

Maximize Your Current Monitoring Using New Data Collection Methods

Shift from monitoring to observability using Dimensional Data

A better engine needs better fuel

IT performance monitoring is undergoing a major transformation thanks to advances in artificial intelligence and real-time analytics. Many IT monitoring platforms promise new capabilities that will move them beyond monitoring to observability. Even the best analytics engines will struggle to fully deliver on that promise unless the industry changes the most prevalent, metric-in-a-vacuum approach to system data collection.

But before we examine why that change is needed, let's first define the goal of observability. In one of the clearest definitions to date, Cindy Sridharan describes observability as a superset of monitoring. Observability brings together monitoring, alerting/visualization, distributed systems tracing infrastructure and log aggregation/analytics to provide better visibility into system health.

Today, most monitoring platforms still rely on an approach to metric collection that emerged about a decade ago when the first cloud-native application performance monitoring platforms emerged. They use thousands of hardwired 1:1 or 1:N endpoint technology-to-platform integrations. These integrations are most-often developed by the monitoring platform, endpoint technology providers or their customers in response to a specific request or use case. Ongoing updates to these integrations are rare—as they're often not an R&D priority for technology providers or orphaned as community commitment fades.

In order to achieve such a lofty goal, the industry must shift metric collection from a focus on flat metrics to Dimensional Data. This paper defines the core elements of Dimensional Data and demonstrates how this improved data-collection model fuels better results from next-gen analytics engines.

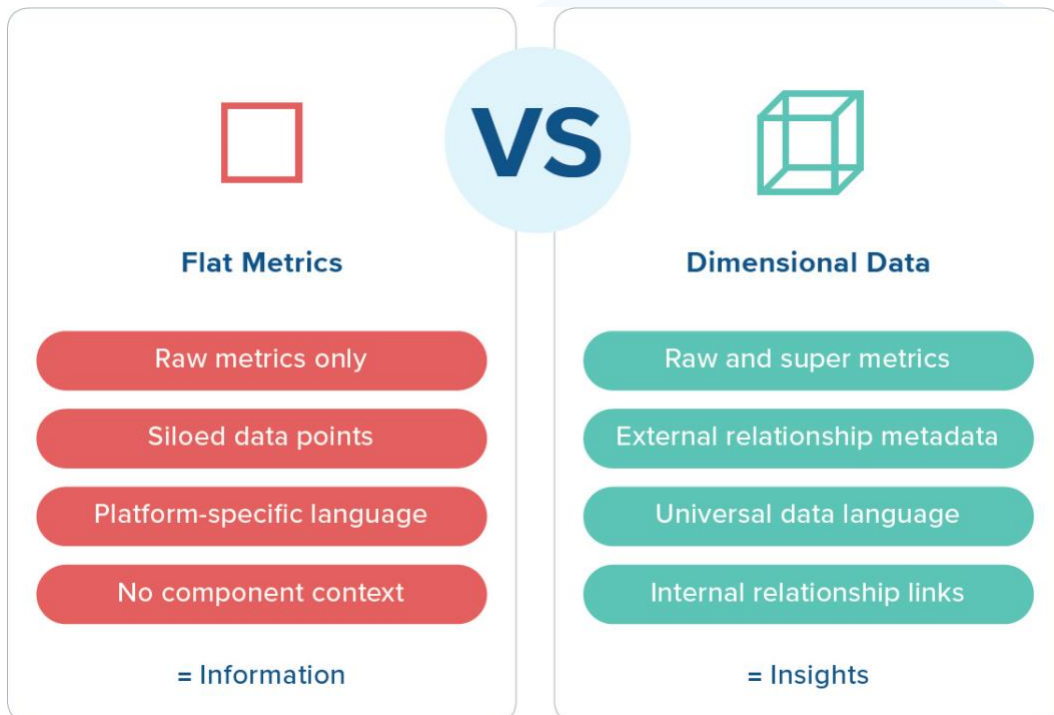
Why current metrics fall “flat”

To understand the need for Dimensional Data, it’s important to first understand the challenges with the current data collection status quo, something we refer to as “flat metrics”.

Flat metrics provide only surface level analysis, without segmenting the pieces of a technology into its various roles and responsibilities. Flat metrics don’t include the context that helps you understand how each resource in your IT stack relates to one another. That context has become critically important in recent years, as newer technologies like containers and serverless have clouded relational visibility. The very abstraction techniques that make them so user-friendly to DevOps teams make them more challenging to monitor and analyze.

As a result, flat metrics can create false-positives in monitoring platforms. For example, a database might appear to be down, but the actual root cause of the problem may be that a lack of available memory on the host is causing it to reject connections.

To avoid that situation, many teams have turned to tools that automatically rebalance workloads to address an application performance issue. This is a short-term solution, much like taking an aspirin for a broken arm. In the long run automated rebalancing without human authorization can be costly—in the form of unexpected cloud service charges.



Four key dimensions of IT system data required for observability

Dimensional Data refers to the stream of information delivered by a next-gen monitoring integration strategy, like monitoring-integration-as-a-service (MaaS). A Dimensional Data stream will include highly-granular behavioral detail—beyond what a single endpoint API connection might include—as well as rich relational context. The term Dimensional Data is specific to IT system health and performance information and should not be confused with dimensional models used in data warehousing or low and high Dimensional Datasets in business analytics.

Key characteristics of flat metrics and dimensional data. In contrast to flat metrics, Dimensional Data brings the standardization, relational visibility and super metrics that move performance monitoring and analytics platforms closer to observability. Some traditional approaches to monitoring integration may cover one or two dimensions of data, but the only way to access all four dimensions of system data is with a MaaS. Over the next few pages, we'll take a closer look at each element of Dimensional Data with specific emphasis on the architecture advantages Blue Medora's BindPlane MaaS has over traditional flat metrics. For each element, we'll use real-world examples to demonstrate how those technical advantages elevate next-gen monitoring and analytics platforms closer to observability.

Step #1

Dimensional Data element: Universal data language

As mentioned previously, the traditional approach to monitoring integrations is to rely on plug-ins, scripts or other single endpoint to single platform connections provided by a variety of sources. These integrations lack a standard data language and even desired functionality. Because of the fact that many of these traditional integrations are hardwired, they require ongoing maintenance, either from the community who

developed them or by the company that deployed them. That maintenance can actually turn into a significant investment, one with the potential to rival application costs. That integration investment can be enough to lock some organizations into a specific monitoring and analytics platform for much longer than they desire.

What's more, most organizations run six or more monitoring tools. Larger organizations relying on traditional integration methods and flat metrics often find themselves deploying (and maintaining) multiple integrations for each endpoint they are trying to monitor.

How a universal data language delivers on observability

A MaaS, like BindPlane, includes a universal data language as part of its Dimensional Data stream. In this case, the BindPlane data provider translates all the metrics from a given endpoint into our proprietary ExUno universal data language, which makes it compatible with any monitoring platform and every use case (alerts, dashboards, reports, etc.). The result is a standard process for integration—an integration layer if you will—that delivers universally accessible insights to every analytics platform within the organization.

This leads to a significant reduction in integration development, customization and ongoing maintenance. If observability is truly a superset of monitoring the first step is ensuring all those tools have access to the same data.



Figure 3. BindPlane data provider translates each endpoint into the ExUno universal data language, making its insights accessible for any monitoring and analytics platform.

Step #2

Dimensional Data element: Internal relationship links

Traditional integration methods also create challenges when it comes to data depth. Most rely on 1-2 hard-wired connections, which can provide enough detail for some monitoring use cases, but when attempting to use for observability, they can end up falling short.

A MaaS architecture features a much more flexible ingestion framework, one that is able to accommodate multiple, varied types of APIs, from REST to SOAP and SNMP to ODBC for every endpoint connected to it. The result is a data provider, like the one included in the BindPlane MaaS, can go more than 6x deeper than community or open sourced plugins, see figure 4. The BindPlane smart collector brings in component-level metrics for a given endpoint and links them together, so you can analyze performance by endpoint (ex. a PostgreSQL DBMS or cluster) or drill down into a specific component (ex. a database instance or node).

FEATURE	TELEGRAF	COLLECTD	MIAAS
Monitored object types	<ul style="list-style-type: none"> • Redshift Cluster • Redshift Node 	<ul style="list-style-type: none"> • Redshift Cluster • Redshift Node 	<ul style="list-style-type: none"> • Redshift Cluster • Redshift Node • Redshift Disk Partition • Redshift Table • Redshift Query • Redshift Query Plan
Metric sources			<ul style="list-style-type: none"> • AWS Cloudwatch • AWS Extended APIs • 3rd Party APIs • Synthetic metrics
Internal relational links	X	X	✓
External relational metadata	X	X	✓
Redshift query level detail	X	X	✓
Total metrics	15 metrics	15 metrics	109 metrics

Figure 4. The flexibility of a MlaaS framework enables the highly granular insights into the behavior of systems along with rich, relational context.

How internal relationship links deliver on observability

Summarizing Sridharan again, she feels good monitoring is simple. It should tell you if your systems are up or down. She continues, “Observability, on the other hand, aims to provide highly granular insights into the behavior of systems along with rich context, perfect for debugging purposes.” Sridharan describes a very clean line between monitoring and system optimization, one that works nicely in theory, but isn’t something we’ve seen in practice often. Many organizations are trying to maximize their investments in monitoring tools by squeezing more insights out of their analytics engines. Dimensional Data can provide the highly-granular behavior detail.

Let’s take a closer look at the example in figure 4. A fairly common scenario of monitoring a Redshift database using a popular open-source technology. Each of these integration technologies will enable your APM platform to alert you if a particular Redshift instance is down. But the APM platform might not give you a great deal more than that, and Sridharan argues, you might not even want it to. The root cause of your problem could be in disk space, table space or your queries. With flat metrics, you won’t have access to deep dive metrics within the database instance in your APM, you also won’t be able to find that depth in Amazon Cloud Watch. You’d need to go into Amazon Redshift Monitoring utility to get object-level data depth.

Step #3

Dimensional Data element: External relational metadata

Internal relationship links can provide the highly-granular detail you need to identify root causes more quickly and accurately, but external relationships are the rich content Sridharan describes. Noise has become a real problem in monitoring. Flat metrics generate false alarms and false positives because they lack the ability to sort root cause from effect. We’ve become so accustomed to this norm that we accept alert fatigue as the only outcome. The issue has become so pervasive that many organizations acknowledge they have at least one alerting subsystem that is entirely ignored.

How external relationship metadata delivers on observability

The third dimension of system data, external relationship metadata, can significantly reduce alert fatigue and clear the way to observability. As the BindPlane MaaS data provider ingests new metrics, it flags any with potential external relationships with a tiny piece of metadata. The BindPlane manager uses this external relationship metadata to auto-discover any new or changed relationship information for that endpoint since the last data collection (as little as 5 seconds ago) and uses that information to update a full-stack relationship map. This map, when shared with the analytics platform through the Dimensional Data stream, helps the platform and its users sound the alarm more accurately.

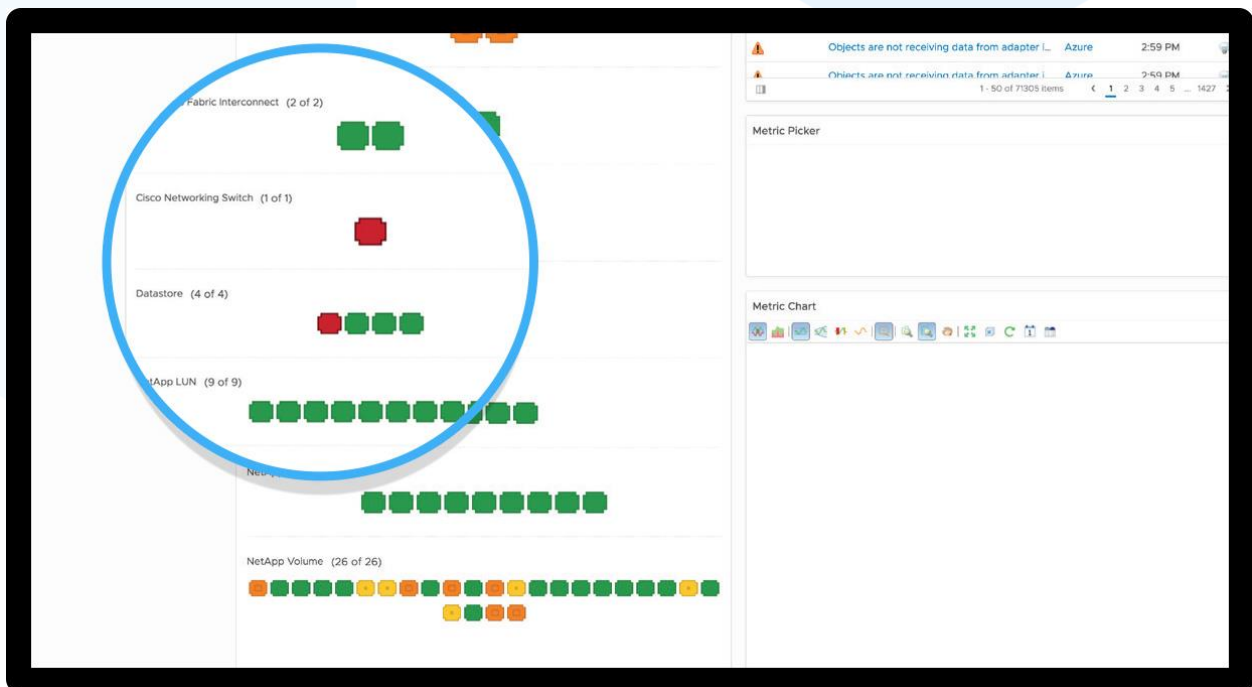


Figure 5. The relational context in a Dimensional Data stream populates this full-stack dashboard in VMware vRealize Operations for faster troubleshooting.

To see how external relational metadata filters out the noise, take the example of a full-stack relationship map delivered as a dashboard inside the VMware vRealize Operations cloud management platform. Each of the objects shown in figure 6 are colored based on their overall health. If an object is red, highlighting it will show you what other objects might be affected. With a double click, you can drill down to see exactly what's causing this red condition (figure 7).

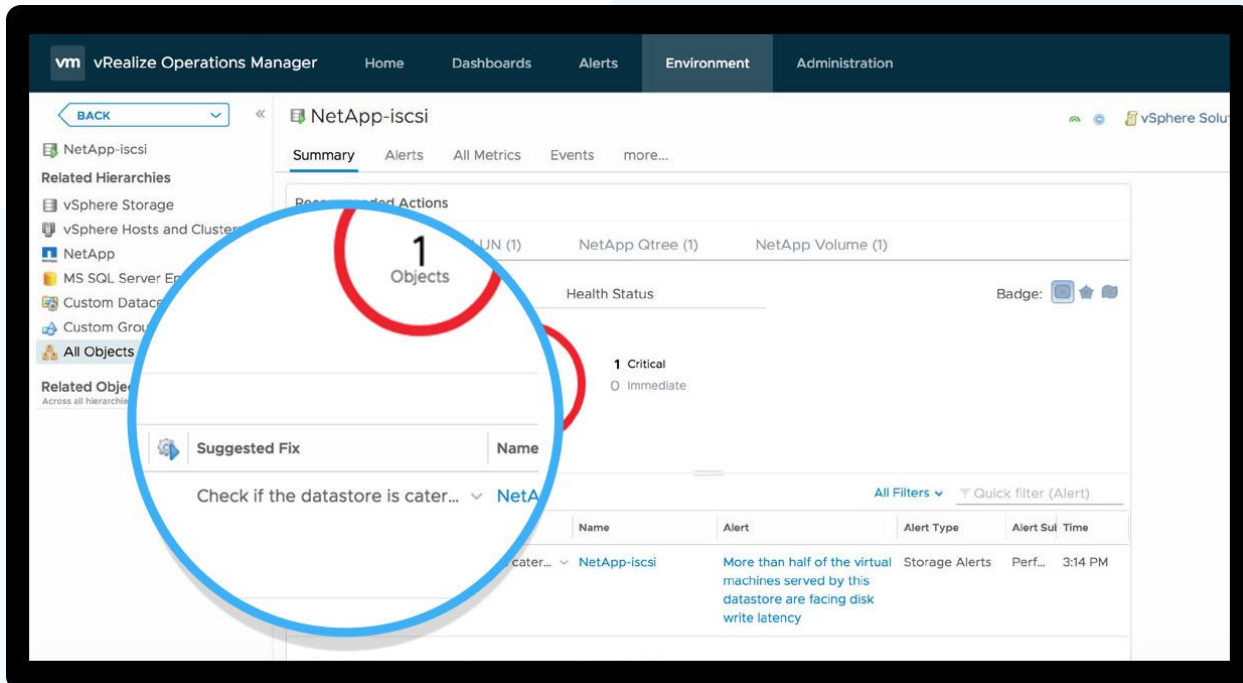


Figure 6. A drill down from the dashboard indicates that this particular database would benefit from more back-end storage and recommends storage vMotion. These are all details provided by the highly-granular intelligence of the Dimensional Data stream.

Step #4

Dimensional Data element: Super metrics

One of the core tenets of observability is meaning, and that's also the driving factor behind the final Dimensional Data element, super metrics. As previously mentioned, flat metrics are a raw feed from the endpoint API. To make use of flat metrics as health and performance measures you—or your platform—has to know what to do with them. That is often taken for granted even as organizations become more DevOps driven. Having a subject matter expert on staff for each of an ever-evolving list of more than 200 technology endpoints is not practical. Neither is relying on a platform to provide them.

Many DevOps teams rely on custom tooling and even some of the of the most cuttingedge real-time analytics engines aren't designed specifically with monitoring in mind.

How super metrics deliver on observability

A Dimensional Data stream from an MaaS provider like BindPlane can deliver raw and synthetic metrics (super metrics). These calculated metrics combine multiple raw metrics to calculate rates or ratios that have more meaning than the original flat metric. Here are a few places super metrics come in handy.

1. Applying functions: If you want to know, for example, what the average is of a metric such as execution time of queries on a database, you can do that with one operation.

2. Performing rollups: Going back to the earlier Redshift database example, a flat metric feed may show you the total number of CPU consumed by a particular node, but you can't understand how close your cluster is to maxing out.

Certainly, many monitoring platforms can enable you to create and or "view" this information, but Dimensional Data simply delivers it, which is a time saver when it comes to things you calculate regularly. The example in Figure 8 shows how super metrics can elevate the users understanding of the overall health of this PostgreSQL database, something that helps simplify both monitoring and observability.

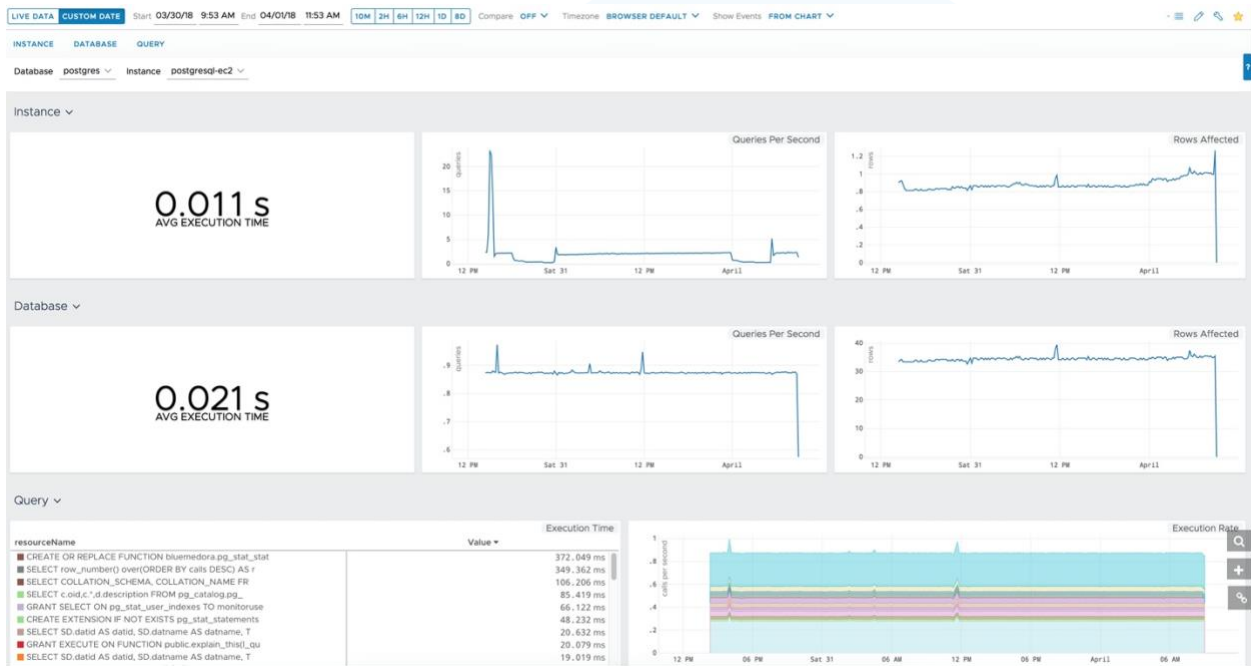


Figure 7. This dashboard features instance and database execution time super metrics, which are aggregates of queries below, provided by the Dimensional Data stream.

Any path to observability requires Dimensional Data

Monitoring purists like Sridharan might argue that the best route to observability is to look for a way to bring together insights from your monitoring, log analytics, tracing and other tools. That works best when each of the tools are drawing from the same well of information and admins can share a common language.

In practice though, some organizations try to get to observability by expanding the capabilities of their existing monitoring tools to take on more observability-like functions. Many monitoring, log analytics and tracing tools are innovating in the hopes of becoming the single platform for observability. While it's unlikely that any platform will become the single source for observability, Dimensional Data deployed through an MlaaS gives any platform the highly-granular intelligence and the relational-context to bring observability to most platforms. Perhaps more importantly though, an MlaaS also gives teams the flexibility to change their path to observability at any time without sacrificing a significant integration investment.



About Blue Medora

Blue Medora's pioneering IT monitoring integration as a service addresses today's IT challenges by easily connecting system health and performance data—no matter its source—with the world's leading monitoring and analytics platforms. Blue Medora helps customers unlock dimensional data across their IT stack, otherwise hidden by traditional approaches to metrics collection. Learn more at [BlueMedora.com](https://www.BlueMedora.com).

Contact

3225 N. Evergreen Drive NE,
Suite 103
Grand Rapids, MI 49525
(616) 719-4550

Follow us:
[@bluemedora](https://twitter.com/bluemedora)
[linkedin.com/company/bluemedora](https://www.linkedin.com/company/bluemedora)

©2019 Blue Medora.
All rights reserved.