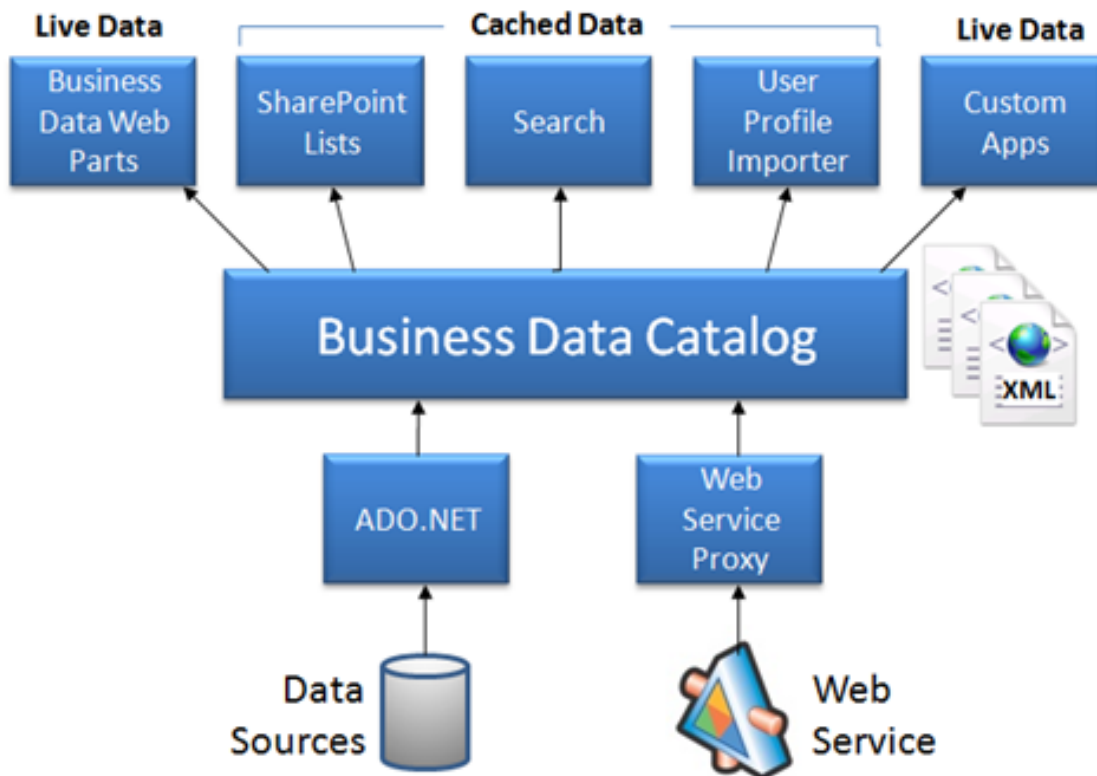


Figure 3.2: Using the data accessed by the BDC.



From an architecture standpoint, the BDC uses ADO.NET and other data-access techniques—such as a Web service proxy—to access back-end data from a variety of sources. It then makes that data available to a variety of SharePoint features. XML-based metadata is stored in SharePoint itself to help manage that data, and some of the back-end data is cached within SharePoint to enable features like search. Figure 3.3 illustrates the basic architecture.



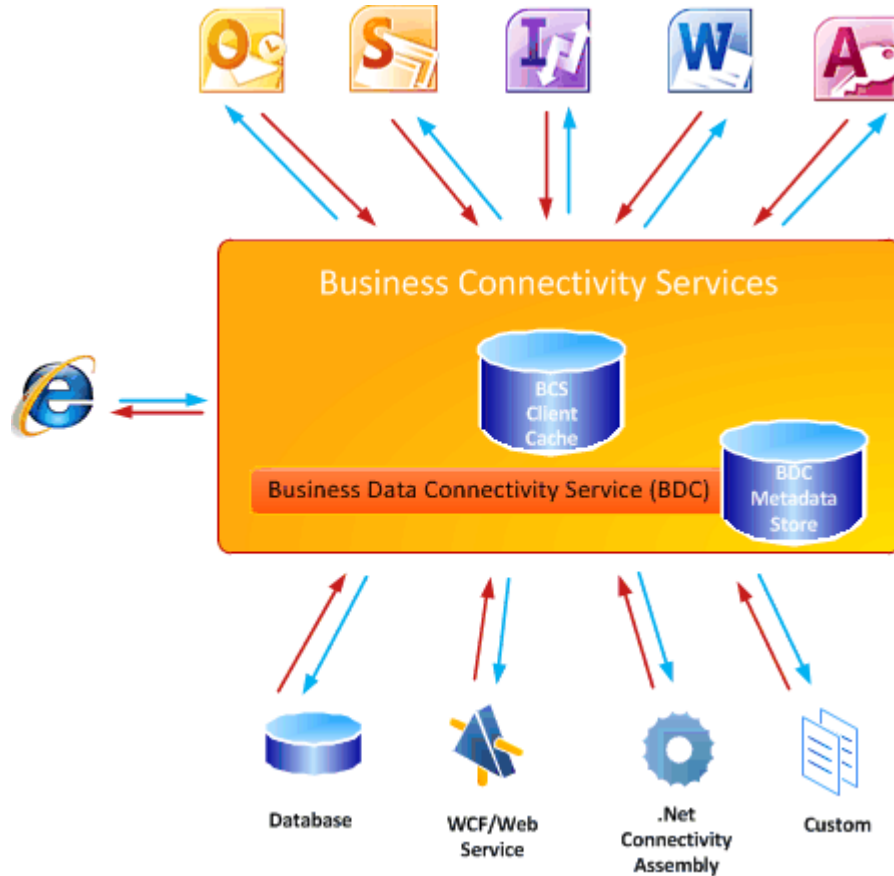
**Figure 3.3: BDC architecture.**

That XML-based metadata is the real functionality of the BDC. It defines the “mapping” between SharePoint and the back-end data, defining back-end entities, data relationships, and so on. You have to define this metadata when connecting the BDC to back-end data; SharePoint includes tools for doing so.

#### Business Connectivity Services or “Two-Way BDC”

For SharePoint 2010, BDC was renamed Business Connectivity Services (BCS), and given the ability to do more than just read data from back-end services. Now, it can also *write* data to the back-end. BCS can also be used from within Office 2010 applications. You might use Word to perform a mail merge from a customer database, for example. In SharePoint 2010, “BDC” now stands for “Business Data Connectivity,” the service that makes BCS work.

You still have to create XML models of your back-end data, and those models help control the ability to write data back to the back-end. As Figure 3.4 shows (drawn from <http://msdn.microsoft.com/en-us/library/ee557658.aspx>), the number of ways in which this data can be consumed is even larger than it was with the old BDC. It even includes robust caching so that users—say, someone using Excel 2010—can access data even if the back-end data source isn't accessible at the moment. BCS can sync read and write operations with the back-end server, once it does become available.



**Figure 3.4: BCS architecture.**

As shown, you can also write custom .NET Framework assemblies to broker connections to back-end data, enabling you to impose and enforce business logic between SharePoint and your back-end data.

#### Resource

You can read more about BCS at <http://msdn.microsoft.com/en-us/library/ee557658.aspx>.

BCS remains a solid way of integrating back-end data into SharePoint. It isn't exactly "no coding" as sometimes implied because you do need to be able to create your data models, but you won't have to actually write programming code in order to create those models.

## Migration

Migration: Moving content from place to place. In this case, permanently moving content from file servers or wherever into SharePoint. This isn't an option with back-end data; I'm not presenting migration as an alternative to the BDC/BCS. But for files, media, and so forth, migration is an option.

### How It's Done

Migration is typically performed using a tool of some kind—and a plethora of them exist from third-party vendors, although migration isn't a space that Microsoft has directly addressed with any major effort.

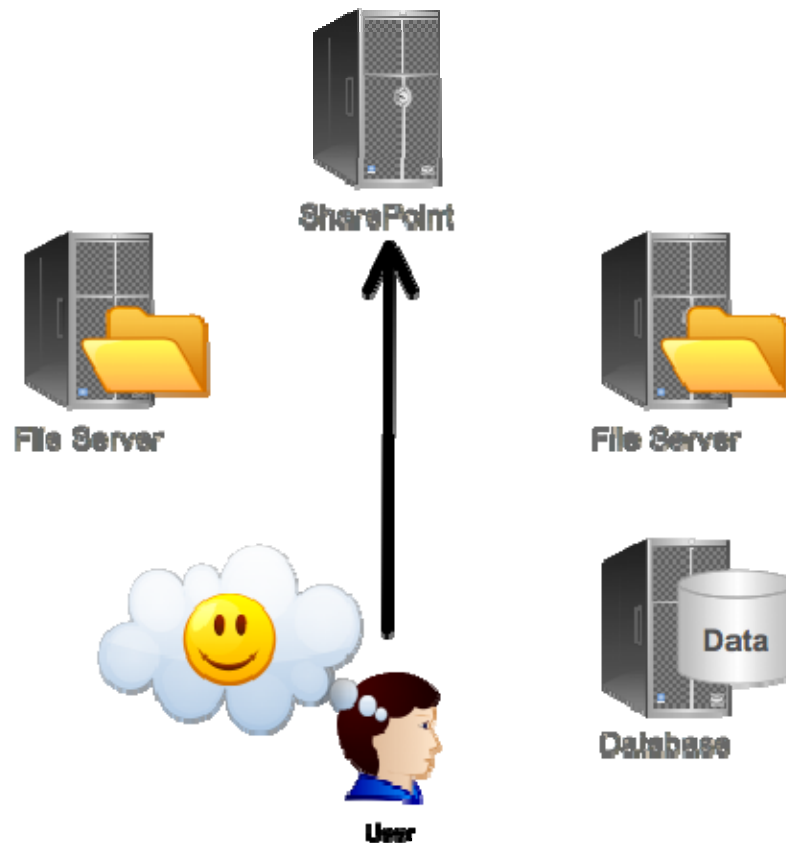
At its simplest, migration can involve simply copying a folder hierarchy into SharePoint, while maintaining whatever access permissions the file had on its file server. A more complex migration might also involve restructuring the data because SharePoint offers organizational metaphors and techniques that aren't present in a simpler, hierarchical folder structure on a file server. Migrations can also help automatically populate document metadata, which helps make those documents easier to find through SharePoint's search facilities.

### Benefits of Migrating Content

The benefits of migrating content into SharePoint are varied and significant. Your content becomes indexed and searchable. You gain the ability to add version control to documents, and to enforce workflow rules around document updates. You can in many cases employ simpler permissions structures, using SharePoint groups rather than individual user permissions—making long-term permission maintenance easier. You can even make it easier for content to be accessed from a variety of places, since SharePoint not only integrates with the Office suite of applications, but also exposes content via a Web-based interface.

But perhaps the biggest benefit of migration is that you can start to end data chaos. Along with BDC/BCS, migration can help bring all of your content *into one place*, as shown in Figure 3.5. Users have *one* place to go to find everything they need. You can start to manage permissions in this one place, report on utilization in this one place, and so on. You won't need to spend days teaching new users where to find everything on the network, and you can stop mapping a dozen network drives in logon scripts. Users begin to utilize SharePoint as—well, as a replacement for file servers, perhaps. Everything lives in one place.

By keeping authoritative copies of data in a single location, you can also start to avoid the bloat that often accompanies file servers. Users won't have as strong a need to make copies of data into their home folders; they can simply keep smaller shortcuts to SharePoint-stored documents. Everyone will be working from a single copy of information, rather than having dozens of copies spread across everyone's home folders.



**Figure 3.5: SharePoint migrations can help reduce data chaos.**

There are a *lot* of strong arguments in favor of migrating *everything* into SharePoint.

#### Downsides of Migrating Content

There are, however, downsides to migrating content, many of which I've outlined already.

- Large content items are going to take up a lot of room in the SharePoint database. That's going to complicate SharePoint performance management, as well as database maintenance tasks like backups, defragmentation, and so on.
- Media items don't work as well from a database—which as I've pointed out isn't designed for streaming data—as they would from a file system.
- Some file server-based data, as I suggested earlier, *needs* to be on a file server in order to interact with other processes. For example, you might have a process that “watches” a given folder for new files, and does something with them; that won't work if the file “lives” in SharePoint.

Ideally, there's some middle ground, where we can get the advantages of having content in SharePoint—without the downsides.

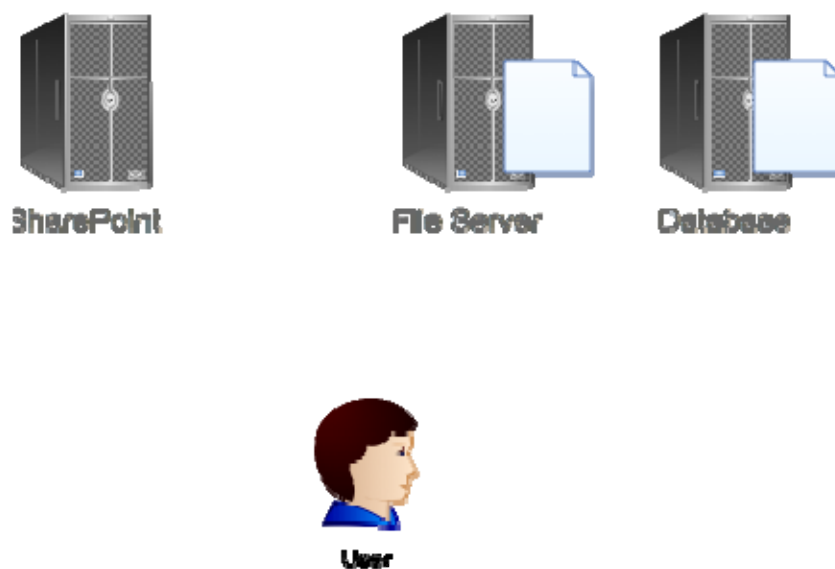
## Goals for External Content

Let's establish some business goals for all of this external content. Let's define—at a business level—what we'd like to do, and then see if it's possible to achieve that.

### Keeping the Content External

Our first goal will be to *keep external content outside of SharePoint*, simply because there are benefits in doing so. We'll maintain our file servers for those processes that rely on them, and we'll be keeping our SharePoint database from becoming overly-bloated with large content items. File servers *are really good at storing files*, after all, so let's let them continue to do so.

Let's start building a picture of what we want this to look like. Figure 3.6 will be our starting point: External content (represented as page icons) will remain external. They'll be accessible to external processes that might need them, and they'll remain secured by whatever security (such as SQL Server or NTFS permissions) that's in place.



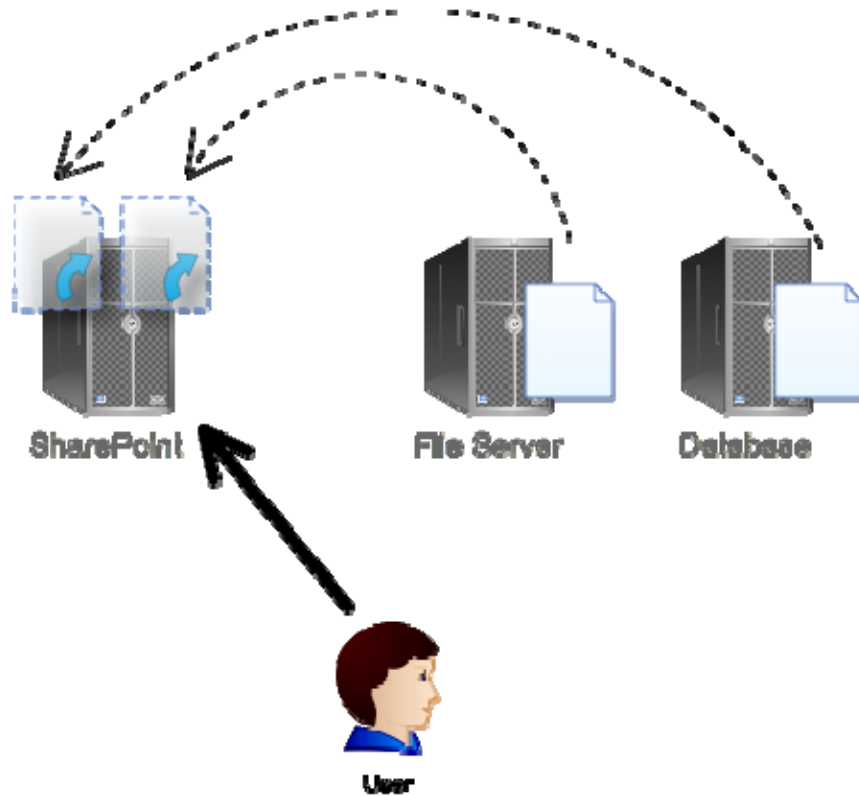
**Figure 3.6: Leaving external content where it is.**

Of course, now we're facing some downsides, because it's like we don't even *have* SharePoint. So we need to move quickly on to our next business goal.

### While Surfacing the Content in SharePoint

We *want* that external content to appear in SharePoint. In the case of database data, that might be through something like a custom application, or a BDC/BCS type connection. The data stays in the database, but SharePoint becomes an interface to it.

Similarly, we want SharePoint to be *aware* of our external files and folders—including media content. We want users to be able to find that content from within SharePoint. SharePoint will thus remain a single point-of-access for our users—helping to kill data chaos—but it will just be “keeping track” of the external content’s location, rather than necessarily storing it in the SharePoint database. Figure 3.7 shows this updated goal, with shortcut icons indicating that the content is *surfaced* in SharePoint, but not actually *present* within it.



**Figure 3.7: Surfacing external content in SharePoint without migrating it.**

Now, we need to be very specific about our business requirements for how this content is treated within SharePoint. It’s not that tough to just create a little shortcut, within SharePoint, to external content; we want a bit more functionality out of the situation than just that.

### And Making External Content a Full SharePoint Citizen

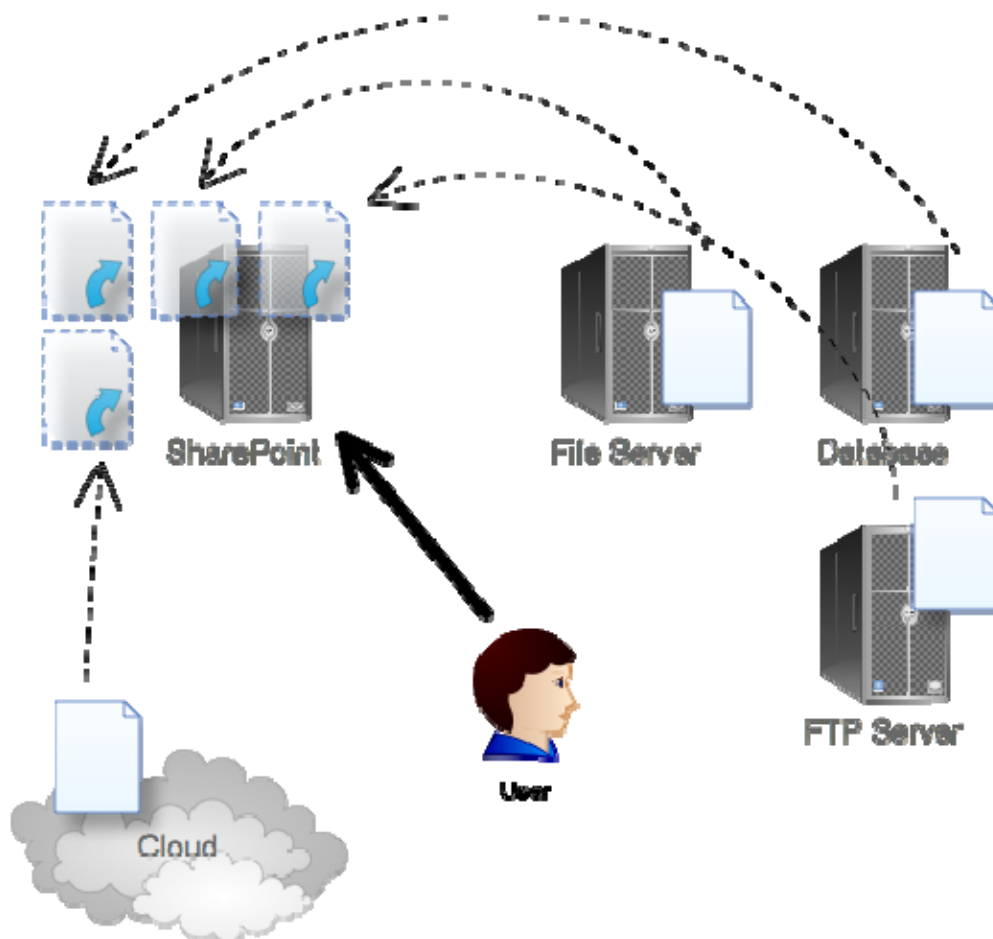
We want our external data to be a “full citizen” of the SharePoint environment. Specifically, we want external data:

- Included in a SharePoint Document Library, not just as a list of external links
- Included in workflow operations
- Available for alerts, so that users can be notified when content changes
- Added to SharePoint’s search index

- Taggable with metadata to improve searchability
- Controlled by SharePoint's permissions (at least, all access via SharePoint should be controlled that way)
- Accessed directly from within SharePoint by SharePoint-aware applications like Microsoft Office. We should be able to have Word open up a SharePoint-based Word document, even if that document is technically still living on a file server someplace. We want to remove users' awareness of those file servers, in other words.

Basically, we're asking a lot: We want our content to act as if it lives in the SharePoint database, even if it doesn't.

**And we want this for more than just file servers.** We also want content located on local and, possibly, remote FTP servers treated this way, along with content that may be located in outsourced, cloud-based storage. In fact, let's update our drawing—in Figure 3.8—to reflect that added requirement.



**Figure 3.8: Including FTP- and cloud-based content in SharePoint.**

We *can* do all of this. We just need to take a more creative approach for getting external content into SharePoint—more creative than a simple migration, that is.



**Note**

From here on out, I'm going to sort of drop the "external content from a database" aspect of the discussion. SharePoint's BDC/BCS functionality serves that need well, and custom applications are needed if you exceed the capabilities of BDC/BCS. I'm going to focus on external file-based content (including media files) living on FTP servers, cloud storage, or file servers. I'm not implying in any way that database-based data isn't important, just that it's a problem with a clearly-defined native solution.

## Creative Approaches for Getting External Content into SharePoint

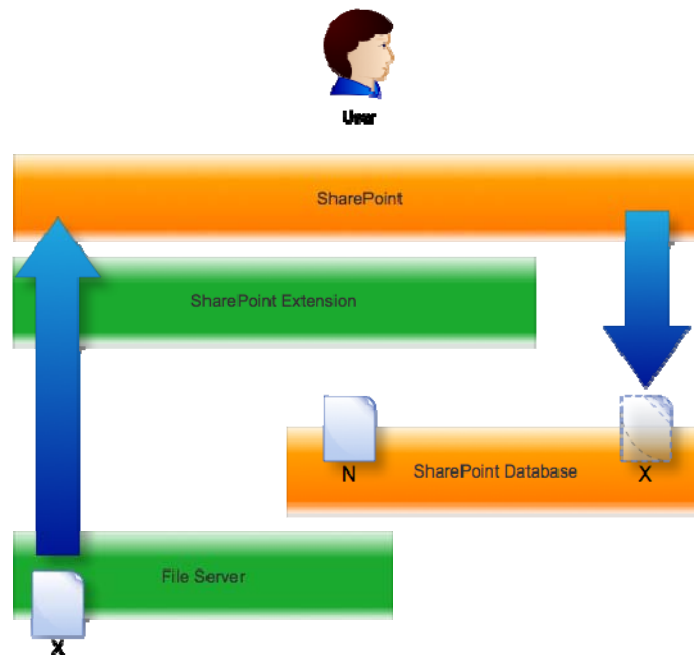
So let's talk about creative approaches. In general, these are things you'll need third-party software to implement; we're not talking about native SharePoint capabilities. In general, there are two things to discuss: Content connectors and the special considerations necessary for those large, streaming-media files.

### Content "Connectors"

A content connector is designed to take external data—living, perhaps, on a file server, FTP server, or some cloud-based storage—and *present* it through SharePoint. Done correctly, this meets all of our requirements: We get to leave our external content outside SharePoint, for a trimmer SharePoint database, but we get all the benefits of having the content "in" SharePoint, like search, workflow, alerts, and so on.

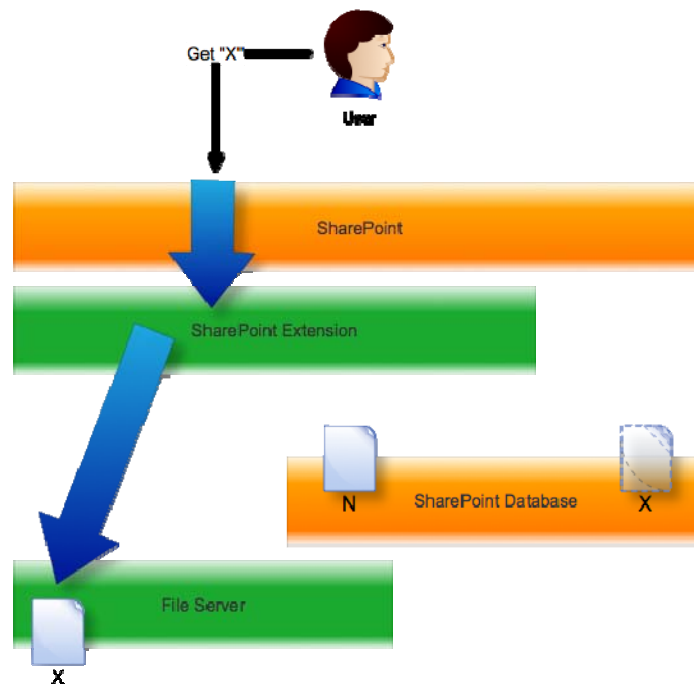
There's a trick, here: It's still possible to access the content from its original location. That means there has to be some kind of two-way synchronization, so that changes in either SharePoint or the native location will be synchronized to the other. For example, if someone changes a filename in SharePoint, we want that mirrored on the file server where the file actually lives. One way to accomplish that synchronization is to have a tool that can monitor both SharePoint and the content's native location for change events. This is a superior approach to time-based synchronization, which simply checks for changes at a fixed interval.

Architecturally, this is accomplished through an extension to SharePoint itself. That extension receives or gathers information about external content, and populates SharePoint's database with the information. Figure 3.9 shows how this might work. Note that I'll use "N" to designate content natively stored in SharePoint's database, and "X" to indicate external content.



**Figure 3.9: Incorporating external content into SharePoint.**

That information includes an identifier that the content is, in fact, external, so that when a user attempts to access the content, the SharePoint extension can physically retrieve it. Figure 3.10 illustrates.



**Figure 3.10: Accessing external content from SharePoint.**

The important fact to remember here is that the user is *only dealing with SharePoint*. Any application—like the Office suite—that works with SharePoint will work with this external content, because the content *appears* to be living in SharePoint. The extension simply makes the fact that it's really located externally transparent.

### Special Considerations for Media Files

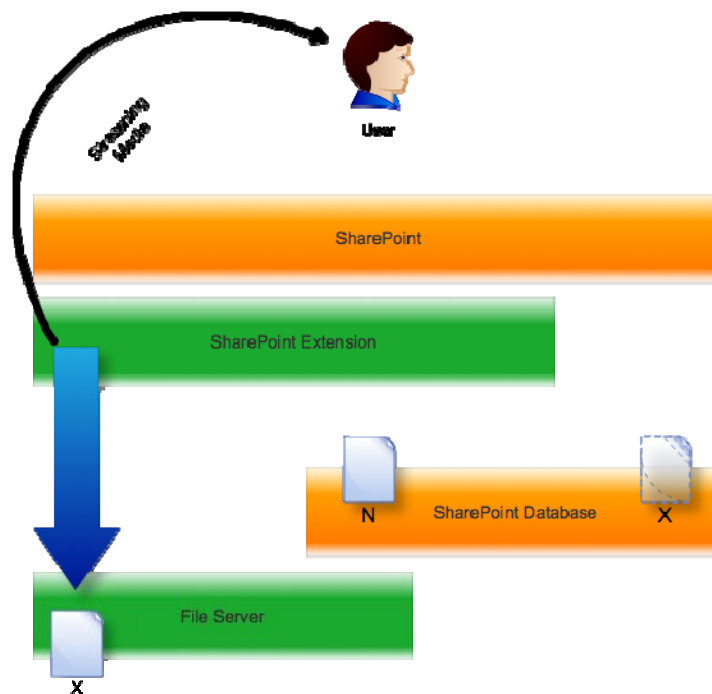
You could do that exact same “connector” trick for media files. However, there's a danger in doing so: You'll be putting a decent burden on SharePoint's Web Front End (WFE). Essentially, the WFE is going to become responsible for streaming the media content to the Web-based user. There's nothing *wrong* with that, since it's what Web applications are more or less designed to handle, but there *is* a better way. Simply bypass the WFE.

You still have a SharePoint extension operating, here. It gathers your external content, be it located in a file share, FTP server, or in the cloud—FTP and cloud being more important when it comes to media, since so much of it may be located externally.

#### Note

Make sure the SharePoint extension you select supports a variety of media types—WMV, WMA, MP3, AAC, VP6, MP4, MPEG, MPG, AVI, WAV, and so forth. You don't necessarily want to be “stuck” just doing Microsoft-friendly media types, or any other vendor's media types for that matter. You want support for them all.

By having the extension stream content *directly to the user's browser*, you offload some work from the SharePoint WFE. That helps keep performance high for other users, and makes SharePoint itself more scalable. Figure 3.11 shows how that might work.

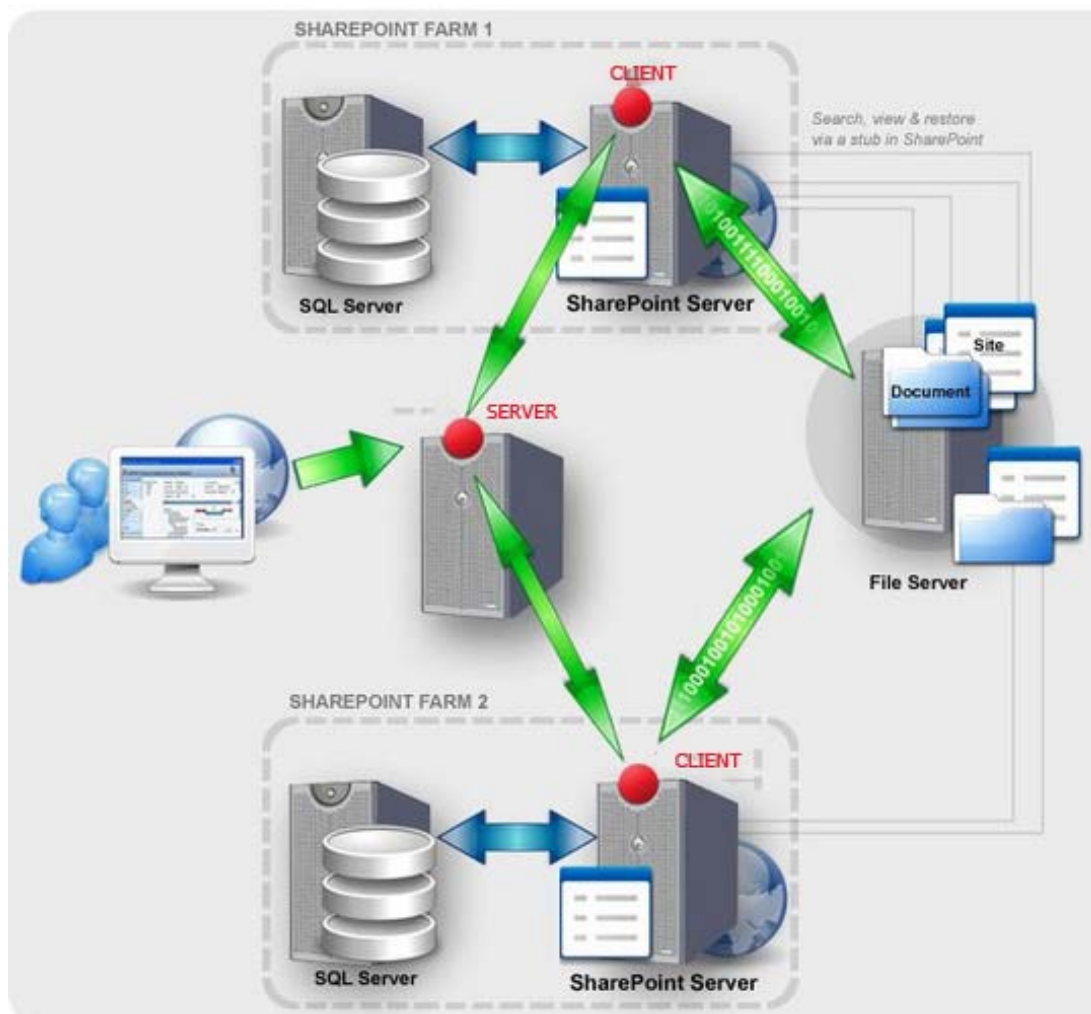


**Figure 3.11: Streaming media directly to the user, bypassing the WFE.**

## How Do You Do It?

There are a number of ways that you could physically implement such a connector. One way would be to simply install an extension of some kind on to each SharePoint server. That might not be optimal in high-load environments, though, since each server would be responsible for maintaining its own list of external content.

A better approach, shown in Figure 3.12, might be to adopt a client-server model. A central “external content server,” shown in the middle, would be responsible for loading “stubs” for external content into SharePoint databases. A client agent, running on each SharePoint server, would handle that communication with the central external content server. By populating SharePoint with the appropriate stubs, the system would enable SharePoint to discover, index, view, and manage the external content. The central server would also be responsible for detecting changes and synchronizing them as I described earlier.



**Figure 3.12: Implementing an external content connector.**

Obviously, this is just a rough sketch of how the implementation might actually work; each vendor producing such a solution would no doubt have a unique approach.

## Coming Up Next

Most businesses are awash in old data, much of which they're required to keep handy. It's a good thing we're closer to the "paperless office" than a decade or two ago because otherwise we'd be drowning in archived content. But "old data" doesn't mean "unneeded data;" you still may have to refer to that dormant, older content from time to time. Keeping it all in SharePoint is a great way to bloat your database, reduce performance, and consume costly storage resources—but removing that data from SharePoint makes the data more difficult to find and access. In the final chapter of this book, we'll look at ways to incorporate dormant and archived content into SharePoint, while still maintaining good performance and responsible utilization of storage resources.

## Download Additional Books from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this book to be informative, we encourage you to download more of our industry-leading technology books and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.