ExtraHop
See IT run.

WHITE PAPER

Designing and Building an Open IT Operations Analytics (ITOA) Architecture

Abstract

This white paper provides a roadmap for designing and building an open IT Operations Analytics (ITOA) architecture. You will learn about a new IT data taxonomy defined by the four data sources of IT visibility: wire, machine, agent, and synthetic data sets. After weighing the role of each IT data source for your organization, you can learn how to combine them in an open ITOA architecture that avoids vendor lock-in, scales out cost-effectively, and unlocks new and unanticipated IT and business insights.

Table of Contents

Executive Summary

This paper explains the need for and benefits of designing an Open IT Operations Analytics (ITOA) architecture based on the principle of streaming vastly different IT data sets into a scalable, non-proprietary data store for exploration and multi-dimensional analysis. Many IT organizations do not realize how they can use rich data sets that they already have for better operational and business decisions. The objective of Open ITOA is to enable the discovery of new valuable relationships and insights derived from these combined data sets. This will drive improved IT operations, add business value, prevent vendor or data lock-in to proprietary systems, and provide a roadmap for the cost-effective expansion of the analytics architecture.

With the aim of providing practical and prescriptive guidance, this paper introduces a taxonomy that covers the primary data sets that serve as the foundation of Open ITOA. This taxonomy not only defines machine data, wire data, agent data, and synthetic data, but also explains each data source's strengths, limitations, and potential uses in an organization. By classifying data sets according to their source, you can select best-of-breed technology on an objective basis. The goal is to put you in control of your ITOA platform by providing vendor choice and minimizing switching costs if you decide to replace any particular data source technology.

Using the four sources taxonomy, you can assess your current toolset to identify and eliminate redundancies, address technology gaps, and evaluate new solutions while building toward a unified and Open ITOA practice. Organizations that have adopted this practice have reduced the number of tools and products in use from an average of 22 products down to less than 8 while increasing operational visibility, IT productivity, improved MTTR, and cross-team collaboration (through fewer and dramatically shorter "war room" sessions). Open ITOA architectures can also augment traditional business intelligence architectures.

The final section of this paper provides some prescriptive guidance for designing an open ITOA architecture, including concrete actions to take when evaluating your existing tool sets, selecting non-proprietary data stores, as well as choosing visualization and analysis tools.

PLAN NOW FOR ITOA

Gartner estimates that by 2017, 15% of enterprises will actively use ITOA solutions, up from just 5% in 2014. IT executives should plan an open architecture for their ITOA solution, which will minimize costs and ensure the best results for their organization.

The Need for an Open Architecture for ITOA

IT organizations purposefully design and build architectures for their applications, networks, storage, cloud, and security systems, but what about IT Operations? If you asked a CIO or VP of IT, "Do you have a network or security architecture?" they would of course answer, "Yes." In most cases, they would be able to describe their strategy, the principles driving that strategy, and the components of those architectures. But if you ask, "Do you have an IT Operations architecture?" most would not be able to describe a strategy let alone the principles behind their tool selection. It is ironic that production IT Operations—the intersection where the technology and business meet—is the one place in the IT organization that lacks a well-defined and unified architecture.

Lacking a planned architecture for gaining visibility, IT organizations purchase monitoring and management products according to the specific needs of specialist groups. The network team will have one set of tools; the application support team

another; DBAs, storage teams, and system administrators will all have others; and so forth. What results is "tool bloat" and a number of disparate views, and information locked in their own data stores, with no easy way to correlate data sets and gain holistic insight.

However, it is not entirely IT organizations' fault for a lack of a unified architecture; IT management vendors are primarily to blame because they rarely, if ever, pursue meaningful integration between siloed products that would enable a unified architecture. Meaningful integration does not mean sharing SNMP traps, alerts, or writing plug-ins. While these can be useful, the only way to achieve ITOA insights from expected relationships while discovering new ones is through unrestricted data integration using a common data store. This data store must be open and non-proprietary. It cannot be controlled by any one vendor because that represents too big of a business risk for the others. The data store serves as a neutral ground where different monitoring products can be fairly compared based on the data they provide. This approach also negates vendors' attempts to control their customers by locking their data in a proprietary data store and then discouraging meaningful integration with products that could be perceived as competing.

	TRADITIONAL IT MONITORING AND ANALYTICS	NEXT-GENERATION IT MONITORING AND ANALYTICS
Context	IT teams purchase their own monitoring tools, each with a data set that cannot be easily correlated across other data sets. IT teams lack context when answering questions.	Open data streams and generous APIs enrich data with context and enable collaboration across multiple IT teams and offer value to business stakeholders.
Flexibility	Traditional proprietary data stores and analytics solutions rely on rigid schemas that require weeks or months to change. Organizations' data is locked in a proprietary data store and cannot be ported.	Modern, non-proprietary data stores enable organizations to easily perform ad hoc queries on unstructured data. Data can be moved from one data store to another without hindrance.
Scalability	Relational database technology does not scale well.	NoSQL data stores are built for scalability.
Cost	Discrete proprietary tools for each IT team and limited scalability increase licensing and maintenance costs.	A rationalized toolset offering the four sources of IT visibility and a non-proprietary data store reduce licensing and maintenance costs.

It is time for IT organizations to take an architectural approach to ITOA, intentionally designing an open system to deliver consistent, high-quality results for performance, availability, security, and business insight. An Open ITOA platform will provide both real-time analysis for what is happening within the environment at any given time as well as the ability to query indexed data on an ad hoc basis for forensic analysis and long-term trending. This platform will primarily provide value for IT Operations, Security, and Application teams, but will also benefit business

stakeholders through business-value dashboards.

With an open architecture for ITOA, IT organizations can:

- Proactively identify and resolve performance and security issues
- Correlate transactions and events across every tier of the infrastructure
- Intelligently plan IT investments and optimize infrastructure
- Facilitate collaboration across various IT teams
- Improve the IT staff productivity and workflow efficiency
- Reduce costs by rationalizing their IT operations management toolset
- Provide line-of-business stakeholders with real-time insights

As stated previously, no single IT management or monitoring vendor will provide this ideal solution, but with a planned and purposefully designed open architecture that incorporates the four sources of IT visibility, you can effectively evaluate different solutions and how they fit into a unified ITOA architecture that is appropriate for your organization.

The Four Sources of IT Visibility: A Taxonomy

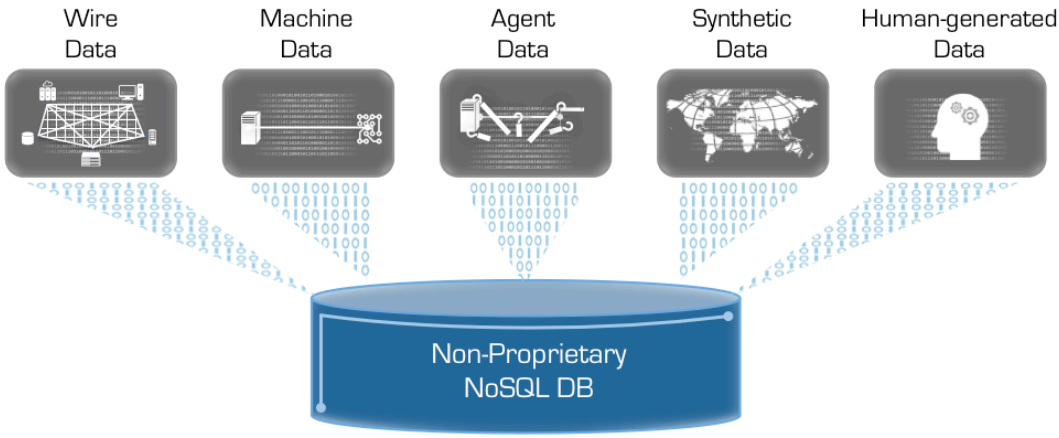
In order to break down the siloes in the IT organization, IT teams need to stop thinking in terms of product categories such as NPM, APM, and EUM and instead think in terms of the data output of those products. Thinking in terms of product categories only leads to greater confusion. Go to any of these vendors' web site and you will see that they all provide application, network, cloud, virtualization, infrastructure, system, database, and storage visibility.

As organizations prepare to adopt ITOA architectures, a new taxonomy can help to clarify what components are required. Borrowing from Big Data principles, a new IT data taxonomy classifies data sets by their source. The benefit of this taxonomy is to accurately set expectations of what types of outputs you can expect from each product.

An IT Operation Analytics (ITOA) architecture must combine the following four IT data sources for complete visibility. Each data source provides important information that makes it well suited for specific objectives and is complementary to the others.

Borrowing from Big Data principles, a new IT data taxonomy classifies data sets by their source to accurately set expectations of what types of outputs you can expect.

1. **Machine data** is system self-reported data, including logging provided by vendors, SNMP, and WMI. This information about system internals and application activity helps IT teams identify overburdened machines, plan capacity, and perform forensic analysis of past events.
2. **Wire data** is all L2-L7 communications between all systems. This source of data has traditionally included including deep packet inspection and header sampling but recent advancements allow for far deeper, real-time wire data analysis.
3. **Agent data** from byte-code instrumentation and call-stack sampling. Code diagnostic tools have traditionally been the purview of Development and QA teams, helping to identify hotspots or errors in software code.
4. **Synthetic data** from synthetic transactions and service checks. This data enables IT teams to test common transactions from locations around the globe or within the datacenter.



The following sections describe each source of data in detail. It is important to understand the capabilities and limitations of each data source so that you can determine its applicability in your organization. This paper will not cover the capabilities of products that consume machine data and only discuss the data source itself and what that source can and cannot provide to assist end-users.

Machine Data

Definition: Machine data is system self-reported information produced by clients, servers, network devices, security appliances, applications, remote sensors, or any data that is self-reported by a machine. Machine data outputs tend to be log files or SNMP and WMI polling and traps. Machine data can be classified as time-series, event-driven data.

Examples: SNMP and WMI are common methods for capturing point-in-time metrics, representing active and passive methods of acquiring machine data. When polling a system, you are asking for information. When using traps, you are waiting for the system to push out information. Log files are ubiquitous across systems and devices and include application logs, call records, file system audit logs, and syslog.

Strengths: Machine data is ubiquitous and can contain a wealth and variety of information, including: application counters, exceptions, methods and service states; CPU, memory, and disk utilization and I/O, NIC performance; Windows and Linux events; firewall and security events, configuration changes, audit logs, and sensor data such as temperature and pressure readings. The variety of information to be mined from machine data is nearly endless which makes it a powerful source of data not only for IT Operations but for the business as well. The information about system internals and self-reported application activity helps IT teams identify overburdened machines, plan capacity, and perform forensic analysis of past events.

Limitations: As the data source implies, you are dependent upon the machine to produce timely, thorough, usable, and accurate information. Because of the wide variety of machines,

systems, and applications in existence from many different vendors, the quality, usability, and thoroughness of machine data can vary greatly depending upon vendor and even vary widely among product teams working for the same company. Another limitation is that if a machine or application is under duress, the system may not log information accurately or at all. Finally, there is valuable information that is either impractical or impossible for machines to log. This would include information found in transaction payloads, database stored procedures, storage transaction performance or any timed measurements between multiple systems.

Key Questions

- Are your systems administrators using tools such as PowerShell to manage the types of machine data collected? The answer to this question will help determine how well your organization is using machine data.
- Do you have a log indexing and analysis solution in place? If so, how much data do you send to it for indexing each day?
- What percentage of your server infrastructure is virtualized? In highly virtualized environments, application performance is less correlated with resource utilization metrics than when applications run on dedicated infrastructure.

Wire Data

Definition: Wire data comprises communication that passes between networked systems. At the most fundamental level, wire data is the raw bits flowing between hosts, but it includes the entire OSI stack from L2 to L7, inclusive of TCP state machines and application information passed in L7 payloads. Wire data can be classified as time-series, globally observed transaction-level data.

Example: Traditional network monitoring tools scan traffic to capture basic L2-L4 metrics such as TCP flags, Type of Service (ToS), and bytes by port, while packet analyzers take a deeper look at point-in-time captures using deep packet inspection (DPI), which enables IT teams to view individual or contiguous packet payloads. New high-speed packet processing capabilities make it possible to fully analyze all L2-L7 communications passing over the wire in real time.

Strengths: Considering that a single 10Gbps link can carry over 100TB in a single day, the communication between servers is easily the most voluminous source of data for an ITOA architecture. This data represents a tremendously rich source of IT and business intelligence providing it can be efficiently extracted and analyzed.

- Wire data can be analyzed passively using a copy of network traffic mirrored off a physical or virtual switch. With this approach, there is no need to deploy agents.
- Wire data is deterministic, observed behavior and not self-reported data. So if a transaction failed because a server crashed, that failure will be observed on the wire but would not be reported by the server.
- Wire data touches each element in the application delivery chain, making it an excellent source of correlated, cross-tier visibility.
- Wire data is real-time and always on. Unlike other data sources that may require configuration with changes, wire data provides an uninterrupted view of dynamic environments.

Limitations: The primary limitation of wire data is that not all information important for performance, security, or business analysis passes over the wire. Resource utilization metrics, policy and configuration changes, and internal application execution details are some examples of important information that does not pass between systems on the network. Additionally, wire data is dependent on the quality of the network feed. Overloaded network infrastructure should prioritize production traffic over mirrored traffic, which can lead to lossy wire data feeds when using port mirroring.

Key Questions:

- What is the volume and speed of the communications carried over the network in your datacenter?
- Do you have methods in place to analyze all of your wire data? Do those methods of analysis require engineering expertise?
- How many applications is your IT Operations team responsible for?

SFLOW, NETFLOW, AND PACKET CAPTURE

sFlow and NetFlow collectors produce basic network data that is a pre-cursor to wire data, but is more accurately self-reported machine data derived from network routers and switches. These systems do not transform raw packets to structured data that can be measured, visualized, alerted upon, or trended. However, because they provide L2-L4 visibility similar to wire data, sFlow and NetFlow are comparable to other packet-based tools. Packet-based tools are not useful in Open ITOA architectures because packet captures of greater than 10GB become inefficient for ad hoc IT or business analytics. It is much more effective to use a wire data analytics platform to stream pre-processed, structured data to the ITOA data store.

Agent Data

Definition: Agent data is derived from bytecode instrumentation and call stack sampling. In custom applications, agents hook into the underlying application runtime environment such as .NET or Java, inserting code at the beginning and end of method calls. Agents can then generate method performance profiles, and using tags or keys, they can trace transactions through various tiers of an application.

Examples: Code diagnostic tools have traditionally been the purview of Development and QA teams, helping to identify hotspots or errors in software code. As agents have become more lightweight, they have increasingly been deployed in production environments to better characterize how application code affects performance.

Strengths: Agent-based monitoring is the microscope in the ITOA architecture. If you have identified a specific application or device that requires code-level monitoring and management, agent-based monitoring tools can be incredibly useful. Agent data is vital to DevOps, where IT teams must integrate seamlessly with the development process.

For custom-developed applications, agents provide unprecedented ability to do root cause analysis at a code level. By dumping the call stack, developers can see exactly where the code failed; you can capture method entries and exits, memory allocation and free events, and build method traces. By identifying hotspots and bottlenecks, you can optimize the overall performance of your application.

Limitations: Agents require trade-offs in terms of performance. Because they exist in the execution path, agents, by design, increase overhead and must also support safe reentrance and multithreading. During runtime, these agents share the same process and address space as your application, which means they have direct access to the call stack and heap and can perform stack traces and memory dumps. You can capture every class and method call, but your application will be too burdened to run well in a production environment. Tailoring agents for specific applications requires an understanding of both the agents and the application, and optimizing agent monitoring must be considered as part of the development and QA process for each application rollout.

Deploying agents requires access to application code so that they provide little value for off-the-shelf applications like Microsoft SharePoint, SAP, or Epic EMR. For those types of applications, do not expect any greater insight with agents than could be gathered from SNMP or WMI machine data. In addition, systems that offer no agent hooks (e.g. ADCs/load balancers, DNS servers, firewalls, routers/switches, and storage) cannot be instrumented with agents.

Key Questions

- Does your organization develop mission-critical applications, does it primarily rely on off-the-shelf applications, or does it use both types?
- Is your IT Operations team able to interpret the application code?
- Do you have staff resources available to configure agents for production and maintain compatibility?

Synthetic Data

Definition: Synthetic data originates outside of your application delivery chain through hosted monitoring tools or as part of active service checks. Firing on a predefined schedule, service checks (pingers) can be anything from simple ICMP pings to fully scripted checks that work through the application flow. To more accurately mimic customer geolocation, probes can be fired from around the globe, representing many points of presence. Synthetic data could be characterized as scheduled and scripted time-series transaction-level data.

Example: Typically, pingers answer the question “Is the system alive?” Most pingers are lightweight and easy to implement. There are many vendors who offer hosted ping services, and many free options available for on-premises deployment. Using services that offer many points of presence, IT teams can understand how geography affects the end user experience.

Strengths: Synthetic data enables IT teams to test common transactions from locations around the globe or within the datacenter, and can quickly identify hard failures. Synthetic monitors are easy to implement for both custom and off-the-shelf applications, and can be deployed broadly without significant impact on the application as long as care is taken in setting up the checks.

Limitations: Because synthetic data is generated outside of the application delivery chain, it does not include insight into why the application is slow or failing. Additionally, synthetic data is generated through sampling and not continuously monitoring real-user transactions; it therefore can miss intermittent problems or problems that only affect a segment of users. Likewise, without an understanding of the success or failure of a transaction, it is easy to mark a service green that is accessible but not functioning correctly.

Key Questions

- Does your application have a geographically distributed user base?
- Do you need to monitor the success or failure of specific, well-defined transactions, such as an online checkout process?

Other Sources – Human-Generated Data

Human-generated data includes text documentation, engineering diagrams, Facebook and Twitter posts, and posts and messages from internal social collaboration software. Analyzing human data can provide the end-user contextual information previously unavailable to IT Operations teams. While Big Data solutions can store and manage this type of unstructured data, solutions that convert this data into actionable intelligence are still in their infancy. This remains largely an experimental data source for ITOA architectures. Many applications and use cases for leveraging human-generated data will begin to surface with greater market adoption, which requires tools that take advantage of processing gains to provide real insight.

Tying It All Together: Open Architecture for ITOA

Once you determine the relative importance of each of the four sources of IT visibility to your organization, you are ready to design an Open ITOA architecture. There are three elements of any Open ITOA platform: extraction of the data, storage and indexing of the data, and presentation or visualization of the data. This section addresses the issues that IT organizations must consider for each element.

Selecting Sources of IT Visibility

When evaluating your current toolset, you may determine that some platforms will work for your Open ITOA architecture by providing one or more of the four sources. Or you may decide that it is time for a refresh. When considering whether an existing or new platform is needed to provide machine data, agent data, synthetic data, and wire data, keep in mind these requirements:

- **Openness –** Integration options, including robust APIs and the possibility for real-time data feeds into a backend data store. Batch data integration will not suffice.
- **Flexibility –** Do not depend on vendors to define new metrics or extend the solution. You should be able to easily adapt the data collected and analyzed to meet new IT and business requirements.
- **Scalability –** Solutions that have onerous licensing or fees for data usage will hinder the growth and utility of your Open ITOA architecture. You should select solutions where you are not constrained by the number of data points or metrics stored based on cost.

Selecting a Data Store for the Open ITOA Platform

The real power of your Open ITOA platform is unlocked when your data sources are combined in an open, non-proprietary data store that enables contextual search and ad hoc queries, such as Elasticsearch or MongoDB. This is a case where the sum becomes greater than the parts.

When selecting a data store, consider the characteristics of Big Data in general, namely the ever-increasing volume, variety, and velocity of data. With that in mind, the considerations for an Open ITOA data store include:

- **Cost –** Plan for twice as much capacity as you initially think adequate. Your data usage will grow with broader adoption and as your organization adds use cases. Selecting a non-proprietary data store will help avoid licensing costs.
- **Vendor lock-in –** Your data is extremely valuable and should be portable should you decide to change your ITOA data store. Be wary of data stores that are closed or limit how you can use your data. Choosing an open data store will ensure that you have the flexibility to adopt the best technologies in the future.
- **Ecosystem –** Data stores that have broad ecosystems will enable you to take advantage of community resources.
- **Scalability –** The data store should be able to scale out storage and processing capacity easily and require minimal ongoing maintenance as the environment grows.

Selecting a Visualization Solution

Many organizations already have visualization solutions in place, such as Tableau, Chartio, Domo, Pentaho, or JSON Studio. It may make sense to examine those and see if these existing visualization solutions can be applied to your ITOA platform, especially if one of the primary goals is to provide increased value for sales, marketing, customer support, and other non-IT stakeholders that are already using those solutions. A key selection criteria in choosing the non-proprietary data store will be the type and variety of visualization solutions currently available for that data store.

Conclusion

Every IT executive today must grapple with building an ITOA platform. It is a strategic issue with significant ramifications, not only in regard to cost but also the competitiveness of the business. According to Gartner, global spending on ITOA products and services is projected to reach $1.6 billion by the end of 2014, representing a 100 percent increase over 2013 levels.[1] Starting out with a purposefully designed architecture for ITOA will ensure cost-effectiveness, scalability, and the best results for your organization.

The first step in planning an open ITOA architecture is understanding the relative importance of the four sources of IT visibility for your organization and then selecting the right-fit solutions for each data source. The second step is to choose an open data store that can consume real-time streams for each data source and does not limit the amount of data you store or how you use it. Finally, you will want to select the best visualization solution that will provide insights to both IT and non-IT stakeholders.

ExtraHop believes that an Open ITOA architecture is the best future direction for IT organizations and where the IT operations management industry is headed. To learn more about how ExtraHop is leading the industry in supporting an Open ITOA approach with our open and scalable wire data analytics platform, visit www.extrahop.com.

[1] Gartner, June 2014: “ITOA at Mid-Year 2014: Five Key Trends Driving Growth” http://www.gartner.com/document/2783217