Realtime publishers

Creating Unified IT Monitoring and Management in Your Environment

Don Jones

sponsored by

Copyright Statement

Chapter 6: Unified Management, Illustrated

In this final chapter of the book, I want to revisit everything from the first five chapters. However, I'm going to do so in the form of case studies. I've been fortunate enough to speak with several consulting clients of mine who've been struggling with the same issues I've outlined, and who've recently been trying solutions that follow the basic approach I've described. They've agreed to let me share their stories (although they've asked that I not use their names or company names) so that you can get a before-and-after look at how this "unified management" thing should work. Along the way, I'll also share some of the challenges and roadblocks they've encountered. A switch to unified management isn't always going to be hassle-free, so I think it's valuable for you to see what they've had to deal with, and how they think they're going to do so.

This chapter will also include some of the practical information on unified management that hasn't made it into the previous chapters. I'll provide a consolidated shopping list of unified management features so that when you start examining solutions, you can have that list in hand to help you. I'll also look at different purchasing models that vendors are offering these days to give you an idea what kind of flexibility you might have for acquiring and implementing a solution.

The Case Studies

A unified management solution has to provide features for what I believe are two distinct broad use cases. The first is in helping you to react to problems, while the second helps you manage non-problem requests—such as requests for changes within the environment. I'm going to provide two distinct stories for each of these. They're actually both drawn from the same consulting customer, although you'll meet different people from those organizations in each narrative.

Detecting and Solving Problems

Lisa is a senior systems administrator, responsible primarily for the Windows-based systems in her environment. Her counterpart, Peter, is responsible for the company's Unix- and Linux-based server infrastructure. Both have considerable areas of overlapping responsibility, as many of the company's line-of-business (LOB) applications rely on both Windows- and *nix-based resources.

"It isn't just the servers, of course," Lisa told me. "It's what's running on those servers: databases, Web services, you name it. Someone else supports those different pieces, so there used to be a lot of time spent arguing about whose fault something was."

I asked her for an example of how things worked in their environment prior to implementing a unified management system. She laughed and brought out a file that she'd clearly held on to for some time. It looked like the text from a Help desk ticket's notes. Here's the complete text, with names edited; I've added some [editorial] notes for items that I had to ask Lisa to explain.

> OPENED BY HelpDesk AT 2009-06-14 13:34
> User states that BOS [an LOB application] is extremely slow. Have several e-mails about this in the q also. Server BOSDB02 responding slowly to pings.
>
> ASSIGNED TO LHarte [this is Lisa]
>
> NOTES BY LHarte AT 2009-06-14 15:26
> BOSDB02 is working fine, apart from the fact that SQL is hogging 100% of the CPU. Passing to DBA.
>
> ASSIGNED TO DShields
>
> NOTES BY DShields AT 2009-06-14 16:53
> Probably the indexes again, SQL is taking longer to complete queries than it should. Will schedule indexes to be rebuilt tonight
>
> NOTES BY HelpDesk at 2009-06-15 10:44
> Still getting calls on this
>
> NOTES BY DShields AT 2009-06-15 11:12
> Indexes rebuilt
>
> ASSIGNED TO HelpDesk
>
> NOTES BY HelpDesk AT 2009-06-15 11:34
> Still getting calls that BOSDB02 is still slow to ping
>
> ASSIGNED TO DShields
>
> NOTES BY DShields AT 2009-06-15 13:12
> SQL is still slow—looks like it is in disk IO. Fragmented disk? Need server support.
>
> ASSIGNED TO LHarte
>
> NOTES BY LHarte AT 2009-06-15 13:47
> Server disk shows less than 2% frag—not the problem. IO is slow because SQL is thrashing the disks. Maybe your DB is fragged. I'll call you.
>
> ASSIGNED TO DShields

The conversation clearly went offline at that point because the next entry simply indicates "problem resolved." Unfortunately, there was no official documentation of what went wrong or what was done to fix it, but Lisa explained. "We kept going back and forth between us—he'd see something in Performance Monitor that looked like the server was slow, and bounce it to me, and I'd tell him that it's because his SQL Server was causing the problem and bounce it right back. I don't even have permission to look inside SQL Server, and he just kept wanting to get the ticket out of his queue.

"In the end, it actually turned out to be a problem with the SAN, which was Peter's problem. Something had gone wrong with our main SAN connection and we were on a slower backup link, and something was wrong with that link's configuration, so it wasn't running at full speed or something. We were seeing it as slow disk IO because Windows obviously thinks that the SAN is just one big locally-attached volume. We were running all kinds of tests on the server and in SQL Server to try and find the problem, but none of our tools were able to realize that the real problem was further under the hood someplace."

Peter recalled the incident. "It was weird because there wasn't anything actually broken, so none of the tools I use to monitor the SAN gave off any alerts. The problem was a configuration problem on several of our hosts. The tools don't see that as broken, of course, although it was causing them to access the SAN a lot slower than they're used to.

"The real problem was that this cropped up on about seven machines all at once. We didn't correlate the problem at first, because every single machine was affected slightly differently because they all use the SAN for different purposes. There's only one major database on that SAN, but there's a small Web farm and a file server. So the symptoms the users saw were different, and the problems were all routed to different people to handle. It was the file server guys who brought the problem to me. They saw the disk queue length going up pretty dramatically, and they knew that had to be the SAN, so I got involved."

"That was the problem we dealt with all the time back then," Lisa said. "We all focused specifically on the bit we were responsible for, but these days there are so many interactions and dependencies that we can't see from a tool level that we'd get all tied up when a problem happened."

I also spoke with Kevin, who manages the company's Help desk. He says those types of problems were especially trying for his team because users would keep calling and the Help desk had no idea what was related and what wasn't, or what the status of anything was. "Users would call in with something that sounded new to whoever answered the phone, so they'd open a new ticket. We were probably slowing down whoever was trying to fix the problem just by loading new tickets on them for the same problem. But we had no real communication. If you answered the phone, you looked to see if there was an open ticket on anything that sounded similar. But there was no one place where we kept track of all the currently-open problems. I finally just had a white board installed in the Help desk office, and outstanding problems would get written up there. So when a call came in you could at least look to see if the problem was already open, then look up that ticket to see what was happening and give some status to the user on the phone."

I asked Lisa how things worked now, after the company had implemented a unified management system. "We've been on it for about a year now," she told me, "and it's completely different." She showed me a ticket for a problem that had occurred recently. "This is what we see, now."

> ALARM 2011-06-14 12:13:42
> NODE Windows Server BOSDB02
> SQL Server Instance DEFAULT
> SYMPTOM: SQL Server response time exceeds threshold
>
> IP: 10.10.15.212
>
> SQL Server database shows 34% free
> SQL Server fragmentation shows <5%
> Disk queue length <1
> Network utilization <40%
> CPU utilization <60%
> Memory utilization <75%
>
> RELATED ALARM 2011-06-14 12:10:52
> NODE Router MBS3667
> Interface fault

"Just looking at that, I can start to guess what the problem is." She showed me the monitoring console that the entire IT team now worked from, which looks similar to Figure 6.1. "You can see that it's basically a network diagram. It shows the servers and the services they run, but it also shows things like routers and switches. So when a server alarms, it'll also look for alarms on any dependencies, like a router. In this case, we had a router interface that was going bad and starting to drop packets. That triggered an alarm right away to the router guy, but it also alarmed all of the servers that use that router to communicate because clients—and the monitoring system—saw the servers' response times go up. Just having that data in front of us saves a ton of time testing for problems. The system basically runs a series of basic checks whenever there's a problem, so it gets those preliminary steps out of the way for us."

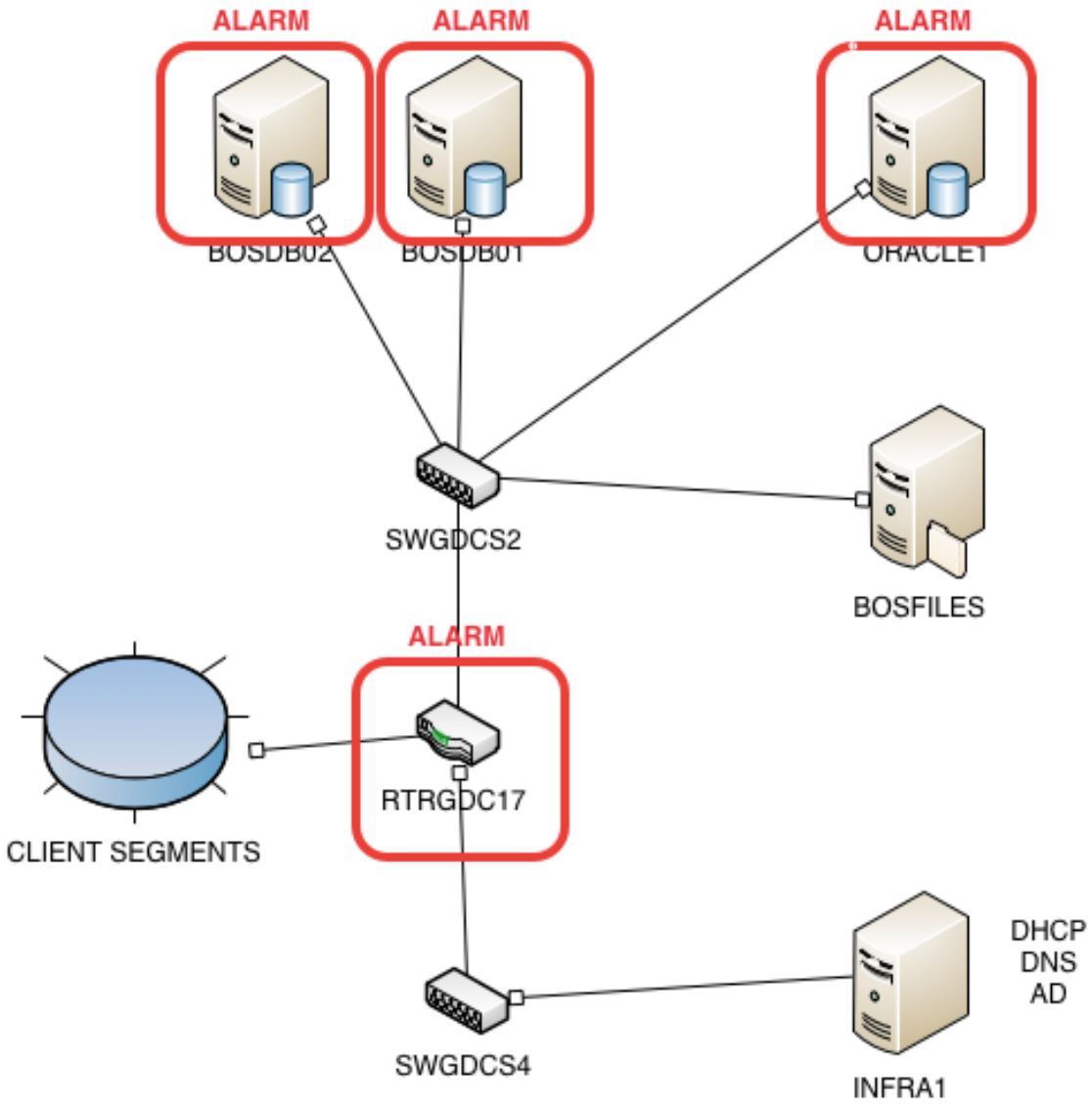

Figure 6.1: Tracing alarms visually.

She said the team spends a lot less time passing problems back and forth because it's usually much clearer where the problem lies when the system is looking at the entire stack.

"This is huge when the problem is actually outside the data center. We have a number of applications that interface with SalesForce.com, and whenever those guys have a problem, or more commonly when our ISP gets a little slow, our users see it as 'our' application being slow. But the monitoring system knows about the dependencies, and it's usually already alarmed us. We'll post a message about the affected applications on our end, and start calling the service provider to log a ticket with them."

Posting a message, Kevin says, has helped the Help desk tremendously. "We have this Web portal where users can log tickets, and current system status is shown right there. So before they even open a ticket, they can see that we know there's a problem. Once we trained them to trust us on that, they stopped logging duplicate tickets."

He admits that the training was a big step. "We didn't do it initially," he said, "but once users realized we were being pretty honest and consistent about posting problems, they started to trust us more. We had a big communications effort, and now there's even a mailing list users can add themselves to so that they get a message whenever a system they use is affected. Being proactive cuts back on the Help desk volume a ton."

The benefits of a unified management system were pretty clear for this team: faster time to resolution, less passing the buck, and more proactive communications with their end users. The biggest challenge they faced?

"A trust thing," Lisa told me. "We had to learn to trust this new system to monitor everything as well as we could with the tools we were familiar with. So the first few times things went wrong, we went right back to what we knew to troubleshoot the problem. Once we realized that we were seeing the same data, we started trusting the new system more, and just started relying on it. We'll still dig out the old tools if we have to dive deep into an affected system, but by the time we do that we *know* the problem is in *that* system, so we're not wasting time. You don't pass the buck to someone else at that point, you stay in *that* problem area until you spot the problem."

Fulfilling User Orders

Kevin provides the link to the other side of the unified management story. "We're not just responsible for opening tickets for problems. We also open tickets when routine changes need to be made." I asked him to give me an example of how this was handled prior to the implementation of their unified management system, and he pulled out an archived ticket.

> OPENED BY HelpDesk AT 2010-08-12 15:50
> User BDOUDS needs a new SharePoint site deployed as intranet/projects/universitybid. User will be site admin.
>
> ASSIGNED TO JHoltz
>
> NOTES BY JHoltz AT 2010-08-13 08:27
> Sent e-mail to Bill's manager confirming. Also sent e-mail to Special Projects confirming.
>
> NOTES BY JHoltz AT 2010-08-16 11:12
> Bill's manager, KHICKEY, confirms. Still waiting to hear from Special Projects.
>
> NOTES BY JHoltz AT 2010-08-18 11:05
> Still waiting to hear from Special Projects. Left VM.
>
> NOTES BY HelpDesk AT 2010-08-20 10:34
> User is asking for status.

> NOTES BY JHoltz AT 2010-08-20 11:34
> Tell him to call Special Projects. I just need them to confirm since this comes out of their budget.
>
> NOTES BY JHoltz AT 2010-08-22 13:11
> Special Projects confirmed. Set up site and assigned BDOUDS as site owner.
>
> STATUS SET TO RESOLVED AT 2010-08-22 13:12

"That kind of thing went on all the time. Someone would call us asking for some access or whatever. We'd assign the ticket to someone in IT, but then they'd spend time figuring out who was responsible. We used to have a big book," he added, pointing to a thick three-ring binder on his shelf, "that told us who was responsible for pretty much everything. Then you'd wait and wait to hear back from them. This one took, what, two weeks to resolve? That's insane, and the whole time the user is calling us to check on the status, when we're not the ones holding things up. This took Jeff 10 minutes to do once he got approval."

And in the world of unified management?

"It's actually pretty cool," Kevin said. "Now we have a big online catalog with everything a user might want. It's kind of like an online store. They submit their request through there, and the system opens a ticket automatically. But each item is associated with a workflow, so IT doesn't even hear about it until the ticket has been routed through the proper approvers and been approved. Once we see it, it's a done deal, so we just implement it. For some things, we're even implementing scripts that do the implementation for us, so it's completely hands-off." The organization worked out, and documented, the desired workflows for each possible product. Kevin provided an example of that documentation, shown in Figure 6.2. "This kind of documentation is important because we worked off of this to implement the workflows. The business owners can come up with these flowcharts on their own, then we just implement them on the designated products in the catalog."

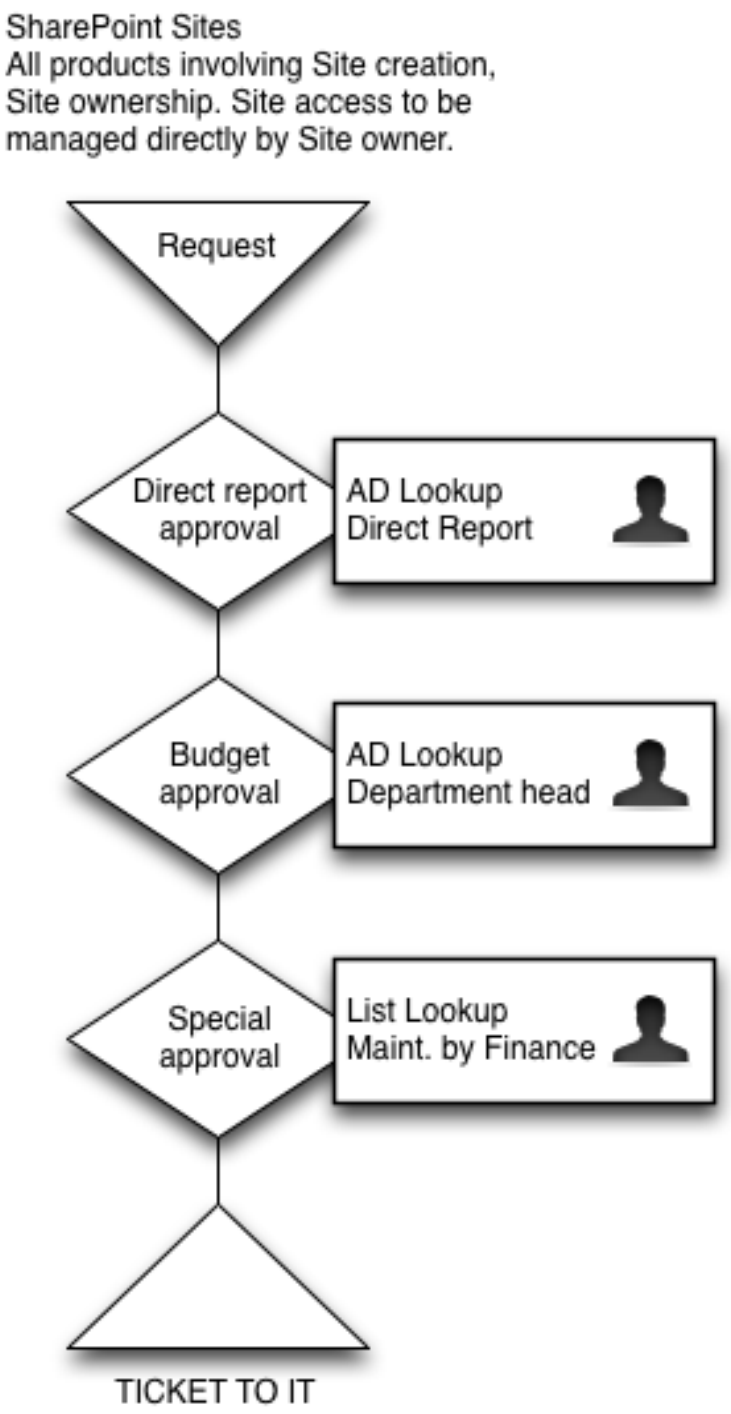

Figure 6.2: Documented workflow used to drive automated review/approvals for catalog requests.

We were discussing access permissions as an example, so I asked what happened when those needed to change. "They never did," Kevin admitted. "Once you had access, you usually kept it until you left the company. We just didn't keep track of it. Now, the catalog keeps track of it. If you don't need something, you 'return' it to the store, it goes through whatever approvals, and we get a ticket to remove your access. Different managers also have to occasionally complete an attestation, which is where they review who has access to their resources and let us know if anyone needs to be removed, or if everyone can stay. We're not the gatekeepers anymore."

I noted that an automated workflow wouldn't necessarily guarantee a speedy response time. "Oh, users still have to wait two weeks for approvals, sometimes. But when they submit their request through the catalog, they can check the status of that request on their own. They can see that it hasn't made it to us, and they can take it on themselves to bug their manager or whatever. We're totally out of the loop until it's approved, and they know that, because the request shows it hasn't even made it to us yet." Such a system does a better job of keeping users informed, and helping them to understand what's really holding things up.

A Shopping List for Unified IT Management

I want to use this section to present a list of what I believe are the must-have features of a true unified management system. As you're evaluating solutions, make sure they offer these features—and make sure the features operate in a way that makes sense for your environment's needs.

- Workflow. Unified management solutions should offer workflows that can help automate responses and service management. Workflow construction should be as drag-and-drop as possible, involving as little programming as possible.
- Agents. I know there's a huge divide between people who are fine with deploying agents, and those who hate the idea; I'd suggest looking for a solution that supports both models. Agentless data collection is fine in some instances, although it can offer less performance and coverage than an installed agent. I think a hybrid approach is probably best for most organizations, and unified monitoring solutions ought to support that.
- Alarm integration. When a problem arises, a unified management solution should obviously tell the designated individuals; it should also open a Help desk ticket and automatically search for related alarms from the past. Doing so will help speed up the time to resolution. This kind of "knowledge automation" is really crucial.
- Approvals. As I've pointed out, "tickets" aren't always for problems—sometimes they're for new work, like change requests. A unified management system should support a review/approval workflow for these requests so that IT can be taken out of its traditional "gatekeeper" role and instead simply work those tickets that have been approved for implementation by the business.

- Discovery and deployment. A unified management solution should help you discover manageable nodes and services and deploy any necessary agents to monitor them. This discovery should happen more or less continuously, or at least be able to be run regularly, so that changes to your environment can be captured.
- Routing. Tickets—whether for problems or for requests—should be automatically routed based on custom business rules that you can define. In other words, tickets should head straight to the correct implementer as quickly as possible.
- Scheduling. A unified management system should have some kind of internal calendar that lets you schedule maintenance tasks. This functionality helps to resolve maintenance window conflicts and schedule work to happen at the right time.
- Catalog. This is a key part of making a unified management solution part of a self-service, managed system. In addition, a catalog helps work toward bringing process compliance—such as ITIL compliance—into your environment. A catalog provides users with a list of "orderable products," not unlike shopping at an online Web host. Users' "purchases" translate into tickets, which go through review/approval prior to being passed to IT for implementation.
- Communications. Users need to be able to submit requests, and users and your team must be able to review them from a familiar place. A Web portal is the traditional way to enable this communication, but systems that can integrate via users' inboxes—which they're in all the time, anyway—is even better.
- Interface. You can't have too many interfaces into a unified management system, and whatever solution you pick should offer both Web-based and mobile-friendly versions of its UI.
- Metering. If you're monitoring actual paying customers, you'll need the ability to charge them for what they use. Even if you're just dealing with internal "customers," being able to perform "charge backs" for their IT resource consumption is going to be critical as business managers advance their management strategies. There's no reason for IT to be seen purely as overhead when resources can—and should—be tracked back to the business components that are consuming them.
- SLAs. A unified management system should assist you in both defining and monitoring service level agreements (SLAs) based on actual historic trends.
- Trends. A unified management solution should include a performance database that lets you track historical performance trends. This database can be used to help define and report on SLAs as well as perform capacity planning.
- Surveys. Closing the loop with your end users is crucial because technical SLAs aren't the only way your success is being measured, whether you know it or not. Being able to poll users helps you define SLAs in their terms, creating more appropriate expectations.

- Reports. Look for reports and dashboards that provide managerial- and executive-level views of items such as workload, SLA compliance, and so forth. Heck, even dashboards that can be exposed to end users, helping them see that the environment is performing as it should, can go a long way toward helping IT be seen as more responsive and engaged with the business.
- Visualization. Being able to visualize your environment can help make root cause analysis and problem resolution faster and easier.
- Everything in one place. As I've written several times in this guide, a unified management system's primary value is unity, or the ability to get all your performance concerns into a single place, using a single set of metrics, alarms, identifiers, and so forth. This singular view helps to break down the traditional domain-based "silos" that IT is built around, and gets everyone focused on the root cause of a problem more quickly.
- Knowledge retention. A unified management system should help your organization retain critical knowledge by turning Help desk tickets into an automated, searchable knowledge base.
- Pre-loading information. When an alarm generates a ticket, that ticket should include whatever details the unified management system can provide: IP addresses, response times, and so forth. The more information included in the ticket, the less the responder has to go look up, and the sooner they can start working on resolving the problem.

This list obviously isn't comprehensive but provides a starting point. If a potential solution offers these features *and* meets your organization's specific needs, that solution is probably worth looking at in detail during an evaluation. Make sure you gain not only a "check mark" on these features but also a detailed explanation of how they're implemented. Also, ensure that the implementation is one that will work within your organization's requirements.

Ways to Buy Your Unified IT

I want to briefly outline different approaches that vendors take for delivering unified management solutions. Let me emphasize up front that I don't regard any of these as "right" or "wrong;" there's merely "what's right for *you,*" which you'll need to decide on your own.

Typically, you'll find that solutions of this kind are priced based on the number of nodes you need to manage, possibly also incorporating the number of users in your organization. A "node" is typically defined as any manageable device: a router, a server, and so forth. Some vendors are more creative than others with this portion of their licensing model; don't let a complex model scare you off. In some cases, more complex license models are actually to your benefit because vendors are trying to precisely accommodate a wide range of scenarios. You should be more concerned about *what* you're licensing.

For example, at one end of the spectrum, you'll find what I call *monolithic* solutions. With these, you get—and pay for—every feature that the vendor offers, regardless of which ones you'll need right away. I think it's hugely important to make sure you're acquiring a solution that *can* do everything you want, although I'm not sure you necessarily want to *pay* for all of that up front. In some cases, you may want to implement a solution in a phased approach, licensing just the functionality you need for each phase, thus allowing yourself to kind of "ramp up" into the full licensing and functionality of a product. The nice thing about monolithic solutions is that they're often well-integrated because everything is delivered to you in a single piece.

There are also *pluggable frameworks.* I tend to view big frameworks like HP OpenView as fitting into this kind of model. With these solutions, you buy a base product, then add in the various bits and pieces you need to speak to your environment. These models offer a ton of flexibility, of course, and if you're going with a big enough vendor, you should be able to find plug-ins for every bit of functionality you need. These solutions run the risk of becoming a massive do-it-yourself project, though, and the plug-ins aren't always as well-integrated as you might like. Licensing can also be really, really complex because you're often licensing the plug-ins separately from the base framework.

Another model is the *pay as you go* approach. With this model, the solution offers all the functionality you might ever need, but you don't "switch it all on" right away. Instead, you turn on the modules, or functionality, that you need immediately, and you just pay for that. As you add more responsibility to the solution, you pay a bit more. This setup is a bit more like a "cloud" model, where you can grow as large as you like but only pay for what you need right now. You're not typically dealing with plug-ins, or if you are, they're usually all delivered by the same solution vendor. I'm seeing more clients considering this approach.

The last thing you'll need to think about is where the solution will live. In this age of "the cloud," you actually have a choice of hosting your monitoring and management solution in your own data center or simply purchasing it as a hosted service that lives in the vendor's data center. Either way, the vendor's agents get installed into your environment. I won't dig into the "on-premise versus hosted" debate; you probably know what's right for you, and you can certainly discuss that option with whatever solution vendors you're investigating. Regardless which side of that debate you're on, I think it's nice to have a solution that offers both options.

Conclusion

Where, there you have it: unified management. The overall idea behind this book was simple: really focusing on the straightforward theme of "get everything in one place, and get everyone on one page." It's only "revolutionary" compared with the disjointed approach that our existing technology tools have more or less forced us into.

Of course, I don't expect you to just rush right out and start switching over to a new monitoring and management framework. These things can be done in small steps so that they create less impact on your organization and allow you to learn to use various techniques and features properly in an organic, rather than disruptive, fashion.

The goal should be there: Stop wasting time with the back-and-forth and instead get yourself onto a single pane of glass for your organization's top-level monitoring. Integrate that with a Help desk system that lets you keep everyone informed and gives you the metrics you need to analyze your IT performance objectively.

Good luck.