BladeSystem Matrix: A Shared Services Engine

It’s been said that IT is a fashion industry. While it’s true we have our fashions—just like any field of human endeavor—we’re generally practical. It’s hard to see either IT executives or technicians as highly subject to whims of style or flights of fancy. The truth is closer to: We’re an evolving industry—one constantly struggling to find better ways.

It’s not easy to grapple with the fantastic, relentless progress afforded by Moore’s Law (on the supply side), nor the constant demand for more capacity, capability, and integration (on the demand side). In a few short decades, IT has undergone a massive shift from an engineering-oriented support role to driving the beating heart of the global economy. IT is now central to large swaths of all human activity. As new technologies and strategies come online—whether network computing, open source, agile development, SOA, cloud, virtualization, or whatever—we seek to employ them to improve our outcomes.

There’s always a bit of experimentation and a bit of hype involved in the early days, but most of the major approaches touted over the past few decades have become work-a-day parts of the IT landscape. When they recede from view, it’s not because they’ve failed, but because they’ve succeeded so well that we don’t need to talk about them anymore.

Each wave builds on the last, incorporating its best parts, weeding out what didn’t work, and often re-emphasizing themes that had appeared years before, but weren’t quite workable at that time.¹

IT is now heading toward a “shared services model.” This marries the latest approaches to building efficient datacenters (e.g. modularity, density, and virtualization) with the best thinking on how to operate IT at scale (e.g. standardization, service-orientation, automation, and change-readiness). More important, it represents a maturation of thinking on what IT is, and what it should be. By organizing IT assets, activities, and processes as deliverable services, it focuses attention on IT as an engine of business outcomes.

¹ Although often using different names. The utility computing, grid, and application service providers (ASPs) of years past, for example, have become the software-as-a-service (SaaS, or more generally, ITaaS) and cloud computing of today.
This Illuminata Insight discusses the shift toward a shared services model, then examines HP’s BladeSystem Matrix, a platform designed for the shared services approach.

**What is the Shared Services Model?**

Roughly speaking, it means adopting a centralized, standardized, streamlined approach to IT. Different organizations often have slightly different takes on the details, but almost all agree with common elements:

**Service-Oriented** – IT as a function is thought of as a provider of services to a business—or, in some cases, multiple businesses. The services IT provides, and how well they map to, and measure up to, business requirements—these are the primary lenses through which every IT process, asset, and outcome is understood, operated, and judged.

The change, compared to the traditional “IT maintains whatever computer stuff we happen to have,” is enormous. It remakes the traditional bottom-up, input-oriented, technical perspective as a top-down, outcome-oriented business perspective. It encourages judging, for example, the performance of whole applications, from the end-user perspective, rather than the performance of individual machines or parts, from an IT staff perspective.

One level up, it encourages measuring the performance of the IT organization and putting it on a competitive footing. If IT can’t implement a service markedly better than alternatives, maybe it should be selectively outsourced to an external service provider. Organizing via services not only lets an organization rationally have the “Should we do this ourselves? Or should we buy it done?” discussion, it structures investments and interfaces to actually engage external services, when and where they make sense.

Viewing IT as an organization building, maintaining, and operating a collection of services—rather than one overseeing a collection of equipment and a pile of code—it’s a radical departure from historical IT-as-a-cost-center approach.

**Shared** – Many organizations traditionally created IT “silos.” Many of these fiefdoms formed around internal line-of-business (LOB) boundaries; each LOB had its own little IT department. Financial services organizations have probably been the most siloed; some have had nearly 100 distinct IT operations. Other silos are formed around technology specialties such as servers, storage, networks, facilities, Unix, Windows, mainframes, virtualization, app development, and so on. Name a technical specialty or platform variety of any significance and there’s a good chance someone, somewhere has an IT silo built around it. So pervasive are these technology silos that every sizeable corporation has had—and many retain—a half-dozen or more.

Unfortunately, silos fragment IT into many little IT shards. They introduce gratuitous complexity and variability. One silo does things this way; another does basically the same thing, but in a different way, or with a different product. Silos don’t necessarily work together with any great enthusiasm or efficiency; even when they have overlapping responsibilities, silos often work at cross-purposes to “protect” their fiefs. IT as a whole cannot enjoy economies of scale in staffing, training, procurement, and operations. Often, the organization as a whole can’t even get visibility into its aggregate choices to see where economies or simplifications might lie.

A shared services approach consolidates operations, eliminating and minimizing silos, and making them actively cooperate. This consolidation is a prerequisite to globally sharing resources and capabilities.

**Standardized and Simplified** – The sharing or consolidation theme allows approaches, vendors, strategies, and procedures to be standardized. Vendor counts are reduced; rather than “one of everything,” rational vendor management approaches can be used, such as strategic dual-sourcing. Standardization leads to competition. Less
variety also means simpler. Instead of various management approaches (many of them ad hoc), “best practices” can be used. Skills can be shared. External resources, learning, and tools can be brought in, rather than homegrown. And because the systems and interfaces used are regularized, they can be increasingly automated, further simplifying the environment. Standardization and simplification is an iterative process, with each pass enabling further standardization, further simplification. Simplification, in turn, eases scaling, as well as whatever technology or process transitions are required over time.

**Agile and Effective** – A variety of phrases are used to describe our desired IT future state: agile, flexible, adaptive, dynamic, and so on. Whichever term you prefer, everyone agrees with the goal of IT’s operating at the speed of business, able to wrangle both internal and external resources to design, build, and run the services the business needs. What business needs will change over time, given new products and business opportunities, mergers and acquisitions, geographic expansion, economic upturns and downturns, and seasonal surges and quiet periods. But whatever changing mix the business needs, IT wants to agilely and efficiently provide it. That’s a tall aspiration, given the relatively static, backroom support orientation of IT of old. But that is the goal of forward-looking IT.

So, in short, a shared services model seeks to create a virtuous cycle between Shared, Service-oriented, and Standardized and Simplified to establish an Agile and Effective IT capability.

These goals, however, are not for everyone—or, at least, not every organization will view them as achievable in the near term. Many haven’t progressed fully enough through the evolution from best-of-breed and piece-parts approaches to the business service provider approach. Change is hard, culturally and politically; moving from traditional silos to a shared function requires a strong executive mandate and much work.

If the shared services model or the attributes described above aren’t compelling and essential in your IT strategy, you might as well stop reading here. The rest of this Insight discusses the infrastructure necessary to achieve this vision, and looks at one platform in particular (HP’s BladeSystem Matrix) that aspires to be the engine for enterprises’ shared services model.

**Spec-ing Out a Services Engine**

Once you’ve decided on a shared services model, a next logical question becomes: So what will IT look like under this model? What type of IT engine will support it?

The good news is that neither captured alien technology nor magic pixie dust is required. Common, off-the-shelf components already found in just about every datacenter will suffice. But the architecture and management style are quite different from the accreted hodge-podge of systems, tools, and approaches seen in traditional IT. Instead, they take on many attributes common within external service providers and Internet datacenters, albeit adapted to enterprise needs. The key attributes of a service-oriented infrastructure or services engine are:

**Consolidated** – IT resources will be consolidated into pools. Resources will then be allocated from these pools, on an on-demand basis, to operate various services. Many workloads will run on the same infrastructure, sharing the same physical servers, storage devices, network switches, and so on. Sharing is key to achieving capital expense (capex) economies, but even more so to operational efficiency and expense (opex). Virtualization is a key enabler; the resources parceled out will often be virtualized servers, storage, networks, and clients. Virtualization provides the ability to pool and allocate resources, and to maintain appropriate performance and security isolation among workloads. Over time, more and more of the IT resources will be virtualized, approaching 100%. But for years to come, while virtualization continues to evolve, important parts of the
infrastructure will also involve physical systems and resources.\(^{2}\)

Consolidation here is not just about where resources are placed, or that they’re allocated from common pools. It’s also a logical concept, embodying the standardized and simplified thought above. IT must limit the variations from system to system, so that pretty much any application can run on any part of the infrastructure. There will of course be some variation. Some applications need more I/O bandwidth, and some need less. Some need extreme reliability, or have other legitimate special requirements. They will use different resources from more run-of-the-mill apps. Multiple classes of service will be used, in order that operations are economically optimized. Rare indeed is the Web app that needs its own 64-processor Superdome or NonStop platform; also rare is the financial trading app that is well-placed on the lowest-cost, lowest-reliability 1-socket x86 server. There will be a handful of different classes of service, or maybe a dozen—not the “every server custom-configured for just one app” historically seen.

**Flexible** – Resources will be allocated to meet demand. Application demands may grow or shrink, whether from predictable daily, weekly, monthly, yearly, or seasonal patterns, from planned events and promotions, or \textit{a priori} unpredictable events such as business news, economic conditions, and unexpected surges. The infrastructure, including how it is designed, measured, and operated, will be organized with such changes specifically in mind. IT will not just be able to change slowly, slightly, and occasionally; rather, it will designed and operated to be actively change-ready: nimble enough to change in real-time.

---

\(^{2}\) HP’s management tools have the advantage of being able to logically manage physical servers (and the network information that defines their connections to the world) as though they were virtual servers. While most workloads are “going virtual” over time, the ability to blend in and orchestrate physical deployments as well is a very nice performance optimization. It also eases administrative concerns, because physical servers don’t need to be managed through “off book” special/different procedures.

**Automated and Orchestrated** – It’s an old claim to say that something is “managed,” “highly manageable,” or “easy to manage.” Management tools, interfaces, APIs, and such are good, as is “ease of management”—but they aren’t sufficient. At the scale that modern IT operates—thousands of composite applications, and many thousands of servers—no one can any longer afford having armies of managers carefully watching and reacting to every event, every system action, and every error. It’s essential that workloads be automatically provisioned and orchestrated. It’s essential they be monitored and measured, from as much an end-to-end perspective as possible, and it’s essential that they be automated as fully as possible, to eliminate human touches. The transition from “hands on” to “everything automated” is a hard one for many practitioners, who for years have managed all the details themselves; it’s essential, however, for IT to move to higher levels of efficiency, reliability, and ease.

**Matrix: HP’s Services Engine**

HP’s BladeSystem Matrix, introduced earlier this year, is ready-to-run infrastructure for shared service environments. Matrix combines off-the-shelf parts such as BladeSystem servers, Virtual Connect interconnects (Flex-10 10 Gbps Ethernet or 8 Gbps Fibre Channel), and Insight management software into a single ‘packaged infrastructure’ offering. Matrix can be configured either with its own integrated SAN, or be connected to existing SAN resources across Fibre Channel.

We call Matrix an \textit{offering} rather than \textit{product} because it’s not just equipment. It’s also a supply chain, mode of purchase, factory configuration, management software stack, testing/qualification as a whole, and on-site installation services that come with it. You could buy any component on the Matrix bill of materials separately in the conventional way—well, any component except the fancy rack bezel that signifies the rack is a Matrix. But to do so would miss the point: escaping the old buy-it-piecemeal, assemble-it-yourself approach.
Technically, Matrix’s two most interesting parts are its networking and its management environment. Unlike traditional servers, networking and management are first-class citizens of the overall system, not bolt-ons.

Virtual Connect’s Flex-10 technology, for example, enables very dense, I/O-rich configurations, with up to four times as many NIC connections as conventionally possible, yet without adding more NICs or managed switches. This lowers both equipment costs and power consumption. Admins can configure the bandwidth of each connection, even on the fly. Most HP blade servers have embedded Flex-10 capable 10 GbE dual-port NICs; if even more I/O is required, Flex-10 mezzanine cards can be installed.

For many users, the need isn’t having 8-10 gigabits/sec of nominal bandwidth available at all times for individual applications. Rather, it’s about using 10 GbE and 8 GbFC as an aggregation mechanism to flexibly pair workloads with the appropriate amount of network bandwidth. In other words, it’s about virtualizing network connections, in much the same way servers have been virtualized.

Beyond cost optimization and virtualization benefits, Virtual Connect is part and parcel of HP’s strategy for the “wire once” datacenter. Coordinating workloads with network connections is hard; the process touches and must integrate with servers, network switches (LAN and SAN), hypervisors, management tools, and so on. With Virtual Connect, admins define the network configurations (such as Fibre Channel Worldwide Names and MAC addresses), security boundaries (VLANs), and policies they want up-front; the system then automatically moves these definitions around and enforces them as workloads migrate in a dynamic environment.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Underlying Product</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Servers</td>
<td>BladeSystem c7000 enclosure and server blades (“c-Class”)</td>
<td>Popular modular server foundation</td>
</tr>
<tr>
<td>Storage</td>
<td>Pre-integrated option with HP StorageWorks EVA4400; or integrate with existing HP and 3rd party SAN storage</td>
<td>SAN storage, as desired</td>
</tr>
<tr>
<td>Network</td>
<td>Virtual Connect Ethernet (Flex-10) and Virtual Connect 8 Gbps Fibre Channel</td>
<td>Cost-efficient high-bandwidth LAN and SAN; flexible allocation across workloads</td>
</tr>
<tr>
<td>Operating System</td>
<td>Customer choice of operating system (Windows, Linux, etc.) and virtualization environment (VMware, Microsoft, etc.)</td>
<td>Runs variety of workloads; essentially anything that x86 runs. Virtualized and physical deployments supported.</td>
</tr>
<tr>
<td>Management Environment</td>
<td>Infrastructure Operating Environment (built on foundation of Insight software, including Systems Insight Manager, Insight Dynamics and related tools)</td>
<td>Resource pool workload provisioning, capacity planning, energy optimization, virtualization management, etc. not added as after-thought add-ons, rather built in as first-class citizen of the system</td>
</tr>
<tr>
<td>Integration</td>
<td>Matrix</td>
<td>Customer-choice on compute nodes and storage targets, but standardized infrastructure factory-assembled, connected, wired, and packaged; new industry high-water mark for ready-to-run</td>
</tr>
</tbody>
</table>
Management is built up from a well-known suite of Insight software. The Matrix management console has extensions for managing physical servers, storage devices, energy consumption, and virtual instances. But what really makes Matrix sing is its ability to also do workload analysis and optimization. Its capacity advisor, for example, looks at historical performance and resource consumption data, creates what-if consolidation scenarios, and makes intelligent recommendations as to what workloads (physical or virtual) should be run where.

The console also embeds a virtualization-aware templating and orchestration engine, HP Operations Orchestration (from its Opsware acquisition). It would be hard to overstate the importance of this engine. It is the fundamental way that workloads are intended to be managed in a Matrix deployment. As with Virtual Connect, the goal is to identify and state the requirements up front, then let automated mechanisms handle moment-to-moment operation. That’s a big change from the status quo in many shops. But moving to automation is why Matrix becomes not just another rack of equipment, but an engine for infrastructure services. You can of course fire up applications by hand if you like, but ideally you shift into a services management mindset and let Matrix fire them up for you.

With any package or bundle, there’s always the question of “will it fit us?” There is, of course, variability. In Matrix you choose what compute/server blades, storage targets and capacity, rack and power distribution, VMware and Microsoft virtual machines (or not), network configuration, security setup, and integration with other management software—whatever makes sense to your application set and desired workload. Matrix is packaged, but also flexible enough to meet varying customer needs for a ready-to-use shared services infrastructure. Just as you can order your car in the color of your choice, with the options you select. You can still do any after-market modifications you like. The key thing is that Matrix is like buying a car as a car. Most customers will select some options, but they won’t make major modifications. The goal is loading your people and stuff into the car, not modifying the car. And so beyond the basic information about the datacenter in which they come to reside, application workloads and local policies for how to run them is the primary thing that customer are expected to load onto Matrix.

Five years ago, Matrix would have been more packaged and more automated than customers were ready for. Indeed, HP had a historical product, Utility Data Center, which tried to provide unified infrastructure, albeit on a much more primitive, unvirtualized basis. It was more an island than an integrated part of a customer’s data center, and it required close cooperation that most storage, server, and network admins weren’t ready for. It sold to a few sophisticated datacenters, but quickly stalled, and was withdrawn. It was ahead of its time; most
shops were simply not ready for the uniform replicated configurations of storage/compute resources, reconfigurable network links, and automated workload coordination needed to make consolidated infrastructure work. Nor were the technical underpinnings—whether in the server, storage, networking, virtualization, or management realms—really ready to all work together in a seamless, automated way. Now, all of those things are feasible. Matrix integrates and interfaces readily into existing environments. It is non-disruptive to IT’s organizational structure. Moreover, Matrix is built on proven, popular technologies like HP BladeSystem, Virtual Connect, and Insight software. It’s a packaged infrastructure, but one that can be flexibly configured and readily connected to the rest of the datacenter.

**Buy Not Build**

Why is Matrix required for an enterprise to operate a shared services model? The short answer: It’s not. One could assemble all the parts of a shared IT infrastructure from components, or even build some of it from scratch, and still operate that infrastructure as a shared, service-oriented facility. Indeed, many service providers have exactly this strategy—custom infrastructure, lovingly hand-crafted and meticulously maintained by on-site experts, in support of the most scalable, efficient IT engine possible. Google, for example, is infamous for designing its own custom servers, development tools, and middleware.

The problem isn’t whether it can be done, in theory; it’s practical—whether it can be done, in reality, by you, with the same level of efficiency. Leaving aside for the moment the question of whether homebrew is a reasonable strategy for the largest service providers on the planet, over the years it’s repeatedly proven itself a poor strategy for everyone else—including even very large enterprises.

Every IT shop has to assemble its own macro platform from the things it either buys on the open market or creates itself. The issue is the level of responsibility involved. If the components are high-level and work well together, IT’s responsibility is basically picking the parts and learning how to use them. If the parts are low-level, or not well-integrated, IT has to not only select and learn the tools, but also to provide its own integration “glue” among the tools. And it has to maintain the glue code and procedures over time. Its level of investment and responsibility inevitably rises; it must keep investing in that home-grown, home-assembled stack, year after year.

Large enterprises used to think they had both the scale of operations and the special requirements that justified developing large swaths of their IT stack, including their own applications. Over time, it became clear that they were, for the most part, wrong. What started as small internal development teams became very large teams. What started as limited, targeted investments became very large investments. Even these large, rich enterprises couldn’t amortize their investments over more than their own IT operations—and that’s not enough.

External developers who could amortize their investments over many enterprises easily outspent even the largest IT shops. Industries like aerospace/defense, telecommunications, and financial services that had access to large capital budgets and had truly special requirements held on to the “we can build it ourselves, we are the experts in these requirements, we should build it ourselves” argument the longest—even they have now largely seen the economic wisdom of commercial off-the-shelf (COTS) procurement. Google may still be unconvinced, but for the other 99.997% of enterprises, if the vast majority of their IT stack isn’t externally sourced, their executives should be asking: Why not?!

Perhaps even more pressing than the money that a roll-your-own strategy ties up is what it does to internal culture and priorities. It focuses IT on becoming the technology expert, the manufacturing expert, the software development expert for the homebrew IT stack—not the expert on the enterprise’s business, processes, requirements, problems, and opportunities. It focuses IT’s attention, time, and energy on creating
and maintaining componentry from which the business cannot achieve meaningful differentiation or value.

This is not to argue that enterprises need do nothing themselves. There will always be applications, business logic, or component integration that doesn’t exist on the market, or that the enterprise is uniquely capable of creating. But, year by year, those parts become less and less—and especially, they become more and more closely tied to unique business processes and approaches of the enterprise, rather than “horizontal” platforms that multiple enterprises might use.

In short, a shared services platform like Matrix isn’t theoretically essential to building a shared services approach. In theory, an enterprise could build shared services using only Turing Machines. Matrix, on the other hand, does answer a very fundamental practical need for those building a shared service: How to get IT out of the componentry business and refocus attention and resources on the services aspect that provides value. We call this “buying infrastructure by the pound,” rather than by the piece part.³

³ See our “Infrastructure by the Pound.”

Conclusion

The shared services model is a mature conception of what IT is, what it provides, and what it should be. It organizes IT around business outcomes—and, one level down, operationally, around the services that IT provides. It depends on essentially all of the technology and management advances of the past two decades—network computing, service-oriented architecture, modular infrastructure, and virtualization, to name just a few—as well as on a maturing of IT’s processes and operational philosophy that aren’t tied to specific technical or product issues. Beyond using these technical and operational improvements, it leverages them into better outcomes. It requires a change in thinking, priorities, and operational style—and change is never easy. Nonetheless, we see organizations everywhere adopting large swaths of the shared services approach—though over time, at different paces, depending on their starting points and the urgency of their evolution.

The shared services approach isn’t tied to any specific technology, vendor, or product set. HP’s BladeSystem Matrix, however, stands out as an early, impressive offering for organizations working towards the shared services approach. It packages and integrates a large number of popular system components, then delivers them as a ready-to-run services engine.