Zenoss®

WHITE PAPER

Zenoss Enterprise Architecture Overview

Zenoss® Enterprise is IT management software designed to meet today's datacenter operations challenges. It is a single product that provides full stack coverage of networks, servers, applications, services, and virtualization. Functionally, it provides complete operational awareness by combining discovery and inventory, availability and performance monitoring, event management, and reporting. And since it's built with open source technology, it has the reliability, cost, and support benefits that come with an active community of thousands of users.

This paper explains the architectural design that collectively distinguishes Zenoss from legacy enterprise management products. You'll be introduced to its five key architectural elements and see how these combine to deliver Dynamic Service Assurance to the Dynamic Datacenter.

Introduction

The new challenges of dynamic virtualization environments and use of external "cloud" service providers fundamentally change the requirements and economics of management.

What's different about today's data center?

- IT organizations no longer subscribes to the one server – one application world view
- Cost of computing has plummeted
- Device count in the data center has exploded
- Operations team is less technology specialized, more business focused
- Virtualization creates an extremely dynamic server environment
- Wide-scale adoption of outsourced applications
- Emerging use of cloud services

Irrespective of the changes, you still need to provide the same service quality.

Many IT organizations are forced to rely on legacy management toolsets. These toolsets generally include many products that must be purchased, licensed, installed, and maintained separately. Each element must be integrated separately, usually into yet another tool. It's just too hard to do not to mention the institutional non-transferable knowledge that accumulates over time.

On the other hand using a loosely integrated set of specialized products selected to increase individual productivity has its own limitations. Without a comprehensive, integrated view of devices, networks, and applications, identifying current status and diagnosing issues can be extremely difficult.

Introducing Zenoss™

Zenoss™ Enterprise is IT management software designed to meet today's dynamic datacenter operations challenges. It is a single product that provides full stack coverage of networks, servers, applications, services, and virtualization. Functionally, it provides Dynamic Service Assurance by combining discovery and inventory, availability and performance monitoring, event management, and reporting. And since it's built with open source technology it has the reliability, cost, and support benefits that come with an active community of thousands of users.

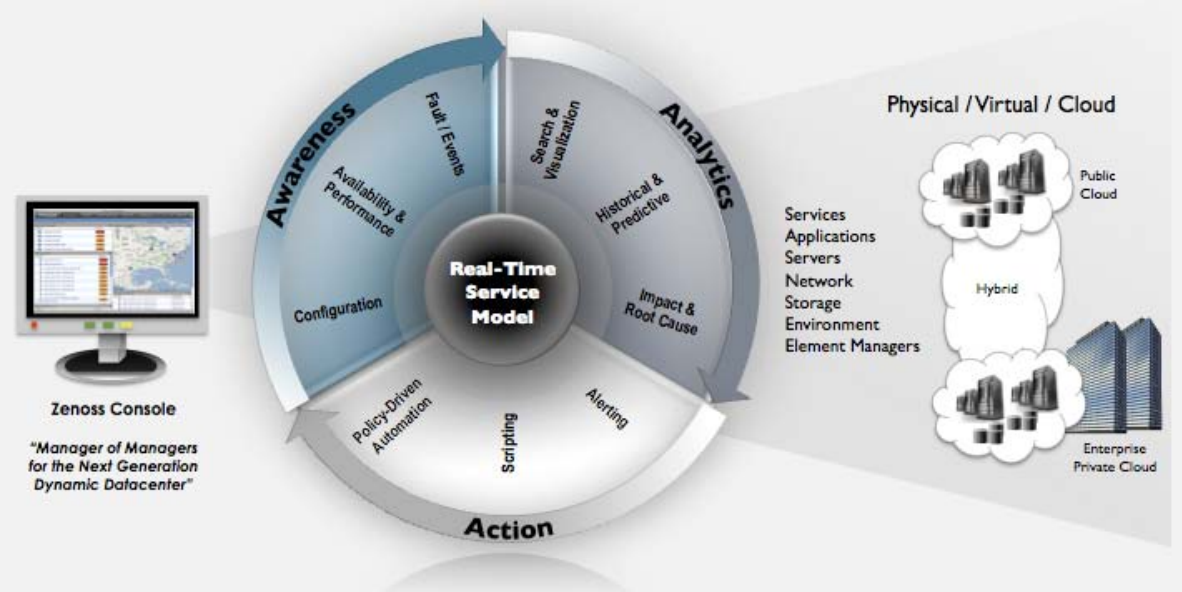
This paper explains the architectural design that collectively distinguishes Zenoss from legacy enterprise management products - making it the best Unlegacy ™ solution for managing your dynamic datacenter. You'll be introduced to its five key architectural elements and see how these combine to handle the most difficult challenges of today's data center.

Zenoss Enterprise: Functional Overview

Zenoss is management software designed for the dynamic datacenter. It automatically discovers and models networks and devices, then applies precise monitoring templates to collect consistent data. Across physical and virtual servers, networks, and applications it:

- measures availability and performance
- monitors key events and faults
- executes synthetic transactions
- provides alerting and remediation
- offers integrated cross-object reporting
- maintains an accurate real-time view of the data center using a dynamic model.

An ajax[1] based, extensible html user interface ties these functions together with precise security controls for individual requirements.



Zenoss Enterprise is based on the open source Zenoss Core project. An active community with more than 8,000 deployments has guided development to an extremely stable, high-quality product with an extraordinary range of device and application support.

Key Features

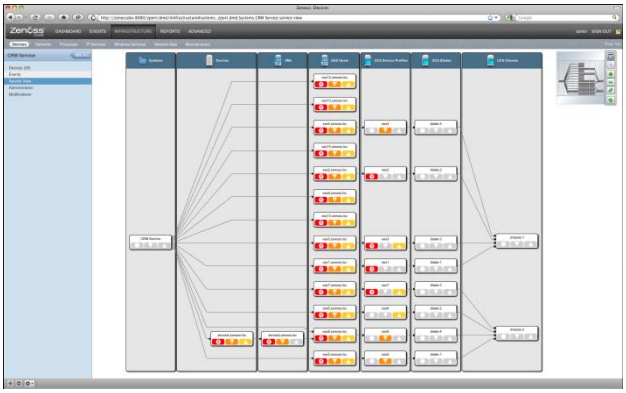
Zenoss provides these key features:

Discovery

Zenoss finds and identifies network-attached devices. You'll always have an accurate inventory of what's there and it is extremely simple to identify new devices. Immediately following initial discovery, Zenoss builds a detailed model of devices, their configuration, network dependencies and routes and then uses that model to enable all other functions.

[1] Ajax Asynchronous Javascript and XML

Dynamic Service Views give you the ability to use these models to see device dependencies in real time. For example, you'll be able to identify shared resources such as ESX servers which may be a source of contention and potential single points of failure such as a Cisco UCS chassis. At each level, you will be able see the status for all devices and their dependencies.



Availability Monitoring

The availability monitoring system provides active testing of the IT infrastructure. Zenoss discovers the routes to each device and builds a network map. It uses this network map to discover relationships among routes and devices so that you won't be flooded with availability failure events from every device behind a failed router.

Simple network presence is a good, basic indicator of service availability. Zenoss goes further, allowing you to easily determine if web, mail, file transfer, and many other services are working.

Performance Monitoring

Collecting and monitoring performance metrics is vital in problem detection and capacity management. Zenoss collects and stores performance data by using several methods including SNMP, WMI, shell commands, and application APIs. It then analyzes the data for user defined threshold violations and makes the data available through analytical charts and reports. You're probably familiar with single value thresholds and predictive thresholds. Zenoss provides for both types, so that you are alerted to issues that need immediate correction and notified early of evolving issues to head them off.

Event Management

Zenoss provides a single event console that brings together events from all managed devices with Zenoss-generated alerts. Excellent notification and escalation management ensures that the right people become aware of critical problems, and automatic up-down correlation helps you focus on real issues, not momentary interruptions.

Keeping up with a large number of events is straightforward in Zenoss. New events are visually distinct from past acknowledged events. Acknowledging events is a powerful and simple way for your support team to collaborate. This allows you to respond to issues rapidly without stepping over each another's efforts. Events that are no longer relevant can be moved to an event history and are automatically moved after a defined time period. You can communicate findings to others, or help yourself remember actions or status by adding comments to an event.

Key Tenets

To understand how and why Zenoss is different, it's useful to understand the core ideas that drove the development of Zenoss. Zenoss was developed by a team of operations professionals, familiar with operations needs and the shortcomings of legacy products. From the pain of managing this gap came a set of core ideas on which Zenoss was built.

Full IT Infrastructure

Unlike other tools, Zenoss' inclusive approach unifies all areas of the IT infrastructure to eliminate the need to use multiple tools, whether from one or multiple vendors.

Zenoss monitors the performance and availability of heterogeneous operating systems (including Windows, Linux, and Unix), SNMP-enabled network devices (such as Cisco), and a variety of software applications (such as WebLogic and VMware). It natively supports virtualized and real servers as well as mass-virtualization enabling infrastructure (such as UCS).

Scale

If a product can't manage all your devices, it cannot be your central monitor. Your monitoring product needs to support as many devices as you choose to watch. While Zenoss can be deployed on a single server to manage hundreds of devices, it also manages large, distributed networks with horizontal scaling. The largest current deployment is over 32,000 devices.

Agentless Data Collection

You probably already know the pain of dealing with agents. Agents need to be deployed, configured, secured, and updated and that takes scarce staff time away from delivering customer value. Worse, you'll find that adding agents to every guest can raise memory overhead in virtualized environments. resulting in fewer guests per host. By contrast, Zenoss relies on agentless data collection, taking full advantage of the management function built into nearly every device today.

Many customers use Zenoss to help manage cloud-based services, where agent deployment is simply impossible. If a legacy system requires agents to function, and the cloud vendor won't allow that access, then you're stuck. Either you're delivering services to end users and cannot monitor availability and performance, or you install separate point products and attempt to integrate the solutions. By removing the agent requirement, Zenoss simplifies your ability to monitor cloud services along with your network devices, real and virtualized servers.

Modeling

Large operations centers know that managing similar devices in a consistent way leads to high efficiency. Zenoss builds and maintains a detailed model your environment enabling it to adapt itself to your operation. The model is object-based and can be easily extended. You apply desired management settings to the model, and Zenoss executes your policy. That's very different than legacy systems in which you configure each agent on each system, then do it again every time you need to change what you're monitoring. In addition, the Zenoss model keeps near real time track of changes that occur in virtualized environments. There is significant loss of service if you just have DR and HA configured in your virtualization infrastructure, but your monitoring tool cannot keep up with changes as they occur.

Extensibility

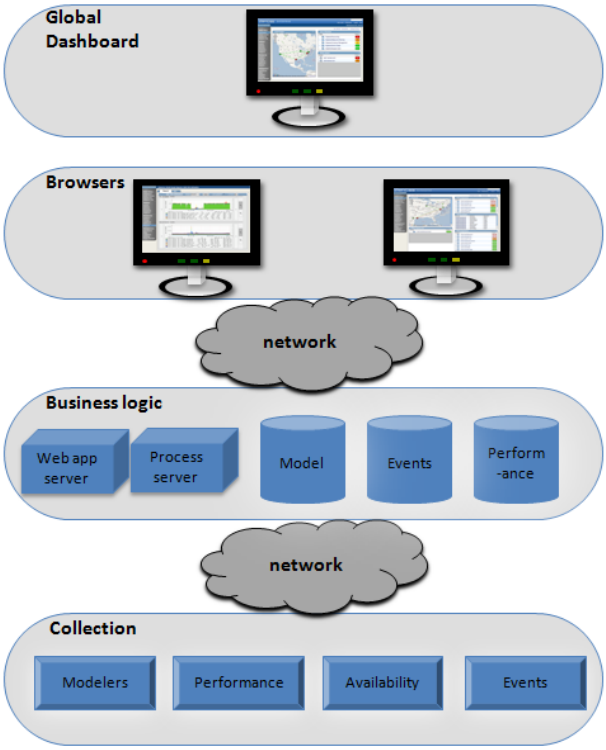
Relying on a vendor to deliver everything you need is a frustrating experience, particularly if the feature you need is "not on the roadmap." Zenoss was designed from the start to be extensible. You have the ability to add to the model, to add new data collection methods, even to modify the source code.

As for device and application support, more than 170 Zenoss ZenPacks allow for immediate productivity. ZenPacks are provided by Zenoss and by the Zenoss community of customer experts. A comprehensive list of supported ZenPacks is available in *Zenoss Extended Monitoring Guide* available at http://www.zenoss.com/community/docs/. In addition, the Zenoss community has contributed a large number of additional ZenPacks that can be reviewed at http://www.zenoss.com/community/projects/zenpacks/.

Zenoss Architecture: Four Tiers; Model Driven

Understanding the architectural design of a system is an extremely useful way of understanding how the functional needs and design tenets come together into a product.

The two key elements of the architecture are its four tier design and model-driven management. Together, these two elements provide scalability that enables management of tens of thousands of devices.



Four-Tier Design

Zenoss' system architecture is a four-tier design. Elements in every tier can be deployed in multiple instances to support scale, security, and separation needs found in complex distributed organizations.

Global Dashboard

The Zenoss Global Dashboard provides consolidated views of information across several separate Zenoss deployments. Typically, environments with very high numbers of devices, or with business needs that call for complete isolation of monitoring environments will use the Global Dashboard.

The Global Dashboard is a Web server application that aggregates event and heartbeat data from associated Zenoss servers. Detailed information remains stored on the individual Zenoss deployments; if you're using the Global Dashboard, you can simply click through to the proper deployment to investigate an issue.

Presentation Layer

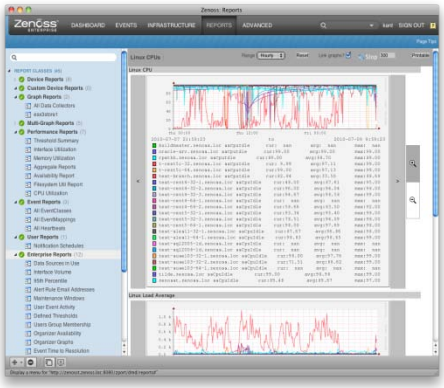
The Zenoss Web interface combines the console and reporting functions in a single Web application. It also enables management of Zenoss itself. Multiple people, using secure custom views, can:

- Monitor status by business system, geographic location, or specific "watch list" devices
- Work with accurate, current device information including performance, availability, and configuration
- Monitor, track, and respond to events
- Use powerful dynamic views for industry leading visualization of managed devices.
- Move devices in or out of production status
- Create and run analytic reports
- Include or exclude devices from management
- Configure, apply, and customize monitoring templates
- Define new Zenoss users and adjust functional and device access control

When you first start the Web interface, you see the Dashboard and its collection of at-a-glance data views. Individuals can choose which of these views ("portlets") are displayed on their Dashboard; you can restrict devices and functions that any Zenoss user can access. The Google Maps portlet shows the status of devices in a location and the logical connections between locations; it's a very popular feature because the location organization and connector status is delivered automatically from the Zenoss model. There's no need to place each device and define each connection, as many legacy systems require.

Zenoss reports are integrated into the Web interface. The basic report categories are device, event, performance, and user.

- Device reports show data over sets of devices, including new devices and configuration changes.
- Aggregate information covering events are in the Event category.
- The Performance category includes reports useful for capacity planning and service level assurance reporting. For example, it's useful to know the percentage of time a device was operating outside of its desired threshold.
- The final User category helps you plan staff notification schedules.



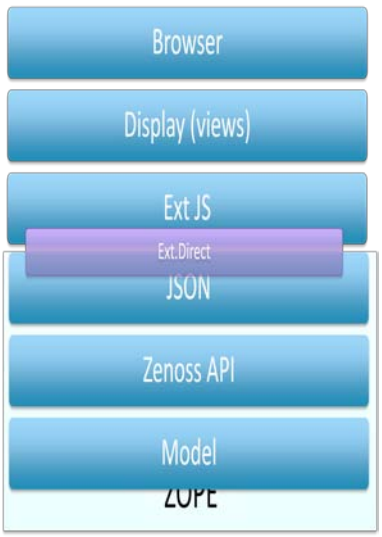
The sample report shown is a multi-device performance report. By comparing selected data from different devices you can identify the root cause, or even solve an application issue. For example you can see which VMware guest could benefit from more resources while simultaneously identifying guests that can be resized to provide those resources.

Business Logic

The Zenoss business logic layer can be thought of as a Web application, a process daemon, and three data sources. These can be implemented on a single server or spread across several servers for higher performance or to meet geographical needs.

Web Application

The Web application is built using a collection of open source presentation technologies on top of the Zope application server. Zope was chosen for its high development productivity, ease of management, object database, and excellent security model. Overall, the Web application supports the user layer with scale-aware design intended to handle very large datasets and perform well even when handling thousands of objects. The Web application is specifically aware of distributed data resources including multiple performance data databases (see below). In addition to Zope the Zenoss web app makes extensive use of the ExtJS toolkit. The front end is extensible and makes very heavy use of ajax to offer desktop like behavior via a web application (such as drag and drop capablities)



According to http://www.zope.com/WhatIsZope, *"Zope is an open source Web application server primarily written in the Python programming language. It features a transactional object database which can store not only content and custom data, but also dynamic HTML templates, scripts, a search engine, and relational database (RDBMS) connections and code. It features a strong through-the-Web development model, allowing you to update your Web site from anywhere in the world. To allow for this, Zope also features a tightly integrated security model. Built around the concept of 'safe delegation of control', Zope's security architecture also allows you to turn control over parts of a Web site to other organizations or individuals. The transactional model applies not only to Zope's object database, but to many relational*

database connectors as well, allowing for strong data integrity. This transaction model happens automatically, ensuring that all data is successfully stored in connected data sources by the time a response is returned to a Web browser or other client."

Process Daemon

The Zenoss process daemon manages communications between the collection and data layers, executes timed on-demand administrative tasks, and performs device management tasks initiated by users. This daemon can be replicated to multiple servers for increased scalability or to meet particular enterprise network design needs, such as multiple sites or overlapping IP address spaces.

The daemon is designed to use event-driven processing wherever possible to avoid unnecessary polling, large batch processes, and overly broad queries. It uses intelligent caching wherever practical to allow for disconnected operation of collectors and consistent performance.

Databases

Zenoss stores data in three databases:

- Zope for the Zenoss object model
- RRDTool for time-series performance data
- MySQL for event data

Each database contributes significant strengths to the overall Zenoss product. In large environments, customers commonly separate the databases onto separate servers to handle high loads and meet separation-of-data needs.

Zope stores data for devices, components, groups, and locations as the core configuration model for Zenoss. As the database is populated in discovery, the Web application automatically adjusts to the discovered data.

Performance data, such as processor usage, has unique storage requirements. RRDTool precisely delivers the requirements needed, including automatic grooming, interpolation for missed data points, and a very compact store. Learn more about the open source RRDTool at http://oss.oetiker.ch/rrdtool/.

Zenoss event data is stored in a MySQL database. MySQL is a traditional relational database, well suited for uses involving the structured textual data that make up events. Zenoss captures event data from multiple sources and maps it to a common database format, transforming raw device data with useful extensions from its model. One particularly useful transform is automatic correlation of up/down messages that eliminates pointless administrative work resulting from short-term issues. MySQL is the world's most popular open source database. Learn more at http://www.mysql.com.

Zenoss Collection Layer

Zenoss uses native management function built into devices by their manufacturers to provide operating data. These functions have now reached a level of maturity and ubiquity where the additional benefits of a proprietary agent were either absent entirely or not worth the costs of agents.

One of the single most effective ways to reduce the costs of management is to eliminate the provisioning, securing, and configuration overhead of agent management. This cost is significantly increased when legacy management systems require different agents.

New agent versions may supply new and valuable features but the opportunity cost of agent deployment as compared with other projects often means that older agents are left in place for years.

Other agent-related issues can also increase costs and reduce benefits. You may need to support multiple agent versions simultaneously. Stable business systems running older operating system versions and new applications on new hardware often require different agent versions. You may need to deploy multiple versions of the same monitoring platform, each working with specific agent versions. Monitoring teams will need education to understand the operational differences, too.

Legacy management systems were generally developed well before management function and communications protocols were standardized, and the costs were a necessary evil. Legacy vendors and their customers typically remain burdened by these vestigial artifacts of the past.

Zenoss' agentless collection layer comprises services that collect and feed data to the data layer. These services are provided by several daemons that perform modeling, monitoring, and event collection functions. The modeling system uses SNMP, SSH, and WMI to collect information from remote machines. The raw information is normalized into a format that matches the core Zenoss model before being inserted into the databases. The multi-daemon design provides significant flexibility to distribute and segment data collection to meet unique requirements. The Zenoss collector layer is architected to be field or customer extensible, it is based on a object oriented model with inheritance delivering most of the glue code needed for a new collector to interface with the core Zenoss process. This way a customer can quickly write a collector that solves their business problem and focus just on the business code and not worry about product specific integration issues.

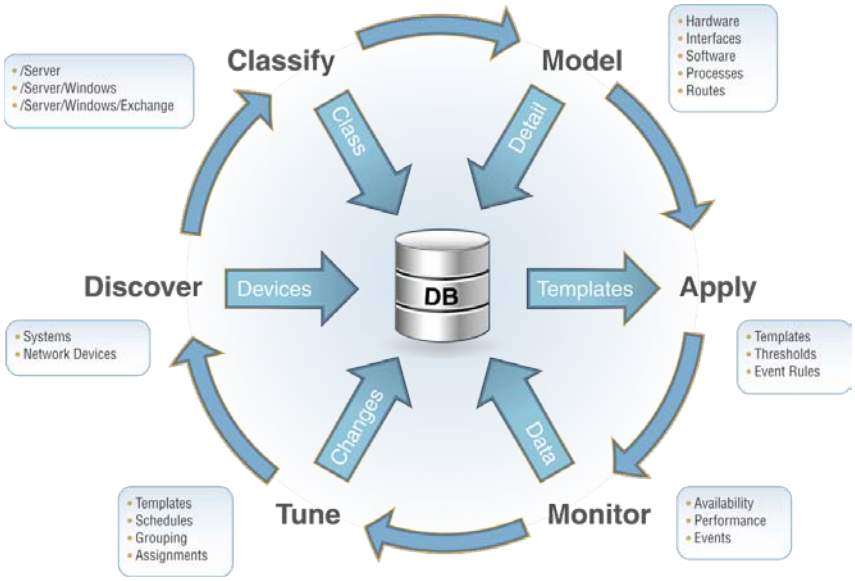
With multiple collector machines, Zenoss customers can manage extremely large numbers of devices, operate efficiently in multi-site wide area network environments, handle difficult networking setups such as overlapping IP spaces, and customize management models for different customer needs.

Model-driven Management

The second key element of the Zenoss architecture is its model-based design. The model enables easy use of consistent monitoring policies. Monitoring policies are enhanced with application, device, and function monitoring templates provided in more than 200 ZenPacks.

The Model

At the heart of Zenoss Enterprise is a unified model of the entire IT environment. Housed in the Zope configuration database, the model uses data abstraction, encapsulation, and inheritance to provide simple initial configuration and ongoing use. This diagram can help you understand how the Zenoss model works and how you can apply it:



Discovery

Zenoss populates its database with devices it discovers, generally through an IP address space scan. You can choose to populate the database with your own device lists. This is useful where address space scans is blocked or when you choose to pick specific devices to monitor.

Classify

You choose a classification for each device either individually or through multi-select. By classifying a device, you tell Zenoss which monitoring behaviors to use with the device. Of course, you can re-classify a device at any time.

Model

Once Zenoss knows the classification of a device, it understands how to collect detailed configuration information from that device. Configuration information is collected in specific ways for each device class. For example, SNMP is used for networking devices, WMI for Windows machines, and the VI API with VMware ESX servers. Collected model data is stored in the configuration management database and can be accessed by other applications by direct database access, XML export, and through a Web service.

Apply

With an understanding of device classification and its model, Zenoss ensures that each device is monitored appropriately and completely using performance templates, thresholds, and event rules. It automatically configures itself so that the proper performance metrics are gathered, thresholds monitored, and events processed without your having to manually manage detailed differences among devices. For example, it can track network utilization on all interfaces without you needing to know precisely how many interfaces exist in advance.

Monitor

Zenoss collectors gather, receive, process, and store management data for all monitored devices and store it into the appropriate Zenoss database. Collected data is processed as it is received using data abstraction rules. For example, performance data is normalized so that values that should be represented as percentages are stored as percentages regardless of the scale of the actual metric, interface speeds are represented consistently, etc. Event records are fully populated by rule with fields filled by configuration data from the model and from data extracted, parsed, and inferred through regular expression matching.

Tune

Although Zenoss works very well without special adjustments you will undoubtedly find that you want to make its operations more closely match your needs. You can adjust the model, and hence Zenoss' behavior, by adding maintenance schedules, device groupings, and operator assignments. You can collect different performance data, change thresholds, or define new event rules for either an entire device class or just for a single device. You can monitor a server-based application by binding applications templates to a server orthogonally to the hierarchical classification. You can even write entirely new collection daemons to take advantage of specific APIs, much as Zenoss has already done for VMware, JMX, and more.

Inheritance drives monitoring policy

Monitoring using inheritance makes it very easy for you to define and use a consistent monitoring policy.

When you apply a template at a point in the hierarchy, you ensure that every sub-device receives the same template. Zenoss uses configuration information collected during modeling to determine what parts of the template to apply – for example using WMI to collect information from Windows and SSH from Linux servers.

If you define a template at the /Devices/Server level to collect CPU utilization, of course all the servers will be collecting CPU utilization. But you'll get a lot more than that!

- Since all the schedules are the same, you'll be able to see how application usage loads affect different servers at the same times
- Zenoss ensures that correct techniques for each operating system are used to collect and normalize data. You won't have any errors from not understanding the difference between a 0-1 scale, and 0-100 scale, and a 0-10000 scale from different servers!

ZenPacks

Collections of Zenoss monitoring function are called Zenpacks. More than 200 Zenpacks are available, provided by Zenoss and by experts in the broad Zenoss community. Zenpacks cover devices such as APC power systems, Cisco routers and switches, and EMC storage; server operating systems from AIX to Windows; applications such as Apache, IIS, and SQL Server; and provide enhanced function like file change monitoring. For a complete list of Zenpacks, please visit http://www.zenoss.com/community/projects/zenpacks.

Zenpacks include model extensions, performance collection and thresholds, availability, event rules, reports, and even new collection daemons. Applying a template to a device will automatically extend the object model for a device, and it will be represented in the user interface immediately

Two specific Zenpack functions are worth discussing individually - predictive thresholds and synthetic transactions. In nearly every legacy management system, these are separate, often costly, products. With Zenoss Enterprise, both are included.

Predictive Thresholds

If you've ever tried to define and tune performance thresholds in legacy systems you've undoubtedly experienced some frustrations. The "rules-of-thumb" published on vendor Web sites usually apply to systems that are actually failing. By the time an individual server hits 90% processor utilization, for example, the business system it is part of has generally slowed to a crawl. The threshold warning comes too late to allow proactive response. Legacy management vendors usually provide thresholds that just aren't right for your environment and adjusting these on a device-by-device basis can be extraordinarily time-consuming.

By contrast, the Holt-Winters Zenpack applies automatic predictive thresholds to any performance metric. It allows you to determine if any performance metric is trending out of normal, even if you have no advanced knowledge of normal ranges. Using the Holt-Winters exponential smoothing algorithm (the Wikipedia article http://en.wikipedia.org/wiki/Exponential_smoothing is a fairly good introduction) Zenoss is able to forecast emerging issues, permitting corrective action to be taken early.

Synthetic Transactions

In today's data center, traditional technical performance metrics may be unavailable or inappropriate. Cloud-based applications such as Salesforce.com generally do not provide monitoring access to their underlying systems. Customer-facing Web applications may deliver unacceptable performance even while underlying systems appear to be working successfully. What's needed is the ability to execute and measure the results of real transactions.

Zenoss includes the ability to record and playback complex Web interactions and use the results for monitoring. You can monitor availability and performance of cloud-based applications, alert when response time thresholds are exceeded, and easily compare metrics from these synthetic transactions with metrics from your production systems.

In addition to Web transactions, Zenoss can also test relational database queries and mail delivery.

Access Control

Access control is built into the Zenoss model as well. With Zenoss, you can give specific users and groups a precisely limited view of the monitoring environment. Zenoss integrates with your enterprise Active Directory or LDAP directory to provide role-based control over the devices users see and the actions they can take.

Many IT operations managers were surprised to discover the extent of the business controls required for Sarbanes-Oxley compliance in IT operations. Unlike many legacy tools, Zenoss was designed with access control and change identification.

Four basic roles can be combined with device permissions to give users precisely the view and power you want them to have. The basic roles are view only, view and act, configure, and event manager. You assign breadth of authority using the Zenoss object model, so you can allow a user access to groups, systems, locations, or devices. The Web client automatically filters device lists to match the scope of restricted users, and limits reports as well.

To ensure business controls are followed, you must watch for device changes, new devices, and changes to the monitoring environment itself. The Zenoss discover-model-monitor approach helps significantly with this, ensuring new devices are identified and easily added to standard monitoring. Standard reports identify device changes and new devices. Changes to the model itself are tracked with time of change, user, and a description of the change. You'll be able to identify changes within Zenoss, and your auditors will have the information they require.

Issues in Today's Dynamic Datacenters

The Zenoss component architecture and model are ideal for dealing with issues of today's dynamic data center. The dramatically lower cost of computing and widespread use of virtualization has combined to drive the number of devices you need to manage in the data center to record numbers. At the same time the rate of change in those devices has greatly increased. Many organizations now provide different levels of service to different customers and some organizations make outsourcing their entire business. Zenoss helps you effectively deal with each of these challenges.

Managing Virtualization

Large and small organizations have rapidly adopted virtualization solutions to increase hardware utilization, improve availability, and enable more dynamic response to problem situations. However, legacy management products face significant challenges in integrating the physical and virtual environments and managing the frequent addition, movement, and deletion of guests. The Zenoss model-driven management solution is a welcome exception, representing virtual and physical infrastructure components in one product. You'll be able to track interactions between physical and virtual machines and drill down from virtual machine, to guest, to guest operating system, to individual applications running on the guests.

A fundamental challenge for legacy management vendors is an inflexible data model based around servers as fundamental devices. But in a virtual world, servers contain other servers. Those broadly adopting virtualization technology have often been forced to deploy separate management systems as a standard practice. With parallel deployments comes poor problem resolution, as staff is forced to switch back and forth between consoles with different management models to attempt to track and diagnose problems. Even as legacy vendors have acquired or built special purpose virtual management products, this dichotomy has continued.

Virtualization Lifecycle with Zenoss

Change Management

Each new guest means two new objects to monitor – the guest in the virtual host and the new operating system instance. Zenoss automatically adds both, and then uses its model to ensure that your monitoring policy is followed.

Performance Debugging

When a new application isn't performing well, you'll often find VMware specialists jumping from the operating system to the ESX Web interface to the ESX command shell, discovering which guests don't cooperate well and should be moved to different hosts. Zenoss brings all the information into one interface and its simple charts make it easy to visually identify problems.

Performance Degradation

The leading cause of performance degradation in a virtual environment is increasing demands on the storage subsystem – and it's the hardest thing to detect, too. While virtual host management tools easily let you see memory and processor over-commitment, they don't give you a view of storage use nor how it has changed over time. Zenoss directly manages common storage devices – and you'll be able to spot how close you are to IO capacity limits in your SAN.

Gartner Research Notes the Importance of Unified Management

"While some environments could support virtual-only clusters for testing, the more common scenario would have, for instance, two virtual elements and one physical one supporting a single IT service. IT still needs to correlate performance metrics and understand the profile of the service that spans the virtual and physical infrastructures."

– Cameron Haight, Gartner Research Vice President

Zenoss is fundamentally different. Its model-driven architecture permits virtual clusters, hosts, guests, and data stores to be represented natively, with full, natural integration between virtual hardware and guest operating systems. It's easy to drill to the root cause of a problem thanks to this native model.

Zenoss collects data from VMWare using collectors that work with Virtual Center and ESX servers. It uses the VI SDK to collect performance and event data, and to discover and populate the data model with current configuration data. The SSH daemon can collect information directly from ESX command shell commands. The Zenoss console uses the model to display ESX servers, Virtual machine guests, ESX clusters, and VMWare data stores.

When a guest moves from one ESX server to another Zenoss uses the same VMWare APIs to track the new location of the guest and ensure that relevant performance and events from the old and new location are grouped. This is true whether the guest movement is due to operator command or automated action. Your monitoring team will see a Web console event indicating the motion, and can easily see if a guest is moving too frequently.

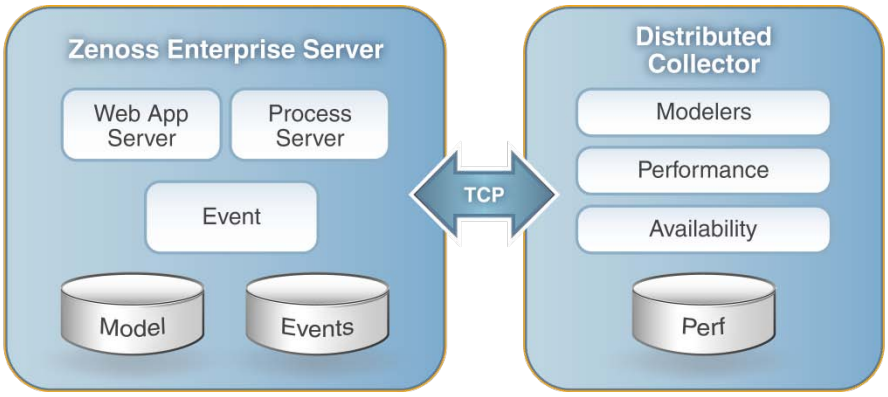
Handling Tens of Thousands of Devices

Zenoss currently manages networks as large as 32,000 devices. Customers appreciate the architectural and licensing flexibility that allows them to easily meet their technical requirements.

Components in the four-tier Zenoss architecture can be deployed in multiple ways to meet different needs. The most common needs are managing large numbers of devices, managing devices in remote locations, and respecting security boundaries.

Networks of a few hundred devices can be supported on a single server running all Zenoss components. With larger networks, or networks requiring higher data collection rates, it is common to install collectors onto multiple servers.

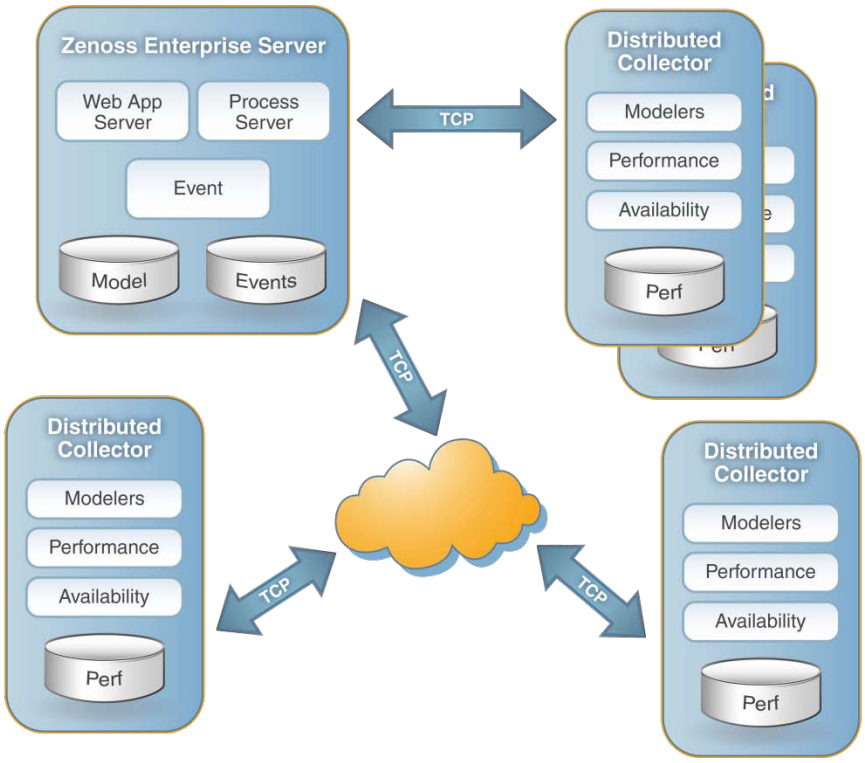
You can also choose to distribute the event database to a separate server to meet lengthy historical event retention requirements, handle very large event volumes such as Web proxy traffic, and enable higher performance event access by other applications.



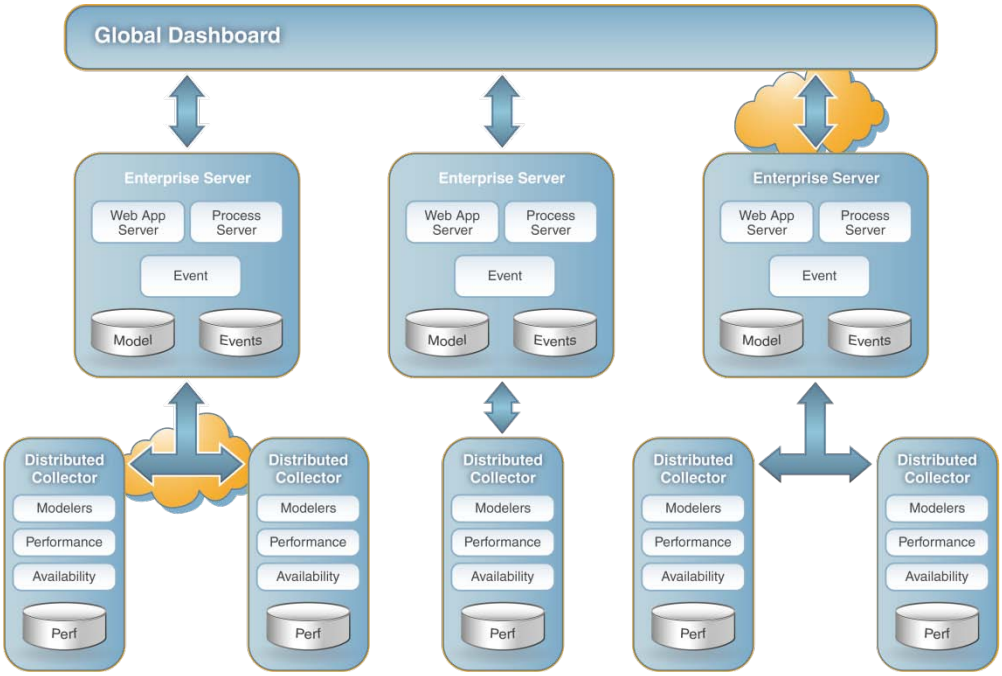
Still larger networks, frequently involving multiple large and geographically distinct locations, generally use a distributed collector in each location. Typical characteristics of this network include one or more of the following:

- networks of more than 1,000 devices
- networks that include devices that are very slow to respond
- remote locations where SNMP and WMI protocols are not allowed on the wide area network
- slow wide area network links

Multiple collectors are also used to manage overlapping IP address spaces, which can occur in mergers or acquisitions, managed service providers, and identically configured branch office networks.



In the very largest of networks, involving tens of thousands of devices, multiple Zenoss Enterprise servers are used with the Zenoss Global Dashboard, which provides consolidated summaries. Dashboard users can drill down through events into the detailed data for any device in the client console associated with the Enterprise Server where that device is monitored.



Deploying for Multiple Customers

IT operations staffs often find that they need to provide disparate service levels to different "customers." Whether those customers are internal to the organization or external, you need to manage their equipment according to individual requirements and/or SLAs. You might need to provide separate consoles, different monitoring policies, or complete data isolation. Zenoss can be used to meet each of these requirements.

Through the directory-integrated access control function, Zenoss can provide completely separate consoles. You can use service grouping within those consoles to help your customers understand how their application is performing, and whether issues are specific to them or part of a supporting function such as e-mail or network services.

Many organizations provide different monitoring levels to different users. This is a straightforward process with Zenoss. For example, you can deliver gold, silver, and bronze levels by simply defining new event classes for each. A bronze service level might offer simple availability monitoring only. Silver could add tcp service monitoring and Web response time testing, while gold might include full operating system performance and event integration. From the console, your staff can easily identify which SLA applies to a device and act accordingly.

If you have geographically distributed customers but need to provide central monitoring, consider placing a collector at each location. It's straightforward to provide a secure virtual private network channel between the collector and the central Zenoss server. A VPN will maintain secure network standards, and distributed collectors reduce WAN traffic significantly. Performance data is stored at the collector, so detailed performance data never needs to cross the WAN.

Finally, for full isolation, you can use a completely separate Zenoss installation for a customer. The Global Dashboard will continue to give you the central view of critical events you need to provide central fault detection.

Why do enterprises pick Zenoss?

Security – Precisely define devices and functions available to Zenoss users, integrated with your enterprise AD or LDAP directory.

Scaling – Deploy distributed collectors to support thousands of devices, honor enterprise or customer network boundaries. Tie together separate Zenoss installations with the Global Dashboard.

Integration – Feed events into Remedy, use bi-directional SNMP, capture network configuration changes, etc.

Analytics – A broad range of reports designed specifically for large organizations.

Application Support – Virtualization extensions and a broad range of supported enterprise applications complement the breadth of support from the community.

Commercial Relationship – Support agreements, installation and customization assistance, tested and certified releases, and intellectual property indemnification.

Summary: Five Ways Zenoss Enterprise is Different

To summarize, here are the five characteristics that make Zenoss Enterprise's unlegacy architecture the right choice for you.

One: Zenoss Enterprise is a single model-driven product

The heart of Zenoss Enterprise is an active, intelligent, auto updating configuration model that unifies and drives all of the operational disciplines of service management up and down your infrastructure stack from servers, to network devices, and applications across physical and virtual realms.

Two: Zenoss Enterprise is agentless

Zenoss Enterprise's agentless, intelligent model-driven architecture allows you to instantly deploy and continuously discover, track, visualize and manage your network. The agentless architecture works equally well to manage physical, virtualized, and cloud-enabled services.

Three: Zenoss Enterprise incorporates open source community power

Zenoss operates on a powerful combination of open source software development and commercial software delivery. This combination allows Zenoss to gain significant development, testing and business leverage.

Four: Zenoss Enterprise adapts to your enterprise

Zenoss is easy to adapt to your network needs. It provides simple user-interface-driven extensibility and APIs for more significant enhancement. Our community has used these methods to deliver support for many different devices and applications with freely-available Zenpacks.

Five: Zenoss Enterprise simply scales

Zenoss Enterprise's modern technology architecture allows it to easily breakthrough the scaling barriers of legacy products today. Today Zenoss customers monitor thousands of servers and tens of thousands of devices.