Realtime
publishers

Tips and Tricks Guide™ To

Managed File Transfer

Don Jones

Volume 3 1

Tip, Trick, Technique 13: What Deployment Models Are Available for Managed File Transfer Solutions? 1

On-Premises 1

Software as a Service 2

Hybrid Deployments 4

Tip, Trick, Technique 14: Can a Managed File Transfer System Integrate with My SEIM Systems? 5

The Push Option 6

The Pull Option 6

The Hybrid Option 7

Tip, Trick, Technique 15: Can I Deploy a Managed File Transfer System Quickly for a Project that Has an Immediate Need? 7

Long-Term Vision, Short-Term Implementation 8

Outsource It 9

Tip, Trick, Technique 16: Apart from My Specific Project Requirements, What Should I Look for in an MFT System? 10

Encryption Standards 10

Protocol Flexibility 11

Workflow Flexibility 11

Data Handling Flexibility 12

Interface Flexibility 13

Tip, Trick, Technique 17: What Will My Auditors Like or Dislike About Managed File Transfer? 13

Happy Auditors 14

The Devil's in the Details 16

In a Hosted Scenario 16

Tip, Trick, Technique 18: We Already Encrypt Files on Disk. Why Would I Need Managed File Transfer? 16

Copyright Statement

Volume 3

Each volume of this Tips and Tricks Guide will present a series of tips, tricks, answers, and best practices around Managed File Transfer.

Tip, Trick, Technique 13: What Deployment Models Are Available for Managed File Transfer Solutions?

Managed File Transfer (MFT) systems are powerful, often complex applications—but they're ultimately just server applications. There's nothing especially magic about their deployment, and with readily-available high-bandwidth connections to employees' homes and to corporate offices, MFT can be deployed in a number of ways.

On-Premises

The most straightforward deployment model, conceptually, is a traditional on-premises deployment, as pictured in Figure 13.1. Here, the MFT solution lives on a server in your own data center.

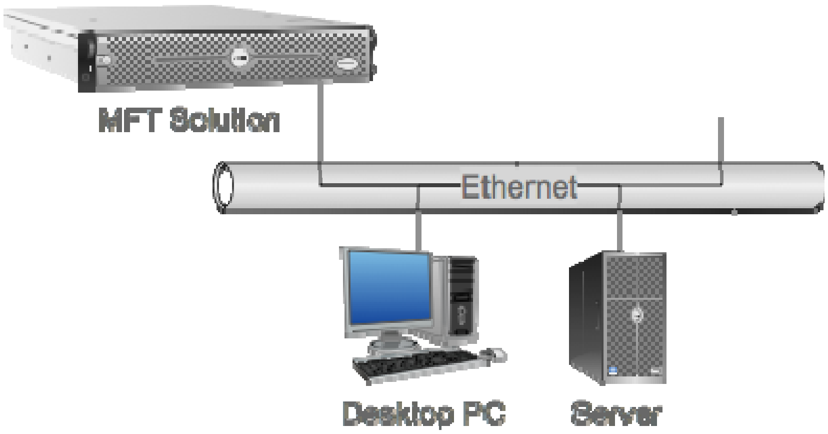


Figure 13.1: On-premises deployment.

There are obviously numerous advantages to this model:

- You're in full control of the MFT assets.
- You can build out options such as high availability to whatever degree you require.
- You're absolutely assured of the security and integrity of the system because it's completely under your control.
- You have the option to combine the MFT software with other server functions on a single physical machine or to virtualize the MFT solution as you see fit.

- Most of your hard costs are up-front, meaning that once you pay for the hardware and software, there are no additional cash flow needs—apart perhaps from software maintenance fees, which are usually annual and predictable.

There are, however, downsides to hosting the solution yourself:

- The capital needed to host the service—including server hardware, data center space, power and connectivity, cooling and other costs—are borne entirely by you.
- You'll have to spend more in order to scale the MFT solution should you outgrow the initial deployment.
- Creating a distributed or highly-available system will cost more both in hardware and in software licensing.
- It's yet another piece of hardware, another operating system (OS), and another piece of software that your IT team will need to patch and maintain over time.
- It's another failure point, meaning you'll have to maintain the hardware, potentially deal with outages, and so forth.
- Deployment often takes longer because you'll have to deploy an entire server.
- It becomes another point in your disaster recovery plan that has to be dealt with in case of a partial or total disaster in the data center.
- If you're not going to be fully-utilizing the MFT solution eventually, then you'll be paying for excess capacity.

Many companies continue to opt for on-premises deployment of MFT solutions simply because it's a familiar, well-understood model. Particularly for companies with security concerns, such as those dealing with industry or regulatory compliance, the self-hosted model offers a level of control and comfort that can't be had any other way.

Software as a Service

Many MFT vendors offer their solutions in an outsourced, or Software as a Service (SaaS), deployment model. As Figure 13.2 shows, in this setup, the solution is hosted at the vendor's data center, and you simply consume MFT as a service, much as you might consume CRM services from an SaaS company like SalesForce.com.

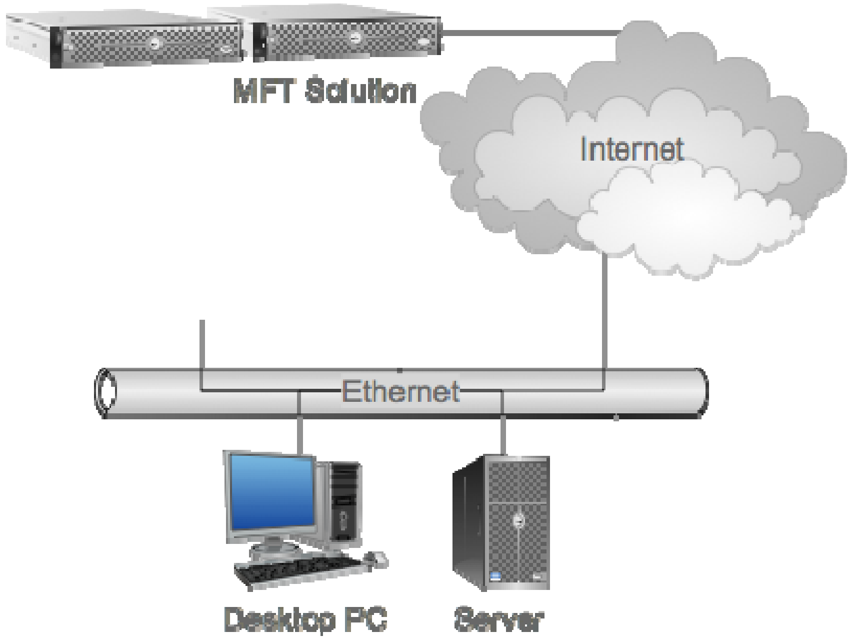


Figure 13.2: SaaS deployment for an MFT solution.

There are, of course, advantages to this model:

- There's typically no capital expense. You either pay flat service rates or you pay for your actual usage.
- Scalability is typically included, meaning that the hosting vendor has to scale their capacity to meet client demand but that your pricing is typically fixed or based on your own consumption.
- High availability is typically built-in to the pricing model.
- You'll have no hardware to maintain, no software to patch, and no OS to maintain.
- There's no additional failure point because the hosting vendor is responsible for whatever level of service is specified in your SLA with them.
- The solution can often be deployed in a few days.
- The solution is available even if you experience a disaster in your data center, easing the management of your disaster recovery plan.
- You're often able to secure pricing that meets your specific point-in-time needs, then buy more capacity as you need it. This can save money over buying a solution outright and hosting it yourself.

Of course, with all of those benefits, there are obviously disadvantages:

- You're not in control of the MFT infrastructure. You need to be comfortable with your vendor's management and maintenance practices.
- You have to be happy with whatever high-availability options your vendor offers, although SLAs typically ensure the availability of the MFT service.
- You have no direct control over the system's security, although good MFT systems can be configured to retain little or no data, and you typically have control over options like transmission encryption.
- You'll be continuously paying for the service rather than making a one-time up-front capital investment. For companies with variable cash flow, this may not be an optimal financial position. Note, however, that some on-premises solutions come with annual maintenance fees, so they aren't always just a one-time investment.
- The MFT market is full of startup companies, and these can obviously experience problems of their own—which could potentially leave you scrambling to replace your MFT service.

The SaaS model is ideal for companies who don't need or want constant, precise control over their MFT solution, and who would rather have a fast deployment and less ongoing maintenance.

Hybrid Deployments

Hybrid deployments seek to combine the best aspects of a self-hosted and SaaS solution. As Figure 13.3 shows, you typically have an on-premises solution, which is often backed up by a hosted solution. There are numerous other ways to create a hybrid deployment, but this is the one we'll examine in this book.

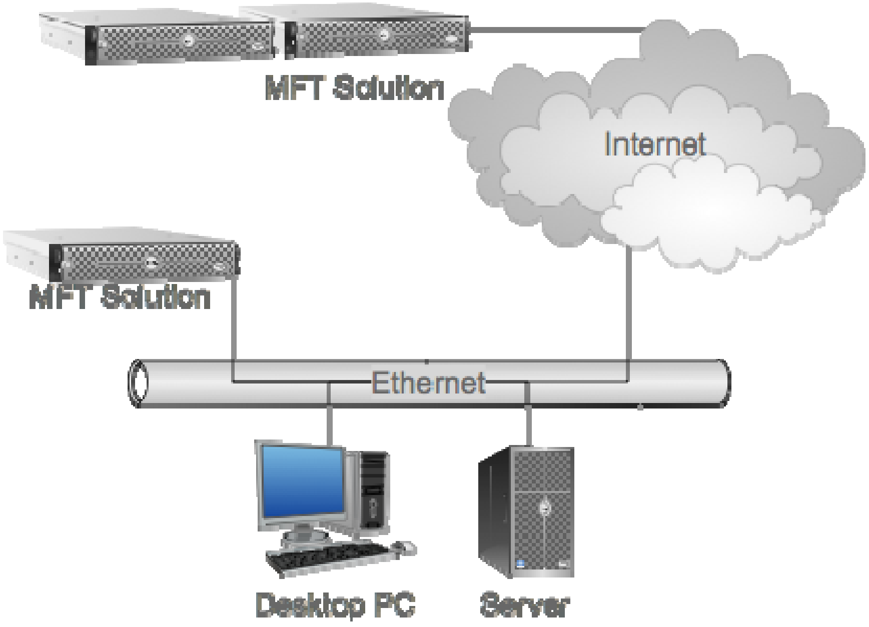


Figure 13.3: Hybrid MFT deployment example.

This type of deployment can help mitigate some of the biggest disadvantages of both the SaaS and self-hosted models. The main advantage is in high availability and disaster recovery: If you're primarily using the hosted solution as a backup, then you've got built-in disaster recovery. If your data center is hit by a meteor, you simply fail over to the hosted solution. In many cases, the hosted component can also serve as a scalability buffer, providing extra on-demand capacity in the event that you need it.

There are still disadvantages:

- You're paying for two solutions. This model can still be feasible if the hosted component has minimal fees when you're not using it, and charges you for the capacity you actually use.
- You still have potential security and control issues when the hosted component is utilized, and you'll need to work with your vendor to understand how those issues can be addressed.
- Because you aren't using the hosted component continuously, it's possible for the vendor to underestimate the capacity you'll need. If you wind up needing to fail over to the hosted component due to an on-premises failure, the hosting vendor might not be able to provide the capacity you need on demand.

Again, there are numerous ways that a hybrid deployment can be used aside from providing a disaster recovery backup, which is how I've outlined it in this example. The moral is to discuss this kind of model with your MFT vendor, see what they can offer, and see how their offering might help address your specific business needs.

Tip, Trick, Technique 14: Can a Managed File Transfer System Integrate with My SEIM Systems?

SEIM stands for *Security Event Information Management*; you'll also see it written as SIEM, or *Security Information Event Management.* Sometimes it's *Security Incident Event Management* and *Security Event & Incident Management.* Regardless of how you define it (I'll use SEIM), these systems are essentially designed to centralize security audit log information from across your enterprise into a single tamperproof or tamper-evident database that can be used for reporting and analysis. These systems often provide event correlation capabilities, enabling them to pull together multiple seemingly-unrelated events into a cohesive picture that can, for example, alert you to suspicious activities or attacks that are presently underway. *Incident detection* is another term you'll hear or see used alongside SEIM.

An SEIM system is designed to store security audit information, and a Managed File Transfer (MFT) system can generate a lot of security audit information. You typically want all of your security audit information in one place (which is why you bought or built an SEIM system in the first place), so it'd be ideal if the MFT solution could integrate with your SEIM system. How might that happen?

The Push Option

One option is for the MFT system to send, or *push*, audit data to the SEIM system. This setup might be accomplished through the use of numerous application programming interfaces (APIs), but one standard and widely-adopted method is the Unix syslog protocol. Provided your SEIM system can act as a syslog server, and the MFT solution can send syslog messages, then you're good to go. Figure 14.1 illustrates this approach.

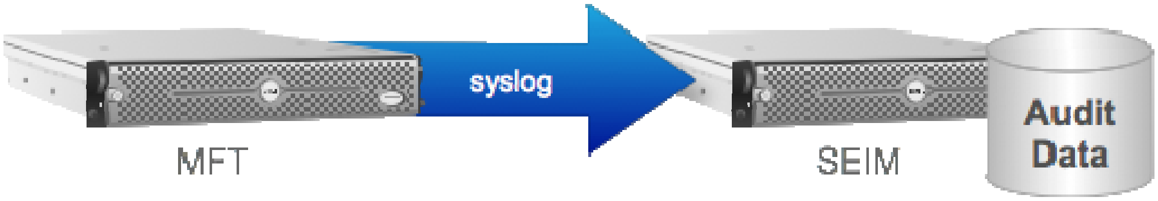


Figure 14.1: Pushing data to the auditing system.

Other protocols might be used. For example, if your MFT solution utilizes a RADIUS server for authentication, it may also be able to send auditing messages to the RADIUS server, which would be responsible for storing them. That might be another way to get auditing information into your centralized SEIM database.

The Pull Option

Most SEIM systems are capable of pulling text-based log files from different systems, and an MFT system that writes its audit data to a log file is certainly a candidate. Figure 14.2 shows how this might work. This is often a less real-time approach to audit log collection because the SEIM system typically pulls the log files on a periodic basis rather than continually receiving real-time audit messages directly from the MFT system.

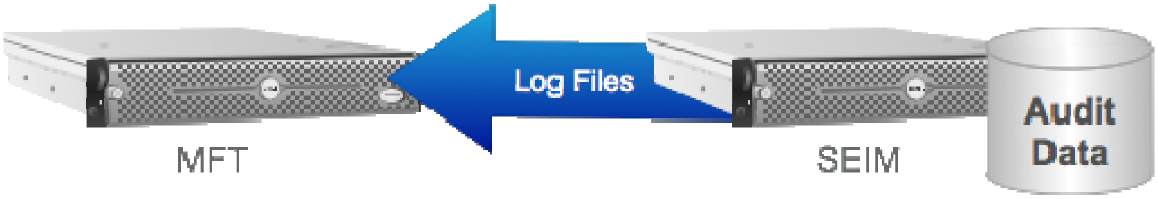


Figure 14.2: Pulling data from the MFT solution.

A pull approach can be used with any log format that the SEIM system and MFT solution have in common. For example, Windows-based MFT solutions might write auditing information to the native Windows event log, and most SEIM systems are capable of pulling events from those logs.

Some SEIM systems utilize locally-installed agents to collect data and send it to the centralized databases. These agents can typically provide more real-time data than an SEIM system that passively pulls logs because the agents can have local awareness of when new messages are written to the log. They can immediately pull the updated log and send the new information to the database. Not necessarily real-time, but awfully close to it.

The Hybrid Option

There are instances when a combination approach might be needed. For example, an MFT solution capable of sending syslog messages might send them to a standalone syslog server. That server might then be polled by your SEIM system, acting as a kind of auditing middleman. Figure 14.3 illustrates this approach.

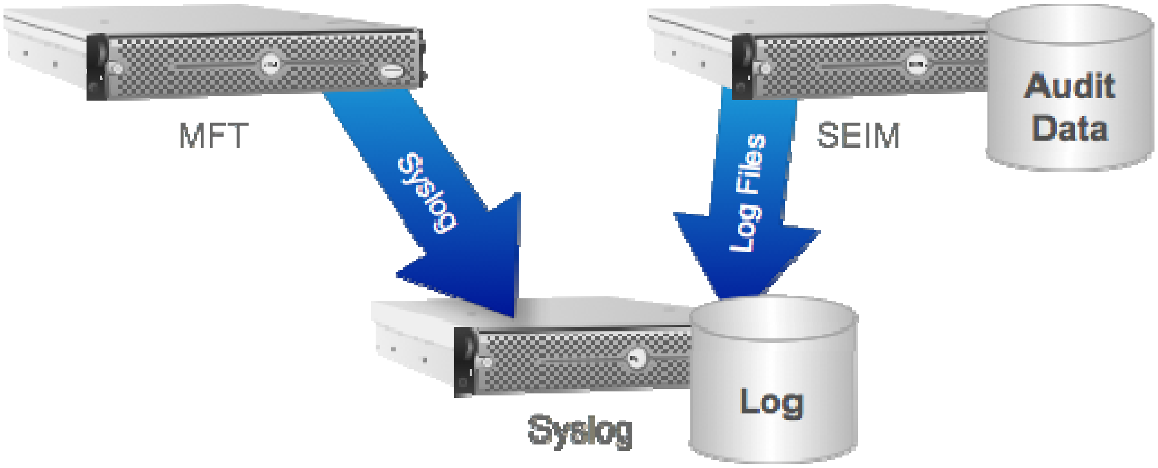


Figure 14.3: Using syslog as a middleman for auditing data.

In the end, there are numerous ways for data to get from your MFT system to a centralized SEIM database. Talk to your MFT solution vendor about the methods they support, and look at your SEIM system to see which of those methods it might be able to utilize. If you're considering an MFT solution that simply doesn't expose its audit log data in a way that can be captured by an external system, well, in that case, you might want to stop considering that particular solution.

Tip, Trick, Technique 15: Can I Deploy a Managed File Transfer System Quickly for a Project that Has an Immediate Need?

There's a general perception that any server software must take a long time to set up and deploy. With Managed File Transfer (MFT) solutions, that's rarely the case—but simply the process of getting server hardware, finding room in the data center, getting approvals for all of the purchases (such as an operating system—OS—license) and other details *can* take a long time. That's not the software's fault, of course, but there it is. Aside from those details, however, there are some tricks for getting an MFT solution up and running quickly.

One thing I'll offer as a caution: *Don't* assume that your immediate project needs are the only ones you'll ever have for an MFT system. Read Tip 16 for more information, and consider planning for a solution that will meet companywide needs—even if all it needs to do *today* is meet the needs for a single project.

Long-Term Vision, Short-Term Implementation

One way to get an MFT solution up and running more rapidly is to think about your long-term vision for the solution, then only implement the portion of it that you have the most immediate need for. For example, consider Figure 15.1. Here, the vision is for a redundant, highly-available installation that includes two servers. There's also a plan to implement a Web-based client side for ad-hoc person-to-person transfers, and to deploy a custom file transfer agent to company desktops.

The most *immediate* need, however, appears to be for a single, non-redundant MFT server that handles only server-to-server transfers. The other portions of the project are marked as "Phase II," indicating that they'll come later. This is a great way to meet a project's immediate needs, while planning for a more robust, flexible infrastructure for the future.

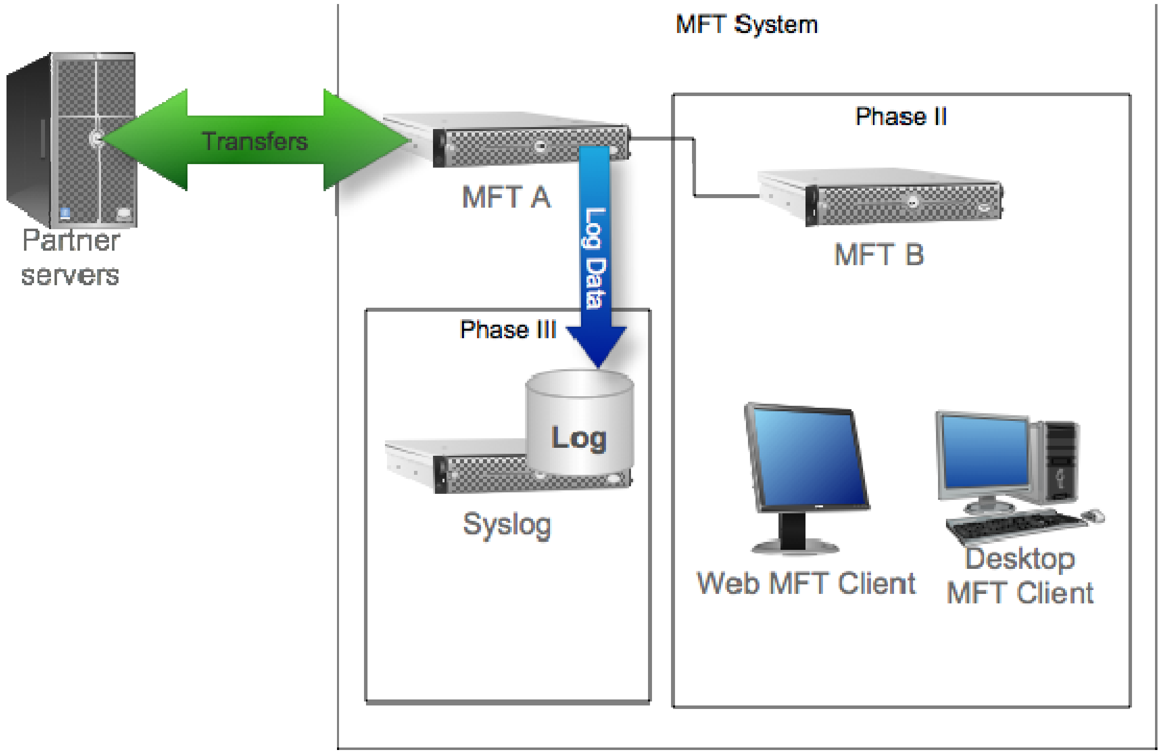


Figure 15.1: Phased deployments get you up and running sooner.

A "Phase III" is also listed, and appears to include the addition of a centralized security audit log by means of a syslog server. Again, if that's not a critical component of the immediate project, it can be planned for a later phase so that the immediate need can be met more quickly.

The trick with this—as will be discussed in Tip 16—is to *plan* for the big, long-term picture. If you only plan for the immediate project need, you might not have the *option* to add a redundant server in Phase II, or you might select a solution incapable of supporting your syslog plans for Phase III.

Outsource It

Of course, the other option is to find an MFT vendor that supports a Software as a Service (SaaS) hosted solution. In this case, you can usually be up and running in a few days—if not hours—to meet those immediate project needs.

However, unless you've looked carefully at the full range of your company's potential requirements and are satisfied that a hosted solution will work for you permanently, *avoid* going with an MFT solution that is *only* available as a hosted offering. Instead, find a vendor that sells a traditional self-hosted solution and makes that solution available in an SaaS offering. That way, you'll be able to more easily migrate from the outsourced solution to a self-hosted system in the future.

In fact, it's wise to discuss that very migration with your vendor *before* you begin hosting with them. How hard is that migration? If you've taken the time to set up custom workflows, data translation, and other features, will those be easy to move from the outsourced platform to the self-hosted one when the time comes? Or will you be re-doing a lot of that work—wasting time and effort in the process? In some cases, it may be as simple as copying a database or a set of files from the hosted solution into your data center to bring all of you configuration work in-house; in other cases, a more complex process might be involved. Either way, you'll want to make sure the vendor supports you, and that you understand exactly what the process will look like.

In my capacity as a consultant, I've worked with clients who had a truly *immediate* project need for MFT. They researched available solutions, and chose a vendor that offered the same solution both as an in-house and an SaaS option. They determined that migrating from one to the other was as easy as copying a few files, and that the vendor had a documented, supported process for doing so. They started the SaaS MFT offering immediately, were online within a couple of days, and within a week had created the workflows and other configuration elements that their project needed. Once that project was off the ground, they immediately began implementing the in-house solution. Because the critical project's needs were already being met, they were able to deploy and extensively test the in-house solution, test the migration process from the SaaS system, and eventually plan a seamless cutover to the in-house solution. This particular client decided to retain a portion of the SaaS offering as a backup to their own in-house solution but ran most of their daily operations from the MFT solution they'd installed in their own data center.

This is definitely an effective approach, provided everything comes together to meet all of your business needs—both immediate, and long-term.

Tip, Trick, Technique 16: Apart from My Specific Project Requirements, What Should I Look for in an MFT System?

As I stated in Tip 15, you should *always* evaluate Managed File Transfer (MFT) systems with an eye toward your organization's *total* needs rather than the needs of a specific project that might be driving an immediate MFT acquisition or deployment.

Although you can absolutely focus on getting just the immediate-need functionality and capabilities online first, having a solution that can grow to accommodate the entire organization is a wise move. *Most* organizations that I've consulted with start with a specific project in mind; almost all of them eventually want to use MFT for more than just that project. Too many of them acquire an MFT system based solely on that initial project's requirements, then end up acquiring a *different* system for the next project, eventually resulting in a sprawl of disparate systems. The more you can think ahead to what the company might need, the more you can focus on a solution that will grow to meet those needs.

In general, you want a solution that offers *flexibility.* That way you don't necessarily have to accurately predict every future need, and can instead rely upon a well-equipped system simply being able to do whatever those future needs might entail.

Encryption Standards

This is an area where you don't want to skimp. Perhaps your current project doesn't involve or require encryption, or perhaps the project requirements don't call for especially strong or powerful encryption because the data being transferred isn't especially sensitive. Fine—but don't choose an MFT solution based on that criteria. Your organization *will* need better encryption someday—even the current project might find its requirements redefined by outside influences like legislation or industry requirements.

The easy, future-proof choice is to select an MFT solution that has independently-certified encryption technology, preferably carrying a FIPS 140-2 validation certification. There are two reasons for that: First, it's the best encryption money can buy. Second, that validation more or less forces a vendor to build their cryptography software as a module, meaning it's easier for them to pull that module and replace it with a better one if encryption standards evolve in the future (which they definitely will do). So you're a little more future-proofed because the vendor will have an easier time updating the code without forcing you to re-deploy an entirely-new solution.

Protocol Flexibility

Whatever you do, choose an MFT solution that offers the maximum number of transfer protocols possible. That way, you're never caught off-guard when some new requirement comes along or you have to deal with a business partner whose system is more limited in its protocol selections. That means choosing a solution that—at a minimum—supports:

- FTP
- SSL/FTPS
- SSH/SFTP/SCP2
- HTTP (and HTTPS)
- SMTP/POP
- CIFS/SMB
- EDINT AS1, AS2, and AS3

Some solutions may make some protocols available as an optional module, which is fine. So long as you have the ability to handle these protocols, you'll likely be able to handle whatever business situation comes your way in the future. A solution that covers all of these is also being produced by a vendor who is intent on providing good protocol coverage; even if someone dreams up some new protocol, a broad-coverage solution will likely be updated to include that new protocol if it catches on in the business world.

Workflow Flexibility

Throughout this book, I'll keep coming back to the idea that MFT is more than just about file transfer. In fact, the "FT" part is really the least technically-challenging part of an MFT solution; it's the "M" that's hard. *Managed* file transfer solutions not only send files from place to place but also move that file through your own systems to complete business processes. It's rarely enough to just have a file plopped into a directory and left there. More commonly, you'll want the file processed in some way. Vendors refer to that capability by various terms, including *integration, coordination,* and so on. I'll use *workflow* because we're using the MFT solution to implement a broader process, typically based on a predefined flow of actions.

So what should you look for in terms of workflow capabilities?

- As little coding as possible. Many solutions now allow you to build fairly complicated processes that require no code, or maybe only a minimal amount of code to customize a particular action. Less coding gets you up and running more quickly.
- Standards-based. Application programming interfaces (APIs) should use standard, industry-wide languages such as C or Java, and data should flow in industry-standard forms like XML.
- Customizable. I've seen some low-end MFT solutions whose idea of "workflow" consists of a few fixed steps. You don't want that. You want a system that offers a blank slate to create whatever kind of process you want. If a system comes with template or example workflows that you can start with, great, but you don't want to be limited by a vendor's idea of how your company operates.

Data Handling Flexibility

Think about what we've covered so far, process-wise: We can transfer files from place to place. We can maneuver those files through a process workflow. At some point, that workflow is likely to involve translating or transforming the data in some fashion, or possibly even doing data validation.

Most companies would be perfectly happy with an MFT system that could simply invoke external components to do data translation, and to then write or acquire those components separately. MFT solutions are, however, evolving to include that capability on their own.

And why not? If the goal is to truly *manage* your data throughout its life cycle, simply receiving it and handing it off to another process isn't covering the entire life cycle. Handling the translation of that data so that it can be handed off to your in-house systems—or translating it so that outside partners can make us of it, or even translating it *between* two different in-house systems—is a sensible part of the data's life cycle, and the right MFT solution can help you do it.

Translation (or transformation, which is a more accurate term) typically takes the form of predefined data maps. "US should be USA" and "CA should be California" are simple examples of data maps that an MFT solution could implement for you. Some MFT systems can also implement rules, kicking out data (and notifying someone) that doesn't meet certain standards.

The trick with this kind of translation is in how the software is actually written. If it's not in a fairly low-level, native language (think C++ or C), it probably isn't going to perform well on massive amounts of data. As part of your evaluation process, make absolutely certain to test the performance of the translation software by giving it a representative set of data in a trial run.

Interface Flexibility

Interface flexibility is the biggest area where I see companies fail to plan for the future. MFT often comes into the organization to solve, as I've said, a specific project need. That project almost always involves server-to-server transfer requirements. It's often transferring data on a schedule between the company and a business partner. That's great, it's what MFT does well, and it's a pretty straightforward scenario with which to get started in MFT.

But it's hardly the only type of file transfer companies need. Ad-hoc transfers, whether server-to-server or person-to-person or person-to-server, are just as important. In fact, *many* companies who start with simpler server-to-server transfers often find that their person-to-person traffic is by far the bulk of the data their MFT solution handles—and few of them expected that or even identified person-to-person transfers as a serious business need.

Here's what happens, though: The company gets started with MFT, doing those server-to-server transfers. They eventually realize that more of their data should be transferred that way—secured, guaranteed delivery, fully audited, and so on. More business processes move to the MFT server. Before long, someone wonders why users are using their non-secure, non-guaranteed, non-audited email as a way of moving files between users. They quite rightly look at the MFT solution and wonder why it can't help manage, secure, and audit those "file attachments" as well. With the right MFT solution, *it can do so.*

That's why *even if your immediate need doesn't include person-to-person transfers,* you should plan for them. Select an MFT solution that not only supports them but that does so *well.* You definitely want a system that can provide a Web-based user interface so that external users can send files into the organization or pick up files sent to them from someone inside the organization. Some MFT systems *additionally* offer rich client software—perhaps a standalone utility or even an Outlook plug-in—to help internal users better take advantage of these ad-hoc, person-to-person transfer capabilities. Outlook plug-ins are really ideal because they provide users a means of accessing ad-hoc transfers that still looks and feels a lot like the "file attachments" they're already accustomed to.

Tip, Trick, Technique 17: What Will My Auditors Like or Dislike About Managed File Transfer?

Initially, auditors tend to be a little wary (if not outright suspicious) of anything new. In fact, if you're still in the process of evaluating Managed File Transfer (MFT) solutions, it's wise to involve your auditors right from the outset so that their concerns can become an integrated part of your evaluation and acquisition process.

That said, most modern MFT solutions offer plenty for auditors to love. Keep in mind that most auditors are dealing with pretty strict security and auditing requirements. Their job is generally to make sure that the right data is being collected in order to prove compliance with those requirements, and that the auditing data itself is secured and either tamper-proof or tamper-evident. Once they're happy with the condition of the auditing information, they use that information to ensure compliance with whatever requirements they're dealing with.

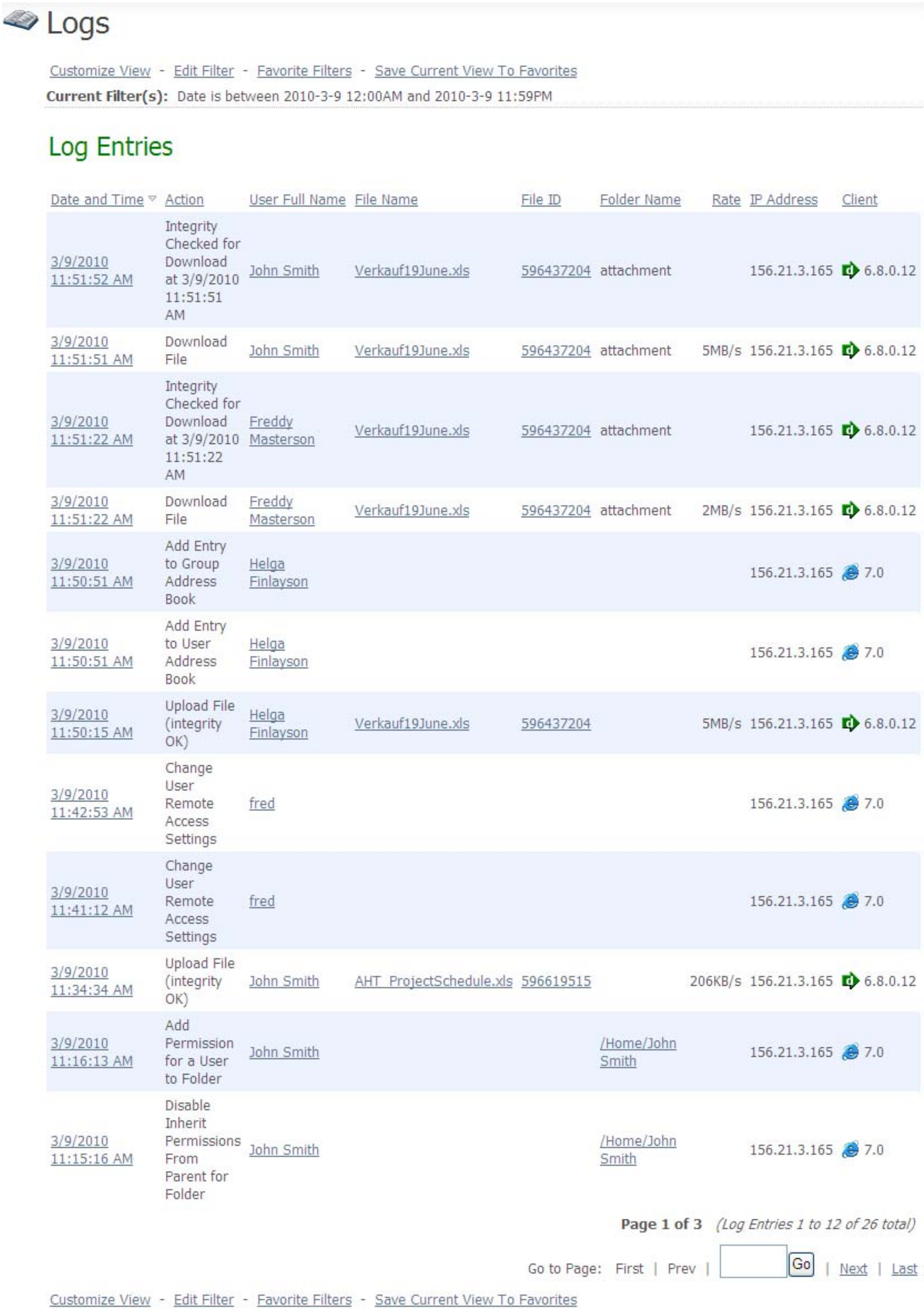
Happy Auditors

Here are some of the major things that a good MFT system will offer that auditors will enjoy:

- Audit records of every file transfer. In some cases, depending upon the protocols and software used, this will include audit records proving receipt of the file as well.
- Audit records of MFT-related activity. This usually includes logons and logoffs, permissions changes, configuration changes, new user account creation, and other activity that occurs entirely within the MFT solution itself.
- Audit records of the security settings—such as encryption level—for each file transfer.
- For solutions that are also coordinating data through a complete workflow process, which may even include translation activity, the solution will generally log each step.
- For solutions that can be configured to securely wipe data after it has been delivered, limit the number of deliveries for a file, or other options, the audit log will typically include details on those activities, enabling an auditor to confirm that they are in fact taking place as configured.

These details are available for *every* file transfer—including transfers between individual users who use an MFT solution's ad-hoc transfer capabilities. That's one reason auditors typically love MFT systems: Auditors get a *lot* more insight into those ad-hoc transfers than they did when those transfers were being conducted by email file attachment.

MFT solutions will vary in how they *expose* this data. Some will offer formal reports, others will require you to write a small script that queries the audit information from a database. Figure 17.1 shows one example.



Logs

Customize View - Edit Filter - Favorite Filters - Save Current View To Favorites

Current Filter(s): Date is between 2010-3-9 12:00AM and 2010-3-9 11:59PM

Log Entries

Date and Time	Action	User Full Name	File Name	File ID	Folder Name	Rate	IP Address	Client
3/9/2010 11:51:52 AM	Integrity Checked for Download at 3/9/2010 11:51:51 AM	John Smith	Verkauf19June.xls	596437204	attachment		156.21.3.165	6.8.0.12
3/9/2010 11:51:51 AM	Download File	John Smith	Verkauf19June.xls	596437204	attachment	5MB/s	156.21.3.165	6.8.0.12
3/9/2010 11:51:22 AM	Integrity Checked for Download at 3/9/2010 11:51:22 AM	Freddy Masterson	Verkauf19June.xls	596437204	attachment		156.21.3.165	6.8.0.12
3/9/2010 11:51:22 AM	Download File	Freddy Masterson	Verkauf19June.xls	596437204	attachment	2MB/s	156.21.3.165	6.8.0.12
3/9/2010 11:50:51 AM	Add Entry to Group Address Book	Helga Finlayson					156.21.3.165	7.0
3/9/2010 11:50:51 AM	Add Entry to User Address Book	Helga Finlayson					156.21.3.165	7.0
3/9/2010 11:50:15 AM	Upload File (integrity OK)	Helga Finlayson	Verkauf19June.xls	596437204		5MB/s	156.21.3.165	6.8.0.12
3/9/2010 11:42:53 AM	Change User Remote Access Settings	fred					156.21.3.165	7.0
3/9/2010 11:41:12 AM	Change User Remote Access Settings	fred					156.21.3.165	7.0
3/9/2010 11:34:34 AM	Upload File (integrity OK)	John Smith	AHT_ProjectSchedule.xls	596619515		206KB/s	156.21.3.165	6.8.0.12
3/9/2010 11:16:13 AM	Add Permission for a User to Folder	John Smith			/Home/John Smith		156.21.3.165	7.0
3/9/2010 11:15:16 AM	Disable Inherit Permissions From Parent for Folder	John Smith			/Home/John Smith		156.21.3.165	7.0

Page 1 of 3 *(Log Entries 1 to 12 of 26 total)*

Customize View - Edit Filter - Favorite Filters - Save Current View To Favorites

Figure 17.1: Example audit log data.

The Devil's in the Details

All in all, there's not much for an auditor to dislike about an effective MFT system. There are, however, subtle differences between solutions that might well make your auditors cringe just a bit. For example, *where* the audit log is stored can be a point of concern. Solutions that use the Windows event logs, for example, are not preferred. That log system isn't designed for high volumes of audit records. Windows event logs aren't natively centralized, meaning you'll have to have something in place to consolidate logs. Administrators can also tamper with the Windows logs fairly easily, and can do so without leaving a trail—and that's one of an auditor's biggest concerns.

Instead, look for an MFT solution that offers an option to log to an external database. That auditing database can be more easily secured, this making it at least tamper-evident, and should be an inherently centralized place where multiple MFT servers could write auditing data, if desired (check with your vendor to make sure that's the case—you don't want to deal with multiple audit databases if you can avoid doing so).

In a Hosted Scenario

Keep in mind that even the best MFT system, from an auditing perspective, may "look" different if it's hosted by someone other than yourself. For example, you may need to explicitly download audit logs, and log retention may be limited (180 days is a common limit). That's not necessarily a deal-breaker, but it is a natural part of the lesser control that you typically get when someone else is maintaining the MFT solution. You can work around any such limitations by making sure you have a process in place to obtain and securely archive any logs that need to be retained for a longer period of time.

Tip, Trick, Technique 18: We Already Encrypt Files on Disk. Why Would I Need Managed File Transfer?

You need Managed File Transfer (MFT) because your data lives in many more places than just on disk. And it may even be living on disks you haven't thought about. Take a look at Figure 18.1, which illustrates a fairly simple server-to-server file transfer operation between two organizations. By the way, this same scenario could apply to a purely-internal transfer of data between two independent systems within the same organization—something MFT systems are often used for.

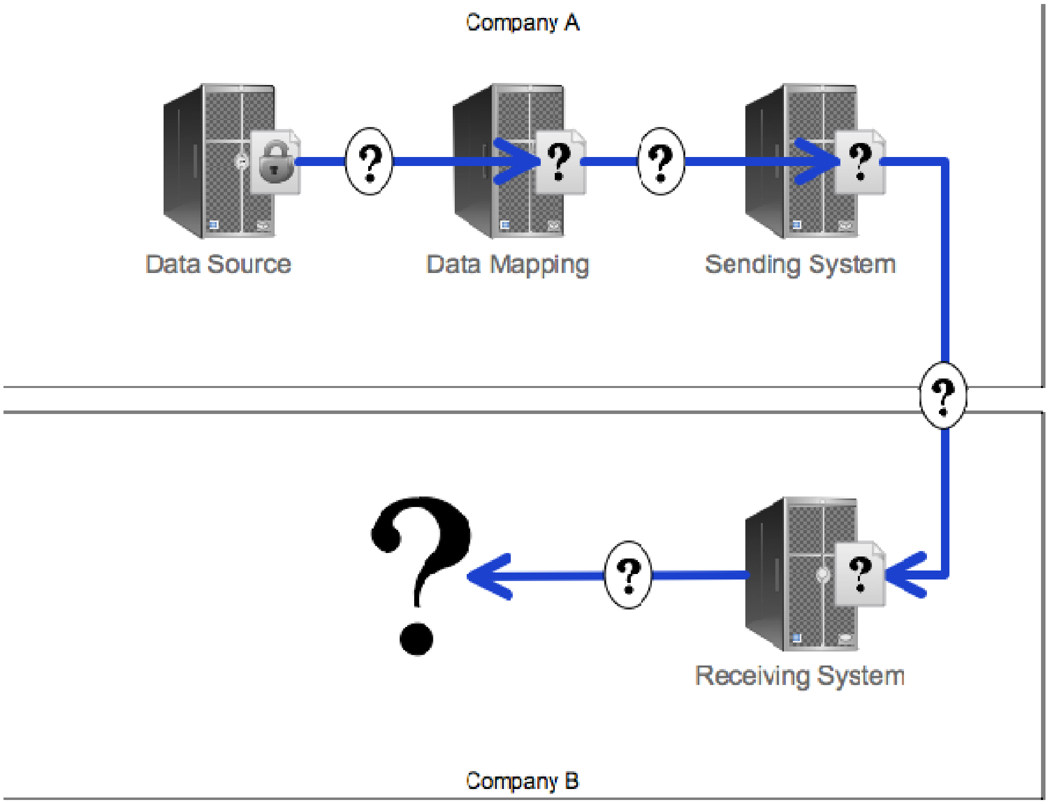


Figure 18.1: Transferring files between organizations and systems.

This example is assuming that you're making minimal use of an MFT system, or that you're actually just using a plain-old FTP server rather than a full MFT solution. Here's what's happening:

1. As shown, the file starts on a data source, where it is indeed encrypted on disk.
2. The file is then transferred—perhaps to a data-mapping and data-translation component. Whether the file remains encrypted during this transit depends. Using a disk encryption system like Windows' EFS or BitLocker, the file is unencrypted as it moves across the network.
3. The file is unencrypted as it is translated/transformed. The translating server will likely store at least temporary files on-disk, and those might or might not be encrypted depending on whether you've enabled disk encryption on that machine.
4. Transmitting the data to the sending server (say, an FTP server) might also take place over an unencrypted channel.
5. The sending server will store a copy of the file on disk, and that might not be encrypted if you haven't enabled disk encryption on that machine.
6. The FTP server might or might not encrypt the data in-transit; the normal FTP protocol doesn't do so.
7. The receiving system is then entirely on its own, of course, and you can't control what happens to the data at that point.

Let's contrast that with what happens when a full MFT system is properly deployed and utilized. Figure 18.2 shows this new model.

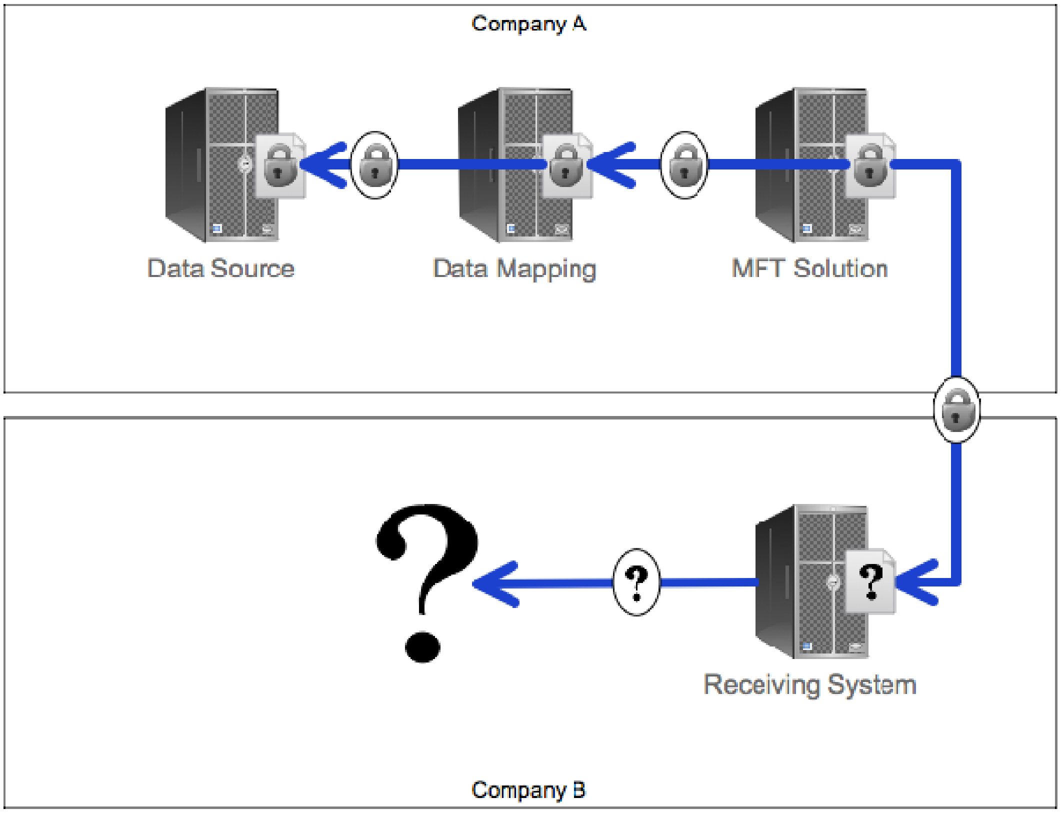


Figure 18.2: Moving data through an MFT solution.

1. Here, the MFT solution *pulls* the data from the data source, over a secured connection.
2. The MFT solution coordinates the data translation or mapping, and can ensure that temp files are in fact encrypted on disk.
3. Transmission to the final sending server (assuming it's a different machine than what handled the data mapping, which isn't necessarily going to be the case) would also be over an encrypted connection, initiated and managed by the MFT solution.
4. The MFT solution encrypts its files on-disk, and can even securely wipe them once sending is complete.
5. The file is transmitted over an encrypted connection, using whatever protocol you've configured.
6. You still can't control what happens on the receiving system—but the file remained encrypted the entire time it was in your hands.

So the short answer is that on-disk encryption is great—but companies don't necessarily deploy that to every single machine. And, even if they do, data is generally transmitted in the clear, which defeats the whole point of encrypting the data on-disk in the first place. By letting an MFT solution manage the entire process, the data can remain encrypted through every step, on every disk, and during every transmission—and the MFT solution can generate audit log information to help prove that the data was properly handled during the entire process.