



A JOINT XOPS AND INFLUXDATA PAPER

The I/O of DevOps Solving the Problem of Information Silos Using a Time Series Database

External Contributors

Larry Gordon

CTO, xOps

May 2018

Introduction

For far too long technology teams have lived in silos – not only organizational but also information silos. Application and service data within companies is often as segregated, if not more, than the teams that build and maintain them. In this paper we discuss the ways in which the many input sources can be brought together to provide a holistic picture of how your applications and services are performing for your customers. Then we discuss how the open source [time series database](#) InfluxDB and its [time series platform](#) TICK Stack can help break down these information silos.

The information silos problem

A study conducted by [xOps](#) – a DevOps open source tool developer and a consulting, tool implementation, and managed services company – to identify the extent of the information silos problem found that \$26.5 billion in revenue is lost each year from IT downtime. In fact, revenue is not the main impact of downtime.

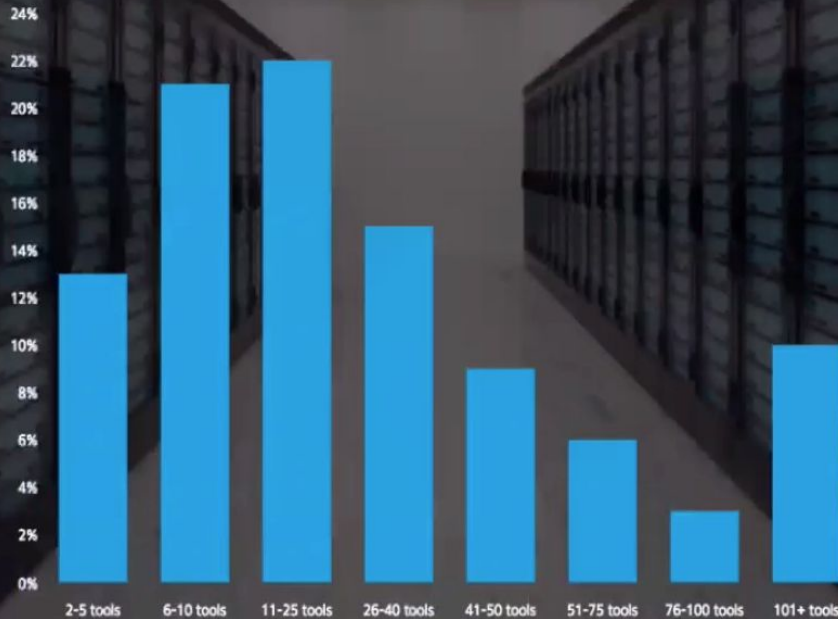
Another study found that 78% of downtime costs relate to reduced worker productivity (very talented and very well-paid engineers who are unable to do their jobs due to IT downtime). Missed sales opportunities and lost revenue make up a substantial 17% of losses from downtime. xOps, in partnership with [InfluxData](#) (the creator of [InfluxDB](#)), is figuring out how to solve this problem.

In DevOps, there is much discussion about the deployment pipeline, automation, continuous integration and continuous deployment, yet automated tests in an ideal scenario don't take into account production environment problems. The reality is that especially in mid to large-sized organizations, enterprises are often faced with legacy systems at all levels of maturity, including long manual deployments and systems that don't have the automation necessary for seamless deployment. Environments need to be monitored so that when things go wrong, enterprise teams can react.

The challenges in today's environment include massive data growth, monitoring tool proliferation, and increasing information silos.

Too Many Monitoring Tools

65% of enterprises have 10+ monitoring tools, despite the fact that siloed tools with limited integration lead to poor end-user experience, reactive performance management, long MTTR, and extensive investments to procure and manage the tools.



APPDYNAMICS

In its survey, AppDynamics found that 65% of enterprises have 10-plus monitoring tools. In fact, xOps encountered one company that had over 50 tools for monitoring their systems and services, some of which were implemented in multiple ways across multiple regions. A red alert in one area didn't mean the same thing as a red alert in another area with the same tool. Within all that noise, there's no real way to figure out how your systems are performing, alerts are often missed or lost, and it's hard to anticipate when a failure is approaching. This problem is exacerbated when information gets siloed and provided to individual pieces of the organization.

In DevOps, informational silos impact system-level availability. In a classic informational silos, developers have access to information about development, and operations are the only ones who have access to information about how those systems that they rely on are performing.



But even within technical teams and within operations teams, information silos is a real problem – especially in larger organizations that have specialized teams, where the network team is the only one who has insight into the network data, and the database team is the only one who can see into database performance. If we're not sharing this information, we can't collaborate and work towards a shared solution or have the shared ownership necessary to drive appropriate behavior. In the DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations (2016), authors Gene Kim, Jez Humble, and Patrick Debois noted this problem:

"For decades, we have ended up with silos of information, where development only creates logging events that are interesting to developers, and operations only monitors whether the environments are up and down...As a result, when inopportune events occur, no one can determine why the entire system is not performing as desired."

To solve the informational silos problem, xOps has focused on developing open source products to revolutionize technical operations by gathering and analyzing all operational metrics, logs, alerts, and business data in one place, in real time using InfluxDB. This allows teams to build self-healing systems through automated issue resolution and provide visibility across business metrics with predictive analytics to determine trends before the competition.

How to “DevOps” your data

There are three key steps to break down information silos, reduce your time to resolve them, and predictively find and prevent issues before they impact customers.

1. Determine data sources and reduce duplications

The first step is to assess and streamline your data situation:

- Eliminate duplicate sources of data: Look at all your data sources and reduce duplicates where possible. Data sources can include systems monitoring tools, application performance monitoring tools, and social media feeds such as Twitter and Instagram. In addition, data sources can and should include data from your IT service management (ITSM) tools.
- Use best-in-class tools for each area (but only find one tool for each area since you don't need five different tools to monitor CPU for example), then standardize and deploy them throughout the enterprise.
- Find the level of diversification and granularity of these tools that suits your organization. That level depends on the organization's size and maturity. Smaller organizations may only need two or three different monitoring tools, or tools that cover two or three different areas, in the absence of specialized teams.

2. Centralize key data

Once you find the right level of granularity for your organization, size, and maturity, you need to centralize key data. This is where tools like InfluxDB, as well as other correlation and presentation tools, come into place. Yet centralizing data is easier said than done since a lot of data sources were designed to contain their data by themselves.

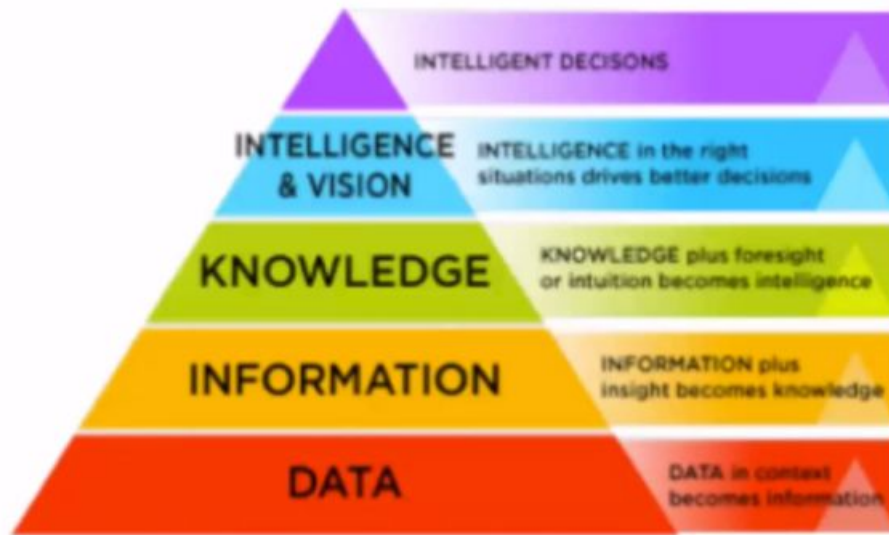
When building a server monitoring tool, developers usually have a very singular purpose in mind, thinking "How do we collect and present the data?" rather than "How do we share that data with something else?" This is why it's critical to examine your data sources and choose the data source store that's right for each data type. Steps to do this correctly are:

- Review your data architecture to examine the different types of incoming data, rationalize them and point them to the same place.
- Determine correlation tool(s) to bring all that data together.
- Clean your input data sources to enhance and improve the raw data.
- Pipe all your data to one place to begin reaping the benefits of data consolidation.

The more disparate data sources brought together and the more data-rich your environment, the more value you have and the more you can break down organizational silos. Transparency is good as a practice, but when necessary, limit it through access controls to present the right data to the right types of users. For example, business unit data may be limited-access due to compliance regulations. Tech and business data may need access-level controls, which can be done through building the correct type of dashboards and presentation layers for the correct type of users, such as delivering detailed as opposed to high-level system performance views to each user group. InfluxDB serves this purpose, as it enables centralizing and managing your data through access controls. It allows you to restrict access by creating individual users and assigning them relevant read and/or write permissions.

3. Uplevel your data

The third step in breaking information silos is to up-level your data so that you can take action based on the information it provides.



Data alone can be noisy and difficult to sort through, but there are several ways to uplevel it:

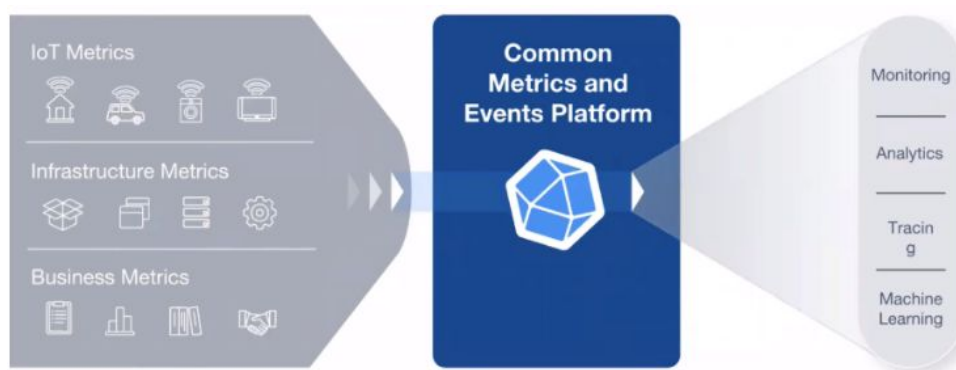
- Look at overall system health by designing dashboards and equations. Application health may mean different things to business and tech users, so design different health equations to present a high-level picture of system health. Service maps can help in this area. If you can map nodes of your services to the overall service or application that it's providing to the customer and then automatically map that to the service or application, you begin to paint a clear picture of system performance.
- Look at real user monitoring and synthetic monitoring, which provides that outside end-picture of how your service is. Tie that in with underlying data to build a rich picture of your business and technical users. By sharing all this data, you can bring them together, all looking at the same information, deriving different values to make your organization run smoothly.
- Use data to apply machine learning so that you can predict when incidents are likely to occur. You can bring together information such as system performance with real user information for synthetic monitoring with information from your social media, and map that against information from your ITSM. You learn and predict when incidents are going to occur, so you can stop them before they impact your customers. Turning data into intelligence will ultimately enable your business to make the right decisions.

One solution that helps “DevOps” your data is InfluxDB and the TICK Stack.

Why the TICK Stack?

The TICK Stack stands for InfluxData’s four open source projects – Telegraf, InfluxDB, Chronograf, and Kapacitor – all written in Go. It is the open source core of InfluxData’s platform. There are several reasons why the TICK Stack is a suitable solution for the DevOps initiatives outlined above:

- The TICK Stack is easy to use - one of the core values at InfluxData is to ensure that developers are happy and find joy in using its products.
- No external dependencies - you can install the TICK Stack, get it up and running, and start using it in just a few minutes.
- Allow for collecting metrics and events - these can be categorized as regular and irregular time series data (actions that occur at a regular interval versus those that may be anomalies).
- Horizontally scalable - InfluxDB can be used by itself, with or without any of the other 3 components.
- Full platform - you can use the collector agents with a number of other databases as well, or you can combine all four projects and use the TICK Stack as a single platform.
- Pull and push for collecting time series - the TICK Stack can collect data either by putting the collector agents out on the various systems that you want to collect metrics on, or can grab that data from containers.



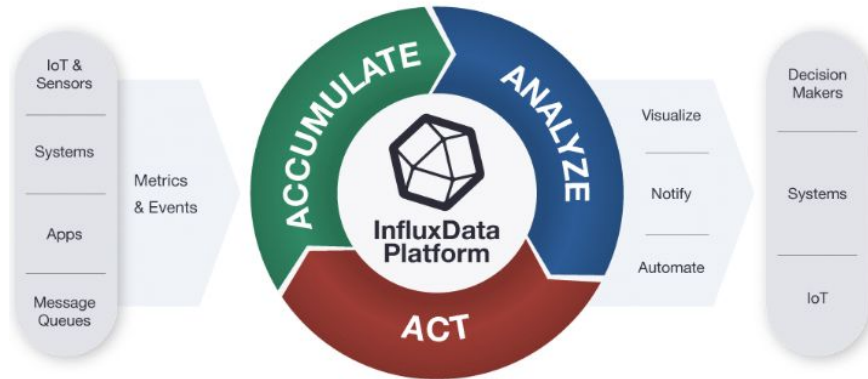
Enterprises use InfluxData to collect metrics and events from a number of sources to gather:

- IoT data from sensors and devices, to optimize solution efficiency and money or energy savings
- Infrastructure data, like the DevOps data described above to measure system health
- Business data, because InfluxData's open source projects are not just tools for the operations team, but also for the development teams. It helps to bring the two teams together through a unified platform that centralizes different kinds of metrics, which ultimately make customers successful if they're able to act on them in the right way.

Once the platform collects the time series data, you can use it to monitor system and application performance and thereby gain visibility. By applying machine learning against the data, you can also make predictions about where your business is going and how much more infrastructure you will need.

The InfluxData platform is a complete platform for handling all time series data, from humans, sensors, or machines. It provides services and functionality to accumulate, analyze, and act on time series data.

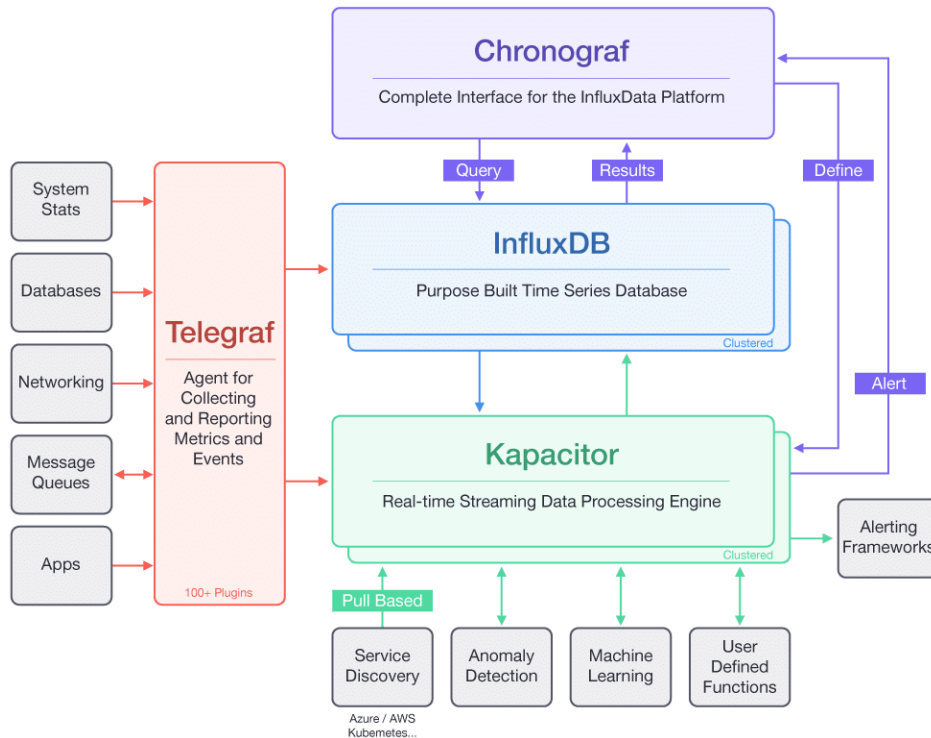
InfluxData Functional Architecture



Following is an overview of InfluxData's four open source projects.

- [Telegraf](#) is a plugin-driven server agent for collecting and reporting metrics. It has over 140 [plugins](#), the overwhelming majority of them contributed by the community, to source a variety of metrics directly from the system it's running on, to pull metrics from third-party APIs, or even to listen for metrics via StatsD and Kafka consumer services. Telegraf also has output plugins to send metrics to a variety of other datastores, services, and message queues.
- [InfluxDB](#) is a Time Series Database built from the ground up to handle high write and query loads. It is a custom high performance datastore written specifically for time-stamped data, including DevOps monitoring, application metrics, IoT sensor data, and real-time analytics. Conserve space on your machine by configuring InfluxDB to keep data for a defined length of time, and automatically expiring and deleting any unwanted data from the system. InfluxDB also offers a SQL-like query language for interacting with data.
- [Chronograf](#) is the administrative user interface and visualization engine of the platform. It makes monitoring and alerting for your infrastructure easy to set up and maintain. It is simple to use and includes templates and libraries to allow you to rapidly build dashboards with real-time visualizations of your data and to easily create alerting and automation rules.
- [Kapacitor](#), the TICK Stack's native data processing engine, can process both stream and batch data from InfluxDB. Kapacitor lets you plug in your own custom logic or user-defined functions to process alerts with dynamic thresholds, match metrics for patterns, compute statistical anomalies, and perform specific actions based on these alerts like dynamic load rebalancing. Kapacitor integrates with HipChat, OpsGenie, Alerta, Sensu, PagerDuty, Slack, and more. Time series data can be processed through Kapacitor against machine learning algorithms to help with forecasting.

InfluxData's four open source projects support DevOps data management and organization, because they can be used for collecting metrics for the functions that you're writing within your application and for metrics on how well any of your systems are performing.



The TICK Stack has been widely adopted by the community as a metrics and events platform largely because at its heart is a purpose-built Time Series Database, which differs from a standard relational database in that the former is designed to handle time series workloads. This is evident in InfluxDB capabilities:

- Ingest of large volumes of points/second - data ingest at very high volumes at millisecond and nanosecond intervals to understand system performance
- Real-time time-function queries on large data sets - real-time view into the data you're collecting
- Rapid eviction and transformation of data - retention policies that automatically expire your data when you need it removed
- Downsampling of high precision data - keeping high-value, high-precision data in memory and lower-value data to disk
- Storage optimization and compression - compacting data to save storage space

Key takeaways

To conclude, below are some takeaways worth considering in implementing the above:

- Stay away from point solutions (when possible): When breaking down information silos and looking at consolidating information in a platform like InfluxData, the more you can generalize within a given space, and the better you'll be able to avoid duplication.
- Avoid vendor lock-in: The need to avoid vendor lock-ins is one reason to choose open source because very rarely do open source platforms force you to use specific tools.

- Ensure that you have access to your data: Very often, you'll get tools in place only to learn that you can't extract that data, which means you have no way to break down information silos or bring that data into a centralized location to visualize it for different users across your business.
- Ensure your tools have complete APIs for automation and integration: Since DevOps requires collaboration, ensure that any tool you choose enables that and helps you share the data.

Remember that DevOps is about building a culture of collaboration – tools alone do not make you “DevOps”. As you push that culture of collaboration, remain focused on developing shared goals, which would then drive shared practices.

InfluxDB documentation, downloads & guides

[Download InfluxDB](#)

[Get documentation](#)

[Additional case studies](#)

[Join the InfluxDB community](#)

[Join the InfluxDB community Slack](#)



799 Market Street
San Francisco, CA 94103
(415) 295-1901
www.InfluxData.com
Twitter: [@InfluxDB](#)
Facebook: [@InfluxDB](#)