

# Consolidating Enterprise Performance Analytics

## A Foundation for Effective End-to-End Enterprise Monitoring



### Introduction

---

With rapid globalization and round-the-clock application availability requirements, IT operations teams, architects and solution centers have proliferated worldwide – along with a corresponding rise in the number and type of tools to monitor application performance and availability. IT organizations are now faced with managing an ever-growing number of toolsets in use across infrastructure, database, middleware and application support organizations. But teams are still unable to bridge the gaps that exist across these silos and get all the performance data they need in one place so they can begin to see relationships and causal chains across events in different layers. Teams instead must manually consolidate metrics, associate related events, and generate actionable intelligence from that data, all by hand.

The flaws in this manual approach are twofold:

- Processes are reactive – incidents and MTTR have not sufficiently improved. Too many people are involved in too many war room meetings. Teams need a “single source of truth” to identify causal issues and eliminate red herrings
- Significant investment in (and switching costs for) incumbent toolsets make rip-and-replace costly and impractical. Tools need to work seamlessly alongside each other, minimize or eliminate redundancy and provide rapid time-to-value for end users.

Thus, a solution must provide a way to seamlessly integrate with existing systems, gathering the relevant metrics unobtrusively and consolidating these metrics – both logically and visually – so they can be seen collectively, and drilled into across multiple directions.

**RTView Enterprise Monitor**<sup>®</sup> addresses these requirements in an inexpensive, easy-to-implement, easy-to-maintain performance analytics solution that can be easily tailored to unique customer environments and user preferences.

### Scalable and Unobtrusive Collection

---

RTView is built on a flexible and efficient data collection approach – leveraging JMX, SQL or other access methods (e.g., SOAP) for collecting data directly from a data source. In this way, RTView can both leverage monitoring data already collected by tools like Hawk, BMC Patrol, Oracle Enterprise Manager, SCOM, or Tivoli, as well as use a variety of built-in collection mechanisms for middleware and other components not monitored by any other solutions. In this way, RTView enables customers to build out a scalable, efficient monitoring architecture that maximizes performance while preserving flexibility without disrupting the usage of incumbent monitoring solutions.

This approach contrasts greatly with typical resource monitoring solutions, which in many cases require agents/probes on each physical server. While common and powerful, agent-based approaches impose performance and administrative overhead and are frequently limited to monitoring compute-level metrics rather than higher-level middleware metrics. RTView enables customers to mix and match collection mechanisms so as to achieve a “total picture” of application resource health while balancing the often competing priorities of performance, flexibility, and ease of administration. RTView Enterprise Monitor addresses this monitoring gap by providing an easy way to leverage existing sources of monitoring data, and bring those into the system alongside additional data captured exclusively by RTView. In this way, customers can enhance existing solutions by providing more holistic detail about component behaviors and begin to understand the complex relationships underlying each incident.

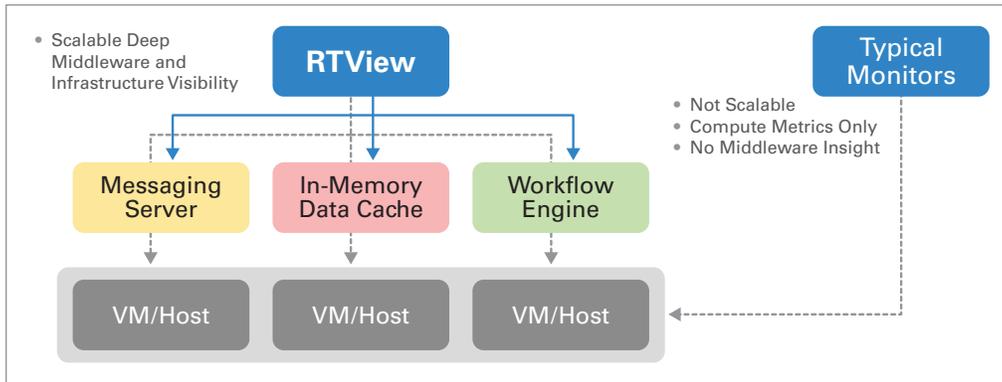


Figure 1 – RTView enables scalable data collection across all components in an application deployment environment, not just servers

## Efficient Localized Storage and Retrieval Architecture

Once collected natively or imported from existing monitoring sources, monitoring data are stored in a set of local caches close to the data source rather than in a centralized repository for rapid, real-time access to performance data. Since storage is local, no network overhead is incurred during collection, and since no central repository exists, no data transfer is required after the data are collected. As it ages, historical data is written out from this cache to a persistent local data store with aging rules completely configurable to customer requirements. When requests come in, RTView uses cache mapping technology to pull only the required data from its local storage location and surface that data subset to users.

The RTView cache is designed for efficient handling of real-time multidimensional data and optimized for the insertion/extraction of time-sequenced data that is indexed or addressed by multiple index columns (time-stamped and multi-dimensional). Other kinds of caches cannot be leveraged to work against time-ordered data, designed instead to store individual objects.

Because RTView caches are column-oriented – a series of rows with columns repeated at regular time intervals – each cache can leverage specialized insert functionality to ensure extremely fast performance. The RTView cache maintains the current indexed data set and thus can accept data asynchronously, automatically creating an in-memory repository for fast access to short-term history.

For different data sources, RTView maintains different caches (e.g., app server sessions vs. JVM data vs. pending message queue) or can combine them into a single cache requiring the fewest number of tables. In this way, RTView’s caching system enables caches to conform to data shape and indexing schemes.

RTView enables monitoring data to be aggregated using group-bys, joins, and deltas to provide an extra level of intelligence to raw table-based, disjointed data.

For example, in order to get an understanding of clustered application server performance for Oracle WebLogic (something not performed, for example, by Oracle Enterprise Manager), RTView uses group-by across web server instances to produce aggregate metrics for the clustered application. In this manner, RTView joins multiple WebLogic monitoring tables to provide a holistic view of clustered web server session behavior.

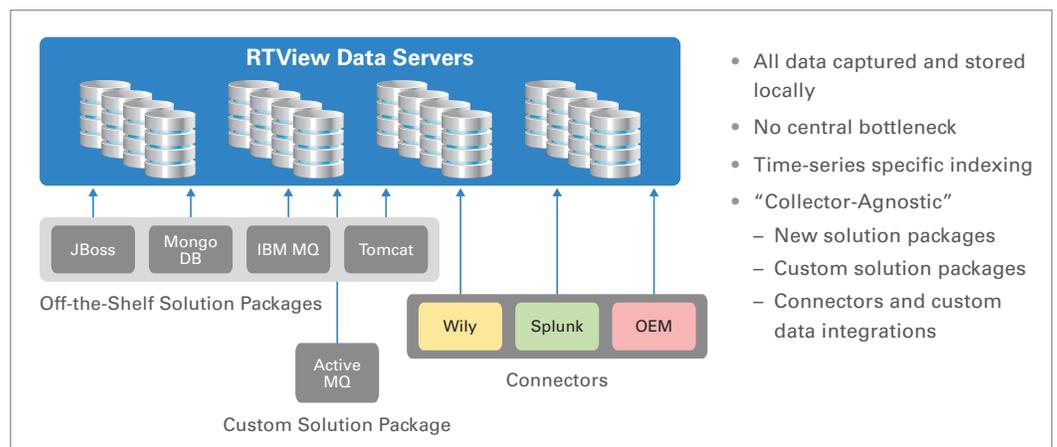


Figure 2 – RTView consolidates metrics from multiple sources and stores these locally in a set of distributed caches

## Service-Based Organization of Monitoring Data

Because of its distributed architecture, RTView Enterprise Monitor enables the aggregation of data across all layers of the application deployment footprint as well as provides facilities for transforming and combining that data in a cache specific to a particular data source. Data from a variety of data sources are then normalized to eliminate variations in data scale and format, allowing for that data to be represented visually and fed into models gauging alert criticality and severity. In this way, RTView Enterprise Monitor can present a *complete* picture of application behavior and performance across any number of previously unrelated resources, presenting all relevant information in a single integrated view, while enabling drill-down to and analysis of the individual component performance data.

However, organizations running a number of different applications across a common pool of hardware and software resources need a way to catalogue individual components, specify dependencies across components and systems, prioritize alerts for specific user roles and applications, and enable results to be auditable for post-situational analysis. This requirement involves the creation of a dynamic service model that associates individual architectural components with business services, whether defined in an existing Configuration Management Database (CMDB), or custom-defined or discovered at runtime.

With RTView Enterprise Monitor, active technology components and system views can be discovered and updated automatically, but in a deterministic fashion. This approach is very different from auto-discovery tools, which impose their own, often faulty associative model on the discovered data. RTView enables users to define and derive relationships between data based on user experience and metadata, such as queue or instance names. The resulting service model produces “logical groupings” in a hierarchy of IT assets that can be navigated through the dashboard or used in alert specifications. In addition, the service model is used to specify the criticality of issues against the configuration items and their parents in the hierarchy: alert criticality, as well as the criticality of a configuration item or service, is used to calculate the impact of system performance on the business.

Without a service model, monitoring solutions cannot understand which fault conditions are related or unrelated to others, or the relative criticality of the component failure to the application that it serves. RTView Enterprise Monitor leverages this capability to aggregate and classify alerts gathered at the component level and display these contextually.

## Summary Visualizations and Dashboards

The UI in RTView Enterprise Monitor is a real-time dashboard, providing current and historical event metrics and representing these as heatmaps for at-a-glance monitoring, and as tables for more comprehensive information views. Mouse-overs and drill-downs enable more detailed information than can normally be displayed in simple line graphs. In addition to heatmaps, RTView Enterprise Monitor provides an array of visualization types, including pie charts, bar charts, strip (or trend) charts, object grids, process flow charts and other unique representations, including combinations of many of the above (e.g., composite objects).

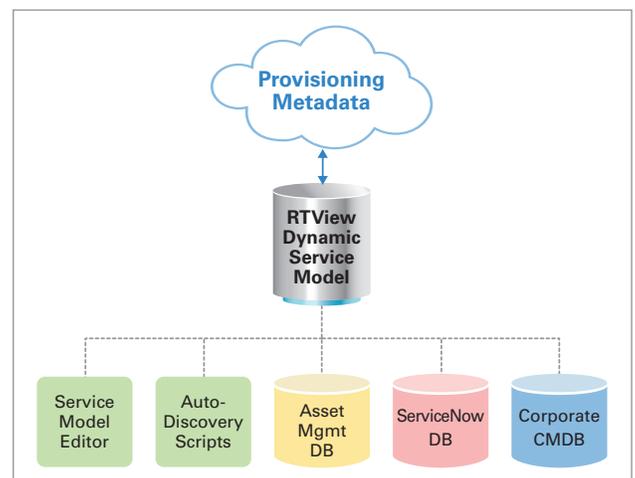


Figure 3 – RTView’s dynamic service model aggregates existing CMDB data and defines new relationships deterministically

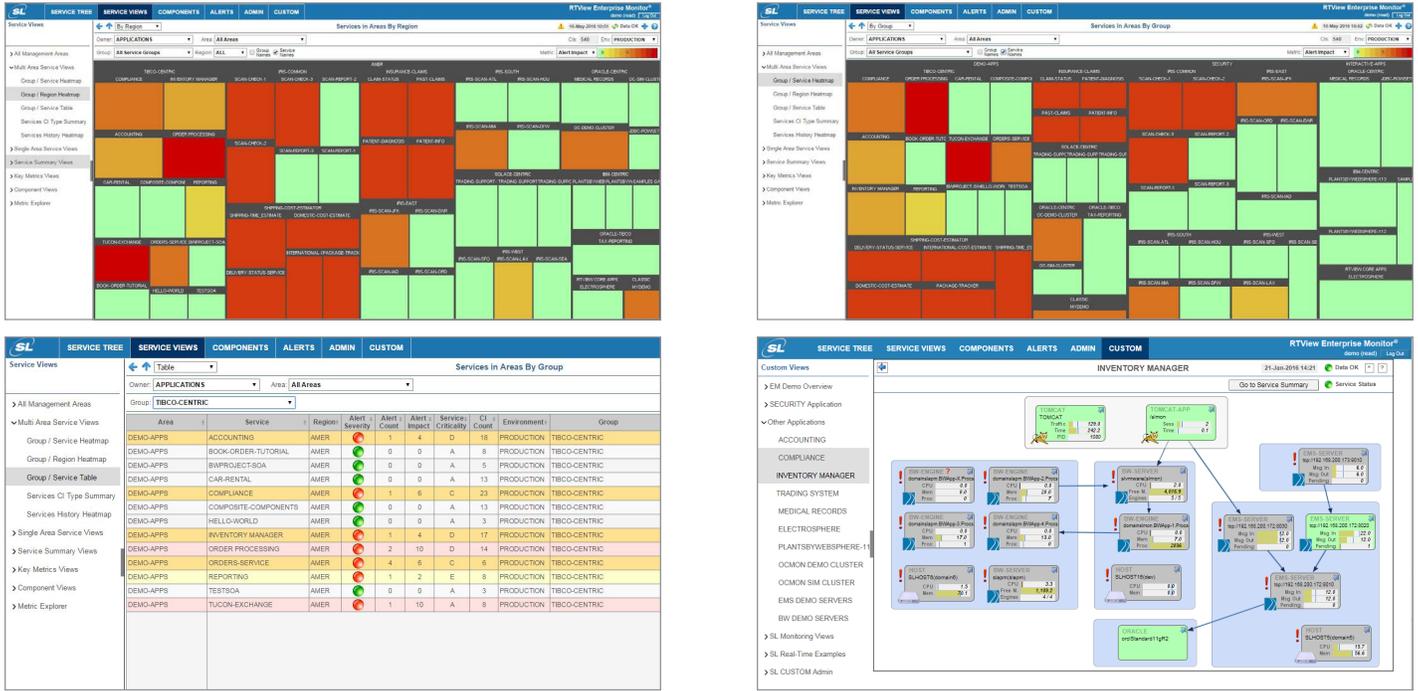


Figure 4 – Consolidated monitoring analytics are provided using advanced context-sensitive visualizations

Display types are also templated, leveraging interactive filters to modify variables within the same view, toggling between metric types or collections of components, so as to focus on those elements that are most important. In this way, the RTView UI provides a means for representing and understanding the huge amount of metrics required for end-to-end monitoring. Customizable displays are also provided for cross-system summary information or organization-specific views within a single pane. RTView Enterprise Monitor users leverage color keys for notification of aggregated issues from which they can drilldown to identify the root cause of the alert and possible or potential fault conditions.

## Correlation, Prediction and Capacity Planning

For most organizations, simply gathering all the data that exists in various silos is a crucial first step in implementing an end-to-end performance analytics solution. RTView Enterprise Monitor not only performs this vital function; it also provides additional intelligence and insight into current and historical performance patterns across components by leveraging the service model to correlate monitoring metrics and apply additional analysis to current AND historical data.

Correlation is achieved with RTView Enterprise Monitor by simply comparing data between/across associated components so that users can make time-based comparisons to determine what's going on at a particular moment between two related events – for example, is a cache down because of a rogue process increasing the number of cache hits or rather a memory problem in the underlying JVM in which that cache is running? Cross-correlation allows users to compare metrics across multiple components and tiers, from the host or VM all the way up to the UI, and see that data in the context of an application service to accurately identify not only the source of a problem, but also the impact to the business. Users can also filter monitoring metrics and alerts so that different teams can see exactly, and only, that data relevant to their role.

RTView Enterprise Monitor provides a number of analytic capabilities designed to leverage correlation to *predict* resource constraints and performance issues. The key metrics facility within the product looks at correlated performance characteristics *across multiple resource tiers over time* – before alert thresholds are reached – and route the appropriate notifications to stakeholders so that they can take corrective steps proactively. Additionally, RTView provides a selection of time-based capacity metrics that enable teams to make capacity planning decisions to ensure optimal performance and reduce technology spend.

## Summary: End-to-End Monitoring With a Consolidated Platform

---

End-to-end application monitoring spans all the components and resources of a deployment architecture, and also integrates application and service-level relationships and dependencies across the underlying architectural elements. The underlying functionality of such systems needs to provide deep visibility into each architectural layer, as well as an easy way to drill up, down or across the application or PaaS runtime environment. This provides development, dev ops and application support teams to address the concurrent challenges of analyzing performance, availability, scalability, capacity, concurrency, usage, and overall system health. The promise of end-to-end monitoring is to deliver comprehensive visibility across all these analytic domains while still remaining easy and inexpensive to implement and use.

SL Corporation's RTView® Enterprise Monitor™ was designed to satisfy these complex and sometimes conflicting requirements. RTView features a large and growing collection of solution packages that unobtrusively collect performance metrics from deployment components and plug seamlessly into the Enterprise Monitor framework. RTView Enterprise Monitor also provides an easy way to connect to existing monitoring systems that collect data on their own, including TIBCO Hawk, Oracle Enterprise Manager, Microsoft System Center, IBM Tivoli, or any other such solution. In this way, RTView Enterprise Monitor acts as a consolidation mechanism for monitoring intelligence across an entire deployment environment – from host to UX – providing an easy way for architects, support teams and application owners to improve holistic visibility, foster closer collaboration, and ultimately, ensure operational application health.

### About SL Corporation:

SL is a San Francisco Bay Area-based software company that develops real-time monitoring, visualization and diagnostic software for Global 1000 and mid-market companies whose businesses depend on complex, high-performance applications. The company has deep expertise in monitoring and management of custom, distributed applications running in heterogeneous middleware environments. SL's RTView® product lines address a broad spectrum of enterprise visibility challenges including end-to-end application monitoring, middleware monitoring, infrastructure monitoring and custom monitoring.

**For more information on SL, please call +1 415-927-8400 or visit [www.sl.com](http://www.sl.com).**

SL Corporation | 240 Tamal Vista Blvd. | Corte Madera, CA 94925 | USA  
415.927.8400 phone | 415.927.8401 fax | [www.sl.com](http://www.sl.com)

© 2016 Sherrill-Lubinski Corporation. All rights reserved. The SL Logo, RTView Enterprise Monitor, BW Monitor and SL-GMS are trademarks or registered trademarks of Sherrill-Lubinski Corporation in the United States and other countries. 05/2016

